

Real-time gross settlement

User detailed functional specifications

Author 4CB

Version 0.4

Date 28 September 2018

Table of contents

Introduction	30
Reader's guide.....	31
Part I - General features of the RTGS component.....	33
1 Overview of RTGS component	33
2 Access to RTGS	35
2.1 Connectivity (U2A/A2A)	35
2.2 Authentication and authorisation process.....	36
2.3 Authentication and authorisation in ESMIG.....	36
2.3.1 Authentication and authorisation concepts.....	37
2.3.1.1 User.....	37
2.3.1.2 Certificate.....	37
2.3.1.3 DN.....	37
2.3.1.4 Technical sender	38
2.3.1.5 Business sender	38
2.3.2 Authentication process	38
2.3.2.1 Authentication of the technical sender	38
2.3.3 Authorisation process	39
2.3.3.1 Authorisation of the technical sender	39
2.4 Security	39
2.4.1 Confidentiality	39
2.4.2 Integrity.....	40
2.4.3 Availability.....	40
2.4.4 Monitoring.....	40
2.4.5 Auditability.....	40
2.5 Graphical User Interface.....	41
2.6 Routing.....	41
2.6.1 ESMIG routing functions.....	45
2.6.1.1 Inbound routing.....	45
2.6.1.2 Outbound routing.....	46
3 Parties and accounts	48

3.1	Parties	48
3.1.1	Setup of parties.....	48
3.1.2	Concept of party in RTGS.....	49
3.1.3	Reference data for parties used by RTGS	50
3.1.4	Participation types.....	51
3.1.5	Blocking/unblocking party	52
3.2	Accounts structure and functionalities	53
3.2.1	Account types	54
3.2.2	Functionalities	58
3.2.3	Messaging.....	62
3.2.4	Blocking/unblocking account	64
3.3	Types of groups	65
3.4	Shared reference data	66
3.5	Interaction between RTGS and CRDM.....	70
4	Business day	72
4.1	T2 Business calendar	72
4.2	Overview	72
4.3	Detailed description of RTGS business day phases	74
4.3.1	SoD.....	74
4.3.2	Settlement windows	75
4.3.3	MWI (maintenance periode)	76
4.3.4	EoD.....	76
4.4	Dependencies to other services or components	77
5	Business and features description	79
5.1	Payment types	79
5.1.1	Overview.....	79
5.1.2	Comparison of different payment types	80
5.1.3	Definition of execution time.....	81
5.1.4	Warehouse functionality	83
5.1.5	Backup payments	84
5.1.5.1	Backup contingency payments.....	84
5.1.5.2	Backup liquidity redistribution payments	87
5.1.5.3	Rules for backup payments	88
5.1.5.3.1	Generation.....	88
5.1.5.3.2	Notification of affected account holder (sender).....	89

5.1.5.3.3	Notification to the receiver	89
5.1.5.3.4	Subsequent delivery of single payments.....	89
5.1.6	Payment priorities	90
5.2	Payments processing and settlement of payments	91
5.2.1	Overview.....	91
5.2.2	Concept of payment submitters	92
5.2.3	Flow of payment related messages	94
5.2.3.1	Payments sent from a direct/indirect RTGS Account Holder to another direct RTGS Account Holder	94
5.2.3.2	Payments sent from a multi-addressee access participant to another RTGS Account Holder.....	100
5.2.3.3	Payments sent from an ancillary system.....	102
5.2.3.4	Payments sent from a CB on behalf of a RTGS Account Holder (mandated payment) to another direct RTGS Account Holder.....	106
5.2.4	Rejection of payments	108
5.2.4.1	Technical validations	109
5.2.4.2	Business validations	111
5.2.5	Amendment of payments.....	113
5.2.6	Revocation of payments	118
5.2.7	Processing of payments	124
5.2.7.1	Entry disposition	125
5.2.7.1.1	General remarks.....	125
5.2.7.1.2	Settlement of payments in the entry disposition.....	126
5.2.7.2	Comprehensive queue management	128
5.2.7.3	Dissolution of the payment queue	132
5.2.7.3.1	Settlement of queued urgent/high payments	132
5.2.7.3.2	Settlement of queued normal payments	133
5.2.7.3.3	Algorithm: "Optimisation on sub-accounts"	138
5.2.7.4	Treatment of backup payments in the settlement process.....	139
5.3	Settlement of ancillary systems	140
5.3.1	Overview.....	140
5.3.2	Ancillary system settlement procedure A	144
5.3.3	Ancillary system settlement procedure B	148
5.3.4	Settlement on dedicated liquidity accounts (ancillary system settlement procedure C and ancillary system settlement procedure D)	151
5.3.4.1	Ancillary system settlement procedure C	153
5.3.4.2	Ancillary system settlement procedure D	161
5.3.4.3	Cross-ancillary system settlement.....	167
5.3.5	Processing of ancillary system transactions using payments	169
5.3.6	Optional connected mechanisms.....	172

5.4	Liquidity management.....	179
5.4.1	Available liquidity	179
5.4.2	Liquidity transfer.....	179
5.4.2.1	Overview.....	179
5.4.2.2	Initiation of liquidity transfers	181
5.4.2.3	Liquidity transfer process.....	182
5.4.2.3.1	Liquidity transfer between two DCAs of the RTGS component	182
5.4.2.3.2	Liquidity transfer from DCA of the RTGS component to CLM MCA ..	184
5.4.2.3.3	Liquidity transfer from DCA of the RTGS component to a DCA in different settlement services	186
5.4.2.3.4	Liquidity transfer from DCA in different settlement service to a DCA of the RTGS component	188
5.4.2.4	Rejection of liquidity transfer orders	190
5.4.2.4.1	Business validations	190
5.4.3	Liquidity management features.....	191
5.4.3.1	Reservation.....	191
5.4.3.1.1	Overview.....	191
5.4.3.1.2	Liquidity reservation and management process.....	193
5.4.3.1.3	Effect and tapping of liquidity reservation	198
5.4.3.2	Limits.....	200
5.4.3.2.1	Overview.....	200
5.4.3.2.2	Process for the definition and management of limits	203
5.4.3.2.3	Effect of limits	204
5.4.3.3	Dedication of liquidity for ancillary system settlement.....	207
5.4.3.4	Floor/ceiling.....	208
5.4.3.4.1	Definition of floor/ceiling threshold	208
5.4.3.4.2	Breach of floor/ceiling threshold - notification	209
5.4.3.4.3	Breach of floor/ceiling threshold - automatic liquidity transfer.....	210
5.5	Information management for RTGS.....	211
5.5.1	RTGS status management	211
5.5.1.1	Concept.....	211
5.5.1.2	Overview.....	212
5.5.1.3	Status management process.....	212
5.5.2	RTGS report generation	228
5.5.2.1	Concept.....	228
5.5.2.2	Overview.....	228
5.5.2.3	Report generation process	229
5.5.3	Query management for RTGS.....	231
5.5.3.1	Concept for RTGS.....	231
5.5.3.2	Overview for RTGS	231

5.5.3.3	Query management process for RTGS.....	231
6	Overview of used common components in RTGS component	234
6.1	CRDM features	234
6.1.1	Concept.....	234
6.1.2	Overview.....	234
6.1.3	Access rights.....	235
6.1.3.1	Access rights concepts.....	235
6.1.3.1.1	User function	235
6.1.3.1.2	Privilege.....	235
6.1.3.1.3	Role.....	247
6.1.3.1.4	User.....	247
6.1.3.1.5	Common reference data objects and the hierarchical party model....	247
6.1.3.1.6	Data scope.....	248
6.1.3.2	Access rights configuration	250
6.1.3.2.1	Configuration of users	250
6.1.3.2.2	Configuration of privileges.....	250
6.1.3.2.3	Configuration of roles	257
6.1.3.3	Access rights configuration process.....	259
6.1.3.3.1	Configuration of access rights at party level	261
6.1.3.3.2	Configuration of access rights at user level	263
6.1.4	Message subscription	263
6.1.4.1	Message subscription configuration	263
6.1.4.2	Message subscription parameter types.....	264
6.1.4.3	Message subscription examples	264
6.1.5	Common reference data maintenance process.....	265
6.1.5.1	Common reference data objects	265
6.1.5.2	Reference data maintenance types.....	268
6.1.5.3	Validity of common reference data objects	269
6.1.5.4	Common reference data archiving and purging	273
6.1.5.5	Lifecycle of common reference data objects.....	274
6.1.5.6	Common reference data propagation.....	277
6.2	DWH.....	282
6.2.1	Introduction.....	282
6.2.2	Scope of DHW	282
6.2.3	Access.....	282
6.2.3.1	Connectivity.....	282
6.2.3.2	Authentication and authorisation	282
6.2.4	User roles and access rights	282
6.2.4.1	Overview.....	282

6.2.4.2	User rights.....	282
6.2.4.3	User profiles.....	282
6.2.5	Data warehouse queries and reports.....	282
6.2.5.1	Overview.....	282
6.2.5.2	Types of queries and reports.....	282
6.2.5.3	Predefined queries and reports.....	282
6.3	Billing.....	282
6.4	Legal archiving.....	283
6.5	ESMIG features.....	283
6.5.1	ESMIG features overview.....	283
6.5.1.1	Authentication of the message sender.....	283
6.5.1.2	Participation to the Closed Group of Users.....	283
6.5.1.3	Validation of the received messages.....	283
6.5.1.4	Message forwarding.....	284
6.5.2	Access to ESMIG.....	284
6.5.2.1	Single access point for the external communication.....	284
6.5.2.2	Network agnostic communication.....	284
6.5.3	ESMIG Portal.....	284
7	Contingency services.....	286
8	Operations and support.....	287
8.1	Business application configuration.....	287
8.2	Calendar management.....	287
8.3	Business day management.....	287
8.4	Business and operations monitoring.....	287
8.5	Possible actions of operator service desk in ESMIG.....	287
8.5.1	Technical monitoring.....	287
8.6	Archiving management.....	287
8.7	Trouble management.....	287
9	Additional information for CBs.....	288
9.1	Role of CBs in the RTGS component.....	288
9.2	Settlement of payments - specific functions for CBs.....	289
9.3	RTGS General Ledger.....	289
9.3.1	RTGS general ledgers production.....	289

9.3.2	RTGS general ledgers content	290
9.4	Query management - specific functions for CBs	290
9.5	Billing - specific functions for CBs.....	291
9.6	Contingency - specific functions for CBs	291
Part II - Dialogue with the RTGS Account Holder		292
10 Processes with RTGS		292
10.1	Send file	295
10.2	Process cash transfer instruction.....	297
10.2.1	Send cash transfer order	297
10.2.2	Revoke/cancel payment	299
10.2.3	Amend payment.....	301
10.2.4	Execute RTGS standing order	303
10.3	Settle RTGS cash transfer orders.....	305
10.3.1	Standard RTGS settlement.....	305
10.3.1.1	Process floor and ceiling	309
10.3.1.2	Process automated liquidity transfer	311
10.3.2	Process RTGS till/reject time instructions	312
10.3.3	Ancillary system payment settlement	314
10.3.3.1	Send ancillary system transfer initiation	314
10.3.3.2	Initiate ancillary system settlement for procedure A and B	316
10.3.3.3	Perform settlement of settlement procedure B.....	318
10.3.3.4	Reverse previously settled debits.....	320
10.3.3.5	Send ancillary system transfer confirmations.....	322
10.3.3.6	Notify guarantee fund mechanism initiation	324
10.3.3.7	Trigger guarantee fund mechanism	326
10.3.3.8	Terminate ancillary system processing	328
10.3.3.9	Execute start of procedure for ancillary system settlement procedures C and D.....	330
10.3.3.10	Execute start of cycle for settlement procedure C and D.....	332
10.3.3.11	Execute end of cycle for settlement procedure C and D.....	334
10.3.3.12	End of procedure for ancillary system settlement procedure C and D..	336
10.3.3.13	Execute settlement in ancillary system settlement procedure C.....	338
10.4	RTGS EoD processing.....	340
10.4.1	Reject payments (EoD).....	340
10.5	Revalidate warehoused payments at SoD	342

10.6	Reference data management	344
10.6.1	Maintain local reference data object	344
10.6.1.1	Maintain reservation	344
10.6.1.2	Maintain limit.....	346
10.7	Information services.....	348
10.7.1	Execute query	348
10.7.2	Receive report.....	351
10.7.3	Receive system notification	352
11	Dialogues and processes	353
11.1	Dialogues and processes between CRDM and CRDM Actor.....	353
11.1.1	A2A Common reference data maintenance and query process.....	353
11.1.1.1	Reference data maintenance process.....	353
11.1.1.1.1	Reference data objects	354
11.1.1.2	Common reference data query.....	355
11.1.1.2.1	Reference data query message coverage	357
11.1.2	DMT file upload.....	358
11.1.2.1	Activity diagram	358
11.1.2.1.1	Upload DMT file.....	359
11.1.2.1.2	DMT file validation	359
11.1.2.1.3	DMT file release	359
11.1.2.1.4	DMT file processing.....	359
11.1.2.1.5	DMT file results provisioning	360
11.1.2.1.6	Download DMT file results	360
11.2	Dialogues and processes between ESMIG and participant	361
11.2.1	Communication processing	361
11.2.1.1	Introduction.....	361
11.2.1.2	Schema validation	361
11.2.1.3	Technical message validation	361
11.2.1.4	Inbound and Outbound messages	361
11.2.1.4.1	Inbound messages	361
11.2.1.4.2	Outbound Messages	361
11.2.1.4.3	ReceiptAcknowledgement (admi.007.001.01)	361
11.3	Dialogues and processes with data warehouse	362
11.4	Dialogues and processes with billing.....	362
	Part III - Catalogue of messages	363

12 Messages – introduction	363
13 Messages - general information	366
13.1 Message validation	366
13.1.1 Structure of ISO 20022 messages.....	366
13.1.2 RTGS-specific schema customisation	368
13.1.3 XML character set.....	370
13.1.3.1 Schema validation	371
13.1.3.1.1 Business validation.....	373
13.2 Communication infrastructure	374
13.2.1 Envelope messages	374
13.2.1.1 Business Application Header	374
13.2.1.2 Business File Header	376
13.2.1.3 Digital Signature managed within the business layer.....	378
13.2.1.4 Time zones.....	378
13.2.1.5 Outbound traffic exceeding given size limitations	379
13.2.1.6 Re-sending of messages.....	380
14 List of messages	381
14.1 Account management (acmt).....	384
14.1.1 AccountQueryList (acmt.025)	384
14.1.1.1 Overview and scope of the message	384
14.1.1.2 Schema.....	384
14.1.1.3 The message in business context	385
14.1.2 AccountListReport (acmt.026)	386
14.1.2.1 Overview and scope of the message	386
14.1.2.2 Schema.....	386
14.1.2.3 The message in business context	387
14.2 Administration (admi)	389
14.2.1 ReportQueryRequest (admi.005).....	389
14.2.1.1 Overview and scope of the message	389
14.2.1.2 Schema.....	389
14.2.1.3 The message in business context	390
14.2.2 ResendRequest (admi.006).....	390
14.2.2.1 Overview and scope of the message	390
14.2.2.2 Schema.....	391
14.2.2.3 The message in business context	391
14.2.3 ReceiptAcknowledgement (admi.007)	391

14.2.3.1	Overview and scope of the message	391
14.2.3.2	Schema.....	392
14.2.3.3	The message in business context	393
14.3	Cash management (camt)	394
14.3.1	GetAccount (camt.003).....	394
14.3.1.1	Overview and scope of the message	394
14.3.1.2	Schema.....	394
14.3.1.3	The message in business context	395
14.3.2	ReturnAccount (camt.004).....	397
14.3.2.1	Overview and scope of the message	397
14.3.2.2	Schema.....	398
14.3.2.3	The message in business context	398
14.3.3	GetTransaction (camt.005)	421
14.3.3.1	Overview and scope of the message	421
14.3.3.2	Schema.....	422
14.3.3.3	The message in business context	423
14.3.4	ReturnTransaction (camt.006).....	431
14.3.4.1	Overview and scope of the message	431
14.3.4.2	Schema.....	431
14.3.4.3	The message in business context	432
14.3.5	ModifyTransaction (camt.007)	437
14.3.5.1	Overview and scope of the message	437
14.3.5.2	Schema.....	438
14.3.5.3	The message in business context	438
14.3.6	GetLimit (camt.009)	440
14.3.6.1	Overview and scope of the message	440
14.3.6.2	Schema.....	440
14.3.6.3	The message in business context	441
14.3.7	ReturnLimit (camt.010)	442
14.3.7.1	Overview and scope of the message	442
14.3.7.2	Schema.....	442
14.3.7.3	The message in business context	443
14.3.8	ModifyLimit (camt.011)	445
14.3.8.1	Overview and scope of the message	445
14.3.8.2	Schema.....	445
14.3.8.3	The message in business context	446
14.3.9	DeleteLimit (camt.012).....	448
14.3.9.1	Overview and scope of the message	448
14.3.9.2	Schema.....	449

14.3.9.3	The message in business context	449
14.3.10	GetBusinessDayInformation (camt.018).....	451
14.3.10.1	Overview and scope of the message	451
14.3.10.2	Schema.....	452
14.3.10.3	The message in business context	452
14.3.11	ReturnBusinessDayInformation (camt.019).....	453
14.3.11.1	Overview and scope of the message	453
14.3.11.2	Schema.....	454
14.3.11.3	The message in business context	455
14.3.12	ReturnGeneralBusinessInformation (camt.021)	458
14.3.12.1	Overview and scope of the message	458
14.3.12.2	Schema.....	458
14.3.12.3	The message in business context	459
14.3.13	ModifyStandingOrder (camt.024).....	469
14.3.13.1	Overview and scope of the message	469
14.3.13.2	Schema.....	470
14.3.13.3	The message in business context	470
14.3.14	Receipt (camt.025).....	474
14.3.14.1	Overview and scope of the message	474
14.3.14.2	Schema.....	475
14.3.14.3	The message in business context	476
14.3.15	ResolutionOfInvestigation (camt.029)	484
14.3.15.1	Overview and scope of the message	484
14.3.15.2	Schema.....	485
14.3.15.3	The message in business context	485
14.3.16	GetReservation (camt.046).....	487
14.3.16.1	Overview and scope of the message	487
14.3.16.2	Schema.....	488
14.3.16.3	The message in business context	488
14.3.17	ReturnReservation (camt.047).....	489
14.3.17.1	Overview and scope of the message	489
14.3.17.2	Schema.....	489
14.3.17.3	The message in business context	490
14.3.18	ModifyReservation (camt.048).....	492
14.3.18.1	Overview and scope of the message	492
14.3.18.2	Schema.....	492
14.3.18.3	The message in business context	493
14.3.19	DeleteReservation (camt.049)	494
14.3.19.1	Overview and scope of the message	494
14.3.19.2	Schema.....	495

14.3.19.3	The message in business context	496
14.3.20	LiquidityCreditTransfer (camt.050)	497
14.3.20.1	Overview and scope of the message	497
14.3.20.2	Schema.....	497
14.3.20.3	The message in business context	498
14.3.21	BankToCustomerStatement (camt.053)	505
14.3.21.1	Overview and scope of the message	505
14.3.21.2	Schema.....	506
14.3.21.3	The message in business context	507
14.3.22	BankToCustomerDebitCreditNotification (camt.054).....	522
14.3.22.1	Overview and scope of the message	522
14.3.22.2	Schema.....	524
14.3.22.3	The message in business context	525
14.3.23	FIToFIPaymentCancellationRequest (camt.056)	536
14.3.23.1	Overview and scope of the message	536
14.3.23.2	Schema.....	537
14.3.23.3	The message in business context	538
14.3.24	GetStandingOrder (camt.069)	542
14.3.24.1	Overview and scope of the message	542
14.3.24.2	Schema.....	542
14.3.24.3	The message in business context	542
14.3.25	ReturnStandingOrder (camt.070)	543
14.3.25.1	Overview and scope of the message	543
14.3.25.2	Schema.....	544
14.3.25.3	The message in business context	545
14.3.26	DeleteStandingOrder (camt.071).....	550
14.3.26.1	Overview and scope of the message	550
14.3.26.2	Schema.....	550
14.3.26.3	The message in business context	551
14.3.27	BillingReportRequest (camt.076).....	552
14.3.27.1	Overview and scope of the message	552
14.3.27.2	Schema.....	552
14.3.27.3	The message in business context	552
14.3.28	BillingReport (camt.077)	552
14.3.28.1	The message in business context	552
14.3.28.2	Schema.....	552
14.3.28.3	The message in business context	552
14.3.29	AuditTrailQuery (camt.097).....	552
14.3.29.1	Overview and scope of the message	552
14.3.29.2	Schema	553

14.3.29.3	The message in business context	553
14.3.30	AuditTrailReport (camt.098).....	554
14.3.30.1	Overview and scope of the message	554
14.3.30.2	Schema.....	555
14.3.30.3	The message in business context	556
14.3.31	DirectDebitMandateQuery (camt.099)	558
14.3.31.1	Overview and scope of the message	558
14.3.31.2	Schema.....	558
14.3.31.3	The message in business context	559
14.3.32	DirectDebitMandateReport(camt.100)	560
14.3.32.1	Overview and scope of the message	560
14.3.32.2	Schema.....	560
14.3.32.3	The message in business context	561
14.4	Headers (head)	564
14.4.1	BusinessApplicationHeader (head.001)	564
14.4.1.1	Overview and scope of the message	564
14.4.1.2	Schema.....	564
14.4.1.3	The message in business context	565
14.4.2	BusinessFileHeader (head.002)	566
14.4.2.1	Overview and scope of the message	566
14.4.2.2	Schema.....	566
14.4.2.3	The message in business context	567
14.5	Payments clearing and settlement (pacs).....	568
14.5.1	PaymentStatusReport (pacs.002).....	568
14.5.1.1	Overview and scope of the message	568
14.5.1.2	Schema.....	568
14.5.1.3	The message in business context	569
14.5.2	PaymentReturn (pacs.004)	571
14.5.2.1	Overview and scope of the message	571
14.5.2.2	Schema.....	572
14.5.2.3	The message in business context	572
14.5.3	CustomerCreditTransfer (pacs.008)	577
14.5.3.1	Overview and scope of the message	577
14.5.3.2	Schema.....	577
14.5.3.3	The message in business context	579
14.5.4	FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009).....	585
14.5.4.1	Overview and scope of the message	585
14.5.4.2	Schema.....	586
14.5.4.3	The message in business context	587

14.5.5	FinancialInstitutionDirectDebit (pacs.010)	603
14.5.5.1	Overview and scope of the message	603
14.5.5.2	Schema.....	604
14.5.5.3	The message in business context	605
14.6	Payment initiation (pain)	610
14.6.1	ASTransferNotice (pain.998)	610
14.6.1.1	Overview and scope of the message	610
14.6.1.2	Schema.....	610
14.6.1.3	The message in business context	611
14.6.2	ASInitiationStatus (pain.998)	620
14.6.2.1	Overview and scope of the message	620
14.6.2.2	Schema.....	621
14.6.2.3	The message in business context	622
14.6.3	ASTransferInitiation (pain.998)	633
14.6.3.1	Overview and scope of the message	633
14.6.3.2	Schema.....	633
14.6.3.3	The message in business context	634
14.7	Reference data (reda).....	651
14.7.1	PartyQuery (reda.015)	651
14.7.1.1	Overview and scope of the message	651
14.7.1.2	Schema.....	651
14.7.1.3	The message in business context	652
14.7.2	PartyReport (reda.017)	652
14.7.2.1	Overview and scope of the message	652
14.7.2.2	Schema.....	653
14.7.2.3	The message in business context	654
14.7.3	CashAccountAuditTrailQuery (reda.039).....	655
14.7.3.1	Overview and scope of the message	655
14.7.3.2	Schema.....	656
14.7.3.3	The message in business context	656
14.7.4	CashAccountAuditTrailReport (reda.040).....	657
14.7.4.1	Overview and scope of the message	657
14.7.4.2	Schema.....	657
14.7.4.3	The message in business context	658
14.7.5	PartyAuditTrailQuery (reda.042).....	661
14.7.5.1	Overview and scope of the message	661
14.7.5.2	Schema.....	661
14.7.5.3	The message in business context	661
14.7.6	PartyAuditTrailReport (reda.043).....	662

14.7.6.1	Overview and scope of the message	662
14.7.6.2	Schema.....	663
14.7.6.3	The message in business context	664
14.7.7	CalendarQuery(reda.064).....	666
14.7.7.1	Overview and scope of the message	666
14.7.7.2	Schema.....	666
14.7.7.3	The message in business context	667
14.7.8	CalendarReport(reda.065).....	667
14.7.8.1	Overview and scope of the message	667
14.7.8.2	Schema.....	667
14.7.8.3	The message in business context	668
Part IV - Appendixes		670
15 Index and digital signature.....		670
15.1	Index of business rules and error codes.....	670
15.2	Digital signature on business layer	730
15.3	Mechanism and introduction for signature constructions	730
15.4	Use of XML and canonicalisation algorithm.....	730
15.5	Message type 1: file with multiple ISO 20022 messages	730
15.6	Message type 2: single ISO 20022 message	736
15.7	ESMIG digital signature services usage of “ds:object”, attribute ID of the “signature” and “keyinfo”, anchor of trust.....	740
16 Glossary 743		
17 List of abbreviations		765

List of figures

Figure 1 - Structure of the RTGS UDFS.....	31
Figure 2 - Connectivity (U2A/A2A)	35
Figure 3 - Technical sender authentication	38
Figure 4 - Routing A.....	44
Figure 5 - Routing B.....	44
Figure 6 - Inbound routing	45
Figure 7 - Outbound routing.....	47
Figure 8 - Linked default MCA DCA	54
Figure 9 - Floor/ceiling breached on MCA1	55
Figure 10 - Settlement windows	75
Figure 11 - pacs.008 – CustomerCreditTransfer / pacs.009 - FinancialInstitutionCreditTransfer.....	95
Figure 12 - pacs.004 - PaymentReturn	97
Figure 13 - pacs.010 FinancialInstitutionDirectDebit.....	99
Figure 14 - pacs.009 – FinancialInstitutionCreditTransfer (sent from a multi-addressee access participant).....	101
Figure 15 - pacs.009 FinancialInstitutionCreditTransfer (submitted by ancillary system).....	103
Figure 16 - pacs.009 FinancialInstitutionCreditTransfer (submitted by ancillary system via file).....	105
Figure 17 - pacs.009 – FinancialInstitutionCreditTransfer – mandated payment (codeword: MANP)	107
Figure 18 - pacs.008/009/010/004 technical validation error	110
Figure 19 - pacs.008/009/010/004 business validation error	112
Figure 20 - camt.007 make amendment of payment (positive).....	115
Figure 21 - camt.056 revocation of payment (positive)	119
Figure 22 - camt.056 FIToFIPaymentCancellationRequest/camt.029 ResolutionOfInvestigation - positive case	121
Figure 23 - camt.056 FIToFIPaymentCancellationRequest / camt.029 ResolutionOfInvestigation - negative case.....	123
Figure 24 - Generic account constellation for an ancillary system settlement bank	142
Figure 25 - Flow standard multilateral settlement	145
Figure 26 - Flow simultaneous multilateral settlement	149
Figure 27 - Flow settlement on dedicated liquidity accounts (ancillary system settlement procedure C)	157
Figure 28 - Flow real-time – start of procedure	164
Figure 29 - Instruction using pacs.009	171
Figure 30 - Flow information period.....	174

Figure 31 - Flow guarantee fund	177
Figure 32 - Liquidity transfer order between two RTGS DCAs in the RTGS component.....	183
Figure 33 - Liquidity transfer from a RTGS DCA to a CLM MCA	185
Figure 34 - Liquidity transfer from an RTGS DCA to a DCA in the T2S Service.....	187
Figure 35 - Liquidity transfer from DCA of the TIPS Service to an RTGS DCA	189
Figure 36 - Reservation management.....	195
Figure 37 - Breach of floor/ceiling threshold - notification	210
Figure 38 - Breach of floor/ceiling threshold – automated liquidity transfer	211
Figure 39 - File state diagram.....	213
Figure 40 - Inbound RTGS message state diagram.....	214
Figure 41 - Status transition diagram A and B.....	217
Figure 42 - Status transition diagram C.....	218
Figure 43 - Status transition diagram D.....	219
Figure 44 - Cash transfer state diagram.....	223
Figure 45 - Task queue order state diagram	226
Figure 46 - Common reference data objects and the hierarchical party model	248
Figure 47 - Data scopes	249
Figure 48 - Access rights configuration steps	251
Figure 49 - Access rights configuration process (A).....	260
Figure 50 - Access rights configuration process (B).....	261
Figure 51 - Example - configuration of access rights at party level by the operator	262
Figure 52 - Configuration of access rights at user level	263
Figure 53 - Example - archiving and purging after deletion of a common reference data object.....	273
Figure 54 - Lifecycle of common reference data objects with unlimited validity period	275
Figure 55 - Lifecycle of common reference data objects with limited validity period	276
Figure 56 - UML conventions – example I.....	293
Figure 57 - UML conventions- example II.....	294
Figure 58 - UML conventions - example III.....	295
Figure 59 - Send file	296
Figure 60 - Send RTGS cash transfer order.....	298
Figure 61 - Revoke/cancel payment.....	300
Figure 62 - Amend payment	302
Figure 63 - Execute RTGS standing order	304

Figure 64 - Standard RTGS settlement	307
Figure 65 - Floor and ceiling processing	310
Figure 66 - Process automated RTGS liquidity transfer order	312
Figure 67 - Process till/reject time instructions	313
Figure 68 - Send ancillary system transfer initiation.....	315
Figure 69 - Initiate ancillary system settlement procedure A and B	317
Figure 70 - Perform settlement procedure B	319
Figure 71 - Reverse debits	321
Figure 72 - Send ancillary system transfer confirmations	323
Figure 73 - Notify guarantee fund mechanism initiation	325
Figure 74 - Trigger guarantee fund mechanism	327
Figure 75 - Terminate ancillary system file processing	329
Figure 76 - Execute ancillary system settlement procedure C and D - start of procedure.....	331
Figure 77 - Execute settlement procedure C and D - start of cycle	333
Figure 78 - Execute settlement procedure C and D - end of cycle	335
Figure 79 - Execute settlement procedure C and D - end of procedure	337
Figure 80 - Execute settlement procedure C.....	339
Figure 81 - Reject payments (EoD)	341
Figure 82 - Revalidate warehoused payments at SoD.....	343
Figure 83 - Maintain reservation RTGS.....	345
Figure 84 - Maintain limit	347
Figure 85 - RTGS send query	349
Figure 86 - RTGS receive report	351
Figure 87 - RTGS receive system notification.....	352
Figure 88 - Common reference data maintenance process	354
Figure 89 - Common reference data query process.....	357
Figure 90 - DMT file upload process	359
Figure 91 - Business application header	375
Figure 92 - BAH extract	376
Figure 93 - Business file header	377
Figure 94 - Message type 1 signature example	732
Figure 95 - Message type 2 signature example	737
Figure 96 - Reference to the BAH (AppHdr).....	738

Figure 97 - Reference to the message (e.g. semt.013).....	738
---	-----

List of tables

Table 1 - Services and components	46
Table 2 - Setup of parties for RTGS	48
Table 3 - Party reference data attributes	51
Table 4 - Comparison of participation types	52
Table 5 - Reference data attributes	58
Table 6 - Direct debit mandate reference data attributes	59
Table 7 - Standing liquidity transfer order reference data attributes	60
Table 8 - Message subscription parameter types	63
Table 9 - Applicable parameter types for outgoing messages	63
Table 10 - Setup of groups for RTGS	65
Table 11 - Attributes of the RTGS directory	67
Table 12 - Attributes of the RTGS scheduled events	68
Table 13 - Attributes of the RTGS currency	69
Table 14 - Attributes of the duplicate check	69
Table 15 - Attributes of the warehoused payment period	70
Table 16 - Backup payments	70
Table 17 - Overview of the main settlement windows in RTGS	74
Table 18 - Overview of payments in the RTGS component	79
Table 19 - Classification of priorities	80
Table 20 - Payments with a set execution time indicators	82
Table 21 - CLS backup payments	85
Table 22 - EURO1 collateral account backup payments	86
Table 23 - EURO1 liquidity bridge backup payment	87
Table 24 - Backup liquidity redistribution payments	88
Table 25 - General procedure for generating backup payments	88
Table 26 - Features to be used for different payment messages	92
Table 27 - Table of possible payment types	94
Table 28 - Payment messaging on the basis of pacs.008/pacs.009	96
Table 29 - Payment messaging on the basis of pacs.004	98
Table 30 - Payment messaging on the basis of pacs.010	100
Table 31 - Payment messaging on the basis of pacs.009	102
Table 32 - Payment messaging on the basis of pacs.009 (submitted by an ancillary system)	104

Table 33 - Payment messaging on the basis of pacs.009 (submitted by an ancillary system via file).....	106
Table 34 - Payment messaging on the basis of pacs.009 (mandated payment)	108
Table 35 - Technical validation failure	111
Table 36 - Business validation failure.....	113
Table 37 - Options for changing the parameters of payments and authorisations.....	113
Table 38 - Amendment of payments	116
Table 39 - Effects of changing the priority	116
Table 40 - Effects of re-ordering the queued payments	117
Table 41 - Effects of changing the execution time	118
Table 42 - Successful revocation of a queued payment	120
Table 43 - Cancellation request for already settled payments – positive case	122
Table 44 - Cancellation request for already settled payments – negative case.....	124
Table 45 - Effective settlement order.....	125
Table 46 - Payments taken into account in the entry disposition	127
Table 47 - Control options for comprehensive queue management	129
Table 48 - Possibilities for changing priorities	129
Table 49 - Effect of changed priority.....	130
Table 50 - Effect of changing the order of queued payments	131
Table 51 - Effect of changing the execution time	132
Table 52 - Possible events for queue resolution	132
Table 53 - Main characteristics of algorithm “partial optimisation”	134
Table 54 - Main characteristics of algorithm “multiple optimisation” – Part 1	135
Table 55 - Main characteristics of algorithm “multiple optimisation” – Part 2.....	135
Table 56 - Main characteristics of algorithm “partial optimisation with ancillary system”	136
Table 57 - Main characteristics of algorithm “optimisation on sub-accounts”	139
Table 58 - Settlement procedures	141
Table 59 - Account types and their ownership	143
Table 60 - Process flow for standard multilateral settlement.....	147
Table 61 - Process flow for standard multilateral settlement.....	151
Table 62 - Accounting.....	152
Table 63 - Amounts taken into account	156
Table 64 - Start of procedure and liquidity provision	161
Table 65 - Amounts taken into account	163

Table 66 - Start of procedure and liquidity provision	167
Table 67 - Peculiarities for pacs.009 sent by ancillary system.....	170
Table 68 - Process description for using plain payments.....	172
Table 69 - Usability of optional connected mechanism per ancillary system processing procedure	173
Table 70 - Process flow information period with disagreement.....	175
Table 71 - Process description	178
Table 72 - Effect of reservations on the available liquidity	179
Table 73 - Liquidity transfer types	181
Table 74 - Execution of liquidity transfers.....	182
Table 75 - Process description	184
Table 76 - Process description	186
Table 77 - Process description	188
Table 78 - Process description	190
Table 79 - Create one-time liquidity reservation with immediate effect.....	196
Table 80 - Modify one-time liquidity reservations with immediate effect	197
Table 81 - “Resetting to zero” of a reservation.....	198
Table 82 - Effect of reservations for payment procession.....	199
Table 83 - Usage of urgent and high reserve – numeric example	200
Table 84 - Limit management – positive validation	203
Table 85 - Limit management – negative validation.....	204
Table 86 - Effects of limits	205
Table 87 - Processing in case of bilateral limit	206
Table 88 - Processing in case of multilateral limits	207
Table 89 - RTGS file status	214
Table 90 - RTGS message status	215
Table 91 - Ancillary system batch message status	221
Table 92 - List of status values per ancillary system procedure.....	222
Table 93 - Cash transfer status	225
Table 94 - Task queue order status.....	227
Table 95 - Report “Statement of accounts”	228
Table 96 - CRDM parameter synthesis	230
Table 97 - Initiating queries for RTGS	232
Table 98 - Access rights management.....	237

Table 99 - Party data management	238
Table 100 - Cash account data management	241
Table 101 - Message subscription configuration	242
Table 102 - Report configuration	242
Table 103 - Reference data queries	246
Table 104 - TIPS functions	247
Table 105 - Other.....	247
Table 106 - User privileges (data scope).....	249
Table 107 - Privilege assignment options.....	252
Table 108 - Assignment of privileges to roles.....	252
Table 109 - Assignment of privileges to users	254
Table 110 - Assignment of privileges to parties	255
Table 111 - Cascade process when revoking privileges	256
Table 112 - Assignment of privilege to party and users	257
Table 113 - Cascade process when revoking roles	258
Table 114 - Assignment of privileges to roles.....	259
Table 115 - Assignment of roles to users	259
Table 116 - Assignment of privileges to users	259
Table 117 - Message subscription parameter types	264
Table 118 - Definition of a new message subscription rule set.....	265
Table 119 - Definition of the rules for a new message subscription rule set.....	265
Table 120 - Common reference data objects	267
Table 121 - Management of reference data objects in DMT	269
Table 122 - Management of reference data objects in A2A mode.....	269
Table 123 - Common reference data objects with unlimited validity period	270
Table 124 - Common reference data objects with limited validity period	273
Table 125 - CRDM data segregation per service/component	282
Table 126 - Tasks of the CBs	288
Table 127 - Summary of queries and actions in U2A and A2A mode for CBs in RTGS.....	289
Table 128 - List of CB specific queries	291
Table 129 - Messages sent by the submitting actor to RTGS component.....	298
Table 130 - Process description revoke/cancel payment.....	301
Table 131 - Process description amend payment	303

Table 132 - Process description execute RTGS standing order	305
Table 133 - Message sent after settlement	308
Table 134 - A2A messages for query processing	351
Table 135 - A2A messages for receiving reports	352
Table 136 - Receive system notification	352
Table 137 - Common reference data maintenance process	354
Table 138 - CRDM messages	355
Table 139 - Common reference data query process	357
Table 140 - Common reference data query messages	358
Table 141 - DMT files specifications	361
Table 142 - List of messages	384
Table 143 - AccountQueryList (acmt.025) – usage case Cash account reference data query	386
Table 144 - AccountListReport (acmt.026) – usage case Cash account reference data query response....	388
Table 145 - AccountListReport (acmt.026) – usage case Error	388
Table 146 - ReportQueryRequest (admi.005) – usage case Account Statement Query	390
Table 147 - ReceiptAcknowledgement (admi.007) – usage case Negative Receipt Acknowledgement.....	393
Table 148 - GetAccount (camt.003) – usage case AccountBalanceQuery	395
Table 149 - ReturnAccount camt.004 – usage case Account Balance Query (Data Response)	401
Table 150 - ReturnAccount (camt.004) – usage case Account Balance Query (Error Response)	402
Table 151 - ReturnAccount (camt.004) – usage case Floor Notification	404
Table 152 - ReturnAccount (camt.004) – usage case Ceiling Notification	406
Table 153 - ReturnAccount (camt.004) - usage case Notification of Credit to all Sub-Accounts	409
Table 154 - ReturnAccount (camt.004) – usage case Notification of Liquidity Blocked on Sub-Accounts ...	412
Table 155 - ReturnAccount (camt.004) – usage case Notification of Execution of Stored Immediate Liquidity Transfers	415
Table 156 - ReturnAccount (camt.004) – usage case Notification of Liquidity Re-transfer	418
Table 157 - ReturnAccount (camt.004) – usage case Notification of Global Amount on Liquidity Account .	421
Table 158 - GetTransaction (camt.005) – usage case Payment Query	430
Table 159 - ReturnTransaction (camt.006) – usage case Payment Query (Data Response)	436
Table 160 - ReturnTransaction (camt.006) – usage case Payment Query (Error Response)	437
Table 161 - ModifyTransaction (camt.007) – usage case Amend Payment	439
Table 162 - GetLimit (camt.009) – usage case Current Limits Query	441
Table 163 - ReturnLimit (camt.010) – usage case Current Limits Query (Data Response)	444

Table 164 - ReturnLimit (camt.010) – usage case Current Limits Query (Error response)	445
Table 165 - ModifyLimit (camt.011) – usage case CRDM Update Standing Order for Limit	447
Table 166 - ModifyLimit (camt.011) – usage case RTGS Update Limit	448
Table 167 - DeleteLimit (camt.012) – usage case CRDM Delete Standing Order for Limit.....	450
Table 168 - DeleteLimit (camt.012) – usage case RTGS Delete Limit	451
Table 169 - GetBusinessDayInformation (camt.018) – usage case Event Query.....	453
Table 170 - GetBusinessDayInformation (camt.018) – usage case System Time Query.....	453
Table 171 - ReturnBusinessDayInformation (camt.019) – usage case System Time Query (Data Response)	455
Table 172 - ReturnBusinessDayInformation (camt.019) – usage case System Time Query (Error Response).....	456
Table 173 - ReturnBusinessDayInformation (camt.019) – Event Query (Data Response).....	456
Table 174 - ReturnBusinessDayInformation (camt.019) – usage case Event Query (Error Response).....	457
Table 175 - ReturnBusinessDayInformation (camt.019) – usage case System Notification.....	457
Table 176 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C – Start Procedure	460
Table 177 - ReturnGeneralBusinessInformation (camt.021) – usage case AS Procedure C – Notify Ancillary System About Start of Procedure.....	461
Table 178 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C – End of Procedure	462
Table 179 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C Cycle – Start of Cycle	463
Table 180 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C Cycle – Notify Ancillary System About Start of Cycle.....	464
Table 181 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C Cycle – End of Cycle	465
Table 182 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure D – Notify Ancillary System About Start of Procedure	466
Table 183 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure D Cycle – Start of Cycle	467
Table 184 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure D Cycle – Notify Ancillary System About Start of Cycle.....	468
Table 185 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure D Cycle – End of Cycle	469
Table 186 - ModifyStandingOrder (camt.024) – usage case RTGS modify standing order.....	471
Table 187 - ModifyStandingOrder (camt.024) – usage case ASI6 RTGS Modify standing order.....	473

Table 188 - ModifyStandingOrder (camt.024) – usage case CLM modify standing order	474
Table 189 - Receipt (camt.025) – usage category case Settlement - Rejected.....	477
Table 190 - Receipt (camt.025) – usage category case Settlement - Settled.....	477
Table 191 - Receipt (camt.025) – usage category case Settlement - Unsettled.....	478
Table 192 - Receipt (camt.025) – usage category case RTGS Status - Rejected.....	479
Table 193 - Receipt (camt.025) usage category case RTGS Status - Confirmed	479
Table 194 - Receipt (camt.025) usage category case RTGS Status – Validation Error	480
Table 195 - Receipt (camt.025) usage category case Liquidity Management - Approved	481
Table 196 - Receipt (camt.025) usage category case CRDM - Rejected	482
Table 197 - Receipt (camt.025) usage category case CRDM - Completed	482
Table 198 - Receipt (camt.025) usage category case CRDM - Queued.....	483
Table 199 - Receipt (camt.025) usage category case Ancillary System Processing – Invoke Guarantee Processing.....	484
Table 200 - ResolutionOfInvestigation (camt.029) – usage case Rejection of Payment Cancellation Request.....	487
Table 201 - GetReservation (camt.046) – usage case Current Reservations Query	489
Table 202 - ReturnReservation (camt.047) – usage case Current Reservations Query (Data Response) ..	491
Table 203 - ReturnReservation (camt.047) – usage case Current Reservations Query (Error Response) ..	492
Table 204 - ModifyReservation (camt.048) – usage case Modify Reservation Request	494
Table 205 - ModifyReservation (camt.048) – usage case Modify Standing Order for Reservation	494
Table 206 - DeleteReservation (camt.049) - usage case Delete Reservation Request	496
Table 207 - DeleteReservation (camt.049) - usage case Delete Standing Order for Reservation	496
Table 208 - LiquidityCreditTransfer (camt.050) – usage case Payment Order Message	499
Table 209 - LiquidityCreditTransfer (camt.050) – usage case Inter-Service Liquidity Transfer Order (Floor Processing).....	500
Table 210 - LiquidityCreditTransfer (camt.050) – usage case Inter-Service Liquidity Transfer Order (Ceiling Processing)	501
Table 211 - LiquidityCreditTransfer (camt.050) – usage case Automated Inter-Service Liquidity Transfer Order.....	502
Table 212 - LiquidityCreditTransfer (camt.050) – usage case Liquidity Adjustment (Ancillary System Settlement Procedure C).....	504
Table 213 - LiquidityCreditTransfer (camt.050) – usage case Liquidity Adjustment (Ancillary System Settlement Procedure D).....	505
Table 214 - BankToCustomerStatement (camt.053) – usage case Query Response Message for Business Data	512

Table 215 - BankToCustomerStatement (camt.053) – usage case Statement of Accounts.....	517
Table 216 - BankToCustomerStatement (camt.053) – usage case CB General Ledger.....	522
Table 217 - BankToCustomerDebitCreditNotification (camt.054) – usage case Payment Settlement Notification (Intra-Service Liquidity Transfer)	527
Table 218 - BankToCustomerDebitCreditNotification (camt.054) – usage case Liquidity Transfer Settlement Notification.....	531
Table 219 - BankToCustomerDebitCreditNotification (camt.054) – usage case Ancillary System Processing	534
Table 220 - FIToFIPaymentCancellationRequest (camt.056) – usage case Cancel Payment.....	541
Table 221 - GetStandingOrder (camt.069) – usage case Get Standing order details	543
Table 222 - ReturnStandingOrder (camt.070) – usage case RTGS return standing order details	546
Table 223 - ReturnStandingOrder (camt.070) – usage case ASI6 RTGS return standing order details	548
Table 224 - ReturnStandingOrder (camt.070) – usage case CLM Return standing order details	549
Table 225 - ReturnStandingOrder (camt.070) – usage case Error	550
Table 226 - DeleteStandingOrder (camt.071) – usage case Delete standing order	551
Table 227 - AuditTrailQuery (camt.097) – usage case Audit Trail for RTGS Query	554
Table 228 - AuditTrailReport (camt.098) – usage case Audit Trail for RTGS Query (Data response)	557
Table 229 - AuditTrailReport (camt.098) – usage case Audit Trail for RTGS Query (Error response).....	558
Table 230 - DirectDebitMandateQuery (camt.099) – usage case Direct debit mandate query	560
Table 231 - DirectDebitMandateReport (camt.100) – usage case Direct debit mandate report.....	563
Table 232 - DirectDebitMandateReport (camt.100) – usage case Error.....	563
Table 233 - PaymentStatusReport (pacs.002) – usage case Payment Rejection Notification	570
Table 234 - PaymentStatusReport (pacs.002) – usage case Payment Settlement Notification	571
Table 235 - PaymentReturn (pacs.004) – usage case Payment Message.....	574
Table 236 - PaymentReturn (pacs.004) – usage case Payment Settlement Notification	577
Table 237 - CustomerCreditTransfer (pacs.008) – usage case Payment Message	580
Table 238 - CustomerCreditTransfer (pacs.008) – usage case Payment Settlement Notification.....	585
Table 239 - FinancialInstutionCreditTransfer(GEN and COV) (pacs.009) – usage case Settlement of an interbank payment	591
Table 240 - FinancialInstutionCreditTransfer (GEN and COV) (pacs.009) – usage case Settlement of an interbank customer cover payment.....	595
Table 241 - FinancialInstutionCreditTransfer (GEN and COV) (pacs.009) – usage case Settlement of ancillary system movement	599
Table 242 - FinancialInstutionCreditTransfer (GEN and COV) (pacs.009) – usage case Liquidity transfer to sub-account	603

Table 243 - FinancialInstitutionDirectDebit (pacs.010) – usage case Payment Message	606
Table 244 - FinancialInstitutionDirectDebit (pacs.010) – usage case Payment Settlement Notification	609
Table 245 - ASTransferNotice (pain.998) – usage case Procedure D – Notify ancillary system of Credit Liquidity Bookings.....	614
Table 246 - ASTransferNotice (pain.998) – usage case Procedure D – Notify ancillary system of Immediate Liquidity Transfer Order Issued by Settlement Bank	617
Table 247 - ASTransferNotice (pain.998) – usage case Procedure D – Notify ancillary system of Execution of Pending Liquidity Transfer Orders	620
Table 248 - ASInitiationStatus (pain.998) – usage category case Settlement - Settled	624
Table 249 - ASInitiationStatus (pain.998) – usage category case Settlement - Rejected	629
Table 250 - ASInitiationStatus (pain.998) – usage category case Settlement - Partial	632
Table 251 - ASTransferInitiation SendASTransferInitiation (pain.998) – usage case Send Ancillary System Transfer Initiation.....	637
Table 252 - ASTransferInitiation SendASTransferInitiation (pain.998) – usage case Procedure C – Liquidity Adjustment	641
Table 253 - ASTransferInitiation SendASTransferInitiation (pain.998) – usage case Procedure C – Instruct the Settlement Transactions	646
Table 254 - ASTransferInitiation SendASTransferInitiation (pain.998) – usage case Procedure D – Liquidity Adjustment	650
Table 255 - PartyQuery (reda.015) – usage case Party reference data query	652
Table 256 - PartyReport (read.017) – usage case Party reference data response	655
Table 257 - PartyReport (read.017) – usage case Error	655
Table 258 - CashAccountAuditTrailQuery (camt.039) – usage case Cash account audit trail query	657
Table 259 - CashAccountAuditTrailReport (reda.040) – usage case Cash account audit trail report	660
Table 260 - CashAccountAuditTrailReport (reda.040) – usage case Error.....	660
Table 261 - PartyAuditTrailQuery (reda.042) – usage case Party audit trail query.....	662
Table 262 - PartyAuditTrailReport (reda.043) – usage case Party audit trail report	665
Table 263 - PartyAuditTrailReport (reda.043) – usage case Error.....	665
Table 264 - CalendarQuery (reda.064) – usage case Calendar query	667
Table 265 - CalendarReport (reda.065) – usage case Calendar report.....	669
Table 266 - CalendarReport (reda.065) – usage case Error.....	669
Table 267 - RTGS validation rules	709
Table 268 - CRDM validation rules.....	729
Table 269 - ESMIG validation rules.....	730

Introduction

This document describes the real-time gross settlement (RTGS) as a business component of T2 and RTGS Account Holders' interactions with other components and services. RTGS settles high value payments and processes transactions of ancillary systems on RTGS dedicated cash accounts (DCAs). The document is intended to guide RTGS Account Holders to the proper understanding of the RTGS component.

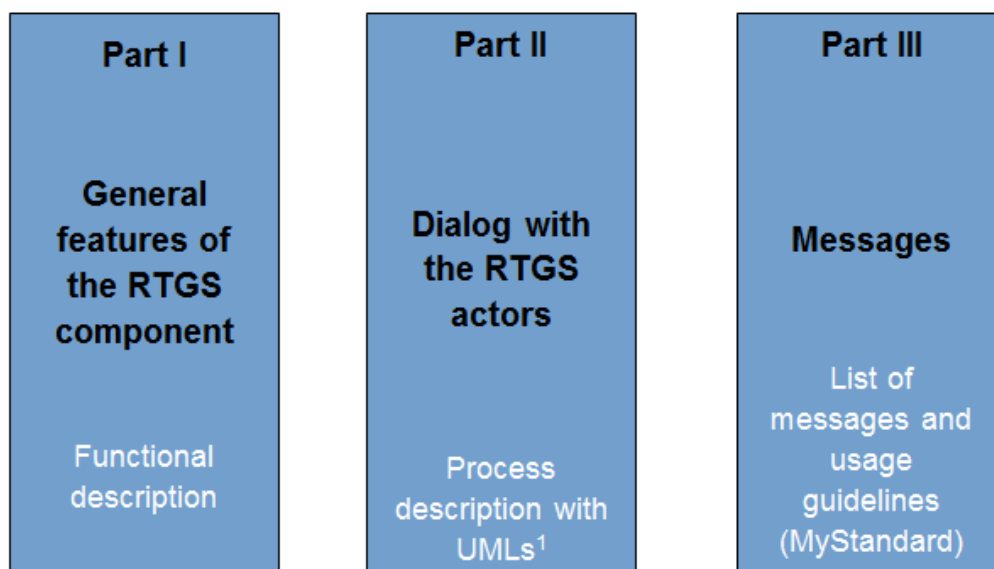
Therefore, the UDFS (user detailed functional specifications) document focuses on the provision of information to RTGS Account Holders to design and build the interface of their business application with RTGS (A2A/U2A). The UDFS RTGS is publicly available.

The document is divided into three parts:

- I The first part provides a full description of all the RTGS features and the related accounts and application processes, functional details concerning access to RTGS and connectivity, dependencies and interactions with other services/components, operations and support features. The background information provided in chapter [Overview of RTGS component](#) [▶ 33] supports the understanding of the RTGS component with its applications and the interaction with common components described in the following chapters. Afterwards it guides the reader through the RTGS features (i.e. participation and accounts). Moreover, it provides an overview of common components used by RTGS (e.g. Common Reference Data Management (CRDM), Data Warehouse (DHW)). The contingency services are explained in chapter [Contingency services](#) [▶ 286] and CB specific information is provided in chapter [Additional information for CBs](#) [▶ 288].
- I The second part provides process descriptions, which allow RTGS Account Holders to interact with RTGS via A2A and gives a functional overview of the U2A processes. This part aims at providing a comprehensive description of all processes being available in RTGS and which the user may instruct. Moreover, the related settlement processes are explained in detail. Furthermore the chapter [Dialogues and processes between CRDM and CRDM Actor](#) [▶ 353] describes the dialogue between CRDM and participants via A2A. Subsequently, also the interaction with ESMIG (Eurosystem Single Market Infrastructure Gateway) is outlined in chapter [Dialogues and processes between ESMIG and participant](#) [▶ 361].
- I The third part provides a detailed description of all XML messages RTGS Account Holders may use to interact in A2A mode with RTGS. The descriptions of the messages include all required elements according to the schema. Wherever a message or its fields are referenced throughout the document, only the reference name is used.

Reader's guide

The document is structured as to guide the readers through the steps of the whole (A2A) interaction and processing details focusing on different user needs, i.e. business experts, IT experts and message experts.



¹ UML = Unified Modelling Language

Figure 1 - Structure of the RTGS UDFS

Different readers may have different needs and priorities and may not need to read the whole book. For instance, business readers, interested mainly in organisational issues, may not wish to enter into the full details of each message description, but they might prefer going through a description of the business processes and the information flows between their own business application(s) and RTGS. On the other hand, technical readers involved in the specification and development of technical interfaces to RTGS may not be interested in the complete description of the features RTGS offers. They would probably search the necessary information to design and build the interface of the RTGS Account Holders' business application with RTGS. The following paragraphs show - with a couple of examples - how business and technical readers may follow different reading patterns in order to fulfil their needs.

All readers, whether business or technical, are invited to read the following UDFS sections, which are providing a minimum functional and technical background to the understanding of any other UDFS chapter:

- | [Overview of RTGS component](#) [▶ 33] which summarizes the RTGS features and functionalities;
- | [Access to RTGS](#) [▶ 35], which focuses on how to connect to RTGS including authentication and authorisation processes and explains the envisaged usage of access rights depending on the respective roles;
- | Parties and accounts, which provides a general description of the main reference data needed for RTGS and the accounts maintained in RTGS, specifying how they are used for the settlement of high value payments and ancillary system transactions (e.g. which parties and RTGS Account Holders are involved and how to set up accounts for different purposes including their usage);

-
- I [Contingency services](#) [▶ 286], which informs how to act in case of a defined contingency situation.

Business oriented perspective

In addition, a business reader may be interested in the way information is structured in RTGS. This user may want to follow the reading plan described below to find further details about the operations possible in RTGS:

- I [Business day](#) [▶ 72], where the business reader finds an overview of the RTGS schedule and respective processes;
- I [Business and features description](#) [▶ 79], which informs about the payments processing and settlement of payments and ancillary systems as well as the liquidity management and information management;
- I [Overview of used common components in RTGS component](#) [▶ 234] describes the common components used by RTGS and the interaction between RTGS and the used common components;
- I [Process cash transfer instruction](#) [▶ 297] to find a description of the processing of a payment instruction and useful information in order to understand the management of RTGS payment settlement;
- I Dialogue between CRDM and CRDM Actors wherein query information may be of relevance;
- I [Index of business rules and error codes](#) [▶ 670] includes the relevant codes to perform functional checks.

Technical oriented perspective

For the technical reader, the following reading plan may be of particular interest:

- I [Processes with RTGS](#) [▶ 292] respectively [Dialogues and processes between CRDM and CRDM Actor](#) [▶ 353], where an overview of the possible A2A dialogue with RTGS is defined. Each sub-chapter of this chapter describes the flows within, to and from RTGS. The reader can focus on the functionality of RTGS , analysing the procedures and main scenarios;
- I Part III - Catalogue of messages, where a detailed description of the content of a given XML message is provided.

Part I - General features of the RTGS component

1 Overview of RTGS component

The primary aim of the RTGS component is the settlement for real-time interbank and customer payments and ancillary system transactions. Any payment which should be processed real-time and in CB money is executed in the RTGS component.

RTGS offers a wide range of features to execute real-time payments and ancillary system transactions in an efficient manner (e.g. reservations for purpose, priorities and optimisation algorithms):

- I The RTGS component is multi-currency enabled, i.e. the settlement services support settlement in different currencies and according to their own calendars; neither the RTGS components nor the other common components of the T2 Services offer conversion between currencies.
- I The A2A communication between credit institutions and the RTGS component are based on the ISO 20022 (international organisation for standardisation) compliant messages.

CB operations (CBOs) are not processed in the RTGS component but in the newly introduced central liquidity management (CLM) component.

While the CLM holds the main cash accounts (MCAs) as the central source of liquidity, the RTGS provides DCA for the settlement of real-time interbank and customer payments and transactions with ancillary system. The available liquidity is transferred to the DCAs of RTGS; like all other DCAs, the RTGS DCA operates on cash-only-basis, e.g. the credit line that is on the MCA may be used to increase the liquidity on the DCA by transferring liquidity from MCA to DCA. A party may open more than one RTGS DCA for a dedicated purpose, depending on its business needs (e.g. for ancillary system transactions, for the payment business of a branch/entity). Furthermore, a participant may open an RTGS dedicated sub-account that is linked to one RTGS DCA. This RTGS sub-account may be dedicated to one ancillary system settlement procedure "settlement on dedicated liquidity accounts (interfaced)". RTGS Account Holders are responsible for their own liquidity management and the monitoring of the settlement processes; otherwise they may also grant access to another party to perform these tasks on its behalf.

RTGS makes use of the following common components:

- I ESMIG provides the central authentication, authorisation and user management features. It is network provider agnostic and thus offers participants the access to all T2 Services through the connection with a single certified network service provider. All network service providers will comply with the same communication interface specifications in application-to-application (A2A) mode (in store-and-forward and real-time communication protocol) and user-to-application (U2A) mode via GUI.
- I CRDM component offers features that allow authorised users to set up, maintain and query all reference data that TARGET Services share for their processing activities. CRDM ensures the consistency and integrity of all reference data, processing and relationships across services. Furthermore, it avoids duplication of reference data or redundant implementation of the same functions in multiple services.

Service-specific reference data objects (or functions) is set up and managed (or implemented) in the respective service. The access to all collected data allows to making use of a billing component as well as queries and reports.

- | DWH component provides the data for historical, statistical and regulatory reporting. It offers predefined queries and reports but also the possibility to design individual reports and queries. The DWH participants may access DWH in U2A mode. The data of previous business days are available in DWH as of the next business day. The business day management offers the schedule and calendar for all components and currencies. A schedule defines the structure of the business day in the T2 Services as well as the events per currency for which participants may configure event-based standing orders and regular reports. The calendar defines the days when a TARGET Service or a common component is opened and follows the defined business day schedule or contrary is closed. Each TARGET Service may have a different calendar per currency.
- | The billing component ensures the preparation and processing of invoices for the different T2 Services and common components. To do so, relevant information for each cash account have to be defined in CRDM (e.g. to whom the invoice is addressed to, which MCA is debited, etc.) and this information is then taken into account during the billing process. Further information on billing and the respective fees is defined in a pricing guide.
- | The legal archiving (LEA) component collects all information, which is subject to LEA requirements The information from TARGET Services and common components is stored in LEA in its original content and format after thirty calendar days and is accessible within its retention period of ten years.
- | The contingency service is used in events where business continuity is impossible or systemically important payments and/ or the settlement of ancillary systems need to be processed during the failover process. Contingency processing is a temporary means that aims at processing limited business only to avoid the creation of systemic risk. The operational tools are provided to the operational staff. Those tools have interfaces to all applications. They support the monitoring and controlling of the RTGS component.

2 Access to RTGS

2.1 Connectivity (U2A/A2A)

RTGS supports access to the service through two different channels: A2A and U2A.

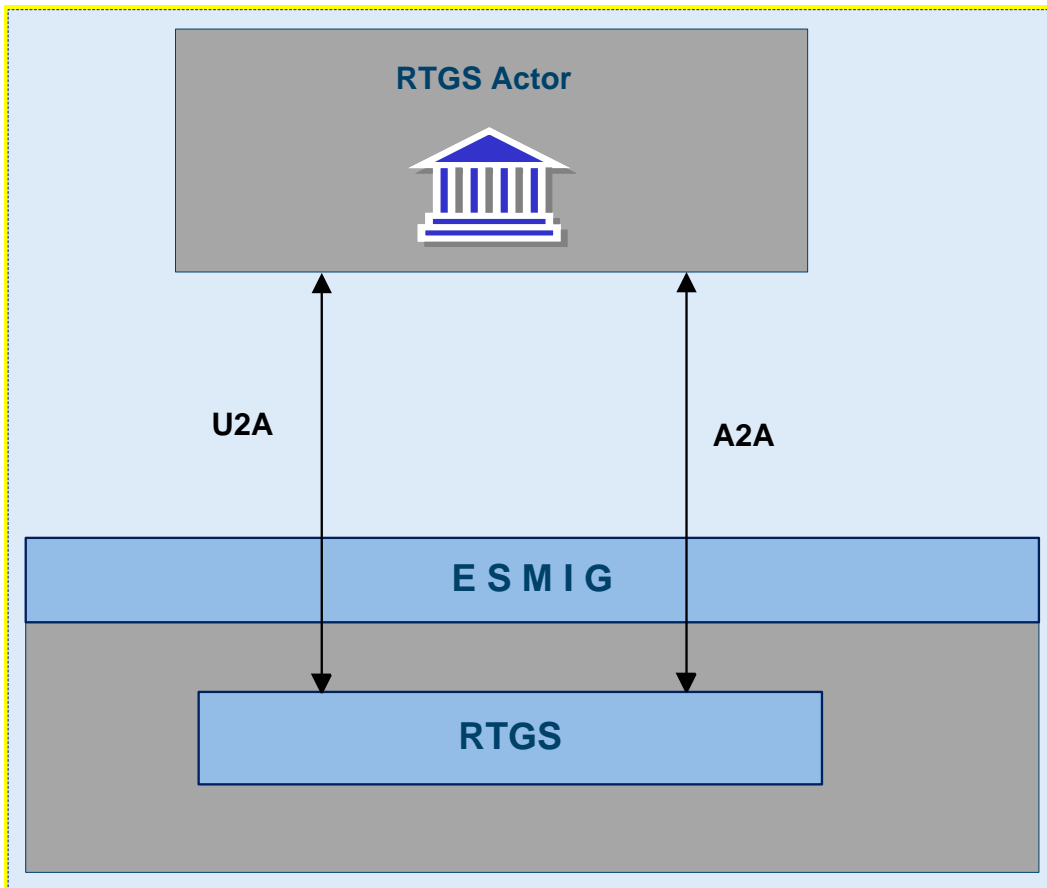


Figure 2 - Connectivity (U2A/A2A)

A2A

Software applications can communicate with RTGS exchanging single messages and multi messages (only inbound to RTGS). A2A communication relies on XML messages, using ISO 20022 standard where applicable, for both inbound and outbound communication.

The A2A supports the following connectivity modes:

- | store-n-forward, message based
- | store-n-forward, file based
- | real-time, message based
- | real-time, file based

The store-n-forward mode allows to send messages also when the receiver is not reachable in the moment the message is sent, in this case a retry mechanism is employed. In contrast real-time communication requires sender and receiver to be available and reachable when the message is sent. In case the real-time message cannot be delivered, no retry mechanism is available for the real-time mode.

Message based and file based in the context of connectivity refer to the maximum allowed size of the business content to be sent. The allowed maximum size of the message based communication is lower than the file based. In case the size of a message exceeds the limitations of message based communication, file based communication needs to be employed.

The connectivity modes are not related to the content, .i.e. single messages can be sent using file based communication (and have to, if they exceed the size limit of message based transmission) and multi messages can be sent using message based communication, if the size limit is not exceeded.

User-to-Application (U2A)

For defined functionalities, the RTGS Actors can access RTGS through a GUI.

2.2 Authentication and authorisation process

Any individual or application interacting with RTGS is identified by a distinguished name (DN). A DN is a sequence of attribute-value assertions separated by commas, e.g. <cn=meier,ou=RTGS,o=bnkacct,o=nspsname>

DNs are uniquely linked to digital certificates, which RTGS Actors assign to their individuals (interacting with RTGS in U2A mode) or applications (interacting with RTGS in A2A mode).

The ESMIG authenticates the RTGS Actor and carries out an authorisation check at service level, in order to verify whether the DN is enabled to submit requests to RTGS. The ESMIG documentation contains exhaustive information on all the checks the ESMIG carries out. If these checks are successful, the request and the sender's DN are forwarded to RTGS.

RTGS then carries out the authorisation of the sender at application level based on the DN's access rights profile. The DN that is used to sign the A2A message or used to access U2A is linked to one user. The user has one or many roles. Roles are defined by the system and contain a fixed set of privileges. According to the role's access privileges the authorisation of the request is checked.

2.3 Authentication and authorisation in ESMIG

This section provides information on the authentication and authorisation processes in ESMIG. More into detail, chapter [Authentication and authorisation concepts](#) [37] presents some basic notions (e.g. user, certificate, DN, technical sender) related to access rights management in the TARGET Services, common com-

ponents and back-office applications. On this basis, chapter [Authentication process](#) [38] and [Authorisation process](#) [39] show respectively how and where the authentication and the authorisation processes take place.

2.3.1 Authentication and authorisation concepts

This section presents the main concepts related to authentication and authorisation processes in ESMIG.

2.3.1.1 User

A user is an individual or application that interacts with ESMIG triggering the available y user functions of TARGET Services, common components and back-office applications. E.g. the set of available user functions stems from the set of privileges of TARGET Services, common components and back-office applications for which the user is grantee. Each user defined in TARGET Services, common components and back-office applications corresponds to an individual or to an application.

2.3.1.2 Certificate

A digital certificate is an electronic document binding an identity to a pair of electronic keys, a private key (used to sign digital information to be sent to a counterpart or to decrypt digital information received from a counterpart) and a public key (used to encrypt digital information to be sent to a counterpart or to perform the authentication and to ensure the integrity of digital information received from a counterpart). Actors assign certificates to their individuals (interacting with ESMIG in U2A mode) and applications (interacting with ESMIG in A2A mode). If an actor uses multiple connectivity providers to connect to TARGET Services, common components or back-office applications, then it has to assign one certificate to each of its individuals and applications for each of these connectivity providers.

2.3.1.3 DN

A DN is a sequence of attribute-value assertions (e.g. "cn=smith") separated by commas, e.g.:

```
<cn=smith,ou=serv-ops,o=bnkacct,o=nsp-1>
```

Each identity bound to a digital certificate is assigned a unique DN (certificate DN). This applies both to individuals and applications. If an actor uses multiple connectivity providers, each of its individuals and applications is assigned one certificate per connectivity provider and hence one certificate DN per connectivity provider.

2.3.1.4 Technical sender

The technical sender is the actor submitting an A2A or an U2A request to TARGET Services, common components and back-office applications. Each technical sender is identified by means of a certificate issued by one of the compliant NSP. The network infrastructure of the Network Service Provider (NSP) authenticates the technical sender on the basis of its certificate, both in A2A mode and in U2A mode. The certificate DN of the technical sender represents the technical address used by the technical sender to connect to TARGET Services, common components or back-office applications.

2.3.1.5 Business sender

The business sender is the actor creating the business payload of an A2A or an U2A request to be submitted to and processed by TARGET Services, common components and back-office applications. In some instructing scenarios, the business sender and the technical sender can be different actors.

2.3.2 Authentication process

The authentication process refers to the authentication of the technical sender.

2.3.2.1 Authentication of the technical sender

The authentication of the technical sender is performed at network infrastructure level and it is based on the certificate used by the actor to establish the technical connection with the network infrastructure itself. This authentication process is under the responsibility of the NSP selected by the actor to connect to the TARGET Services, common components and back-office applications.

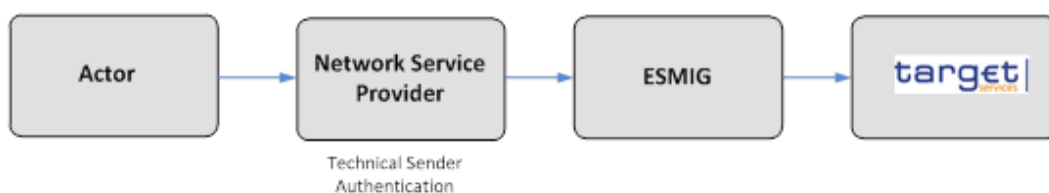


Figure 3 - Technical sender authentication

In case of successful authentication of the technical sender, the TARGET Services, common components or back-office applications gets the certificate DN of the technical sender. The TARGET Services, specific/common components or back-office applications may use this certificate DN later on, during the authorisation process (see chapter [Authorisation of the technical sender](#) [> 39]).

2.3.3 Authorisation process

In case of successful authentication of the technical sender, the TARGET Services, common components or back-office applications gets the certificate DN of the technical sender. The TARGET Services, common components or back-office applications uses this certificate DN later on, during the authorisation process (see chapter [Authorisation of the technical sender](#) [. 39]).

2.3.3.1 Authorisation of the technical sender

ESMIG checks whether the technical sender is allowed to access the service/component, making use of the CGU feature provided at NSP level.

The authorisation of the technical sender is performed at application level, when required by the component. The TARGET Services, common components or back-office applications authorises the technical sender for a given request only if the certificate DN (i.e. the technical address) of the same technical sender is in the list of the party technical addresses of the business sender which are linked to the NSP used to submit the request.

2.4 Security

This section aims at describing the main processes performed by RTGS in terms of principles applied to ensure RTGS actors can securely exchange information with RTGS.

It means that the following security conditions are met:

- I **Confidentiality:** ensuring that information is accessible only to authenticated and authorised RTGS Actors
- I **Integrity:** safeguarding the accuracy and completeness of information
- I **Availability:** ensuring that authorised users have access to information and associated assets when required
- I **Monitoring:** detecting operational and technical problems and recording appropriate information for crisis management scenarios and future investigations
- I **Auditability:** ensuring the possibility to establish and monitor whether a system is functioning properly and that it has worked properly

2.4.1 Confidentiality

The confidentiality of data is ensured by the possibility to grant specific access rights for any given set of data. In conjunction with mechanisms of authentication and authorisation applied to all requests received by

RTGS in both A2A and U2A mode, this guarantees that each RTGS Actor's data is treated confidentially and is not accessible to non-authorized actors.

2.4.2 Integrity

Within RTGS, various business validations ensure the integrity of information. If a business validation fails, RTGS has a concept of error handling in place. The requested action is not processed and RTGS provides the user with detailed information regarding the nature of the error.

In U2A mode, RTGS offers users in addition the possibility to further ensure the data integrity via usage of a dual authorisation concept, the four-eyes principle. In case this option is chosen for a specified set of RTGS operations, a second independent verification and confirmation is required before an operation becomes active in RTGS.

2.4.3 Availability

The overall availability of the RTGS component is ensured by the functional design, and a centralised technical architecture. This, together with a high level of inherent infrastructure redundancy and dedicated IT resources ensure the maximum availability for the RTGS component.

2.4.4 Monitoring

RTGS operational monitoring provides tools to the operator for the detection in real-time of functional or operational problems. Technical monitoring allows for the detection of hardware and software problems via real-time monitoring of the technical components involved in the processing, including the network connections.

2.4.5 Auditability

RTGS provides an audit trail with which it is possible to retrace user activities, exceptions and information security events. More in detail, the following data are collected:

- | payment transaction and liquidity transfer records;
- | authentication successes and failures of normal and privileged users;
- | security related notifications (e.g. changes of access rights, alerts and exceptional events).

2.5 Graphical User Interface

The RTGS Graphical User Interface (GUI) allows users to perform business functions based on their respective access rights. It allows users to enter and maintain business data as well as to retrieve business information.

The RTGS user handbook provides exhaustive information on each of the business functions that the RTGS GUI provides.

2.6 Routing

Communication channels can be categorised as follows:

- store-n-forward and
- real-time.

With the distinction of message based and file based network services this allows four network service types:

- store-n-forward message based network service
- store-n-forward file based network service
- real-time message based network service
- real-time file based network service.

The communication channel is part of the **technical address** that represents the core element for the routing of messages. The communication channel depends on the type of exchanged business data which can be categorised as follows:

- Instructions** are messages that intend to create or change data in CLM/RTGS. Instructions are only sent by external actors to CLM / RTGS in store-n-forward mode,
- Queries** are messages that intend to retrieve data from CLM/RTGS. Queries are only sent using real-time mode.
- Reports** are messages that intend to provide data in push mode from CLM/RTGS in store-n-forward mode.
- Notifications** are messages that intend to provide status information in push mode from CLM/RTGS. Notifications will be provided in store-n-forward mode as result of an instruction.

The following table summarises how the main types of CLM/RTGS business data exchanges are mapped against the technical features of the different network services for inbound and outbound communication including multi-messages.

CLM/RTGS Business Data Exchanges	Inbound communication	Outbound communication
Instructions	Store-n-forward message-based, store-n-forward file-based	Store-n-forward message-based (payments and notifications), store-n-forward file-based
Queries	Real-time message-based, real-time file-based	Store-n-forward message-based, real-time file-based In case of timeout and oversize store-n-forward message-based, store-n-forward file-based
Reports	N/A	Store-n-forward message-based, store-n-forward file-based
Notifications	N/A	Store-n-forward message-based, store-n-forward file-based

A **technical address** consists of 3 items

1. a technical receiver name which is represented by a DN;
2. a NSP;
3. a channel. Possible values for a channel are
 - store-n-forward message-based;
 - store-n-forward file-based;
 - real-time message-based;
 - real-time file-based.

The technical address for a message sent by RTGS is derived as follows:

RTGS Business Data Exchanges	Communication channel	Deduction of technical address
Notifications as response to instructions	Store-n-forward message	A notification as response to an instruction is sent to the same network service and technical address which were used for sending the related inbound communication.
Notifications being not as response to an instruction but belonging to a business case triggered by an instruction, e.g. camt.054	Store-n-forward message	The store-n-forward notification being not as response to an instruction is sent to the technical address that is defined in the routing configuration.

RTGS Business Data Exchanges	Communication channel	Deduction of technical address
Payments and cancellation requests (only RTGS)	Store-n-forward message	Payments and cancellation requests are sent to the technical address which is derived from the addressed business parties. The receiving business party is identified by to BIC (business identifier code) in the BAH.
Responses to queries	Real-time message, real-time file In case of time-out store-n-forward message, store-n-forward file	Responses to real-time messages are sent to the technical address of the sender of the query. In case of timeout and or oversize additional messages are sent using the store-n-forward message channel or store-n-forward file channel for the same technical receiver and the same network provider.
Reports	Store-n-forward file Store-n-forward message	Reports are sent in store-n-forward mode to the technical address that is defined in the routing configuration.

For notifications as response to instructions and responses to queries no routing configuration in CRDM is needed -and therefore also not available – as the notifications are always returned to the technical sender of the initial message.

The CRDM routing configuration applies to notifications not being a response to an instruction, and to reports.

Each party can define for each account and message type exactly one technical address the message shall be sent to. RTGS will identify the channel (message-based or file-based) depending on the size of the message to be sent.

Addressing & routing of payments and cancellation requests

Status quo T2

Routing of payments is based on BIC. T2 uses the Y-copy mode, i.e. the sending and receiving party in the MT header are BICs of participants, in particular the “Addressee” BIC provided in the T2 directory.

Future RTGS

The messages pacs.004/008/009/010, camt.056 and camt.029 (only if received as inbound message from an external party in case of negative result of resolution) are the RTGS inbound messages relevant for this topic.

For future RTGS the perpetuation of the current T2 directory was agreed between 4CB and their community.

The addressing is based on business level on the business application header (BAH, head.001) attributes <From> and <To>. The addressing in the BAH reflects the addressing from the business perspective, i.e. the sending party BIC is indicated in <From> and receiving party BIC in <To>.

BAH inbound and outbound:

Fr: BIC Bank A
To: BIC Bank B

Figure 4 - Routing A

As the future RTGS directory is derived from T2 directory it can happen that the sending/receiving party is deviant from the BIC account holder of the account to be used for debit/credit booking, e.g. multi-addressee.

The technical addressing in the consolidation project changes compared to T2. BICs cannot longer be used as technical addresses. Instead the technical addressing is based on DN, i.e. in case an inbound pacs message the RTGS component as DN receiver must be addressed in the technical header and in outbound messages the sender DN is the RTGS component.

Technical header inbound:

Sender: DN Bank A
Receiver: DN RTGS

Technical header outbound:

Sender: DN RTGS
Receiver: DN Bank B

Figure 5 - Routing B

The RTGS directory provides BICs as addresses. For routing purposes technical addresses consisting of DNs and NSPs are needed. Therefore a link between the BICs in RTGS directory and the technical addresses is established in CRDM. In case there is no link defined in CRDM the inbound message will be rejected as it cannot be forwarded to the intended recipient.

Connectivity requirements for actors due to routing

Store-n-forward mode:

- | Each external actor sending store-n-forward traffic to CLM/RTGS must be able to receive store-n-forward traffic with the sender DN and NSP for message-base and file-based network channel.
- | According to the routing configuration technical receiver name and the NSP are defined for receiving store-n-forward traffic from CLM/RTGS. The external actor must support message and file channel.

Real-time mode:

- | Each external actors sending real-time traffic to CLM/RTGS must be able to receive real-time and store-n-forward traffic with the sender DN and NSP for message and file channel.

2.6.1 ESMIG routing functions

The ESMIG routing functions are related to both inbound and outbound traffic. In this context ESMIG is able to route messages/files (i) to the addressed service/component for inbound traffic and (ii) to NSP and network channel for outbound traffic.

2.6.1.1 Inbound routing

ESMIG is in charge to manage the provisioning of messages and files received from the NSPs to the different market infrastructure services/components.

ESMIG routes incoming messages and files to the addressed market infrastructure service/component. ESMIG identifies and selects the appropriated service/component on the basis of information provided as part of the communication. In this respect, an enhancement of the DEP protocol is required to transport supplementary information to infer whether the target of the inbound communication is a market infrastructure service, a common component or a specific component.

Furthermore, ESMIG passes to the business interface of the relevant service/component the DN of the sender (as result of authentication process) and a predefined list of parameters.

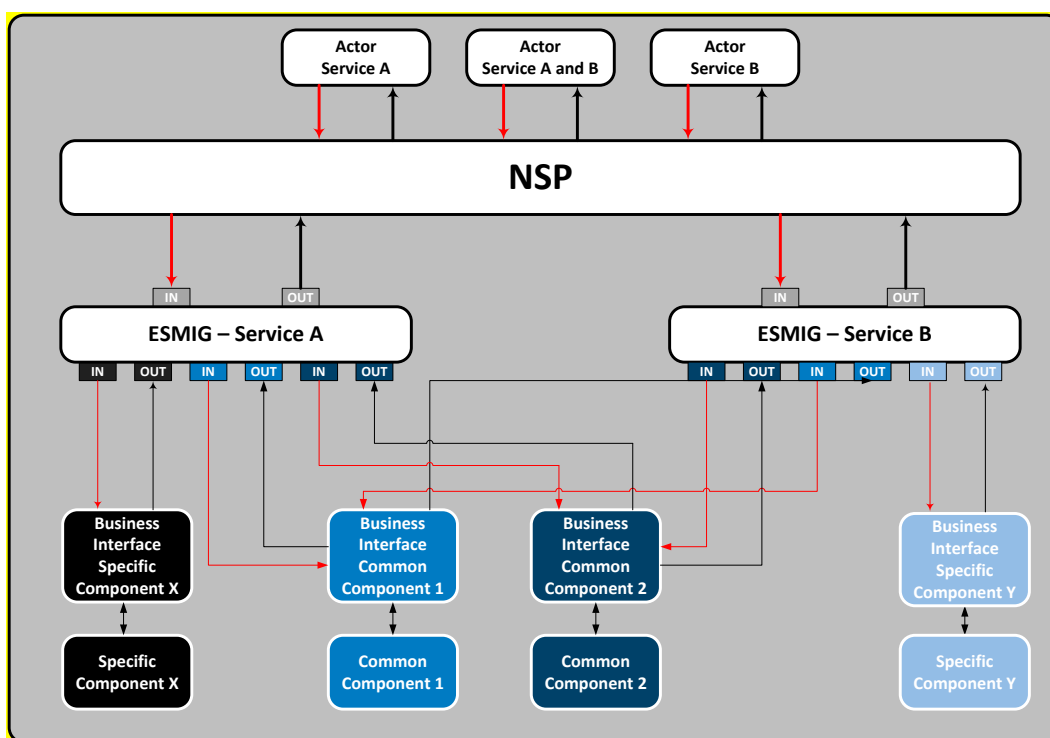


Figure 6 - Inbound routing

The interface between Eurosystem market infrastructure counterparties and the NSP are defined by the relevant NSP protocol documentation (DEP protocol is used only between NSP and ESMIG). In this context,

the NSP interface shall ensure that at least a minimum set of information is provided by the counterparties to be compliant with the DEP protocol.

Business service	Component	Communication mode and protocol
T2	CLM	A2A: MSGSNF, MSGRT, FILESNF, FILERT U2A
T2	RTGS	A2A: MSGSNF, MSGRT, FILESNF, FILERT U2A
T2	CRDM	A2A: MSGSNF, MSGRT, FILESNF, FILERT U2A
ECMS	ECMS	A2A: MSGSNF, MSGRT, FILESNF, FILERT U2A
T2S	T2S	A2A: MSGSNF, MSGRT, FILESNF, FILERT U2A
T2S	CRDM	A2A: MSGSNF, MSGRT, FILESNF, FILERT U2A

Table 1 - Services and components

2.6.1.2 Outbound routing

ESMIG routes messages and files to the external party using

- | the network provider,
- | the address used by the NSP to identify the external party,
- | the communication mode,
- | the protocol.

The above mentioned information is provided by the market infrastructure services/components (i.e. right external user address) to ESMIG.

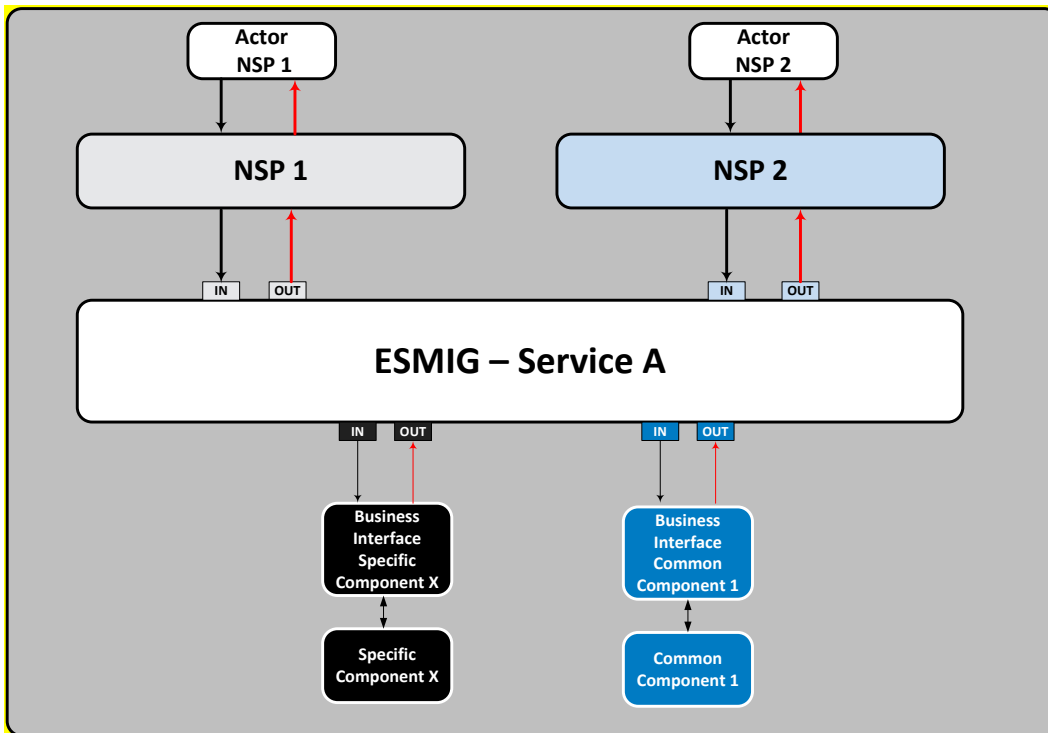


Figure 7 - Outbound routing

3 Parties and accounts

3.1 Parties

The RTGS participation model envisions different types of RTGS Actors, with different roles and responsibilities, as outlined in chapter [Concept of party in RTGS](#) [▶ 49]. T2 Actors that interact with RTGS are defined as different parties in CRDM.

This chapter provides a detailed description of all the objects and attributes that CRDM stores and RTGS uses for its participating T2 Actors. This chapter focuses in particular on the reference data in the context of parties used in RTGS. In [Overview of used common components in RTGS component](#) [▶ 234] the main focus is on CRDM features: setup of objects, the access rights concept and CRDM specific reference data.

More in detail, chapter [Setup of parties](#) [▶ 48] identifies the reference data related to the setup of RTGS Actors and it provides detailed information as to who is responsible for the setup of these reference data. Chapter [Concept of party in RTGS](#) [▶ 49] defines the concept of party in CRDM and the way this concept relates with the different types of legal entities that can interact with RTGS. In addition this chapter describes the so-called hierarchical party model, i.e. the organisational structure of parties in CRDM. The chapter [Reference data for parties used by RTGS](#) [▶ 50] illustrates the reference data required by RTGS for each T2 Actor, i.e. the way a party can be identified in RTGS and which attributes have to be stored for each RTGS Actor.

3.1.1 Setup of parties

A party is defined as any legal entity or organisation interacting with T2. The setup of RTGS parties takes place in CRDM.

The operator is responsible for setting up and maintaining party reference data for all CBs in RTGS. CBs are responsible for setting up and maintaining party reference data for the parties of their community.

The following table summarises the configuration responsibilities for each reference data object related to parties in RTGS and specifies the required communication mode.

Reference data object	Responsible actor	Mode
Party (CB)	Operator	U2A
Party (RTGS Participant)	CB	U2A
Party (ancillary system)	CB	U2A

Table 2 - Setup of parties for RTGS

RTGS imposes a constraint in the assignment of BICs related to its parties, due to the fact that the settlement process must be able to determine the accounts to be debited and credited by a payment based on the

BICs of the RTGS Account Holder. This implies the need to ensure that any given BIC can only be assigned to one RTGS Account Holder or ancillary system. Different RTGS Account Holders or ancillary systems must be assigned to a unique BIC.

For this reason, CRDM prevents two different parties to be defined as RTGS Participant or ancillary system if they are identified by the same 11-character BIC. Therefore, in order to allow a given party to be defined as two different RTGS Account Holders or ancillary systems (e.g. by the same CB or by two different CBs), the same party must be defined in CRDM as two RTGS Participants which will be identified by two different 11-character BICs.

3.1.2 Concept of party in RTGS

The party model of RTGS is based on a hierarchical three-level structure. The operator is the only party at the first level of the hierarchy and is responsible for the setup of each party of the second level, i.e. each CB in RTGS. Similarly, each party belonging to the second level is responsible for the setup of all parties of its community (i.e. RTGS Account Holders or ancillary systems), represented by parties of the third level.

This means that each CB is responsible for the reference data of its community. The RTGS Account Holders and ancillary systems are responsible for their own reference data. Further information about the hierarchical model can be found in [Common reference data objects and the hierarchical party model](#) [▶ 247] and information about the data scope are included in [Data scope](#) [▶ 248].

Each party belongs to one of the following party types according to the above mentioned hierarchical party model:

- | operator;
- | CB,
- | RTGS Participant
- | ancillary system.

The **operators** are the organisational entity that operates i.a. RTGS. They are responsible for the initial setup and day-to-day operations of RTGS and acts as single point of contact for CBs in case of technical issues. They are monitoring the system and carry out corrective actions in case of incidents or in case of service/component unavailability. The operators are also responsible for setting up and maintaining the reference data of the CBs in CRDM. Upon request of the respective CB the operators may operate RTGS functions on behalf of any RTGS Actor. They have full access to all live and all archived reference data and transactional data in RTGS.

CBs are responsible for setting up and maintaining reference data in CRDM for all RTGS Actors belonging to their community. CBs can also act as RTGS Participants themselves. In addition they can act on behalf of one of their parties on the third level in case of need.

In its CB role, it may only own CB accounts (see [Glossary](#) [743] for the definition of a CB account); all other account types need to be owned under its RTGS Participant role.

RTGS Participants represent RTGS Actors that own accounts (RTGS DCA and/or RTGS sub-accounts) in RTGS and are identified by a BIC11. RTGS Participants are responsible for their own liquidity management. They are responsible for instructing liquidity transfers and monitoring the liquidity usage. However, the setup and maintenance of the RTGS DCAs and RTGS sub-accounts is done by CBs on request of the respective RTGS Account Holder.

Ancillary systems can be given the right to submit instructions via the RTGS DCA of a RTGS Account Holder on its behalf or via a RTGS sub-account dedicated to the ancillary system. In principle they shall not own a RTGS DCA. However, there may be exceptions in order to cover certain market conditions. Moreover, ancillary systems are responsible for monitoring their ancillary system technical accounts (used for ancillary system settlement procedures A, B, C and) and guarantee funds accounts (used for settlement procedures A and B). The setup and maintenance of these accounts is done by CBs on request of the respective ancillary system.

3.1.3 Reference data for parties used by RTGS

The following table shows a non-exhaustive list of party reference data attributes that RTGS receives from the CRDM and stores in its Local Reference Data Management (LRDM).

Attribute	Description
Party BIC	It specifies the BIC11 to uniquely identify the party in RTGS.
Institutional sector code	It identifies the financial corporation's sector classification to which the party belongs with respect to the nature of its business.
Party status	It specifies the business status of a party for processing in the system (e.g. active).
Opening date	The date on which the contractual relationship with the party was legally established.
Banking Group identifier	It specifies the unique technical identifier of the Banking Group to which the party belongs to.
LEI	It specifies the unique identifier of the legal entity in accordance with the ISO 17442 standard.
Monetary financial institution (MFI)	It specifies the MFI with which the party is associated for the calculation of minimum reserves via a pool.
MFI leader BIC	It specifies the BIC of the party designated as the MFI leader where minimum reserves are managed in a pool.

Attribute	Description
Closing date	The date that the contractual relationship with the party has legally ended.
Currency code	It specifies the currency associated with a CB.
Country code	It specifies the two-character ISO country code (ISO3166-1) identifying the country code of the address.

Table 3 - Party reference data attributes

3.1.4 Participation types

There are several possibilities to participate in RTGS. This chapter will give an overview of these participation types.

Direct participants

Direct participants have direct access to RTGS and hold their own RTGS DCAs and/or RTGS sub-accounts. They can provide indirect access to RTGS for other credit institutions and offer them additional services. They are responsible for their own reference data and for their own liquidity management in RTGS and for monitoring the settlement process. Furthermore, they are responsible for all cash transfers sent or received on their cash account by any entity registered through them.

Indirect participants

Only supervised credit institutions established in the EEA and EU CBs participating directly in RTGS are allowed to intermediate for credit institutions in the EEA to have their liquidity settled without connecting directly to it.

Indirect participants:

- | are directly linked to one direct participant only (that can be located also in another country);
- | can be indirectly addressed in RTGS and
- | have no own RTGS DCA and/or RTGS sub-account.

Each Indirect Participant needs a published BIC (8 or 11-digit).

The Indirect Participant sends cash transfers to/receives cash transfers from RTGS via the direct participants. The booking is done on the RTGS DCA of the direct participant. The relevant direct participant also manages the liquidity of its Indirect Participants, and has accepted to represent the respective Indirect Participant. T2 recognises Indirect Participants allowing them to benefit from the protection of the settlement finality directive (in countries where such protection has been granted).

Multi-addressee access

Direct Participants are able to authorise their branches and credit institutions belonging to their group located in the EEA countries to channel payments through the RTGS DCA of the direct participant without its involvement by submitting/receiving cash transfers directly to/from RTGS.

The liquidity is settled on the RTGS DCA of the direct participant.

Access as correspondent

Any correspondent (or branch of a correspondent) of a direct participant that holds a BIC is eligible to be listed in the RTGS directory irrespective of its place of establishment. It is the responsibility of the direct participant to forward the relevant information to the respective CB for inclusion in the RTGS directory. These BICs can only send and receive cash transfers to/from RTGS via the direct participant. Their cash transfers are settled in the RTGS DCA of the respective direct participant.

Technically there is no difference between Indirect Participants and the access as a correspondent. However, in legal terms, T2 will not recognise access as correspondents. The latter will therefore not benefit from the protection of the settlement finality directive (in countries where such protection has been granted).

The following table summarizes the conditions and features of the above mentioned participation types.

Feature	Direct participant	Indirect Participant / access as correspondent	Multi-addressee access
Sending and receiving cash transfers	Directly	Via direct participant	Directly
Own cash account in RTGS	Yes	No	No
Liquidity provisioning	On its cash account in RTGS	By direct participant	By direct participant
Liquidity control	By itself	By direct participant	By direct participant
Addressability	Directly	By direct participant	Directly
Publication in RTGS directory	As direct participant	As indirect participant	As multi-addressee access

Table 4 - Comparison of participation types

3.1.5 Blocking/unblocking party

The blocking/unblocking of RTGS Account Holders and ancillary systems is possible. It is up to CBs or any other authority in charge to declare actions to:

- I restrain the disposal of the assets and

- | withdraw the license.

As a consequence of this declaration or withdrawal the affected RTGS Account Holder or ancillary system is blocked in RTGS. The blocking is under the full responsibility of the respective CB. The CB initiates the blocking at party level (via a restriction type) via the CRDM GUI.

When blocking a party in CRDM the blocking request can include a valid from date and time. This value indicates the calendar date and time as of which the party will be blocked. If not stated, the next calendar date will be used by default. If the valid from date and time is specified as immediate, the blocking becomes effective immediately in all services the party is linked to. The same behaviour is applicable for the unblocking of parties.

As soon as a RTGS Account Holder or an ancillary system is blocked at party level, all linked cash accounts across all settlement services/components are blocked too. For further information on account blocking please refer to chapter [Blocking/unblocking account](#) [▶ 64]

3.2 Accounts structure and functionalities

Accounts are opened in RTGS for the settlement of real-time interbank and customer payments and ancillary system transfers. This chapter provides a detailed description of all the reference data CRDM stores and RTGS uses for all its accounts.

The operator and/or CBs set up and maintain the following categories of accounts ¹ in CRDM:

- | RTGS DCAs
- | RTGS sub-accounts
- | RTGS dedicated transit account
- | RTGS CB accounts
- | ancillary system guarantee funds accounts
- | ancillary system technical accounts.

Moreover, RTGS Account Holders can set up the following functionalities on their RTGS DCAs:

- | floor/ceiling
- | standing liquidity transfer orders
- | standing orders for reservation
- | current reservation(s)
- | standing orders for limits
- | current limit(s)

¹ Due to ongoing discussions regarding ECONSII an additional account type could be added in a later version

- | message subscription
- | report configuration.

Even if defined by the RTGS Account Holder, the setup and maintenance of the direct debit mandate is done by CBs.

The following chapters define the above mentioned reference data objects.

3.2.1 Account types

This chapter will give an overview of all account types used in RTGS.

RTGS DCA

RTGS DCA are used for the settlement of real-time interbank and customer payments and ancillary system transfers. They shall either have a zero or a positive balance.

A RTGS Actor may have several RTGS DCAs but shall ensure that each of these RTGS DCAs is identified with a unique BIC11 (in addition to different account numbers).

Each RTGS DCS is linked to one and only one RTGS Account Holder (i.e. the DCA owner). It is up to CBs to set up and maintain RTGS DCAs for their RTGS Account Holders.

A link between a MCA in CLM and a RTGS DCA is the precondition for floor/ceiling and automated liquidity transfers due to CBO. The RTGS DCA involved in this scenario is defined as linked and default RTGS DCA. In case there are several RTGS DCAs linked to one MCA in CLM, only one of the RTGS DCAs is the default one.

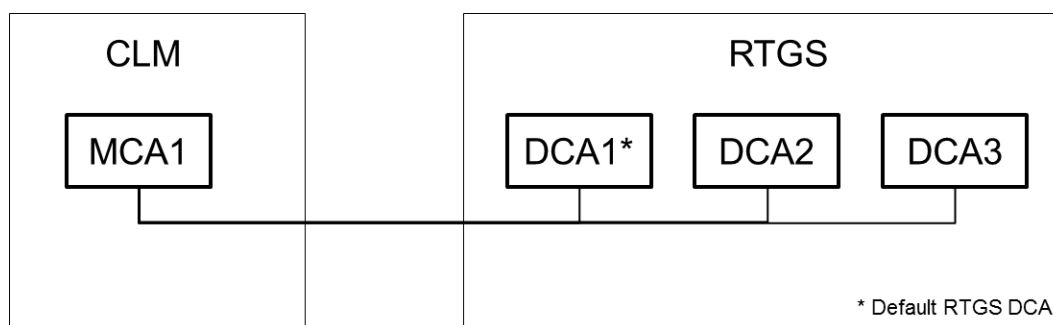


Figure 8 - Linked default MCA DCA

In the event the floor or ceiling amount on a DCA is breached (after the settlement of a cash transfer) and if the RTGS Account Holder has opted for the rule-based liquidity transfer order generation, RTGS generates automatically an inter-service liquidity transfer order. Subsequently, cash is either

- | pulled from the linked MCA in CLM (in the event the floor is breached); or

- I pushed to the linked MCA in CLM (in the event the ceiling is breached).

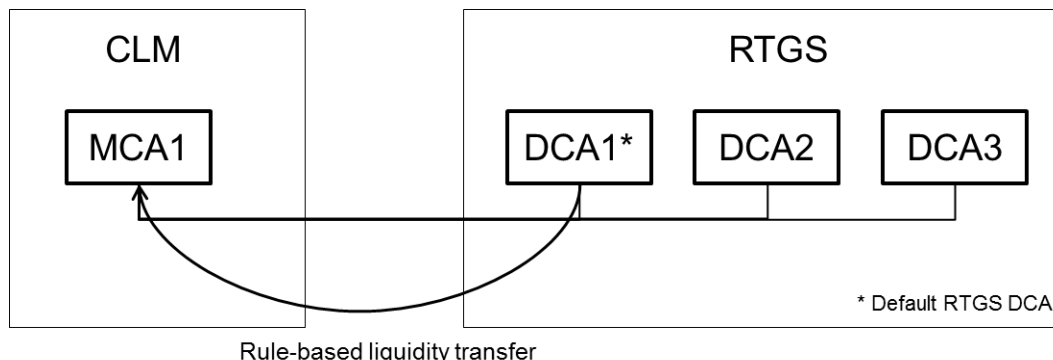


Figure 9 - Floor/ceiling breached on MCA1

It is up to RTGS Account Holders to decide which RTGS DCA should be the default one. The setup and maintenance of the links between MCAs and DCAs will be done by the CB in CRDM.

Furthermore, each RTGS DCA may be linked to one or many Liquidity Transfer Groups and to one or many account monitoring groups.

RTGS sub-account

A RTGS sub-account belongs to a RTGS DCA and holds dedicated liquidity to allow the settlement of an ancillary system under ancillary system procedure “settlement on dedicated liquidity accounts”. A RTGS sub-account shall either have a zero or a positive balance.

The RTGS sub-account is identified by an account number and directly linked to one and only one RTGS DCA, the latter being identified by a unique BIC11. Moreover, one or many RTGS sub-accounts may be linked to one and only one RTGS DCA.

It is up to CBs to set up and maintain RTGS sub-accounts for their RTGS Account Holders.

RTGS dedicated transit account

The RTGS dedicated transit accounts are accounts owned by CBs. They shall either have a zero or a negative balance as they reflect any movement of liquidity from/to CLM. They are technical accounts involved in the inter-service liquidity transfer process and cannot be involved in the settlement of payments and ancillary system transfers. The RTGS dedicated transit account cannot be directly addressed by the RTGS Account Holders in a liquidity transfer.

There is only one RTGS dedicated transit account per settlement currency. The RTGS dedicated transit account for euro belongs to the ECB.

However, the monitoring of RTGS dedicated transit accounts is conducted by the operator.

It is up to the operator to set up and maintain the RTGS dedicated transit account.

RTGS CB accounts

A RTGS CB account is a cash account owned by a CB that is allowed to have negative balance.

The purpose of the account is to provide liquidity to RTGS Account Holders in case of contingency scenarios.

A CB account in RTGS is identified by a BIC11. CBs have the possibility to open more than one CB account, each one being identified by a unique BIC11.

It is up to the operator to set up and maintain the CB accounts.

Ancillary system guarantee funds account

An ancillary system guarantee account is an account in RTGS for maintaining funds allocated to the settlement of balances of an ancillary system in case of failure of settlement bank(s). It applies to ancillary system settlement procedures A and B.

The ancillary system guarantee funds account shall either have a zero or a positive balance.

It is up to CBs to set up and maintain the ancillary system guarantee funds accounts.

Ancillary system technical account

An ancillary system technical account is an account used in the context of ancillary system transfers. It is an intermediary account for the collection of debits/credits resulting from the settlement of balances. Furthermore it can be used as a liquidity bridge for transferring funds from the RTGS into the ancillary system and vice versa.

The ancillary system technical account shall either have a zero or a positive balance.

It is up to CBs to set up and maintain the ancillary system technical accounts.

The following table provides an overview of the attributes of the account reference data objects and does not give any indication on the structure of CRDM reference data tables.

Attribute	Description
Account number	It specifies the number of the account (unique across all services).
Cash account type	Type of account. The exhaustive list of account types in RTGS is as follows: <ul style="list-style-type: none"> RTGS DCA RTGS sub-account RTGS dedicated transit account RTGS CB account

Attribute	Description
	<ul style="list-style-type: none"> ancillary system guarantee funds account ancillary system technical account
Currency code	It specifies the currency of the account.
Account owner	It specifies the BIC11 of the party owning the account.
Status	<p>Blocking status for the account. Exhaustive list of possible values:</p> <ul style="list-style-type: none"> blocked for credit blocked for debit blocked for credit and debit unblocked
Opening date	Opening date of the account.
Floor	<p>If defined, it specifies the behaviour the system applies in case the floor on a RTGS DCA is breached: Exhaustive list of possible values:</p> <ul style="list-style-type: none"> Send notification Initiate liquidity transfers
Ceiling	<p>If defined, it specifies the behaviour the system applies in case the ceiling on a RTGS DCA is breached: Exhaustive list of possible values:</p> <ul style="list-style-type: none"> Send notification Initiate liquidity transfers
Floor threshold	It specifies a lower threshold which may trigger the sending of a notification message and/or a liquidity transfer order if it is breached from above (absolute numbers).
Ceiling threshold	It specifies an upper threshold which may trigger the sending of a notification message and/or a liquidity transfer order if it is breached from below (absolute numbers).
Target amount after breaching floor	It specifies the target amount to be reached if the floor is breached.
Target amount after breaching ceiling	It specifies the target amount to be reached if the ceiling is breached.
Ancillary system used	It specifies the ancillary system used. This applies only to ancillary system guarantee funds accounts and ancillary system technical accounts.
Ancillary system model used	It specifies the ancillary system model used. This applies only to ancillary system guarantee funds accounts and ancillary system technical accounts.
Management of minimum reserve	<p>It specifies the method by which the minimum reserve is managed. Possible values are:</p> <ul style="list-style-type: none"> direct

Attribute	Description
	<ul style="list-style-type: none"> indirect pool no
Minimum reserve party	It specifies the party for which this account is included for minimum reserve calculation.
Linked MCA	It specifies the linked MCA.
Closing date	Closing date of the account.

Table 5 - Reference data attributes

3.2.2 Functionalities

This chapter describes the functionalities available at RTGS DCAs level.

Direct debit mandate

The direct debit functionality in RTGS can be used by RTGS DCA and sub-account holders as well as ancillary systems.

Each RTGS DCA and sub-account holder needs to agree with the parties allowed to debit its accounts on the terms and conditions for using this service. T2 only offers the general framework.

The RTGS Account Holder authorises another RTGS Account Holder or ancillary system to issue a direct debit order. This RTGS Account Holder shall inform its CB, which is then responsible to record and administrate the direct debit mandates.

For each RTGS Account Holder CRDM manages the information about the direct debit(s) this RTGS DCA or sub-account holder has authorised and the related attributes (e.g. maximum amounts).

It is up to CBs to set up and maintain in CRDM the direct debit mandate(s) of a RTGS Account Holder, while the definition is done by the RTGS DCA or sub-account holder. All actions (setup, modify, delete) will become effective as of the next business day or on the activation date of the DCA or sub-account if this is later than the next business day.

The following table shows a list of direct debit reference data attributes.

Attribute	Description
Account number	It specifies the RTGS DCA or RTGS sub-account on which the direct debits are authorised.
Payee party identifier	It specifies the party whose payment requests are authorised under this

Attribute	Description
	mandate and to whom the corresponding payments are made.
Payee reference	The reference provided by the payee party to be included in the payment details for recognition of the payment.
Maximum amount (counterpart)	It specifies the maximum amount the authorised issuer is able to direct debit during the single business day.
Maximum amount per payment	It specifies the maximum amount the authorised issuer is able to debit directly in a single direct debit.
Maximum amount for direct debit per day	It specifies the maximum amount of direct debits which can be debited each day on the RTGS DCA or the RTGS sub-account.
Valid from date	It specifies the date as of which the direct debit instruction is valid.
Valid to date	It specifies the date until which the direct debit instruction is valid.

Table 6 - Direct debit mandate reference data attributes

Floor/ceiling

For each RTGS DCA, a RTGS Account Holder can define a minimum (“floor”) and/or a maximum (“ceiling”) amount in CRDM that shall remain on the respective account. The RTGS Account Holder can choose how RTGS shall respond in case the floor or ceiling on an account is breached (after the settlement of payments):

1. RTGS generates a notification that is sent to the RTGS Account Holder informing about the floor/ceiling breach (upon which the RTGS Account Holder can actively take action); or
2. RTGS generates a rule-based inter-service liquidity transfer order to pull cash from the linked MCA in CLM (in the event the floor is breached) or push cash to the linked MCA in CLM (in the event the ceiling is breached).

It is up to RTGS Account Holders to set up and maintain the floor/ceiling information in CRDM. All actions (setup, modify, delete) will become effective as of the next business day or on the activation date of the RTGS DCA if this is later than the next business day.

Standing liquidity transfer order

A standing liquidity transfer order is an instruction of a RTGS Account Holder to transfer regularly (e.g. daily, weekly) an amount of liquidity from a RTGS DCA to another account (a MCA in CLM, a RTGS DCA or a DCA in another settlement service/component) over a period with or without a predefined end date. Either a specific amount or the whole balance could be transferred from the RTGS DCA.

This information is defined at the level of the RTGS DCA and it is up to the RTGS Account Holder to set up and manage its standing liquidity transfer orders information in CRDM.

The following table shows a list of the standing liquidity transfer order reference data attributes.

Attribute	Description
Transfer Type	It specifies the type of the liquidity transfer. The exhaustive list of transfer type options in RTGS is as follows: <ul style="list-style-type: none"> inter-service liquidity transfer from RTGS DCA to MCA intra-service liquidity transfer to another RTGS DCA inter-service liquidity transfer to DCA in another settlement service / component
Reference of instruction	It specifies the reference given by the original instructor of the liquidity transfer.
Transfer amount	It specifies the amount to be debited with the liquidity transfer.
Whole balance	It specifies if the whole balance is transferred.
RTGS DCA to be debited	It specifies the DCA to be debited in RTGS.
Account to be credited	It specifies the account (MCA or another DCA) to be credited.
Trigger event	It specifies the event type that will trigger the transfer of liquidity.
Whole balance	It specifies if the whole balance is transferred.
Valid from date	It specifies the date from which the standing order is valid.
Valid to date	It specifies the date until which the standing order is valid.

Table 7 - Standing liquidity transfer order reference data attributes

Standing order for reservation

A standing order for reservation is an instruction of a RTGS Account Holder to set up an urgent or high reservation of a fixed amount for a business day on a RTGS DCA without a predefined end date.

An existing standing order for reservation can be modified or deleted. All actions (setup, modify, delete) will become effective as of the next business day or on the activation date of the RTGS DCA if this is later than the next business day. The reservation will remain valid until it is modified or deleted.

It is up to the RTGS Account Holder to set up and maintain its standing order for reservation information in CRDM.

Current reservation

For the execution of cash transfers with certain priority classes (urgent or high) RTGS Account Holders can set up a current reservation on liquidity in RTGS. An existing reservation can be modified and/or deleted. All activities (setup, modify, delete) will become effective immediately. In case the amount changes to "0" the reservation will be removed automatically. The reactivation will however be possible during the business day.

This information is defined at the level of the RTGS DCA and it is up to RTGS Account Holders to set up and maintain the current reservations in RTGS.

Standing order for limit

A standing order for limit is an instruction of a RTGS Account Holder to define bilateral and/or multilateral limits of a fixed amount for a business day on a RTGS DCA without a predefined end date. These limits are processed during the Start-of-Day (SoD) procedure of the following business day.

A RTGS Account Holder can define the following types of limits in CRDM:

- | bilateral standing order for limits
- | multilateral standing order for limits:

A bilateral standing order for limits is defined vis-à-vis a different RTGS Account Holder. A multilateral standing order for limits is defined vis-à-vis all RTGS Account Holders without bilateral limit. It will become effective the next business day or on the activation date of the related RTGS DCA if this is later than the next business day. A standing order for limits can be modified or deleted. Modification or deletion will become effective on the next business day or on the activation date of the related RTGS DCA if it is later than the next business day.

The setting to "0" of the amount of a standing order for limit will have a similar effect as the deletion of a standing order limit. With a single request a RTGS Account Holder can modify all or several bilateral standing order limits which were defined in the past and/or define several standing order limits. The deletion of all or several standing order limits with a single request is also possible.

This information is defined at the level of the DCA and it is up to the RTGS Account Holder to set up and maintain its standing orders for limit in CRDM.

Current limit

A limit is the maximum amount for payments with a priority class "normal" that a RTGS Account Holder is willing to pay to another RTGS Account Holder per day (bilateral limit), or to all other RTGS Account Holders (excluding those with whom a bilateral limit is defined) per day (multilateral limit).

A RTGS Account Holder can define a new value for the following existing limits in RTGS:

- | bilateral current limits
- | multilateral current limits:

The limits are debit limits and not credit limits.

To take a limit (bilateral or multilateral) into account during the settlement process, it has to be defined before the end of the previous business day with a standing order for limit. Once a limit is defined, it can be changed with current limits. Current limits will be valid only for the current business day and will become effective immediately.

A deletion will also become effective immediately but for the current business day only. If the amount for a current limit is changed to "0", it will have the same effect as if the current limit is deleted.

Current limits are defined at the level of the DCA and it is up to RTGS Account Holders to define current limits in RTGS.

3.2.3 Messaging

This chapter gives a rough overview about the RTGS specification regarding message subscription, report configuration and routing configuration.

Message subscription

Message subscription is defined as a service that allows RTGS Account Holders to subscribe for the receipt of certain message types, based on a set of predefined parameters.

This information is defined at the level of the RTGS DCA and it is up to the RTGS Account Holder to set up and maintain the message subscription in CRDM.

Changes to the message subscription in CRDM will become effective with validity from the next business day.

The following outgoing messages are subject to message subscription:

- | camt.004
- | camt.019
- | camt.029
- | camt.054
- | pacs.002.

The table below describes a list of parameter types that authorised RTGS Actors can use for configuring their message subscription.

Parameter type	Description
Message type	It specifies the type of message to which the authorised RTGS Actor wants to subscribe. The exhaustive list of possible message types is listed above.
Cash account	It specifies the cash account in RTGS to which the subscription applies.
Business case	It specifies the business case for which the message is subscribed.
Priority	It specifies the priority of the original payment instruction.
Recipient	It specifies the BIC of the message recipient.
Underlying message type	It specifies the message type of the original message sent to the service.

Table 8 - Message subscription parameter types

The following table provides the mapping between the outgoing message types subject to subscription and the applicable parameter types.

Message type	Cash Account	Business case	Priority	Multi-addressee BIC	Underlying message type
camt.004	Yes	No	No	No	No
camt.019	No	No	No	No	No
camt.029	Yes	No	No	Yes	No
camt.054	Yes	Yes	No	No	No
pacs.002	Yes	No	Yes	Yes	Yes

Table 9 - Applicable parameter types for outgoing messages

If only the parameter “message type” is subscribed, all messages of this message type (e.g. camt.029, pacs.002) are sent to the respective recipient. In case the RTGS Account Holder wants to receive only messages related to a specific cash account he has to define the message type and the cash account to which the subscription applies.

For general information about message subscription please refer to chapter [Message subscription](#) [▶ 263].

Report configuration

RTGS Actors can configure one standard report (statement of accounts) that RTGS shall create at a specific business day event (EoD). RTGS Actors can specify in their report configuration, whether such report shall be sent to the recipient immediately in A2A mode (push) or be stored for later querying in A2A mode or downloading via GUI (pull). Created reports are available for later querying (A2A) and downloading (U2A) until the next report based on the same configuration is created.

Report configuration shall also allow a RTGS Actor to configure another RTGS Actor to receive the report either instead or in addition.

This information is defined at the level of the cash account and it is up to the RTGS Actor to set up and maintain the report configuration in CRDM.

For further information about the report generation please refer to chapter [RTGS report generation](#) [▶ 228].

Routing configuration

The routing configuration defines the technical address to which reports, notifications and forwarded payment messages are sent to. This does not to pacs.002 as this message is always returned to the technical sending address of the underlying message (if subscribed).

Routing for each message type is configured at the level of the cash account and it is up to the RTGS Actor to set up and maintain the report configuration in CRDM.

In the case of multi-addressee access, it is possible to configure a separate DN per RTGS Account Holder for the receipt of forwarded payment messages, reports and notifications.

3.2.4 Blocking/unblocking account

It is possible to block cash accounts in RTGS. The blocking of cash accounts is possible for:

- | credit and debit
- | debit
- | credit.

When blocking a cash account in CRDM the blocking request can include a valid from date and time. This value indicates the calendar date from which on the cash account will be blocked. If not stated, the next calendar date will apply by default. If the valid from date and time is specified as immediate, the blocking becomes effective immediately. The same behaviour is applicable for the unblocking of cash accounts.

- | In case CRDM marks the cash account as blocked for credit and debit, credits and debits are not allowed on the cash account. If the cash account is blocked for debit, credits are still allowed on this cash account. The reverse logic applies in case of blocking for credit (debits are still allowed).

The procedure in RTGS is the following for the blocking of RTGS DCAs:

- | The RTGS DCA of the RTGS Account Holder is earmarked immediately. As a consequence no cash transfers (depending on the kind of blocking) can settle automatically on this cash account.
- | All cash transfers pending in the queue after the blocking became effective require confirmation by the CB before they will can settle on the RTGS DCA.

- I Cash transfers involved in a running settlement process are not affected by the blocking and the algorithm is not stopped. If the algorithm
 - is successful, the involved cash transfers of the blocked RTGS Account Holder will become final.
 - fails, the cash transfers of the blocked RTGS Account Holder will return to the queue. They require confirmation by the CB before they can settle in one of the next running algorithms.
- I Payments (credit transfers or direct debits) sent by the blocked RTGS Account Holder are stored for confirmation by the CB. If the CB
 - gives its confirmation, the payments will run through the entry disposition. If they cannot be settled in the entry disposition, they will be queued and included in the process of dissolution of the payment queue.
 - disagrees, the payments will be rejected.

Note:

- I The confirmation of payments is physically done by the CB of the blocked RTGS Account Holder via the GUI. Nevertheless organisational rules outside T2 can be implemented to involve other bodies (eg the official receiver), but it is up to the legal requirements of each country.
- I Independent from the blocking of a RTGS Account Holder it will be possible to close the account of a RTGS Account Holder. This closure is a regular process. It becomes effective the next business day or at a predefined business day in the future.

3.3 Types of groups

It is possible to set up and maintain a Banking Group and a Liquidity Transfer Group in RTGS.

The following table summarises the configuration responsibilities for each reference data object related to groups in RTGS and specifies the required communication mode.

Reference data object	Responsible actor	Mode
Banking Group	CB	U2A
Liquidity Transfer Group	CB	U2A

Table 10 - Setup of groups for RTGS

Banking Group

A **Banking Group** allows a number of parties (belonging to one or multiple CBs) to be viewed collectively for certain business purposes, such as oversight and regulation. CBs can set up a Banking Group and specify the name of this group. All actions (setup, modify, delete) will become effective as of the next business day.

Liquidity Transfer Group

A Liquidity Transfer Group is an optional grouping of DCAs. CBs can setup Liquidity Transfer Groups for the purpose of arranging intra-RTGS liquidity transfers between them. A RTGS DCA can be included in one or several Liquidity Transfer Groups.

It is up to CBs to set up and maintain the Liquidity Transfer Groups, while the link of the RTGS DCAs to the respective Liquidity Transfer Group is the responsibility of the RTGS Account Holder. All actions (setup, modify, delete) will become effective as of the next business day.

3.4 Shared reference data

RTGS directory

The RTGS directory provides information on all RTGS Participants that are reachable for payments via a Eurosystem market infrastructure. There is a dedicated directory for all participants in RTGS. This RTGS directory stores all the needed information in order to support the routing of payments.

The structure of the RTGS directory is described later in this chapter. A party can also request that its BIC is not published in the directory. In such a case, its counterparts can make payments to the account linked to this BIC only if the party has previously provided the BIC to them.

RTGS actors may receive the RTGS directory in two ways:

- | Push mode: T2 sends after the EoD processing the full version or the delta version of the RTGS directory to all RTGS Actors with an appropriate report configuration; or
- | Pull mode: at any time during the service hours of CRDM, a RTGS Actor may download either the full version or the delta version of the RTGS directory.

Both, the full and the delta (updated) version will be generated and sent every business day.

The following rules apply to the RTGS directory:

- | RTGS Participants (direct and indirect) with a SWIFT BIC or Non-SWIFT BIC will be issued;
- | The Direct RTGS Participant's correspondents can be listed in the RTGS directory;
- | Every RTGS Participant's BIC/Non-SWIFT-BIC is only listed once, while addressee's and RTGS DCA Holder's ones may occur several times with reference to different RTGS Participants. However, the correspondent can be related to only one RTGS Account Holder; and
- | The publication of an indirect-direct relation does not prevent to route payments to another Direct Participant as mentioned in the RTGS directory when a different routing is known.

The RTGS directory will have the following structure.

Attribute	Description
BIC	It specifies the BIC of the RTGS Participant.
Addressee	It specifies the BIC to be used in the message business header.
Account Holder	It specifies the BIC identifying the RTGS Account Holder.
Institution name	It specifies the RTGS Participant's company name.
City heading	It specifies the RTGS Participant's establishment.
National sorting code	It optionally specifies the RTGS Account Holder's national sorting code.
Main BIC flag	It optionally specifies if the BIC could be used to address the payments if the sender has no other information where to send to.
Type of change	It specifies the type of change with the following possible values: A: added M: modified D: deleted U: unchanged
Valid from	It specifies the date from which the entry is valid.
Valid till	It specifies the date up to which the entry is valid.
Participation type	It specifies the participation type in RTGS with the following possible values: 01 - "Direct" 02 - "Indirect" 03 - multi addressee - Credit institutions 04 - multi addressee - Branch of Direct Participant 05 – access as correspondent – Correspondent (including CB customer) 06 – access as correspondent - Branch of Direct Participant 07 – access as correspondent - Branch of Indirect Participant 08 – access as correspondent - Branch of correspondent

Table 11 - Attributes of the RTGS directory

RTGS calendar

The RTGS calendar specifies the calendar days when RTGS is open and follows the defined business day schedule. Different calendars per currency are set up to operate different closing days. It is up to the operator to set up and maintain the RTGS calendars. All actions (setup, modify, delete) will become effective as of the next business day.

RTGS scheduled events

The RTGS scheduled events specifies the scheduled events that will automatically trigger a specified process within the RTGS component. Each trigger event might trigger one or several other events. The other way round each event might have one or several trigger events. The following table shows the attributes of the RTGS scheduled events. It is up to the operator to set up and maintain the RTGS scheduled events. All actions (setup, modify, delete) will become effective as of the next business day.

Attribute	Description
Scheduled event identifier	It specifies the unique technical identifier of a scheduled event.
Process description	It describes the business process behind the scheduled event.
Scheduled event status	It indicates whether the scheduled event has occurred and the business process has been initiated.
Event triggered timestamp	It specifies the system date and time at which the scheduled event occurred and the business process was triggered.
Repeat flag	It indicates whether another instance of the scheduled event should be created when this instance has occurred.
Trigger date	It specifies either the trigger date and trigger time or the trigger event identifier must be populated.
Trigger event identifier	It specifies the unique technical identifier of another scheduled event that shall trigger this scheduled event when it occurs.

Table 12 - Attributes of the RTGS scheduled events

RTGS currency

The RTGS currency specifies the available settlement currencies in RTGS. It is up to the operator to set up and maintain the settlement currencies. All actions (setup, modify, delete) will become effective as of the next business day.

The following table shows the attributes of the currency in RTGS.

Attribute	Description
Currency code	It specifies the three-character ISO currency identifying the currency.
Currency name	It specifies the name of the currency.
Number of decimals	It specifies the number of decimals for the currency.

Table 13 - Attributes of the RTGS currency

Duplicate check

There are duplicate checks on:

- | files and individual messages received (for A2A communication only); and
- | cash transfer orders at business validation level.

The system parameters regarding duplicate checks for inbound files / messages and cash transfers is defined in the table below.

It is up to the operator to set up and maintain the duplicate check parameter. All actions (setup, modify, delete) will become effective as of the next business day.

Concerned Process	Parameter	Created by	Updated by	Mandatory/Optional	Standard or default value
Message/File duplicate check	Number of business days in the past for duplicate check on files and individual messages	Operator	Operator	M	1 day (same business day)
Cash transfer order duplicate check	Number of business days in the past for duplicate check on cash transfer orders	Operator	Operator	M	3 business days

Table 14 - Attributes of the duplicate check

Warehoused payment period

It is possible to send warehoused payments a few days in advance to RTGS. The payment message shall pass technical and business validation and shall be warehoused until RTGS opens for that date. The system parameter regarding the warehoused payment period is defined in the table below. It is up to the operator to set up and maintain the warehouse payment period parameter. All actions (setup, modify, delete) will become effective as of the next business day. No specific configuration by the T2 Actor is required.

Concerned process	Parameter	Created by	Updated by	Mandatory/optional	Standard or default value
Warehoused payment period	Number of business days in the future for warehoused payments	Operator	Operator	M	ten calendar days

Table 15 - Attributes of the warehoused payment period

Backup payments

This functionality can only be used, once the responsible CB has authorised the affected RTGS Account Holder upon his request to use this functionality.

Activity	Description	Responsibility
Activate backup payments	Activation of back up payment for the RTGS Account Holder.	CB, operator
Deactivate backup payments	Revocation of the allowance to send backup payments.	CB, operator
Deactivate value date	This allows the deactivation of the value date check for the selected RTGS Account Holder.	CB, operator
Reactivate value date	Value date check is reactivated again (no payment with previous date is possible)	CB, operator

Table 16 - Backup payments

For further information please refer to chapter [Backup payments](#) [▶ 84]

3.5 Interaction between RTGS and CRDM

CRDM provides features that allow duly authorised users to setup, update, delete and query all reference data that are shared by multiple services/components (e.g. CLM or RTGS) for their processing activities.

It is ensured that CRDM propagates common reference data (and their changes) to the relevant services and components in a timely and consistent way. Further detailed information can be found in [CRDM features](#) [▶ 234].

As far as RTGS is concerned, all reference data setup and maintenance operations are performed in the CRDM with the exception of changes on local data which are performed in RTGS directly.

Local reference data maintenance within RTGS is limited to the following set of operations with immediate effect:

- | creation of reservation;
- | modification of reservation;
- | deletion of reservation in chapter [Maintain reservation](#) [▶ 344];
- | modification of limits;

I deletion of limits in chapter [Maintain limit](#) [▶ 346].

The reference data stored in CRDM are propagated from the CRDM to RTGS asynchronously, on a daily basis. Only exception is the blocking and unblocking of parties and accounts. This is done in CRDM and is propagated immediately to RTGS.

Every CRDM opening day, an ad hoc event triggers the propagation of all RTGS reference data from CRDM to RTGS. The event takes place at the end of day phase of CRDM business day, so to ensure smooth and complete reference data propagation before RTGS receives the notification that a new business day is starting. The propagated reference data will be loaded into RTGS during the start of day phase.

4 Business day

4.1 T2 Business calendar

The calendar days when a TARGET Service or a common component is opened and follows the defined business day schedule or, contrary, is closed is defined in the common calendar for Euro currency. Each TARGET Service may have a different calendar per currency.

For settlement in Euro currency, CLM, RTGS, T2S and common components are closed on the following days, in addition to Saturdays and Sundays

- New Year's Day (01 January)

- Good Friday (Catholic/Protestant)

- Easter Monday (Catholic/Protestant)

- Labour Day (01 May)

- Christmas Day (25 December)

- Boxing Day (26 December)

For settlement in non-euro currency, T2S and the common components may still be opened in any of the above days if any of the T2S settlement currency RTGS is opened (e.g. on Labour Day (01 May) for settlement in Danish Krone).

On the calendar day which is followed by a CLM closing day, the daily schedule of the next business day runs until the start of the maintenance window. The same business day continues on the next calendar day that is an opening day of CLM and RTGS by finishing the maintenance window.

For example: If the 1st of January is a Tuesday and CLM, RTGS and T2S is closed, the business day 02nd of January already starts on calendar day the 31th of December at 18:45 CET until the maintenance window start on Monday the 1st of January 0:30. The end of the maintenance window is on Tuesday the 02nd of January at 02:30

4.2 Overview

The business day management ensures the proper business day processing for all services and components. There is a schedule for each service and component. Therefore depending on the EoD procedures in a specific service/component, the change of the business day may take place at different times in different services and components. Still, the system allows any settlement or interaction between the services and components only when they are in the same business day.

For example: the business day in TIPS is changed shortly after 18:00. From thereof all instant payments settle with BD+1. However, CLM is change the business day around 18:45 once it has finished with the CLM EoD procedures. Only when CLM has also finished with its SoD procedures, the service becomes available for users with business day BD+1. Then the party can transfer additional liquidity to TIPS for settlement of instant payments.

All business day events are defined and stored in the business day management. The purpose of the business day management is to manage the processes of the different MIS and if necessary to initiate and coordinate overarching processes between different services and components e.g. RTGS, CLM, TIPS, T2S or ECMS. This is achieved by recognising the trigger events associated with the processes and then sending triggers to the relevant services or component to start these required processes. The details of each process to be initiated, and the criteria that define when this should happen, is created and maintained in business day management in a scheduler list.

The business day management process is constantly monitor the scheduler list in order to recognise when the date and time has been reached, or the defined criteria are satisfied, to initiate a defined process. A trigger is then sent immediately to the appropriate service or component for the required process to be initiated within that service or component. For some events, when required, the business day management wait for a feedback from the triggered process (e.g. processes on the critical path), for other events the business day management is not awaiting a feedback. The single business day events are defined in the UDFS of the common component business day management.

All times in the business day management are central European time (CET). The effective time of a business day event is the time of the actual occurrence of the event during the current settlement day. Due to dependencies and interconnections between different business day events, the effective time can be differ from the scheduled time.

The business day management defines also the events upon which the parties can configure event-based standing orders and regular reports.

The table below provides the overview of the main windows during the business day RTGS.

Business day phases	Time
Change of business day	18:45
SOD procedures	18:45-19:00
settlement window for ancillary system settlement and liquidity transfers	19:30-18:00 (interrupted by maintenance window)
Maintenance window	00:30-02:30
Revalidation window for warehoused payments	02:30-03:00
settlement window for customer payments	03:00-17:00

Business day phases	Time
Cut-off for customer payments	17:00
Settlement window for bank to bank payments	03:00 – 18:00
Cut-off bank to bank payments	18:00
EoD procedures	18:00 – 18:45

Table 17 - Overview of the main settlement windows in RTGS

4.3 Detailed description of RTGS business day phases

4.3.1 SoD

The SoD process describes the tasks to be performed by RTGS during this period of the business day as from the opening time of the new business day until begin of availability for users.

The current business day (d) is opened in the evening of the previous TARGET working day.

Actions:

Settlement of ancillary system transactions and liquidity transfer orders are possible. The standing orders from the RTGS DCAs to the ancillary system sub-accounts are settled at the beginning of the ancillary system settlement at 19:30 CET.

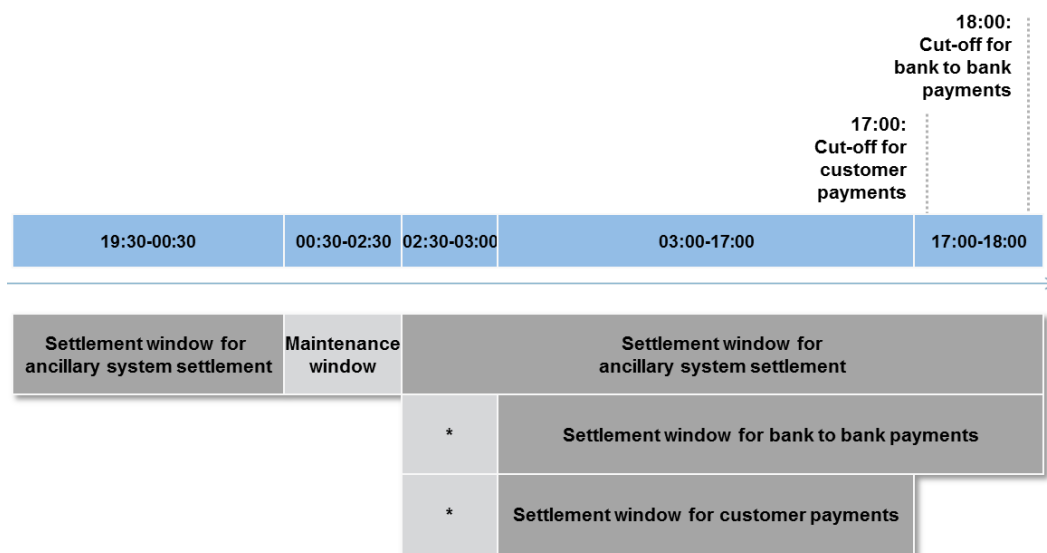
Events:

18:45 SoD procedures

19:30 Start of settlement window of ancillary system transactions and liquidity transfer orders

4.3.2 Settlement windows

This section presents the three settlement windows during the RTGS availability times.



* Revalidation window for warehoused payments

Figure 10 - Settlement windows

Liquidity transfer orders to and from CLM are possible during all settlement windows.

Settlement window for ancillary system settlement

The settlement window for ancillary system settlement starts after the successful completion of SoD processes and ends with the start of EoD processes. It is interrupted by the maintenance window.

During this settlement window RTGS processes all ancillary system settlement procedures.

Settlement window for bank to bank payments

The settlement window for bank to bank payments starts after the successful completion of the rearrangement window for warehoused payments and ends with the cut-off for bank to bank payments at 18:00.

During this settlement window RTGS processes

- | bank to bank payments using the messages pacs.009 FinancialInstitutionCreditTransfer and pacs.009COV FinancialInstitutionCreditTransferCOV
- | bank to bank direct debits using the message pacs.010 FinancialInstitutionDirectDebit
- | warehoused bank to bank payments (originally send via message pacs.009, pacs.009COV or pacs.10) in case the indicated settlement date is reached

Settlement window for customer payments

The settlement window for customer payments starts after the successful completion of the rearrangement window for warehoused payments and ends with the cut-off for customer payments at 17:00.

During this settlement window RTGS processes

- customer payments using the message pacs.008 CustomerCreditTransfer and
- warehoused customer payments (originally send via message pacs.008)

4.3.3 MWI (maintenance periode)

Each TARGET Service or component (CLM, RTGS, T2S and TIPS) has its own opening times, while the change of business day is synchronised across all services. ² The timing of the maintenance windows is also synchronised in all TARGET Services and common components from 00:30 until 02:30 CET, with the exception of TIPS, which operates 24/7/365 and thus has no maintenance window.

During the maintenance window all settlement windows are closed and the access via A2A or U2A is not available. ³

4.3.4 EoD

The EoD process describes the tasks to be performed by RTGS during this period of the business day including the change of business day.

Actions:

- Closure for payment and liquidity transfers(no new payments and liquidity transfers are accepted and rejected).
- Inform business day management about the closure of RTGS.
- Rejection of pending payments and liquidity transfers not executed by the start of the EoD process of the current business day.
- Stop processing of pending reservation order and the remaining reserved amounts are released.
- Rejection of pending verifications for creations, amendments or deletions in four-eyes-principles related to payments
- EoD reporting for RTGS Account Holders depending on the report configuration.

² TIPS is changing the business day after the start of EoD process in CLM.

³ For the sake of efficiency, the Eurosystem will align the maintenance windows across the different TARGET Services/components (i.e. CLM, RTGS, T2S and common components). The indicated timing of the maintenance window (00:30-02:30) is the proposal of the payment community, while the securities community (T2S) is currently used to with the maintenance window between 03:00-05:00. However, the exact timing shall be agreed among all involved communities.

Generate a general ledger file (camt.053) and send it to CLM.

Inform business day management about the EoD processing of RTGS.

Change of business day when EoD processing is finished.

Events:

18:00 cut off for interbank payments and liquidity transfers.

18:45 change of business day.⁴

Shortly after 18:00 (cut-off for interbank payments), when the last settlement algorithm in RTGS is finished, the EoD procedures start with stopping the processing of any pending liquidity transfer orders, payments or modification requests. The parties are notified of the failure and the orders/requests are cancelled accordingly.

Afterwards RTGS produces a general ledger file with all EoD balances of DCAs and send it directly to CLM. At 18:45 the change of business day takes place.⁵ When all tasks of EoD have been completed, the current business day is closed by sending a notification to the business day management.

4.4 Dependencies to other services or components

The business day schedule covers events and phases for all services and components. The following breakdown shows the dependencies between CLM and the other services/components.

Data propagation from CRDM

CRDM offers the possibility to change reference data during the whole business day with the exception of the maintenance window. Any changes are distributed to the other services/components once a day by 17:00. Reference data which needs to be effective on the following business day, have to be entered before the cut-off. Apart from account information like addresses, BICs etc., this also includes liquidity management features, e.g. the management of standing orders and the setting of floor/ceiling thresholds.

Liquidity transfers from/to other services/components

Liquidity transfers from or to other services/components can only be processed, when the value date is the current business day.

⁴ 19:00 on the last day of the reserve maintenance period

⁵ 19:00 at the last day of the minimum reserve maintenance period

Automated liquidity transfers for pending CBO from CLM

In case there is not enough liquidity on an MCA to fully execute a CBO (e.g. overnight deposit, open market operation), CLM pull liquidity from the connected DCA in RTGS with an automated liquidity transfer. Liquidity transfer orders of this kind always have a higher priority in RTGS than immediate or rule-based liquidity transfer orders in RTGS.

Generate and forward general ledger file von RTGS to CLM

To be able to carry out certain processes (e.g. for minimum reserve management and automatic marginal lending), CLM needs to receive the balances from all other services/components e.g. RTGS in a general ledger file (camt.053). A delay of any of the other services/components could lead to a delay of the CLM business day schedule.

5 Business and features description

5.1 Payment types

5.1.1 Overview

The RTGS component enables the settlement of real-time interbank payments, customer payments and liquidity transfers as well as the settlement of ancillary system related payment instructions. The term cash transfer encompasses direct debit instructions, credit transfer orders as well as liquidity transfers. A payment is a credit transfer or direct debit which has been processed in RTGS.

The following types of payments can be submitted by an RTGS Account Holder or ancillary system and are processed in the RTGS component.

Message	Message Name
pacs.004	PaymentReturn
pacs.008	CustomerCreditTransfer
pacs.009COV	FinancialInstitutionCreditTransferCOV
pacs.009	FinancialInstitutionCreditTransfer
pacs.010	FinancialInstitutionDirectDebit

Table 18 - Overview of payments in the RTGS component

Besides payments, also liquidity transfers ([LiquidityCreditTransfer \(camt.050\)](#) [▶ 497]) are processed in the RTGS component. Different users can initiate liquidity transfers. Further details on liquidity management can be found in chapter [Liquidity management](#) [▶ 179].

The sender of a payment, provided that appropriate privileges have been granted, can be:

- | the owner of the account to be debited
- | the owner of the account to be credited (in case of direct debits)
- | a third party (e.g. in case of debtor/creditor of a non-financial institution)
- | a CB acting on behalf of a credit institution (e.g. mandated payments sent by a CB)
- | an ancillary system using interbank payments

Depending on the message subscription certain notifications will be created by RTGS. The relevant message type used by the RTGS component may also depend on who has sent the payment to the RTGS component (i.e. in case the Account Holder is different from the actor submitting the payment).

In general, the sender of a payment receives at maximum one notification related to a payment sent:

- I notification on failure/rejection (mandatory)
- I success notification (optional)

In case payments are sent in a file, the RTGS component checks the validity of the file and splits it into single messages for settlement. Moreover, notifications are provided for the individual messages.

5.1.2 Comparison of different payment types

Class of priority	Urgent	High	Normal
Submission by...	Ancillary systems CBs RTGS Account Holders (to ancillary system technical account procedure D) RTGS Account Holders (CLS payments)	CBs RTGS Account Holders	CBs RTGS Account Holders
Characteristic	Settlement of transactions/ group of transactions between participants of an Ancillary system Transfer of liquidity to ancillary system procedure CLS (continuous linked settlement) pay-ins and pay-outs	Priority payments Processing taking into account the extensive and fast consideration of bilateral payment flow	Highly liquidity saving due to extensive consideration of mutual payment flow Claim of real-time processing takes second place to liquidity saving processing

Table 19 - Classification of priorities

For change of payment types (priority) refer to chapter [Amendment of payments](#) [▶ 113].

Payments instructions functionality for ancillary systems

An ancillary system can send payments to initiate debits of the settlement bank's accounts against credits of other settlement banks (transactions similar to mandated payments). Ancillary systems are allowed to debit the RTGS DCAs of the settlement banks which have authorised the ancillary system.

The ancillary system settlement procedures are described in detail in chapter [Settlement of ancillary systems](#) [▶ 140] and [Ancillary system payment settlement](#) [▶ 314].

Direct debit functionality

Direct debits in RTGS are intended for wholesale purposes only and are restricted to interbank transactions.

The direct debit functionality, which is only available between account holders in the RTGS, can be used by:

- credit institutions
- CBs

The direct debit functionality enables an account holder to debit another account holder's RTGS DCA and credit its own account. In particular, it might be used to offer an efficient cash management service within a group of credit institutions or between different branches of a credit institution.

Direct debits used by credit institutions

In any case, the respective RTGS Account Holders have to agree with the parties allowing debiting their accounts on the terms and conditions for using this service. T2 offers only the general framework.

The RTGS Account Holder authorises another RTGS Account Holder's to issue a direct debit order. He also has to inform his CB, which is responsible to record and administrate the pre-agreements in the CRDM.

The following parameters are used in connection with the direct debit scheme:

- direct debit issuer;
- account to be debited;
- maximum amount per day (for all direct debits independent from the counterparty);
- maximum amount per counterparty;
- maximum amount of any single payment per counterparty.

The RTGS ensures that the conditions mentioned above (if chosen) are met before processing a direct debit request.

Mandated payments submitted by CBs

The mandated payment facility can be used by CBs in case of contingency situations at the level of the RTGS Account Holder. In this situation, the CB can send a credit transfer (with specific message content) on behalf of the failed RTGS Account Holder.

5.1.3 Definition of execution time

RTGS Account Holders have the possibility to determine the settlement time of their payments. The following options are available.

- payments with an "earliest debit time indicator"

l payments with a “latest debit time indicator”

The following table describes payments with a set execution time.

	Earliest debit time indicator	Latest debit time indicator
Features	Payments to be executed from a certain time (message element: FromTime)	<ul style="list-style-type: none"> l Option a: payments to be executed up to a certain time (message element: RejectTime) l Option b: payments which should be executed up to a certain time (only warning indicator) (message element: TillTime)
Effect	<ul style="list-style-type: none"> l The payment is stored until the indicated time (with status earmarked). l At the earliest debit time, the payment runs through the entry disposition. 	<ul style="list-style-type: none"> l Setting a latest debit time only means a special identification via the U2A or A2A query. l In the entry disposition, the payment is treated like any other payment of the same priority.
Processing	If the payment cannot be settled at the earliest debit time, it is queued till cut-off time for payment type is reached (or the payment is revoked).	<p>If the payment cannot be settled until the latest indicated debit time,</p> <ul style="list-style-type: none"> l Option a: the payment is rejected and a failure notification is sent. l Option b: the payment remains in the queue until the cut-off for the respective payment type is reached (or the payment is revoked).

Table 20 - Payments with a set execution time indicators

In case a payment with a “latest debit time indicator” is not executed 15 minutes prior to the defined time, an automatic notification / broadcast in the GUI is triggered. The (optional) notification is directly displayed on top of all screens of the RTGS Account Holder whose account is debited. Further details are provided in the RTGS user handbook.

Note: In case the message element CLSTime is used, the payment is treated in the same way as a payment with a “latest debit time indicator”, option b.

It is possible to combine the “earliest debit time indicator” with the “latest debit time indicator” (either option a or option b). In case of option a, the payment is meant to be executed during the indicated period.

The defined execution time of a payment can be changed if the payment is not executed yet. Further details on the effect of changing the settlement time can be found in the chapter [Amendment of payments](#) [▶ 113].

If TillTime and RejectTime are both provided in the payment then only the TillTime is considered. Therefore, it is strictly recommended to provide only one of the two possible “latest debit time indicators” in a payment. It is not possible to change the “earliest debit time indicator” of a payment which is already queued due to the fact that the original “earliest debit time indicator” had been reached and it was already tried to settle this payment.

5.1.4 Warehouse functionality

Basics

It is possible to submit payments up to 10 calendar days in advance to the intended settlement date. In this case, the payment message is warehoused until the RTGS component opens for that business date.

Ancillary system payment instructions as well as liquidity transfers cannot be sent as warehoused payments.

Note: In case a change of ISO 20022 standards or formats is performed, warehoused payments with an execution time beyond this point in time cannot be stored in the RTGS component. This is ensured by the RTGS component. The affected payments are rejected on the effective date of that change with an appropriate error code.

Rules

The validation of warehoused payments is a layered approach:

- | ESMIG check whether the payment message is well-formed on the day of submission;
- | schema validation by the RTGS component already on the day of submission;
- | validation of the indicated settlement date;
- | content check (e.g. valid BICs) on the indicated settlement day.

The last two validations are carried out at SoD of every day of the warehoused period including the settlement day.

Processing on settlement day

On the indicated settlement day with the SoD trade phase the warehoused payments are processed by the RTGS component. These payments are processed with an entry timestamp identical to the start of day time settlement phase and on top of the queue of incoming payments which have the same priority. They are immediately settled if enough liquidity is available (normal processing of payments in the entry disposition, see chapter [Entry disposition](#) [▶ 125]). Otherwise they are queued until the settlement attempt is successful (see chapter [Dissolution of the payment queue](#) [▶ 132]).

Exception: Warehoused payments with an “earliest debit time indicator” are queued until the set execution time is reached.

Information and control functions

Warehoused payments benefit from the same functionality via U2A or A2A as queued payments:

- | transparency about the status and other detailed information about the payment;
- | revocation;
- | change of priority;
- | change of execution time (“earliest and latest debit time indicator”) if set in the warehoused payment.

5.1.5 Backup payments

In case of a technical system outage a RTGS Account Holder might not be in a position to send payments and to receive payments from the RTGS component.

Such breakdown on the side of the RTGS Account Holder may result in

1. pay-in obligations in other systems like CLS cannot be met
2. liquidity being built up on the affected RTGS Account Holder’s DCA in case other RTGS Account Holders submitted or continue to submit payments in favour of the affected account holder.

In order to give the affected account holder a possibility to reduce the business impact of the outage, the RTGS component offers a functionality to generate payments by using the so-called backup payment functionality in U2A mode (and not A2A).

In general, this functionality is blocked and it can only be used, once the responsible CB has authorised its usage upon request of the affected RTGS Account Holder. There are two categories of backup payments available:

1. backup contingency payments to CLS/EURO1; and
2. backup liquidity redistribution payments to other RTGS Account Holders.

If need be, the CB responsible for the affected RTGS Account Holder can act on behalf. Further details are provided in the RTGS user handbook.

5.1.5.1 Backup contingency payments

Objective

Backup contingency payments are intended to meet obligations and demands arising from the settlement and funding process of other systems. In case the functionality is “switched on” by the responsible CB, pre-defined templates are available in the GUI (for CLS pay-ins, payments to the EURO1 collateral account, pay-ins to the EURO1 prefunding account related to the liquidity bridge between the RTGS component and EURO1).

Rules for CLS payments

The table below gives the rules for backup contingency payments to CLS.

Payment priority	Urgent
Generation	Via the GUI
Message type	pacs.009
Sender of this message	RTGS
Receiver of this message	CLS
Fields for input via GUI	<p>EndToEndId: End-to-end identification</p> <p>IntrBkSttlmAmt: Currency, Amount</p> <p>Dbtr/FinInstnId/BICFI: BIC of the ordering institution</p> <p>Cdtr/FinInstnId/BICFI: BIC of the receiver in CLS (ordering party)</p> <p>CLSTm (format: ISOTime hh:mm): optional; CLS time if captured is treated as latest debit time</p> <p>Note: Element will not be used in pacs.009 if no time has been entered.</p>
Fields predefined (cannot be changed)	InstrId: Instruction Identification - assigned by RTGS
Tag in the payment message	LclInstrm: BACP
Tag in the statement message	LclInstrm: BACP
Tag in the A2A payment queue	PmtTp: BACP
Tag in the U2A payment queue	Backup payment

Table 21 - CLS backup payments

Rules for backup contingency payments to EURO1 collateral account

The table below gives the rules for backup contingency payments to the EBA related to EURO1 collateral account.

Payment priority	High
Generation	Via the GUI
Message type	pacs.009

Sender of this message	RTGS
Receiver of this message	EBA (for collateral account)
Fields for input via GUI	EndToEndId: End-to-end identification IntrBkSttlmAmt: Currency, Amount Dbtr/FinInstnId/BICFI: BIC of the ordering institution
Fields predefined (cannot be changed)	InstrId: Instruction Identification - assigned by RTGS Cdtr/FinInstnId/BICFI:EBA BIC (related to collateral account or pre
Tag in the payment message	LclInstrm: BACP
Tag in the statement message	LclInstrm: BACP
Tag in the A2A payment queue	PmtTp: BACP
Tag in the U2A payment queue	Backup payment

Table 22 - EURO1 collateral account backup payments

Rules for backup contingency payments to EURO1 pre-settlement account (liquidity bridge)

The table below gives the rules for backup contingency payments to the EURO1 pre-settlement account (liquidity bridge between the RTGS component and EURO1).

Payment priority	High
Generation	Via the GUI
Message type	pacs.009
Sender of this message	RTGS
Receiver of this message	EBA (for pre-settlement account)
Fields for input via GUI	EndToEndId: End-to-end identification IntrBkSttlmAmt: Currency, Amount Dbtr/FinInstnId/BICFI: BIC of the ordering institution
Fields predefined (cannot be changed)	InstrId: TRN assigned by RTGS Cdtr/FinInstnId/BICFI: EBA BIC (related to pre-settlement account) pre
Tag in the payment message	LclInstrm: BACP

Tag in the statement message	LclInstrm: BACP
Tag in the A2A payment queue	PmtTp: BACP
Tag in the U2A payment queue	Backup payment

Table 23 - EURO1 liquidity bridge backup payment

5.1.5.2 Backup liquidity redistribution payments

Objective

Backup liquidity redistribution payments are used for the provision of excess liquidity accumulated on the RTGS DCA of the affected RTGS Account Holder. It aims to reduce the likelihood of possible liquidity shortage within the RTGS component.

As the recipient can be any other RTGS Account Holder, such payments can also be used to cater for obligations and demands arising from the settlement and funding processes for other systems than those explicitly covered by the backup contingency payments described above.

Rules for backup liquidity redistribution payments

The table below gives the rules for backup liquidity redistribution payments.

Redistributing liquidity payments can be transferred to...	RTGS Account Holders
Payment priority	High
Generation	Via the GUI
Message type	pacs.009
Sender of this message	RTGS DN
Receiver of this message	According to the routing configuration of the instructed agent
Fields for input via GUI	IntrBkStlmAmt: Amount Cdtr/FinInstnId/BICFI: BIC of the credited party
Fields predefined (cannot be changed)	InstrId: Instruction Identification - assigned by RTGS EndToEndId: End-to-end identification (same content as InstrId) Dbtr/FinInstnId/BICFI: BIC of the ordering institution
Tag in the payment message	LclInstrm: BACP

Tag in the statement message	LclInstrm: BACP
Tag in the A2A payment queue	PmtTp: BACP
Tag in the U2A payment queue	Backup payment

Table 24 - Backup liquidity redistribution payments

5.1.5.3 Rules for backup payments

5.1.5.3.1 Generation

Both, backup contingency and backup liquidity redistribution payments are generated according to the following procedure.

Step	Action
1	Information to the CB responsible about the affected RTGS Account Holder . Result: The CB activates the backup functionality in the GUI for the RTGS Account Holder concerned.
2	GUI users of the affected RTGS Account Holder have to re-login to the GUI before being able to open the backup functionality. Subsequent generation of backup contingency and backup liquidity redistribution payments in the GUI by users of the affected RTGS Account Holder or by the CB acting on behalf of affected RTGS Account Holder.

Table 25 - General procedure for generating backup payments

Further information on the GUI interactions can be found in the RTGS user handbook.

Protection against unauthorised generation of backup payments, including backup contingency as well as backup liquidity redistribution payments is ensured because

- | the generation of backup payments must first be activated by the CB responsible for the RTGS Account Holder facing technical problems (i.e. affected RTGS Account Holder),
- | the number of people authorised to generate these payments, can be kept small (separate role in the GUI),
- | the “four eyes“ principle (different people responsible for initial recording and release) is obligatory,
- | as far as possible, backup payments are generated automatically in the RTGS component, e.g. references, instructed/instructing agents etc. Only fields where an input is needed from the RTGS Account Holder will be provided via GUI.

5.1.5.3.2 Notification of affected account holder (sender)

On request, the affected RTGS Account Holder as sender of a backup contingency or backup liquidity redistribution payment receives a notification ([BankToCustomerDebitCreditNotification \(camt.054\)](#) [522]). Such notification includes the code word BACP. The debit notification reaches it as soon as its connection is operational again.

5.1.5.3.3 Notification to the receiver

The receiver gets a payment, i.e. a pacs.009 which includes the code word BACP.

5.1.5.3.4 Subsequent delivery of single payments

Basic principles

Backup contingency payments as well as liquidity redistribution payments using the backup functionality are considered as payments on their own. This means that when resuming normal operations there is no need to resend the same or a similar payment via the standard channel to confirm the backup payment.

If, following the recovery of the failed account holder, the original payments, which may have already been queued within the RTGS Account Holder's internal environment, are still released by the affected RTGS Account Holder by mistake, there is no control in the RTGS component which prevents these payments from being processed. It is in the sole responsibility of the affected RTGS Account Holder as sender to follow up on these payments with the receiver of the funds.

If the affected RTGS Account Holder resumes normal processing on the same day before the closing of the day-trade phase, payments still to be processed on the account holder's side can be released towards the RTGS component.

If the affected RTGS Account Holder resumes normal operations only on the following day or later, it may choose between two options for the pending payments still to be processed depending on the set-up of its processing engine.

- | transmission of the pending payments with the current (new) settlement date in the tag interbank settlement date; or
- | transmission of the pending payments with the past (original) settlement date in the tag interbank settlement date

Independent from the date contained in the tag interbank settlement date, on the RTGS DCAs all payments are booked with the business day applicable at the time when these payments arrive and are settled, as the RTGS component provides only for same day settlement.

Transmission of unprocessed payments with new settlement date

These payments are released by the affected account holder after resuming normal operations like any other new payments; there is no special treatment of these payments necessary.

Transmission of unprocessed payments with original settlement date

Choosing this option, the affected RTGS Account Holder has to take into account the following process for executing the payments with original settlement date.

- | The affected RTGS Account Holder must request the temporary lifting of the settlement date check to the CB which switches off the settlement date check for the current business day.
- | If more than the current business day is required for dealing with the unprocessed payments with old settlement date, the lifting of the settlement date check for any consecutive business day has to be requested separately at the beginning of the concerned day trade phases.
- | Once having completed the sending of payments with original (past) settlement date, the affected RTGS Account Holder should inform the CB in order to reactivate the settlement date check with immediate effect.

Note: Not all recipients may be in the position to process payments with a settlement date in the past. Further information in this respect can be found in the Information Guide for TARGET2 users.

Account statement sorted by settlement date

The bookings in the RTGS component are sorted by payment settlement date in the tag interbank settlement date of the payment message. One account statement (camt.053) is issued and all settled payments are included.

5.1.6 Payment priorities

In general, all payments submitted to the RTGS component are settled immediately, provided that sufficient liquidity is available on the RTGS DCA of the RTGS Account Holder and other relevant conditions (e.g. limits) are met.

Depending on their urgency of their settlement, payments can be submitted by the sender using one of the following priorities in the RTGS component:

- | urgent
- | high
- | normal

All priorities have specific characteristics.

Some of the priorities can only be used by certain users. Within a priority no further prioritization is possible (no sub-priorities). That means “urgent payments” are settled following the principles of entry disposition and execution of offsetting payments (see chapter [Settlement of payments in the entry disposition](#) [▶ 126]).

If no priority class is indicated in the payment, the payment is handled as payment with normal priority.

The priority class “urgent” is only available for

- | ancillary systems;
- | CLS pay-ins sent by a RTGS Account Holder;
- | immediate liquidity transfers generated in the RTGS component to transfer liquidity to another service/component;
- | automated inter-service liquidity transfers generated in CLM due to pending CBOs to “pull” liquidity from the RTGS component;
- | rule-based liquidity transfer orders
- | standing order liquidity transfers.

Note: Automated inter-service liquidity transfers to transfer liquidity from the RTGS component to CLM due to pending CBO always have the top priority (i.e. top of the urgent queue).

Further details on changing the priority of a payment can be found in chapter [Comprehensive queue management](#) [▶ 128].

5.2 Payments processing and settlement of payments

5.2.1 Overview

The aim of the process is to illustrate the authorised parties to submit payments. A payment can be submitted to and received from the RTGS component by

- | the owner of the account to be debited
- | the owner of the account to be credited (in case of direct debits)
- | a third party (e.g. in case of an ancillary system)
- | a CB acting on behalf of a RTGS Account Holder (mandated payments).
- | a multi-addressee; the so-called multi-addressee access implies that an entity is authorised to submit and receive payments directly without having an own RTGS DCA.

Addressable BICs as well as indirect participants do not send directly any payments to the RTGS component. This is always done via the RTGS Account Holder.

The following table provides an overview of the features for payment messages linked with the way of initiation.

Name	Customer payment	Bank to bank payment	Direct debit	Payment return
Message for A2A initiation	pacs.008	pacs.009/pacs.009 COV	pacs.010	pacs.004
U2A mode initiation	Not provided	Only for backup contingency payments or backup liquidity redistribution payments	Not provided	Not provided
Possible priority	High Normal	Urgent (CB, ancillary system , CLS pay-ins) High Normal	Urgent (CBs only) High Normal	Normal
Settlement time	Earliest debit time indicator (FromTime) Latest debit time indicator (TillTime) (RejectTime)	Earliest debit time indicator (FromTime) Latest debit time indicator (TillTime) (RejectTime)	Earliest debit time indicator (FromTime) Latest debit time indicator (TillTime) (RejectTime)	No indication possible

Table 26 - Features to be used for different payment messages

The RTGS DCAs to be debited and credited are not necessarily linked to the BICs mentioned in BAH. They have to be taken from the respective payment (pacs or camt). After simultaneous booking on the RTGS DCAs, the payment/the liquidity transfer is final and irrevocable.

Note: A payment included in a running algorithm cannot be revoked - although it might not yet be final.

5.2.2 Concept of payment submitters

General aspects

The RTGS component enables different types of participation. Depending on the participation type ([Concept of party in RTGS](#) [49]) different ways of payment submission to RTGS are possible. Aim of this chapter is to illustrate the message flows in case a payment is submitted by

- a Direct Participant;

a participant with indirect access/a participant as an “addressable BIC“. If the originator of the payment is a RTGS Account Holder with indirect access/a participant as an “addressable BIC” he has to instruct the related direct RTGS Account Holder to submit the payment;

a multi-addressee access participant;

an ancillary system sending plain payments (acting as so called third party);

CB on behalf of a participant (mandated payments).

The following table aims to provide an overview about way of payment submissions and related messages.

Originator of payment	Submission of payment	Payment types			
		pac.004 PaymentRe- turn	pac.008 CustomerCredit- Transfer	pac.009 FinancialInstitu- tionCreditTrans- fer (COV)	pac.010 FinancialInstitu- tionDirectDebit
Direct Participant	Directly	Y	Y	Y	Y
indirect access/a participant as an “addressable BIC“	Via Direct Participant	Y	Y	Y	Y
Multi-addressee access participant	Directly	Y	Y	Y	Y
Ancillary system	Directly	N	N	Y	N
CB on behalf of a participant (mandated payment)	Directly	Y	Y	Y	Y

Table 27 - Table of possible payment types

5.2.3 Flow of payment related messages

General remarks

The chapter provides some examples of relevant cases for flows of payment messages and related notifications including respective details.

Due to the fact that the payment flows between sending and receiving RTGS Party can be mirrored the following flow descriptions will be limited to only one direction of the flow, e.g. for the business case "Payment message from a direct RTGS Account Holder to an indirect participant the opposite direction "from an indirect participant to a direct RTGS Account Holder" is not illustrated.

In order to ease the readability pacs.009COV is not mentioned separately, but included in the description for [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585]

5.2.3.1 Payments sent from a direct/indirect RTGS Account Holder to another direct RTGS Account Holder

Note: The following message flows and corresponding process descriptions are based on the interaction between RTGS Account Holders. If the originator of the payment is an indirect participant one step has to be added before the current step 1. If the payment is in favour of an indirect participant one step has to be added after the direct RTGS Account Holder has received the payment message (i.e. after step 4).

Case 1: payment credit message with positive validation and settlement

The following payment flow illustrates the payment messaging on basis of a pacs.008/pacs.009 and with regard to the RTGS component.

Message flow

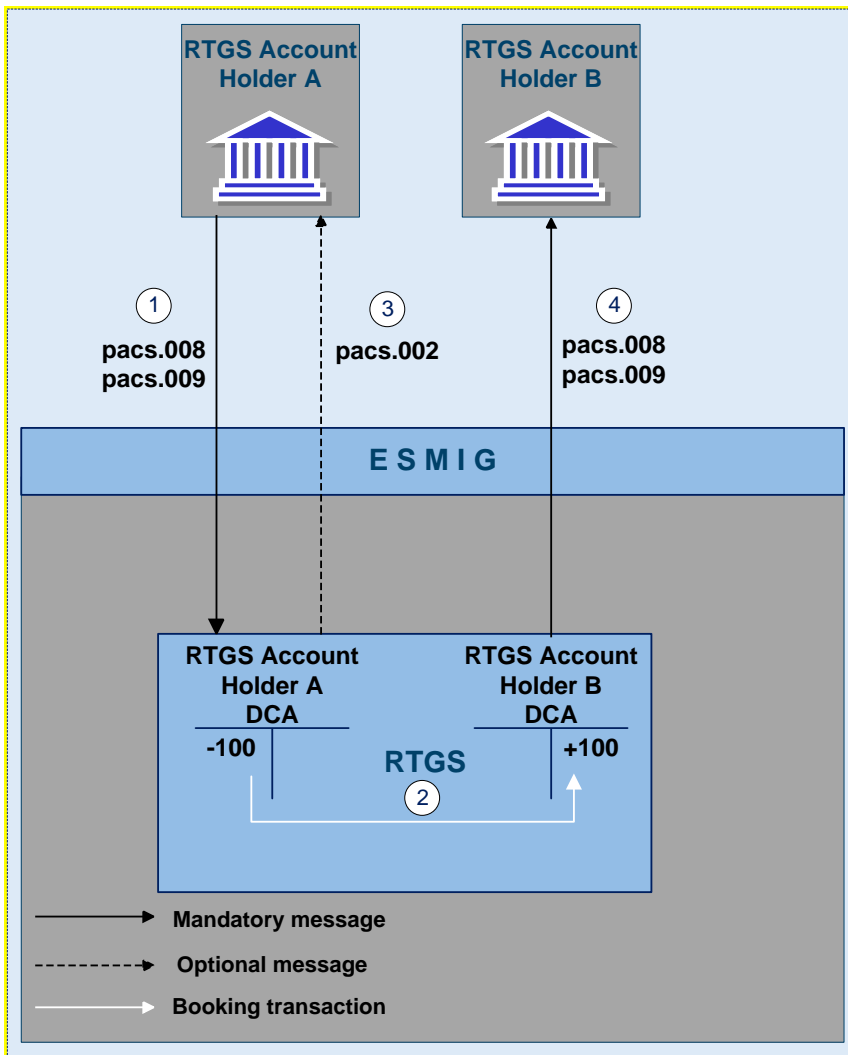


Figure 11 - pacs.008 – CustomerCreditTransfer / pacs.009 - FinancialInstitutionCreditTransfer

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	The RTGS Account Holder A sends a pacs.008/pacs.009 via ESMIG to the RTGS component.
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCAs of RTGS Account Holders A and B
3	RTGS component via ESMIG to RTGS Account Holder A	Booking confirmation pacs.002 to RTGS Account Holder A generated by the RTGS component (optional)
4	RTGS component via ESMIG to RTGS Account Holder B	Creation and forwarding of pacs.008/pacs.009 to RTGS Account Holder B generated by the RTGS component (mandatory)

Table 28 - Payment messaging on the basis of pacs.008/pacs.009

Used messages

- | [CustomerCreditTransfer \(pacs.008\)](#) [▶ 577]
- | [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585]
- | [PaymentStatusReport \(pacs.002\)](#) [▶ 568]

Case 2: payment return message with positive validation and settlement

The following payment flow illustrates the payment messaging on basis of a pacs.004 and with regard to the RTGS component. The pacs.004 is used to return an already settled pac.008 or pacs.009.

Message flow

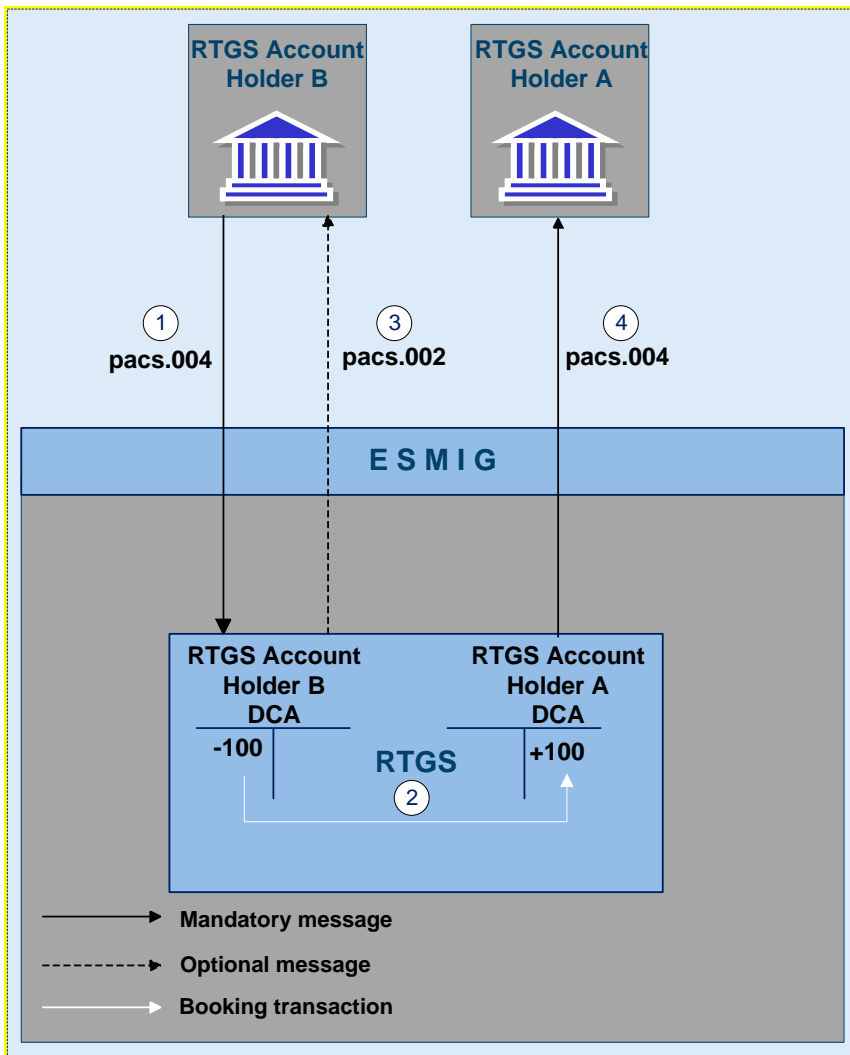


Figure 12 - pacs.004 - PaymentReturn

Process description

Step	Processing in/between	Description
1	RTGS Account Holder B via ESMIG to the RTGS component	RTGS Account Holder B sends a pacs.004 via ESMIG to the RTGS component.
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCAs of Account Holder s B and A
3	RTGS component via ESMIG to RTGS Account Holder B	Booking confirmation pacs.002 by the RTGS component (optional) via ESMIG to RTGS Account Holder B
4	RTGS component via ESMIG to RTGS Account Holder A	Creation and forwarding of pacs.004 by the RTGS component via ESMIG to RTGS Account Holder A (mandatory)

Table 29 - Payment messaging on the basis of pacs.004

Used messages

- ▶ [PaymentReturn \(pacs.004\)](#) [571]
- ▶ [PaymentStatusReport \(pacs.002\)](#) [568]

Case 3: payment debit message with positive validation and settlement

The following payment flow illustrates the payment messaging on basis of a pacs.010 and with regard to the RTGS component.

Message flow

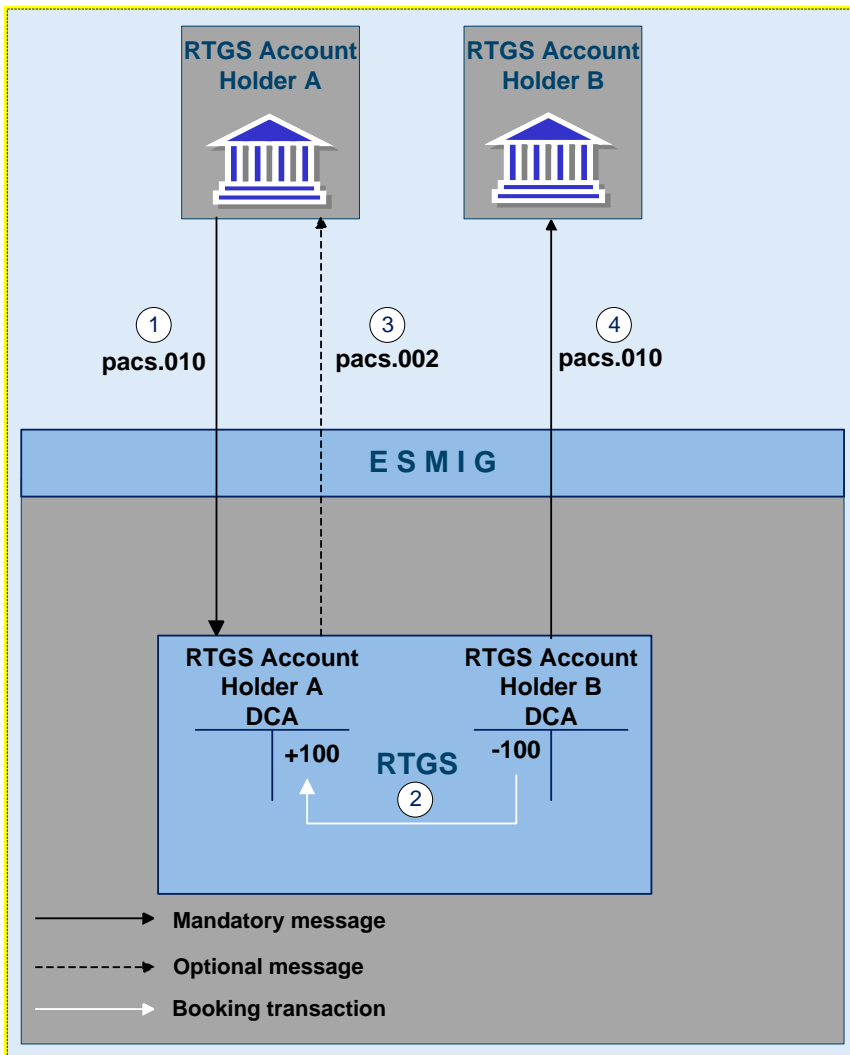


Figure 13 - pacs.010 FinancialInstitutionDirectDebit

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	RTGS Account Holder A sends a pacs.010 via ESMIG to the RTGS component.
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCAs of RTGS Account Holders A and B
3	RTGS component via ESMIG to RTGS Account Holder A	Booking confirmation pacs.002 by the RTGS component (optional) via ESMIG to RTGS Account Holder A
4	RTGS component via ESMIG to RTGS Account Holder B	Creation and forwarding of pacs.010 by the RTGS component via ESMIG to RTGS Account Holder B (mandatory)

Table 30 - Payment messaging on the basis of pacs.010

Used messages

- 1 [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 603]
- 1 [PaymentStatusReport \(pacs.002\)](#) [▶ 568]

5.2.3.2 Payments sent from a multi-addressee access participant to another RTGS Account Holder

Multi-addressee access participants are authorised by a RTGS Account Holder to submit and receive payments which are settled on the RTGS DCA of the RTGS Account Holder.

Note: The following message flow and process description illustrates a payment messaging on the basis of pacs.009 send by a multi-addressee access participant. The multi-addressee access participant is authorised by the RTGS Account Holder A. In case of receiving payments in favour of the multi-addressee access participant the flow is to be understood vice versa.

Message flow

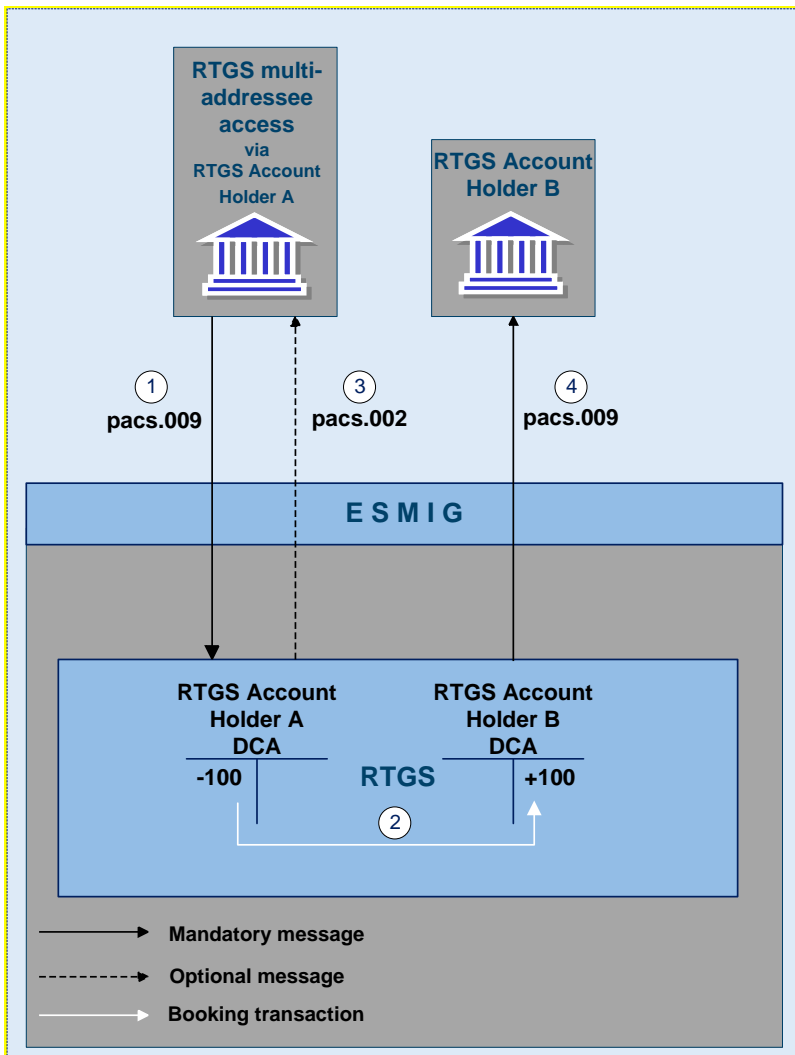


Figure 14 - pacs.009 – FinancialInstitutionCreditTransfer (sent from a multi-addressee access participant)

Process description

Step	Processing in/between	Description
1	RTGS multi-addressee access participant via ESMIG to the RTGS component	The RTGS multi-addressee access participant sends a pacs.009 via ESMIG to the RTGS component.
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCAs of RTGS Account Holders A and B
3	RTGS component via ESMIG to RTGS multi-addressee access participant	Booking confirmation pacs.002 to RTGS multi-addressee access participant generated by the RTGS component (optional)
4	RTGS component via ESMIG to RTGS Account Holder B	Creation and forwarding of pacs.009 to RTGS Account Holder B generated by the RTGS component (mandatory)

Table 31 - Payment messaging on the basis of pacs.009

Used messages

- ▮ [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\) \[▶ 585\]](#)
- ▮ [PaymentStatusReport \(pacs.002\) \[▶ 568\]](#)

5.2.3.3 Payments sent from an ancillary system

Case 1: payment credit message with positive validation and settlement

The following payment flow illustrates the payment messaging on basis of a pacs.009 and with regard to the RTGS component.

Message flow

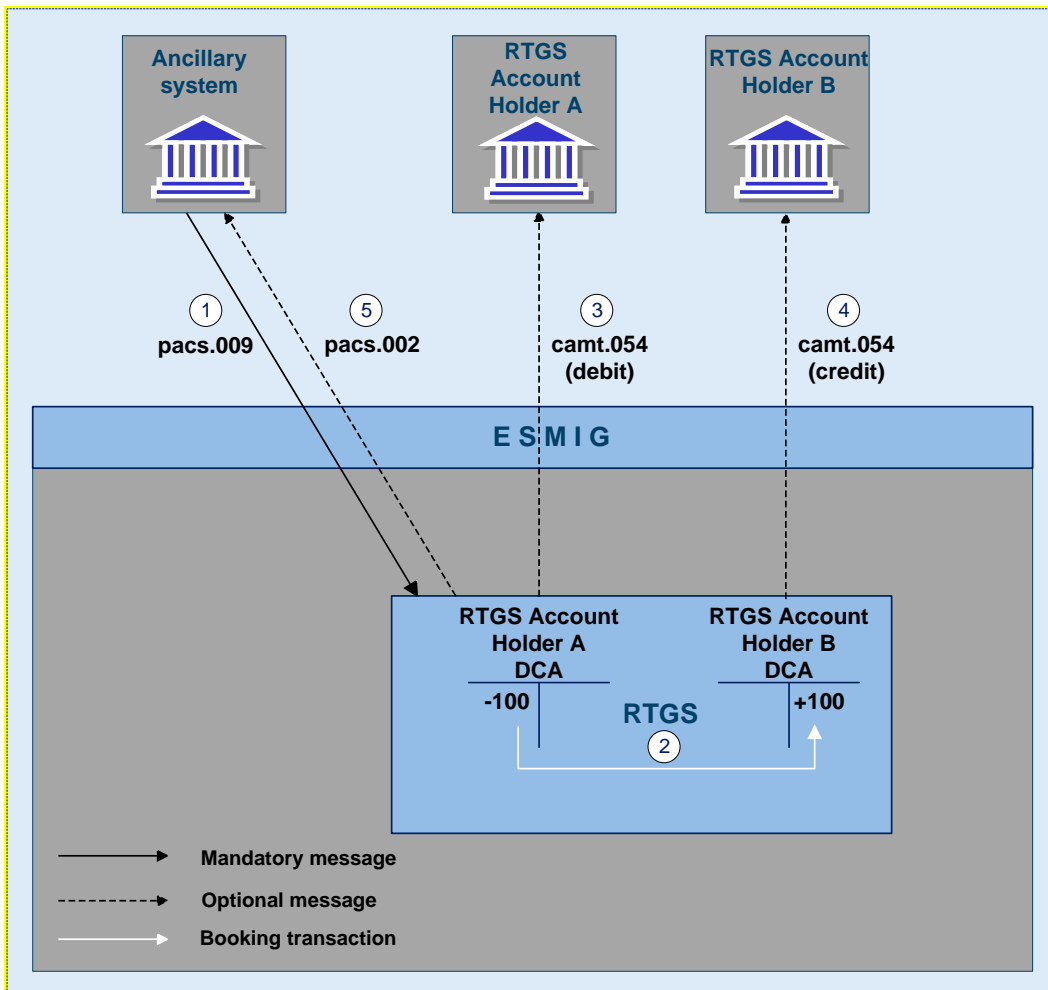


Figure 15 - pacs.009 FinancialInstitutionCreditTransfer (submitted by ancillary system)

Process description

Step	Processing in/between	Description
1	Ancillary system via ESMIG to the RTGS component	The ancillary system sends a pacs.009 via ESMIG to the RTGS component
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCAs of RTGS Account Holders A and B

Step	Processing in/between	Description
3	RTGS component via ESMIG to RTGS Account Holder A	Creation and forwarding of camt.054 (debit) to RTGS Account Holder A (optional)
4	RTGS component via ESMIG to RTGS Account Holder B	Creation and forwarding of camt.054 (credit) to RTGS Account Holder B (optional)
5	RTGS component via ESMIG to ancillary system	Booking confirmation pacs.002 to ancillary system generated by the RTGS component (optional)

Table 32 - Payment messaging on the basis of pacs.009 (submitted by an ancillary system)

Used messages

- ▮ [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\) \[585\]](#)
- ▮ [PaymentStatusReport \(pacs.002\) \[568\]](#)
- ▮ [BankToCustomerDebitCreditNotification \(camt.054\) \[522\]](#)

Case 2: file submission from an ancillary system

An ancillary system as submitting actor sends a file including business file header and several pacs.009 messages via ESMIG to the RTGS component.

Message flow

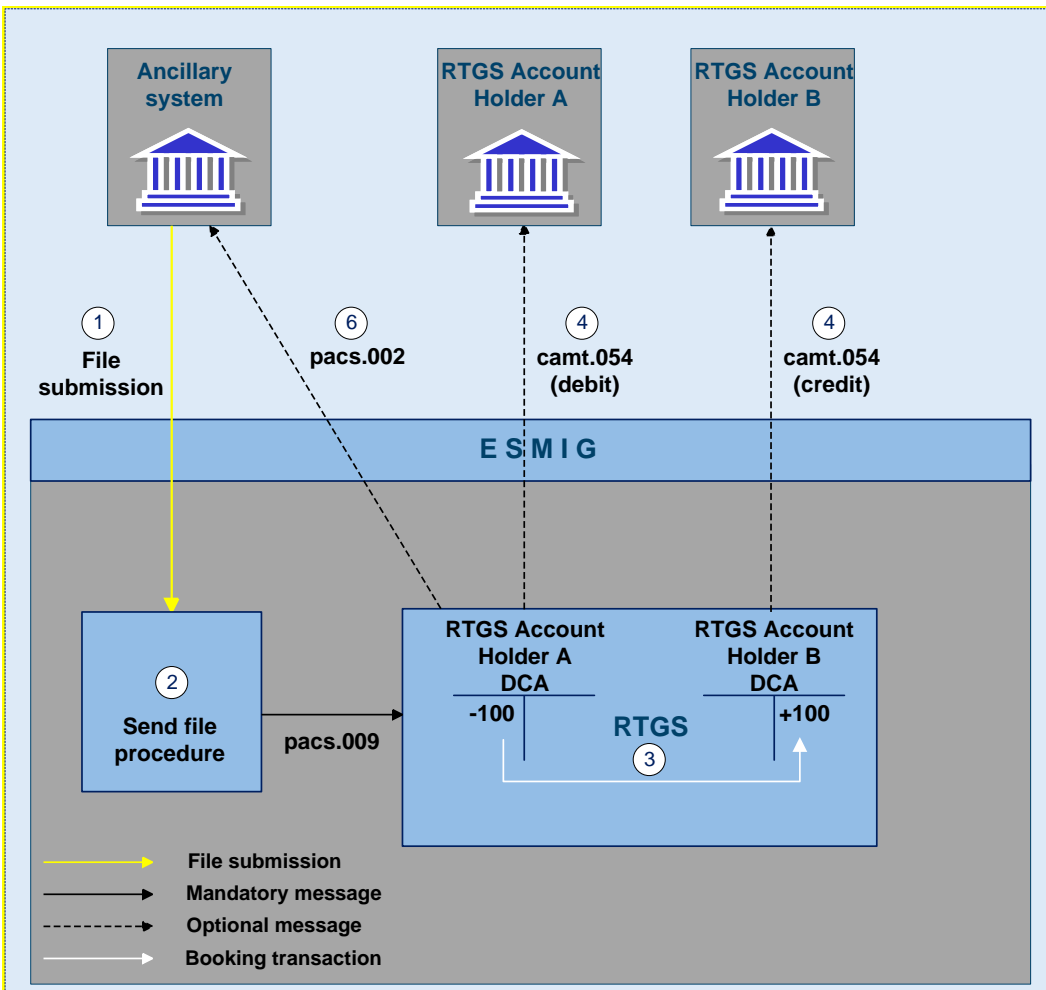


Figure 16 - pacs.009 FinancialInstitutionCreditTransfer (submitted by ancillary system via file)

Process description

Step	Processing in/between	Description
1	Ancillary system via ESMIG to the RTGS component	The ancillary system sends a file via ESMIG to the RTGS component
2	RTGS component	<p>"Send file" procedure applies -</p> <p>Schema validation and business validation in the RTGS component successful</p> <p>File splitting and submission of the single pacs.009 messages for further processing</p>
3	RTGS component	Simultaneous booking on the RTGS DCAs of RTGS Account Holders A and B

Step	Processing in/between	Description
4	RTGS component via ESMIG to RTGS Account Holder A	Creation and forwarding of camt.054 (debit) to RTGS Account Holder A (optional)
5	RTGS component via ESMIG to RTGS Account Holder B	Creation and forwarding of camt.054 (credit) to RTGS Account Holder B (optional)
6	RTGS component via ESMIG to ancillary system	Booking confirmation pacs.002 to the ancillary system generated by the RTGS component (optional)

Table 33 - Payment messaging on the basis of pacs.009 (submitted by an ancillary system via file)

Used messages

- | [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\) \[▶ 585\]](#)
- | [PaymentStatusReport \(pacs.002\) \[▶ 568\]](#)
- | [BankToCustomerDebitCreditNotification \(camt.054\) \[▶ 522\]](#)

5.2.3.4 Payments sent from a CB on behalf of a RTGS Account Holder (mandated payment) to another direct RTGS Account Holder

The mandated payment is sent by the responsible CB on behalf of its direct RTGS Account Holder in the case of contingency situations. In this situation, the CB can send payments (either credit transfer or direct debit) on behalf of the failed direct RTGS Account Holder. As indication of a mandated payment the codeword MANP is used.

The following payment flow illustrates a pacs.009 – mandated payment (codeword: MANP)

Message flow

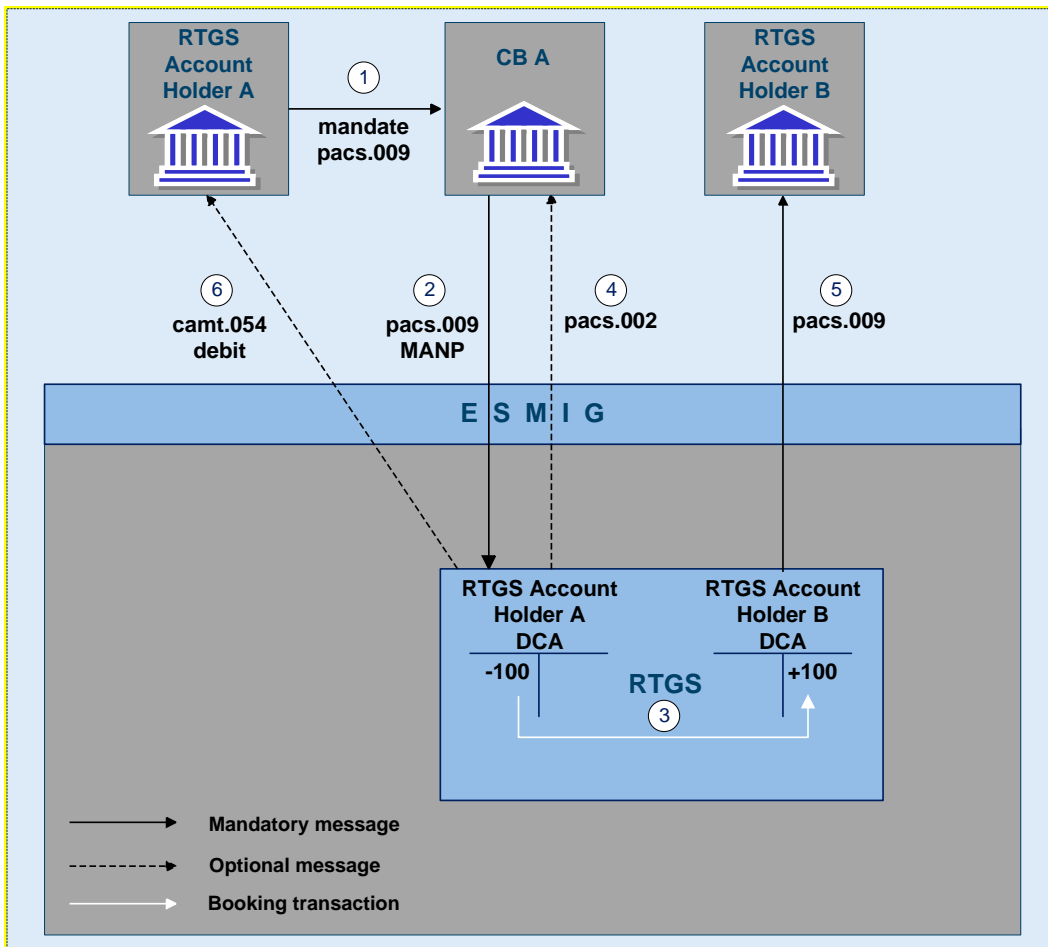


Figure 17 - pacs.009 – FinancialInstitutionCreditTransfer – mandated payment (codeword: MANP)

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A to the responsible CB A	The RTGS Account Holder A mandates its responsible CB A to initiate a pacs.009 on behalf
2	CB A via ESMIG to the RTGS component	The CB A sends a pacs.009 (flagged with codeword MANP) on behalf of the RTGS Account Holder A via ESMIG to the RTGS component.
3	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCAs of RTGS Account Holders A and B

Step	Processing in/between	Description
4	RTGS component via ESMIG to CB A	Booking confirmation pacs.002 to CB A generated by the RTGS component (optional)
5	RTGS component via ESMIG to RTGS Account Holder B	Creation and forwarding of pacs.009 to RTGS Account Holder B generated by the RTGS component (mandatory)
6	RTGS component via ESMIG to RTGS Account Holder A	Creation and forwarding of camt.054 (debit) to RTGS DCA Holder A (optional)

Table 34 - Payment messaging on the basis of pacs.009 (mandated payment)

Used messages

- | [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585]
- | [PaymentStatusReport \(pacs.002\)](#) [▶ 568]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522]

5.2.4 Rejection of payments

The term “rejection” refers to the rejection of a payment by the RTGS component and for different reasons a payment can be rejected.

In case

- | the technical validation in the RTGS component fails, the RTGS component creates and forwards a notification ([ReceiptAcknowledgement \(admi.007\)](#) [▶ 391]) to the submitter of the payment.
- | the business validation in the RTGS component fails, the RTGS component creates and forwards a rejection notification ([PaymentStatusReport \(pacs.002\)](#) [▶ 568]) to the submitter of the payment. The pacs.002 refers to the original instruction by means of references and a set of elements from the original instruction.
- | the EoD phase is started, the RTGS component creates and forwards a rejection notification (xx) to the submitter of the payment.
- | **Note:** The sending of a negative notifications is mandatory and not subject to message subscription.

Technical validation

The following technical validations are inter alia performed in CLM interface:

- | Schema validation - syntax, format and structure of the message are compliant (e.g. check that all mandatory field in the message are populated)

In general CLM continues the technical validation even if a first error has been detected. In case the technical validation was not successful an adm007 is sent to the instructing party (meaning the CB) indicating which error occurred (all negative results in form of error codes are included).

In case the CB instructed the CBO via U2A, the rejection notification is displayed directly on the screen. For further details please refer to the CLM user handbook.

The RTGS component performs various checks during the business validation and does not stop after the first negative validation result, but goes on with the business validation as there could be further negative validation results in the subsequent checks. Consequently, the rejection notification sent by the RTGS component includes the error codes for all negative business validations.

The following business validations are inter alia performed in the RTGS component:

- | payment type specific checks
- | duplicate check
- | process specific authorisation checks
- | value date check
- | field and reference data checks
- | direct debit check
- | check of backup payments
- | mandated payment check
- | account checks

Further information on the relevant business rules and the respective error codes are listed in chapter [Index of business rules and error codes](#) [▶ 670].

5.2.4.1 Technical validations

A file has to be delivered with a file header. A message has to be delivered including a BAH.

The following payment flow illustrates a technical validation failure in the RTGS component on basis of an underlying pacs.008/pacs.009/pacs.010/pacs.004.

Message flow

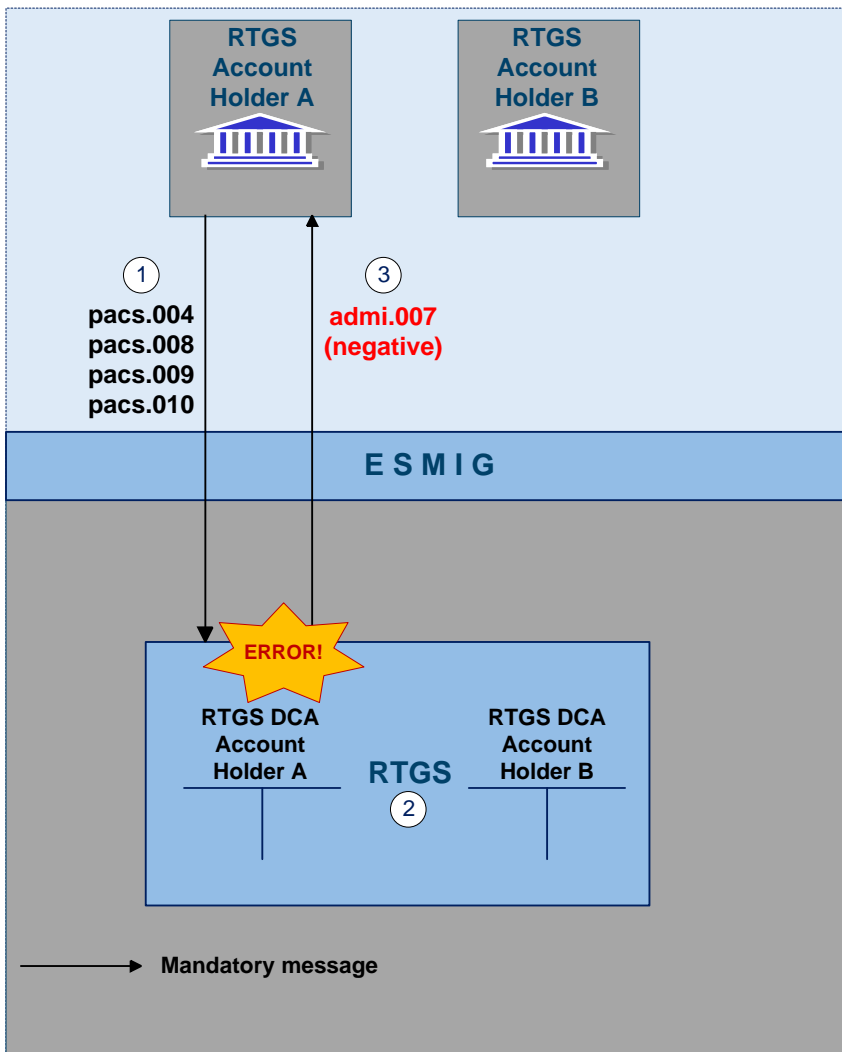


Figure 18 - pacs.008/009/010/004 technical validation error

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	RTGS Account Holder A sends a pacs.008/pacs.009/pacs.010/pacs.004 via ESMIG to the RTGS component.
2	RTGS component	Negative technical validation check in the RTGS component
3	RTGS component via ESMIG to RTGS Account Holder A	RTGS component sends an admi.007 (mandatory) in case of a negative technical validation via ESMIG back to the RTGS Account Holder A.

Table 35 - Technical validation failure**Used messages**

- | [CustomerCreditTransfer \(pacs.008\)](#) [▶ 577]
- | [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585]
- | [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 603]
- | [PaymentReturn \(pacs.004\)](#) [▶ 571]
- | [ReceiptAcknowledgement \(admi.007\)](#) [▶ 391]

5.2.4.2 Business validations

The following paragraph illustrates the payment flow in case of a validation failure in the RTGS component on the basis of an underlying pacs.008/pacs.009/ pacs.010/pacs.004.

Note: The RTGS component performs various checks during the business validation and does not stop after the first negative validation result, but continues with the business validation as subsequent checks could result in further negative validations

Message flow

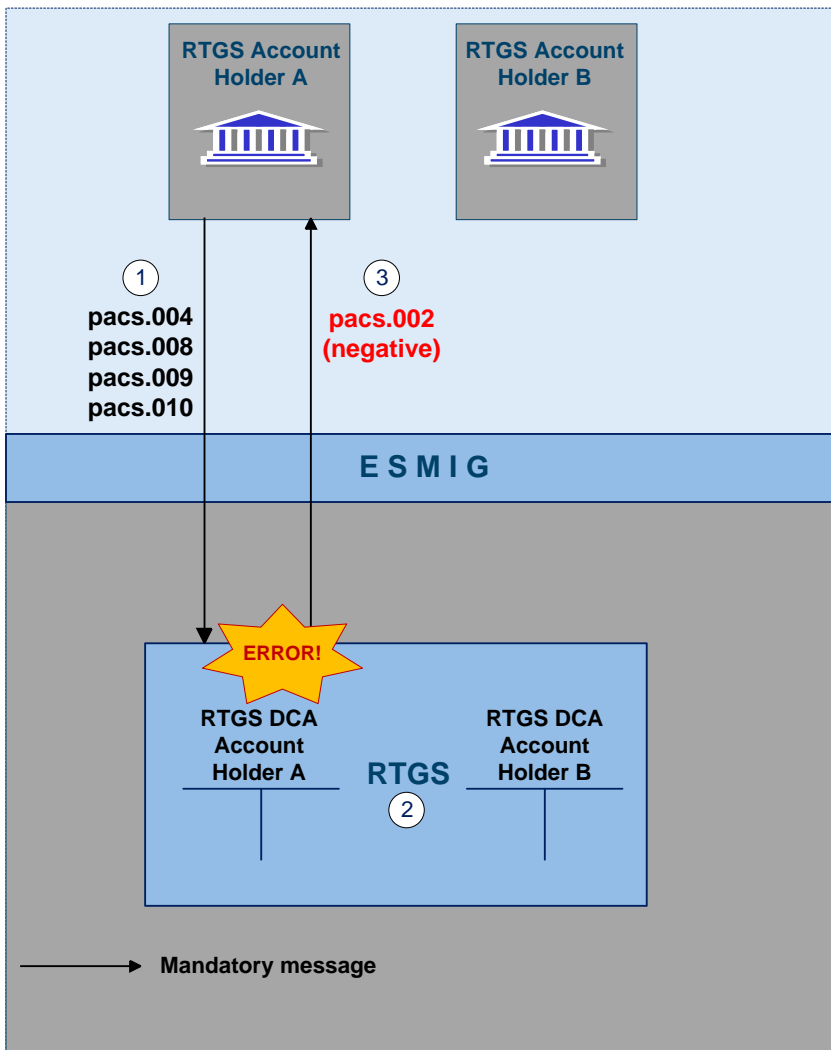


Figure 19 - pacs.008/009/010/004 business validation error

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	RTGS Account Holder A sends a pacs.008/pacs.009/pacs.009COV/pacs.010/pacs.004 via ESMIG to the RTGS component.
2	RTGS component	Negative business validation check in the RTGS component
3	RTGS component via ESMIG to RTGS Account Holder A	RTGS component sends a pacs.002 (mandatory) in case of negative business validation via ESMIG back to the RTGS Account Holder A.

Table 36 - Business validation failure

Used messages

- | [CustomerCreditTransfer \(pacs.008\)](#) [▶ 577]
- | [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585]
- | [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 603]
- | [PaymentReturn \(pacs.004\)](#) [▶ 571]
- | [PaymentStatusReport \(pacs.002\)](#) [▶ 568]

5.2.5 Amendment of payments

As long as a payment is not settled (including warehoused payments), an authorised system user has the possibility to change the relevant parameters of this payment.

Various control options are offered.

Action	authorised party
Change priority	RTGS Account Holder to be debited CB on behalf
Re-ordering (increase / decrease)	RTGS Account Holder to be debited CB on behalf
Change of set execution time (if defined before sending to the RTGS component)	RTGS Account Holder sending the payment or debtor of payment (only in case of ancillary system payments) CB on behalf

Table 37 - Options for changing the parameters of payments and authorisations

These features are necessary to enable RTGS Account Holders to react on changed liquidity conditions during the day. The consequences for the settlement of the affected payments can be found in chapter [Comprehensive queue management](#) [▶ 128].

Note: It is not possible for an authorised system user to use these control options for queued automated liquidity transfer from CLM due to pending CBO. Such liquidity transfers aiming at pulling liquidity from the RTGS DCA in the RTGS component to CLM, remain always on top of the urgent queue until they are settled or replaced by another automated liquidity transfer from CLM due to pending CBO.

The following rules apply in principle:

- | Interventions must be made via the business interface of the RTGS component in U2A and A2A. A description of individual U2A processes can be found in the RTGS user handbook.

- | Several payments together can be modified at the same time.
- | The business interface shows receipt and execution or non-execution of a modified order.

In case of intervention at payment level, processes are started to resolve the queues.

The following payment flow illustrates the amendment of a queued pacs.004/pacs.008/pacs.009/pacs.010.

Message flow

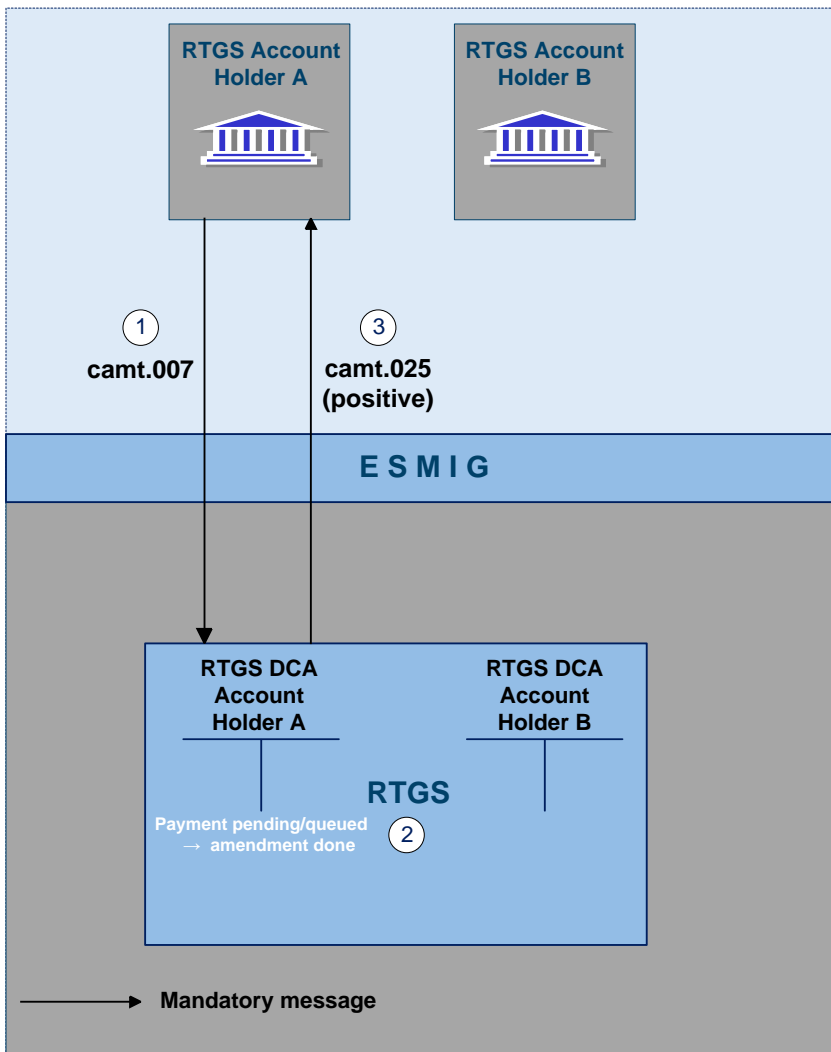


Figure 20 - camt.007 make amendment of payment (positive)

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	An RTGS Account Holder A sends a camt.007 via ESMIG to the RTGS component.
2	RTGS component	Message check and validation in the RTGS component with positive or negative result
3	RTGS component via ESMIG to the RTGS Account Holder A	Mandatory feedback to the RTGS Account Holder A via camt.025

Table 38 - Amendment of payments

Used messages

- | [ModifyTransaction \(camt.007\)](#) [▶ 437]
- | [Receipt \(camt.025\)](#) [▶ 474]

Case 1: changing priority

The following options for changing the priority exist:

- | from normal to high
- | from high to normal

It is not possible to change an urgent priority.

The payment priority can be changed at any time during the day time settlement phase. The updated priority can be checked by querying the payment queue.

The modified payment

- | keeps the original submission time;
- | is placed in the queue according to the (new) priority and the (old) submission time;
- | is processed according to the rules of the (new) priority.

Action	Effect
Change of the first queued high payment into a normal payment	<ul style="list-style-type: none"> If no urgent payment is queued immediate attempt to settle the remaining high payments following the FIFO-principle. If urgent payments are queued no immediate attempt to settle any high payments.
Change of a normal payment into a high payment	<ul style="list-style-type: none"> If the payment changed from normal to high moves to the top of the queued high payments and no urgent payments are queued, immediate attempt to settle high payments following the FIFO-principle (first in first out). Otherwise, no immediate attempt to settle high payments.

Table 39 - Effects of changing the priority

Case 2: re-ordering the queued payments

An authorised system user can change the queue position for an individual or for a sequence of payments. The selected payment or payments can be placed

- | to the top of the queued payments with the same priority
- | to the end of the queued payments with the same priority

The re-ordering can be done at any time during the business day. A detailed description of the process and the effect of the re-ordering can be found in chapter "Comprehensive queue management." The updated payment can be checked by querying the payment queue.

The following table shows the effect of changing the order in the queue.

Action	Effect
Moving an urgent payment to the top of the queued urgent payments	Immediate check whether the first payment in the queue can be executed
Moving an urgent payment from the top to the end of the queued urgent payments	
Moving a high payment to the top of the queued high payments and no urgent payment is queued	
Moving a high payment from the top to the end of the queued high payments and no urgent payment is queued	
Moving an urgent payment which is not at the top of the queued urgent payments to the end	It is taken into account during the next settlement process - no immediate check whether the first payment in the queue can be executed
Moving a high payment which is not at the top of the queued high payments to the end	
Moving a normal payment to the top or the end of the queued normal payments	

Table 40 - Effects of re-ordering the queued payments

Note: The re-ordering of queued payments is in principle available for all payment types including urgent payments.

However, it is not possible for an RTGS Account Holder to re-order queued liquidity transfers. The only liquidity transfers in the RTGS component which can be queued are automated inter-service liquidity transfers from CLM due to pending CBOs in case only partial settlement was possible. In case an automated inter-service liquidity transfer from CLM due to pending CBOs is queued, it always stays at the top of the urgent queue and no re-ordering is possible. Further details can be found in chapter [Liquidity transfer](#) [▶ 179].

Case 3: changing the execution time

Payments can include a time that indicates as of when they should be settled (payments with an “earliest debit time indicator”) and/or a time that indicates by when they should have been settled (payments with a “latest debit time indicator”).

The execution time may be changed in the RTGS component (A2A or U2A). The change has no impact on the payment processing, but on the queue management as the time indication supports the RTGS DCA holder's queue management. The updated execution time can be checked by querying the payment queue.

Changing the execution time has the following impact on the queue management.

Action	Effect
Deleting the execution time of an urgent payment ("FromTime")	Immediate settlement attempt, if the payment reaches the top of the queued urgent payments.
Deleting the execution time of a high payment ("FromTime")	Immediate settlement attempt, if the payment reaches the top of the queued high payments and no urgent payments are queued.
Deleting the execution time of a normal payment	Including the payment in the next settlement process.
Changing the execution time of a urgent, high or normal payment	Including the payment from the new indicated time.

Table 41 - Effects of changing the execution time

5.2.6 Revocation of payments

Revocation of a queued payment

An authorised system user who has sent a payment message has the ability to initiate the revocation of a payment using a PaymentCancellationRequest (camt.056).

A revocation of a payment is only possible as long as the payment is not settled on the RTGS DCA. It is also possible to revoke warehoused payments. A successful processing of the PaymentCancellationRequest results in the revocation of the payment (see case 1). In case the payment is already settled, the RTGS component forwards the PaymentCancellationRequest for further processing to the receiving RTGS Account Holder (see case 2).

The revocation can be done in the RTGS component in U2A or A2A. A description of individual U2A processes can be found in the related user handbook.

A cancellation request can be sent to revoke the following types of payments:

- | [CustomerCreditTransfer \(pacs.008\)](#) [▶ 577]
- | [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585]
- | [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 603]

For each payment submitted a dedicated `PaymentCancellationRequest` ([FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 536]) needs to be sent. In case of a direct debit, only the RTGS Account Holder to be credited can send the `PaymentCancellationRequest`.

The RTGS component informs about the execution or non-execution of a revocation via `camt.029`. The revocation can be initiated at any time during the day trade settlement phase until the COT for the respective payment type. The revoked payment can be viewed through the payment queue query.

Case 1: Successful revocation of a queued payment

Message flow

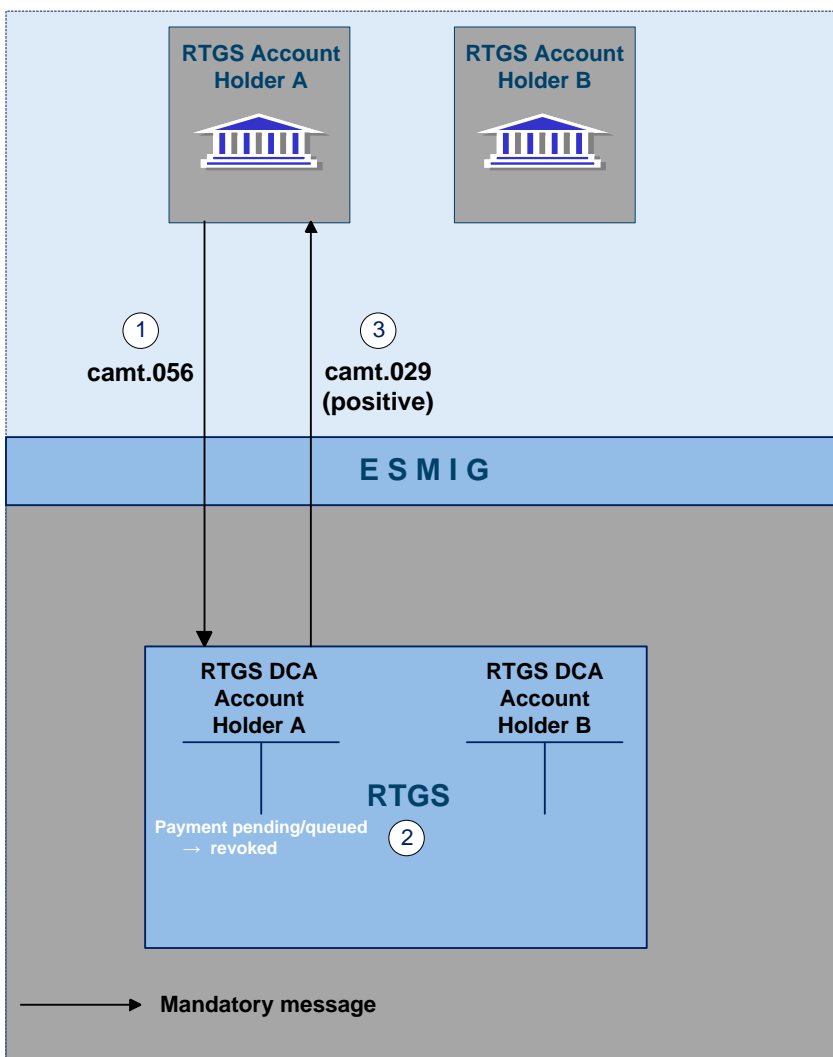


Figure 21 - camt.056 revocation of payment (positive)

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	RTGS Account Holder A sends a camt.056 to the RTGS component.
2	RTGS component	Message check and validation in the RTGS component positive. Underlying payment identified as being in a non- final status.
3	RTGS component via ESMIG to the RTGS Account Holder A	Mandatory feedback to RTGS Account Holder via camt.029

Table 42 - Successful revocation of a queued payment

Used messages

- | [FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 536]
- | [ResolutionOfInvestigation \(camt.029\)](#) [▶ 484]

Case 2: Cancellation request for already settled payments

In case the payment already settled on the RTGS DCA, it is no longer possible for the RTGS Account Holder A to revoke the payment. However, for pacs.008 and pacs.009 the RTGS Account Holder A can send the PaymentCancellationRequest to the RTGS and the RTGS component forwards the cancellation request to the relevant RTGS Account Holder B (i.e. the counterparty of the already settled payment). RTGS Account Holder B checks the cancellation request and sends

- | either a negative reply (i.e. camt.029) or
- | returns the funds by using the payment return message (pacs.004).

In case the RTGS Account Holder B sends

- | a negative reply, this negative reply is forwarded to the RTGS Account Holder A who sent the PaymentCancellationRequest;
- | a payment return message, this payment triggers the booking on the RTGS DCAs involved and which – after successful settlement - is sent to the RTGS Account Holder A.

In addition, RTGS Account Holder B can receive on an optional basis a payment status report.

In case the PaymentCancellationRequest is sent to revoke a pacs.010 which is already in a final status, the RTGS does not forward the PaymentCancellationRequest to the receiving RTGS Account Holder B. In such case, the RTGS Account Holder A is informed about the final status of the payment and the fact that the revocation is not possible. A cancellation request cannot be sent by the debited participant, but only by the initiator of the initial payment (BoP RTGS 10/81).

Message flow

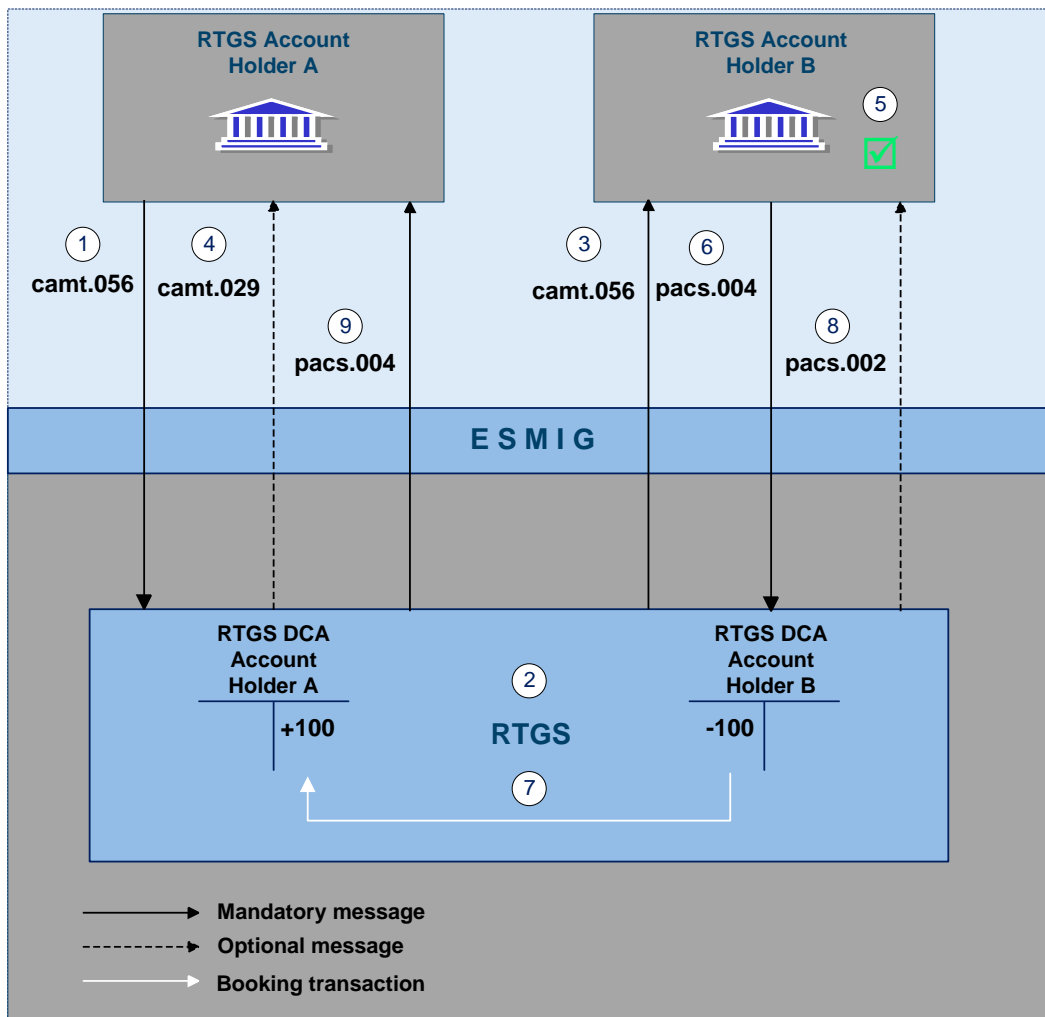


Figure 22 - camt.056 FIToFIPaymentCancellationRequest/camt.029 ResolutionOfInvestigation - positive case

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	RTGS Account Holder A sends a camt.056 via ESMIG to the RTGS component to request the revocation of an already sent payment.
2	RTGS component	Message check and validation in the RTGS component positive. Underlying payment (pacs.008 or pacs.009) identified as being settled on the RTGS DCA.
3	RTGS component via ESMIG to the RTGS Account Holder B	RTGS component sends a camt.056 via ESMIG to the RTGS Account Holder B.
4	RTGS component via ESMIG to the RTGS Account Holder A	RTGS component sends a negative camt.029 via ESMIG to the RTGS Account Holder A.

Step	Processing in/between	Description
5	RTGS Account Holder B	RTGS Account Holder B processes the requested revocation.
6	RTGS Account Holder B via ESMIG to the RTGS component	RTGS Account Holder B sends a pacs.004 via ESMIG to the RTGS component.
7	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCAs of RTGS Account Holders A and B
8	RTGS component via ESMIG to the RTGS Account Holder B	Creation and forwarding of a pacs.002 (optional) by the RTGS component via ESMIG to the RTGS Account Holder B.
9	RTGS component via ESMIG to the RTGS Account Holder A	Creation and forwarding of a pacs.004 by the RTGS component via ESMIG to the RTGS Account Holder A.

Table 43 - Cancellation request for already settled payments – positive case

Used messages

- | [FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 536]
- | [ResolutionOfInvestigation \(camt.029\)](#) [▶ 484]
- | [PaymentStatusReport \(pacs.002\)](#) [▶ 568]
- | [PaymentReturn \(pacs.004\)](#) [▶ 571]

Message flow

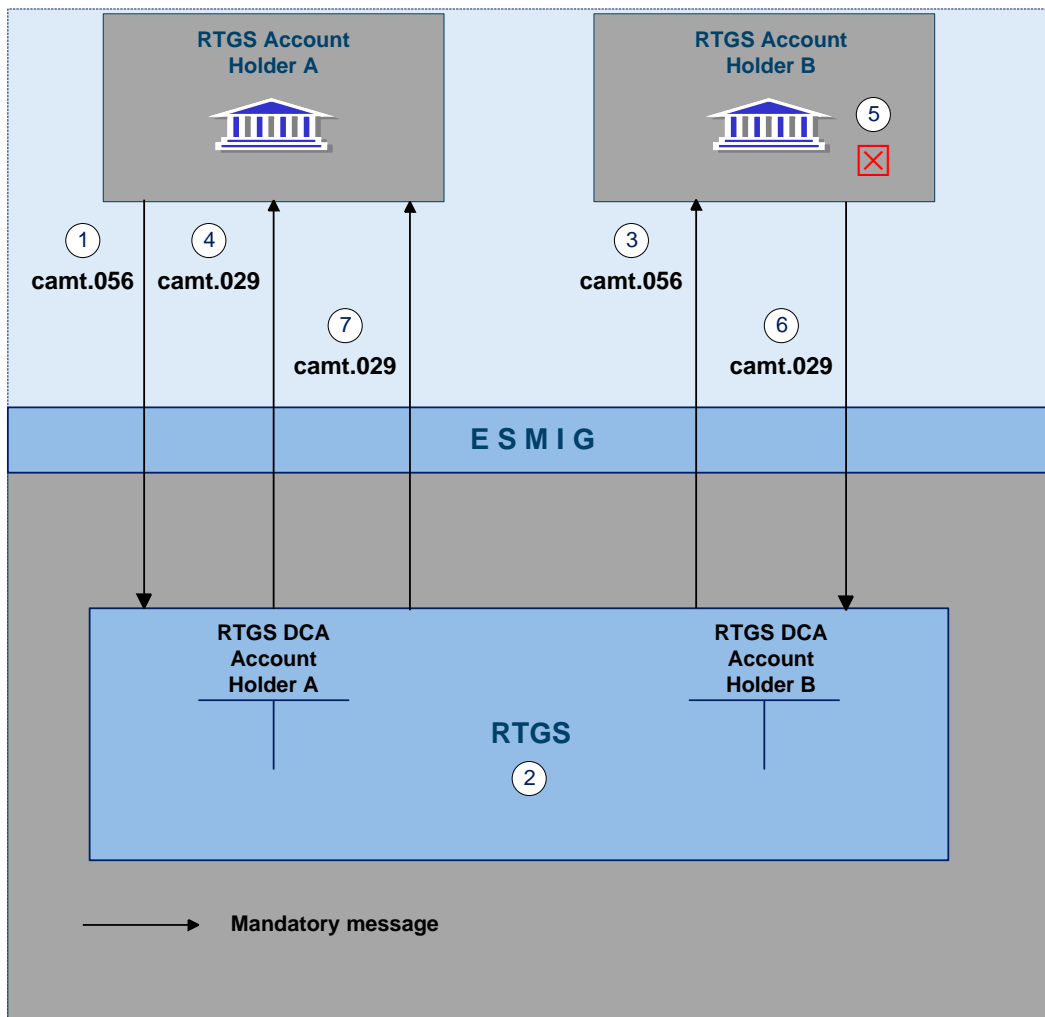


Figure 23 - camt.056 FIToFIPaymentCancellationRequest / camt.029 ResolutionOfInvestigation - negative case

Process description

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	RTGS Account Holder A sends a camt.056 via ESMIG to the RTGS component to request the revocation of an already sent payment.
2	RTGS component	Message check and validation in the RTGS component positive. Underlying payment (pacs.008 or pacs.009) identified as being settled on the RTGSDCAs.
3	RTGS component via ESMIG to the RTGS Account Holder B	RTGS component sends a camt.056 via ESMIG to the RTGS Account Holder B.
4	RTGS component via ESMIG to the RTGS Account Holder A	RTGS component sends a negative camt.029 via ESMIG to the RTGS Account Holder A.

Step	Processing in/between	Description
5	RTGS Account Holder B	RTGS Account Holder B cannot process the requested revocation.
6	RTGS Account Holder B via ESMIG to the RTGS component	RTGS Account Holder B sends a camt.029 (negative) via ESMIG to the RTGS component.
7	RTGS component via ESMIG to the RTGS Account Holder A	RTGS component forwards the camt.029 (negative) via ESMIG to the RTGS Account Holder A.

Table 44 - Cancellation request for already settled payments – negative case

Used messages

- | [FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 536]
- | [ResolutionOfInvestigation \(camt.029\)](#) [▶ 484]

5.2.7 Processing of payments

Effective settlement order

The order of settlement in general is depending on the settlement priority. In addition also the kind of business as trigger of the related cash transfer will be taken into consideration. The following table aims to illustrate the effective settlement order depending on settlement priority and underlying business:

Effective settlement order	Settlement priority	Business case
Debits on RTGS DCA		
1	Urgent	Automated liquidity transfer
2	Urgent	U payment/ancillary system transaction
3	Urgent	Immediate, rule-based or standing order liquidity transfer
4	Urgent	H payment
5	Urgent	N payment

Table 45 - Effective settlement order

5.2.7.1 Entry disposition

5.2.7.1.1 General remarks

Basics

The efficient management of liquidity and the settlement of payments in an optimised manner are of key importance. Therefore, offering a broad set of liquidity management features helps fulfilling the main functionality of the RTGS component.

These features may inter alia

- | result in faster settlement, with a reduced amount of liquidity;
- | help to avoid potential systemic risk owing, e.g. to gridlock situations;
- | increase transparency for RTGS Account Holders;
- | contribute to achieve a higher degree of efficiency.

The features are implemented in the RTGS component on a flexible and optional basis. This is to meet each RTGS Account Holder's individual needs, i.e. each RTGS Account Holder can individually decide whether to use certain tools or not.

Settlement of payments as main functionality

The aim of the processing in the RTGS component is a fast and liquidity-saving gross settlement of payments with the following characteristics:

- | cover for single payments or the balance of a group of payments
- | settlement in CB money
- | immediate, irrevocable booking of settled payments

Influencing factors

The payment processing in the RTGS component is inter alia influenced by the following factors:

- | balance on the RTGS DCA
- | defined limits
- | used priority
- | order of submitted payments
- | opposing payments and synchronisation of submitted payments
- | set execution time

Basic principles

The following basic principles apply to the processing of payments in the RTGS component:

- | Every payment should be marked as “normal”, “high “ or “urgent“. If no priority class is selected, payments will be handled as normal payments.
- | Attempt to settle single or group of payment immediately after their submission, with the exception of payments with a defined earliest debit time indicator (FromTime) and with the exception of warehoused payment prior to their execution date. In case a FromTime is defined, these payments are included in the settlement process from the time indicated as earliest debit time.
- | Offsetting payments are used to save liquidity (bilateral optimisation mechanism).
- | Payments to be settled are simultaneously booked on the RTGS DCA linked to the sender (debit: camt.050 and pacs.004/008/009/009COV; credit: pacs.010) and the counterparty RTGS DCA in the RTGS component (credit: camt.050 and pacs.004/008/009/009COV; debit: pacs.010).
- | Only payments which are not yet executed (i.e. queued) may be revoked.
- | Queuing of payment orders which cannot be settled immediately, according to their priority in different queues (urgent queue, high queue, normal queue).
- | In case of an automated inter-service liquidity transfer stemming from CLM due to pending CBOs which was only partially executed in the RTGS component, an inter-service liquidity transfer with the remaining amount is placed on top of the urgent queue in the RTGS component. This is the only scenario in which liquidity transfers are queued in the RTGS component.
Note: This automated inter-service liquidity transfer which aims at transferring liquidity from the RTGS component to CLM will be put on top of the urgent queue.
- | Continuous attempt to settle payments in the queues.
- | The entry disposition and the optimisation procedures for queues can run at the same time.

5.2.7.1.2 Settlement of payments in the entry disposition

For urgent payments the FIFO-principle applies.

High and normal payments are not settled in the case urgent payments are queued. The only exception is that payments with a lower priority are executed before, if - and only if -, this allows an offsetting payment to be settled and the overall effect of this offsetting is a liquidity increase for that RTGS Account Holder.

For high payments the FIFO-principle applies, too.

Normal payments are not settled if high payments are queued. The only exception is that payments with a lower priority can be executed before, if - and only if -, this allows an offsetting payment order to be settled and the overall effect of this offsetting is a liquidity increase for that RTGS Account Holder.

Normal payment orders are processed according to the “FIFO by-passing” principle.

In order to save as much liquidity as possible, the FIFO-principle would not be the optimal one; i.e. normal payments submitted may be executed even if other previously sent normal payments are still in the queue (provided that the balance on the RTGS DCA is sufficient).

The entry disposition takes offsetting payments into account. The balance available on the account of the RTGS Account Holder is taken into account. In addition, in the case of normal payments, limits defined are considered.

The following table shows which payment orders are taken into account during the entry disposition for the RTGS DCA of the debtor and/or the creditor.

Debtor	Creditor
Submitted payment	All offsetting urgent, high and normal payment in the queues

Table 46 - Payments taken into account in the entry disposition

Unsuccessful entry disposition

If a submitted payment cannot be settled in the entry disposition, it is placed into the urgent, high or normal queue - depending on the priority of the payment.

Note: In general, liquidity transfers are not placed into a queue and are rejected with appropriate error code in case the liquidity is not sufficient or none of the above mentioned criteria for FIFO by-passing can be met. Exceptions are related to automated inter-service liquidity transfers stemming from CLM due to pending CBOs which were not or only partially executed in the RTGS component. In such case the RTGS component creates an inter-service liquidity transfer with the remaining amount and this liquidity transfer is placed on top of the urgent queue.

Detailed sequence of settlement checks

In a first step the RTGS component checks whether there are already payments of an equal or higher priority level in the queue (exception: if the submitted payment is a normal one, it is not checked whether the “normal” queue is empty, because the FIFO principle can be breached for normal payments).

If the urgent and high queue are **not** empty, a bilateral offsetting check with potential liquidity increase takes place. This offsetting check is only successful if offsetting payments from the RTGS DCA to be credited are available and the RTGS DCA to be debited with the payment afterwards has an increased liquidity position. If offsetting payments exist, it is checked if the submitted payment fulfils the other settlement criteria (i.e. bilateral/multilateral [Limits](#) [▶ 200] and liquidity reservations not breached). If no such offsetting payments exist, the payment is put in the queue.

If the urgent and the high queue are empty, an offsetting check called “offsetting position 1 check” takes place. This offsetting check is only successful if offsetting payments on top of the queue of the RTGS DCA to be credited are available. If the offsetting check is successful, it is checked if the submitted payment fulfils the other settlement criteria (i.e. bilateral/multilateral limit and liquidity reservations not breached).

If the offsetting check is not successful, an extended offsetting check takes place. This extended offsetting check is only successful if offsetting payments related to the RTGS DCA to be credited (not only on top of its queue) are available and the RTGS DCA to be credited afterwards has an increased liquidity position. If the extended offsetting check is successful, it is checked if the submitted payment fulfils the other settlement criteria (i.e. bilateral/ multilateral limit and liquidity reservations not breached). If the extended offsetting check is not successful, the payment is put in the queue.

If the other settlement criteria (i.e. bilateral/multilateral limit and liquidity reservations not breached) are fulfilled, then the operation(s) is (are) settled on the RTGS DCA (i.e. debit as well as credit booking on the respective RTGS DCAs take place). If the other settlement criteria are not fulfilled, then the payment(s) is (are) put in the queue until sufficient liquidity is available and the other settlement criteria are fulfilled (details on the dissolution of the queues are given in chapter [Dissolution of the payment queue](#) [▶ 132]).

If there is not sufficient liquidity available and/or the other settlement criteria are not fulfilled until the end of the day, the payments not yet settled are rejected.

Note: In case of direct debits, the RTGS Account Holder sending the payment expects a liquidity increase on its RTGS DCA and the RTGS DCA of the receiver is debited.

Rejection during EoD processing

If queued payments cannot be settled during optimisation procedures and are still queued by the end of the day due to lack of liquidity or insufficient limits, these payments are rejected during the EoD processing.

5.2.7.2 Comprehensive queue management

If a submitted payment cannot be settled in the entry disposition, it is placed into the urgent, high or normal queue, depending on its priority. Moreover, in case of partially settled automated liquidity transfers stemming from CLM due to pending CBOs, the remaining part of such automated liquidity transfer are also queued on top of the urgent payment queue.

As long as a payment is not settled, the RTGS Account Holder has the ability to change the relevant parameters of the payment. Further details on amending payments can be found in chapter [Amendment of payments](#) [▶ 113].

Note: Depending on the configuration chosen by the RTGS Account Holder, in case of pending urgent or high payments an inter-service liquidity transfers might be triggered in order to transfer liquidity from the

linked MCA to the RTGS DCA. Further details on such inter-service liquidity transfers can be found in chapter [Liquidity transfer](#) [▶ 179].

In case of queued payments, four different control options for the comprehensive queue management are offered:

Action	RTGS Account Holder
Change priority Exception 1: It is not possible to change the priority of urgent payments Exception 2: In case of pacs.010 the receiver (i.e. the debtor) has the ability to change the priority	RTGS Account Holder to be debited
Re-ordering (increase / decrease) Exception: in case of pacs.010 the receiver (i.e. the debtor) has the ability to re-order)	RTGS Account Holder to be debited
Change of set execution time (if defined before sending to the RTGS component)	RTGS Account Holder sending the payment
Revocation (Revocation of payments [118])	RTGS Account Holder sending the payment

Table 47 - Control options for comprehensive queue management

These control options enable an RTGS Account Holder to react on changed liquidity conditions during the day. It is possible to modify a single payment or several payments at the same time. In case it is not possible to execute a modification the RTGS Account Holder is notified accordingly. Amendments are possible in U2A via the GUI.

In case of successful interventions, processes are started to resolve the queue(s). Further details on the interventions done in U2A can be found in the RTGS user handbook.

Changing the priority of a payment

Priority of a payment		
Urgent	High	Normal
	⇨	⇩

Table 48 - Possibilities for changing priorities

It is not possible to change the priority of a queued urgent payment. The priority of queued payments can be changed at any time during the day trade settlement phase and the RTGS Account Holders involved can see the changed payment priority.

In case of such change, the payment

- | keeps its original submission time;
- | is placed in the queue according to the (new) priority and the initial submission time;
- | is processed according to the rules of the (new) priority.

Action	Effect
Change of the first queued high payment into a normal payment	<ul style="list-style-type: none"> If no urgent payment is queued immediate attempt to settle the remaining high payments following the FIFO principle If urgent payments are queued no immediate attempt to settle any high payment
Change of a normal payment into a high payment	<ul style="list-style-type: none"> If the payment changed from normal to high moves to the top of the queued high payments and no urgent payments are queued, immediate attempt to settle high payments following the FIFO principle Otherwise, no immediate attempt to settle urgent payments

Table 49 - Effect of changed priority

Re-ordering of queued payments

The RTGS Account Holder sending the payments (exception: pacs.010) can change the queue position for a single or a sequence of payments via U2A and A2A. The payment(s) selected can be placed on:

- | to the top of the queue payment with the same priority;
- | to the end of the queued payments with the same priority.

Action	Effect
Moving an urgent payment to the top of the queued urgent payments	Immediate check whether payments can be executed
Moving an urgent payment from the top to the end of the queued urgent payment	
Moving a high payment to the top of the queued high payments and no urgent payment is queued	
Moving a high payment from the top to the end of the queued high payments and no urgent payments are queued	

Action	Effect
Moving an urgent payment which is not at the top of the queued urgent payments to the end	The action is taken into account during the next settlement process – no immediate attempt to settle
Moving a high payment which is not at the top of the queued high payments to the end	
Moving a normal payment to the top or the end of the queued normal payments	

Table 50 - Effect of changing the order of queued payments

The re-ordering of queued payments is possible for all priorities, including urgent payments. However, it is not possible to re-order queued automated liquidity transfers which were triggered in CLM due to pending CBOs which aim at transferring liquidity from the RTGS DCA to the MCA. Such a liquidity transfer remains on top of the urgent queue and in this case it is not possible to put any other queued urgent payment on top of the urgent queue.

Changing the defined execution time

In principle, RTGS Account Holders can submit payments with a defined execution time. It is possible to include an earliest debit time indicator and/or a latest debit time indicator (see chapter [Definition of execution time](#) [▶ 81]).

In case a submitted payment includes an earliest debit time indicator and/or a latest debit time indicator it is possible to change the earliest debit time indicator and/or the latest debit time indicator via A2A or U2A. Such a change has no impact on the payment processing, but on the queue management as the time indication only support the queue management of the RTGS Account Holder.

Action	Effect
Deleting the earliest debit time indicator of an urgent payment (FromTime)	Immediate settlement attempt, if the payment reaches the top of the queued urgent payment
Deleting the earliest debit time indicator of a high payment (FromTime)	Immediate settlement attempt, if the payment reaches the top of the queued high payments and no urgent payments are queued
Deleting the earliest debit time indicator of a normal payment	Including the payment in the next settlement process – no immediate attempt to settle
Changing the earliest debit time indicator of a urgent, high or normal payment	Including the payment from the new indicated time onwards

Table 51 - Effect of changing the execution time

Revocation of a queued payment

In case a payment is not yet settled, the RTGS Account Holder can revoke the payment via A2A or U2A.

Details on the revocation via A2A using a PaymentCancellationRequest (camt.056) can be found in chapter [Revocation of payments](#) [▶ 118].

5.2.7.3 Dissolution of the payment queue

5.2.7.3.1 Settlement of queued urgent/high payments

The queues for payments with urgent or high priority are resolved in an event-oriented way starting with the payment at the top.

Events	by ...
Liquidity increase	<ul style="list-style-type: none"> incoming settled payment (i.e. credits) incoming settled intra-service liquidity transfers incoming inter-service liquidity transfer from other services/components (i.e. credits)
Intervention on queue level	<ul style="list-style-type: none"> If the payment on the top of the urgent/high queue is changed (change of order, change of priority, revocation)

Table 52 - Possible events for queue resolution

Resolving the urgent/high queue and the entry disposition are handled in the same way. If a single urgent or high payment cannot be settled, it remains in the queue (at maximum until the end of the business day).

Continuously resolving of the queue

The urgent/high queue is continuously resolved by the sequentially run of algorithms for the resolving of queued normal payments.

Optimisation for the processing on sub-accounts

For optimisation of the processing of urgent ancillary system payment instructions on the sub-accounts of settlement banks a special algorithm is used. It can be seen as an exception of the below described algorithms for the settlement of queued normal payments. Further details on the settlement of ancillary system payment instructions can be found in chapter [Settlement of ancillary systems](#) [▶ 140].

5.2.7.3.2 Settlement of queued normal payments

Principles

The normal queue is continuously resolved by including queued urgent and high payments as well as the queued part of automated inter-service liquidity transfers from CLM due to pending CBOs. There are three different algorithms available:

- | partial optimisation
- | multiple optimisation
- | partial optimisation with ancillary system

The single algorithms are used either sequentially or according to the situation in order to respond in a flexible way to changed liquidity conditions during the day trade settlement phase.

The algorithms can run in parallel to the “entry disposition” of the RTGS component, which means that payments entering the system after the start of any algorithm can be settled immediately if the positions and limits of the accounts concerned are compatible with both the settlement of these payments and the settlement of payments taken into account in the current optimisation.

However, two algorithms cannot run in parallel to each other.

Sequence of algorithms

During the business day the algorithms run sequentially,

- | while there is no pending simultaneous multilateral settlement of an ancillary system (see chapter [Ancillary system settlement procedure B](#) [▶ 148]):
 - first algorithm “partial optimisation” then algorithm “multiple optimisation”...
 - if algorithm “partial optimisation” succeeds then two algorithm schedule options are in place, i.e. either algorithm “multiple optimisation” runs always after algorithm “partial optimisation” or algorithm “partial optimisation” runs again.
 - changes of the algorithm schedule lie within the sole responsibility of the operator in order to be able to react in a flexible way to changed liquidity conditions.
- | while there is a pending simultaneous multilateral settlement of an ancillary system:
 - algorithm “partial optimisation with ancillary system”

The algorithms run in a flexible way by defining a time lag (i.e. a parameter) between the executions of different algorithms to have a minimum interval between two runs of algorithms. The temporal sequence is automatically controlled by the RTGS component. Manual intervention is possible by the operator.

Consequences of a running algorithm

During a running algorithm a payment is “locked“. That means it cannot be re-ordered, revoked, etc. If the payment is settled during the run of the algorithm the request of an RTGS Account Holder to e.g. re-order the payment cannot be taken into account anymore. If the payment is still pending after the end of the algorithm, the request of the RTGS Account Holder is taken immediately into account.

Algorithm: “Partial optimisation”

This algorithm calculates in a first step the total positions of each and every RTGS DCA. In a second step, it removes individual payments in order to avoid insufficient cover. This earmarking of payments for removal (i.e. maintaining payments in the payment queue) is limited to RTGS DCAs for which an uncovered position was calculated as result out of the calculation of the total liquidity position.

Step	Description
1	For each RTGS DCA, the total position is calculated. It consists of the sum of actual balance, + incoming pending payments (i.e. credits), ./. outgoing pending payments (i.e. debits). All total positions are checked for cover.
2	If all total positions are covered, all payments are settled.
3	If merely one total position of an RTGS DCA is not covered, single payments are retained until the liquidity of the DCA is sufficient for covering its total position. Retained payments are included in the next settlement process. The executable payments are settled.

Table 53 - Main characteristics of algorithm “partial optimisation”

For the retaining of transactions the following rules apply.

- | The selection process runs for a short period of time only.
- | Payments at the end of the queue with lowest priority are first checked concerning retaining.
- | The selection is started with the RTGS DCA with the highest uncovered total-debit position.

If run of this algorithm does not succeed, the algorithm “multiple optimisation” is activated.

Algorithm: “Multiple optimisation”

The aim of this algorithm is resolving of the queues with the highest possible settlement volume and low liquidity demand.

This optimisation process consists of two parts following one after another. It starts with resolving of bilateral relationships and ends with resolving of the multilateral relations.

Part 1

Payments which should be processed bilaterally (i.e. between two RTGS Account Holders of which at least one has defined a bilateral limit towards the other) are cleared as follows.

Step	Description
1	Determine the objective sequence of how the bilateral queue should be worked through: first, the pairs of transactions with the best offsetting and then then the other pairs of payments.
2	Check the bilateral positions regarding coverage. If the settlement of a payment is not possible due to a lack of liquidity or breached limits, single payments retains in the queue.
3	The identified covered transactions are immediately settled before the algorithm continues with the next pairs of payments.

Table 54 - Main characteristics of algorithm “multiple optimisation” – Part 1

If the settlement of a pair of queues is not possible due to lack of liquidity or breached limits, single payments retains in the queues (under consideration of the FIFO-principle).

Part 2

The check of bilateral relations is followed by the check of multilateral relations (between one RTGS DCA and others towards which a multilateral limit is defined): how the remaining payments influence the balance of each RTGS DCA. Uncovered payments or payments which breach defined limits are retained (in the same manner as in algorithm “partial optimisation”).

Payments which should be processed multilaterally are handled as follows (step 1 - 3 are repeated until each uncovered multilateral position is checked):

Step	Description
1	Check the multilateral position regarding coverage.
2	If the settlement of a payment is not possible due to a lack of liquidity or breached limits, single payments retains in the queue.
3	The identified executable payments are settled.

Table 55 - Main characteristics of algorithm “multiple optimisation” – Part 2

Algorithm: Partial optimisation with ancillary system

Algorithm “partial optimisation with ancillary system” is developed to support the simultaneous multilateral settlement of ancillary system (see chapter [Ancillary system settlement procedure B](#) [▶ 148]). It ensures an efficient and fast processing of the related ancillary system payment instructions. In order to smoothen the

settlement and to reduce the overall liquidity needed, other “urgent“ payments as well as “high“ and “normal“ ones are also included.

Ancillary system payment instructions which shall be settled using simultaneous multilateral settlement, bypass the entry disposition and are kept in the RTGS component separately until the end of the current optimisation process. This separation is necessary as otherwise they would block the settlement of other payments because of their priority.

Note: As long as no ancillary system simultaneous multilateral settlement is queued and payments are pending, the other algorithms run successively. See below for more details on the sequence of algorithms.

Step	Description
1	For each RTGS DCA, the total position is calculated. All total positions are checked for cover.
2	If all total positions are covered, all payments and ancillary system payment instructions are settled.
3	If just one total position of an RTGS DCA is not covered, single payment orders are retained until the liquidity of the DCA is sufficient for covering its total position. During the selection procedure the ancillary system position remains unchanged (i.e. ancillary system payment instructions (debits) are never retained). Retained payments are included in the next settlement process.

Table 56 - Main characteristics of algorithm “partial optimisation with ancillary system”

Inclusion of all pending payments:

Algorithm partial optimisation with ancillary system takes all pending payments and ancillary system payment instructions into account. The inclusion is independent

- | on whether the RTGS Account Holders owning the debited and credited RTGS DCAs are settlement banks of an ancillary system using the simultaneous multilateral settlement or not;
- | of the priority of a payment (urgent, high, normal).

This broad approach is chosen in order to keep the whole settlement process running in the RTGS component. It also helps to smoothen the settlement process by taking into account offsetting payments.

Ordering of ancillary system payment instructions in the queue

Payments to be settled by the use of algorithm “partial optimisation with ancillary system” are ordered

- | by their priority (urgent, high, normal);
- | within the priority following
 - the time they have entered the RTGS component (FIFO principle);
 - their earliest debit time - if defined (exception 1);

- the time of the start of the settlement period (exception 2 - only for ancillary system payment instructions (see chapter [Settlement of ancillary systems](#) [▶ 140])).

Several ancillary system involved in one running algorithm “partial optimisation with ancillary system”

In the same run of algorithm “partial optimisation with ancillary system” several ancillary system using ancillary system settlement procedure B (see chapter [Ancillary system settlement procedure B](#) [▶ 148]) is included if they intend to settle at the same time.

Settlement process in detail

The algorithm “partial optimisation with ancillary system” calculates the position of each RTGS DCA including all pending payments and ancillary system payment instructions. For debit positions, it is checked whether sufficient liquidity is available.

If at least one RTGS DCA does not have sufficient liquidity, algorithm “partial optimisation with ancillary system” selects the RTGS DCA with the largest uncovered debit position; then it retains payments of this RTGS DCA for optimisation until its position is covered (same retaining rules as algorithm “multiple optimisation”).

If the selected payment is an ancillary system payment instruction using ancillary system procedure simultaneous multilateral settlement also all other payments of the respective ancillary system file is retained from the optimisation process.

As long as there are still ancillary system payment instructions stemming from other ancillary systems using the procedure simultaneous multilateral settlement pending in the RTGS component, algorithm “partial optimisation with ancillary system” continues running (= a further loop within the same run starts). In this further loop, also those payments are included that were retained before, with exception of retained ancillary system payment instructions using the procedure simultaneous multilateral settlement.

Algorithm “partial optimisation with ancillary system” ends

- l a) if there are no ancillary system payment instructions for simultaneous multilateral settlement included in the settlement process anymore; or
- l b) the time defined as maximum for a run of algorithm “partial optimisation with ancillary system” has elapsed; or
- l c) all debit positions are covered.

In case a) and b) all payments included in the optimisation return to their previous status. In case c) all payments that are not retained are settled.

Note: Owing to the fact that also normal payments are included in the optimisation process it is also checked during the run of algorithm “partial optimisation with ancillary system” that no limits are breached. Otherwise, the payment breaching a limit has to be retained independent of the availability of liquidity.

Sequence of the various algorithms

At the entry time of an ancillary system settlement following simultaneous multilateral settlement, algorithm “partial optimisation with ancillary system” starts. In case an algorithm is running at the beginning of the settlement period algorithm “partial optimisation with ancillary system” waits until the running algorithm ends and then starts immediately.

If algorithm “partial optimisation with ancillary system” is successful the simultaneous multilateral settlement is finished. The sequence of the other algorithms continues.

If algorithm “partial optimisation with ancillary system” is not successful or only partially successful in the first run, the next run of algorithm “partial optimisation with ancillary system” starts after a predefined period of time. In the meantime the other algorithms can run and settle payments. The reason for this is not to stop the whole payment processing for a longer period of time.

The time period is a parameter defined in the RTGS component to have a minimum interval between two runs. It is the same for the other algorithms. There is also a minimum interval defined between the runs of these algorithms.

If algorithm “partial optimisation with ancillary system” is running and during this time the entry time of another ancillary system using ancillary system settlement procedure B is reached, the ancillary system payment instructions remains waiting until the current algorithm “partial optimisation with ancillary system” ends and the next one starts after the minimum interval.

5.2.7.3.3 Algorithm: “Optimisation on sub-accounts”

In order to settle ancillary system payment instructions on sub-accounts in the RTGS component, a dedicated algorithm is available.

This algorithm aims at resolving ancillary system payment instructions using dedicated liquidity on sub-accounts. The algorithm only checks sub-accounts instead of RTGS DCAs and only covered ancillary system payment instructions are settled. In case of uncovered ancillary system payment instructions, these ancillary system payment instructions are put back in the queue of the single sub-account. The algorithm runs only once during a business day. .

Note: Owing to the fact that algorithm “optimisation on sub-accounts” only takes into account ancillary system payment instructions to be settled on sub-accounts there is no need to consider any limits or reservations.

Step	Description
1	For each RTGS DCA, the total position is calculated. It consists of the sum of actual balance on one sub-account + incoming ancillary system payment instructions (i.e. credits) ./. outgoing ancillary system payment instructions (i.e. debits) for this sub-account.
2	If all total positions are covered, all ancillary system payment instructions are settled).
3	Ancillary system payment instructions which are not covered are put back in the queue.
4	At the end of the cycle, all ancillary system payment instructions debiting the same sub-account with insufficient liquidity for their settlement are rejected even if only one ancillary system payment instruction cannot be settled.

Table 57 - Main characteristics of algorithm “optimisation on sub-accounts”

5.2.7.4 Treatment of backup payments in the settlement process

Backup contingency and backup liquidity redistribution payments are transferred to the RTGS component in the order in which they were generated.

These payments go through the same clearing and settlement process (entry management, queue dissolution) in the RTGS component as any other regularly submitted urgent payments (in case of backup contingency payments in favour of CLS) or high payments (in case of backup contingency and backup liquidity redistribution payments).

They are visible in the display of pending payments in the U2A. Further details can be found in the RTGS user handbook.

In general, it is also possible to query pending payments via A2A.

If backup payments are in the queue for urgent (in case of CLS backup contingency payments) or high (in case of other backup contingency and backup liquidity redistribution payments) payments, they are treated in the RTGS component as any other payment. As a consequence, revocation (see chapter [Revocation of payments](#) [▶ 118]) as well as queue management (see chapter [Comprehensive queue management](#) [▶ 128]) is possible.

5.3 Settlement of ancillary systems

5.3.1 Overview

To settle ancillary system transfers in CB money the needed functionalities are offered in the RTGS component. These allow the ancillary systems to have i) a broader accessibility of account holders and ii) a broad range of streamlined functionalities.

Advantages for ancillary system settlement banks (i.e. RTGS Account Holder participating in the settlement of ancillary systems) and ancillary systems are:

- | choice to use only one RTGS DCA for payments and the settlement of ancillary system transfers or to open one or more dedicated RTGS DCAs for one or several ancillary system(s)
- | cross-border usage – one RTGS DCA held with one CB can be used for settling ancillary system transfers stemming from ancillary systems from other countries
- | integration with normal payment business
- | urgent priority for ancillary system transfers

Types of ancillary systems are:

- | retail payment systems
- | large value payment systems
- | foreign exchange systems
- | money market systems
- | clearing houses (CCP) and
- | securities settlement systems (SSS)

Settlement procedures

The settlement of ancillary system transfers takes place in different settlement procedures. The table below is an overview of the settlement procedures. Details of the procedures can be found in the following chapters.

Procedure	Description
Ancillary system settlement procedure A	<p>Procedure based on "Debits first" booking:</p> <p>Ancillary system sends simultaneously debits and credits. All debits have to be booked before credits are settled.</p>
Ancillary system settlement procedure B	<p>Procedure based on "All or nothing" booking:</p> <p>Ancillary system sends simultaneously debits and credits to the RTGS component. All debits and credits are simultaneously checked for settlement.</p> <p>If this check is passed all debits and credits are booked simultaneously.</p>
Ancillary system settlement procedure C	<p>Procedure based on settlement on sub-account:</p> <p>Ancillary system settlement bank can dedicate a liquidity amount to settle balances coming from a specific ancillary system. The dedication is achieved by setting aside the needed liquidity on a specific sub-account. Such a settlement procedure can be used for mandatory and optional procedure.</p>
Ancillary system settlement procedure D	<p>Procedure based on prefunding of technical account:</p> <p>Ancillary system settlement bank can dedicate a liquidity amount to settle balances coming from a specific ancillary system. The dedication is achieved by setting aside the needed liquidity on the ancillary system technical account..</p>

Table 58 - Settlement procedures

For all settlement procedures the settlement date of the ancillary system transfers (irrespective of the message used for instructing them) has to be the current business date. There is no possibility to use warehoused payments.

Account types for ancillary systems

The following diagram depicts a generic account constellation for an ancillary system settlement bank, e.g. a settlement bank with various types of settlement business and with accounts opened in the book of one CB.

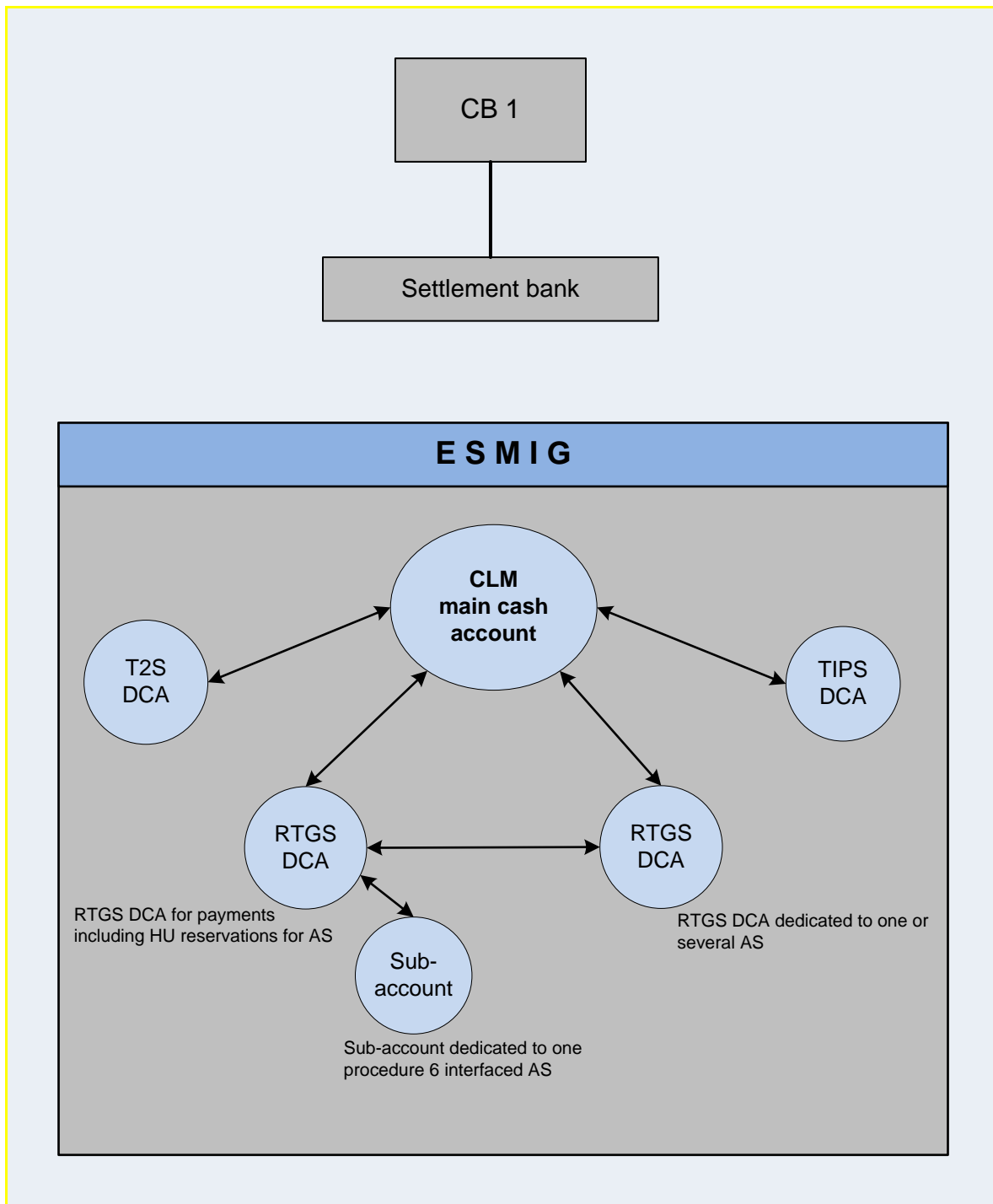


Figure 24 - Generic account constellation for an ancillary system settlement bank

Besides the DCAs for securities (i.e. T2S DCA) and instant payments settlement in CB money (i.e. TIPS DCA) , the ancillary system settlement bank in the example above has an RTGS DCA for high value payments (with a reserved amount for urgent ancillary system transfers) and two further accounts for ancillary system transfers: one account for ancillary system procedure C as a sub-account of the RTGS DCA for high value payments and the second account (for other ancillary systems) as an RTGS DCA dedicated to one or several ancillary systems.

Account type	Account holder	Description	Procedure
RTGS DCA	Ancillary system settlement bank	Used as an RTGS DCA for the settlement of high value payments and ancillary system transfers	Can be used in all procedures except for ancillary system settlement procedure C on debit side
RTGS DCA dedicated to ancillary system	Ancillary system settlement bank	Used as an RTGS DCA specifically for the settlement of one or several ancillary system(s)	Can be used in all procedures except for ancillary system settlement procedure C on debit side
Sub-account	Ancillary system settlement bank	Used to set aside liquidity for exclusive settlement of a specific ancillary system and needs to be mapped to the RTGS DCA	Ancillary system settlement procedure C only
Guarantee funds account	Guarantor, CB or the ancillary system	Used in case the optional guarantee mechanism has to be activated by an ancillary system or the CB on its behalf. The same guarantee account can be used for both procedures (settlement procedure A and B); it is also possible to use two different ones	Ancillary system settlement procedures A and B
Ancillary system technical account	Ancillary system	Used as intermediary account for the collection of debits and credits resulting from the settlement of ancillary system transfers and for prefunding in the context of ancillary system settlement procedure D	One dedicated ancillary system technical account is to be opened for each ancillary system settlement procedure used.

Table 59 - Account types and their ownership

Liquidity used for settlement of ancillary system transfers

The necessary liquidity used for settlement originates from different accounts. Sources of liquidity and liquidity transfer order types are described in chapter [Dedication of liquidity for ancillary system settlement](#) [207].

Monitoring of ancillary system settlement

Ancillary systems and ancillary system settlement banks can rely on a comprehensive information flow for a full visibility on the status of payments/net balances issued at any time during the entire process.

In addition to the information on individual payments/net balances the RTGS component provides ancillary systems, CBs and ancillary system settlement banks with aggregated data. These aggregated data are:

- | number and amount of ancillary system transfers;
- | ancillary system transfers queued because of lack of liquidity;
- | uncovered ancillary system transfers shortly before a settlement period ends;
- | rejected, revoked or reversed ancillary system transfers; and
- | settled ancillary system transfers

5.3.2 Ancillary system settlement procedure A

Basics

Ancillary systems can settle a set of multilateral balances (debits and credits) on RTGS DCAs in a batch mode.

The RTGS component will be responsible for settling first all debits and, once all of them have been settled, to execute also all credits at once. The identification of debits or credits is made from the ancillary system technical account perspective. Whenever an ancillary system settlement bank's RTGS DCA is debited and the ancillary system technical account is credited, the transaction is considered as a debit whilst debiting the ancillary system technical account and crediting the ancillary system settlement bank's RTGS DCA will be considered a credit. In turn, for ancillary system settlement procedure A the usage of the ancillary system technical account is mandatory.

Due to the peculiarities of the settlement, i.e. in order to ensure that after the settlement of debits the needed amount is present on the technical account and not used for other purposes in the framework ancillary system transfer processing, a dedicated ancillary system technical account for ancillary system settlement procedure A is to be used and cannot be reused for any other settlement procedure. Additionally, the sum of all debits must be equal to the sum of credits within one Ancillary system batch message.

Taking into account above mentioned links between the Ancillary system transfers, a failure in settlement of one or more debit leg will result in a reversal of already executed debits and non-settlement of any credit. In order to limit the negative impact of failed settlement, the ancillary system can make use of the guarantee fund mechanism.

Optional connected mechanisms

The ancillary system settlement procedure A may include optional connected mechanisms:

- information period
- settlement period ("till")
- guarantee fund mechanism

For further details on the usage and functionalities offered by the optional connected mechanisms please refer to chapter [Optional connected mechanisms](#) [172]

Process description

The ancillary system settlement procedure A consists of the following steps:

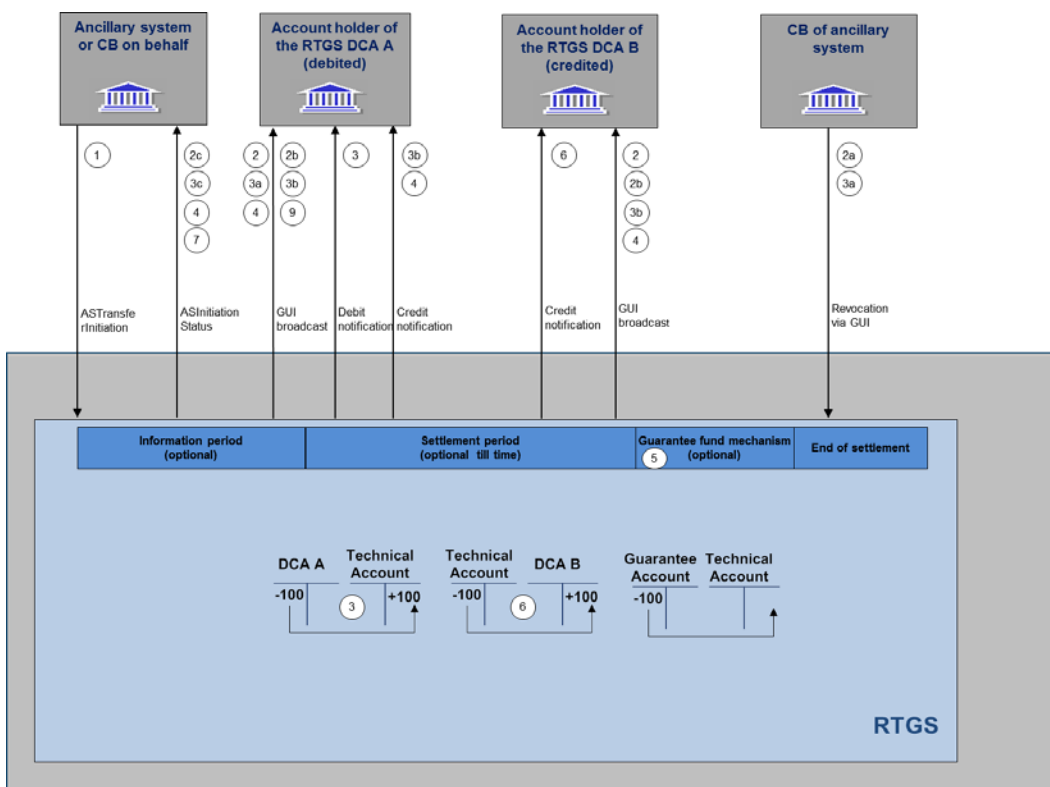


Figure 25 - Flow standard multilateral settlement

Phase	Step	Processing in/between	Description
Initiation	1	Ancillary system via ESMIG to RTGS	The ancillary system (or the relevant CB on its behalf) sends a ancillary system batch message (ASTransferInitiation) with all multilateral balances to be debited and credited on the Ancillary system settlement banks' RTGS DCAs
Information period	2	RTGS	If the "Information Period" option is used, the involved ancillary system settlement banks receive via GUI the broadcast notification on the start of the information period. If no ancillary system settlement bank disagrees (the suitable communication means has to be agreed within the contrac-

Phase	Step	Processing in/between	Description
			tual relationship with the ancillary system) during the information period the processing will continue
	2a	RTGS	If an ancillary system settlement bank disagrees, no settlement is triggered. The relevant CB will revoke the full ancillary system batch message via GUI
	2b	RTGS via ESMIG to ancillary system settlement banks	After disagreement all involved ancillary system settlement banks are informed via GUI broadcast about failure of settlement due to disagreement
	2c	RTGS via ESMIG to ancillary system	The ancillary system is informed about the settlement failure due to disagreement via ASInitiationStatus message. Processing stops
Settlement of debit positions	3	RTGS	Debits are processed for settlement. Once all of them have been settled, the credits will be processed immediately after. The settlement takes place with debiting the related ancillary system settlement banks' RTGS DCAs and crediting the Ancillary system technical account. Each debit ancillary system transfer is checked against the liquidity available in the related ancillary system settlement banks' RTGS DCAs. If the liquidity covers the needed amount, the ancillary system transfer is booked. The ancillary system settlement banks receive a debit notification (camt.054) after successful execution of their debit. If liquidity is not sufficient the ancillary system transfer is posted in queue
	3a	RTGS	The ancillary system settlement banks are informed about queuing by a GUI broadcast message. Immediately after putting the group of debits in the queue the optimisation process starts (settlement algorithms). Pending ancillary system transfers are settled by resolving the queue. The CB of the ancillary system is allowed to revoke the Ancillary system batch message as long as not final
	3b	RTGS via ESMIG to ancillary system settlement banks	A GUI broadcast is sent to all the involved ancillary system settlement banks informing about the settlement failure due to revocation. Already settled ancillary system transfers will be reversed and a credit notification (camt.054) is sent to the previously debited ancillary system settlement banks, if subscribed
	3c	RTGS via ESMIG to ancillary system	The ancillary system is informed about the settlement failure due to revocation via ASInitiationStatus message
	4	RTGS	If the ancillary system (or the relevant CB on its behalf) has indicated a "Settlement Period("till")" time, RTGS - if related ancillary system trans-

Phase	Step	Processing in/between	Description
			<p>fers are still pending - continuously checks whether the time limit is reached. If the time limit is exceeded, and guarantee fund mechanism is not set up, the settlement fails and the whole ancillary system batch message is rejected. Consequently RTGS will trigger the reversing procedure. The ancillary system technical account has to be debited and the ancillary system settlement banks' RTGS DCAs credited (only for those ancillary system transfers which were settled during the interrupted settlement cycle).</p> <p>The ancillary system is notified about the settlement failure with an ASInitiationStatus message, the ancillary system settlement banks receive a GUI broadcast informing about the failed settlement</p>
	5	RTGS	<p>If the time limit is exceeded and the guarantee fund mechanism is set up, it can be activated according to the agreed procedures. For details such as the involved messages and notifications please refer to chapter Optional connected mechanisms [172]</p>
Settlement of credit positions	6	RTGS	<p>RTGS processes all credits. The ancillary system settlement banks are informed via a credit notification (camt.054) on an optional basis</p>
End of settlement	7	RTGS via ESMIG to ancillary system	<p>After all ancillary system transfers have been settled the ancillary system (or the relevant CB on its behalf) receives a notification (ASInitiationStatus), confirming the settlement of the entire ancillary system batch message</p>

Table 60 - Process flow for standard multilateral settlement

At each step throughout the process information for ancillary system settlement banks and ancillary systems is available, please refer to chapter [Overview](#) [140].

Used messages

- | [ASTransferInitiation \(pain.998\)](#) [633], [ASInitiationStatus \(pain.998\)](#) [620]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [522]
- | [ReceiptAcknowledgement \(admi.007\)](#) [391]

5.3.3 Ancillary system settlement procedure B

Basics

Ancillary systems can settle a set of multilateral balances (debits and credits) on RTGS DCAs in a batch mode.

The RTGS component will be responsible for settling all debits and credits received in such a set of ancillary system transfers simultaneously. The identification of debit or credit ancillary system transfers is made from the ancillary system technical account perspective. Whenever an ancillary system settlement bank's RTGS DCA is debited and the ancillary system technical account is credited, the ancillary system transfer is considered as a debit whilst debiting the ancillary system technical account and crediting the ancillary system settlement bank's RTGS DCA will be considered a credit. Additionally the sum of all debits must be equal to the sum of credits within one ancillary system batch message. The usage of the ancillary system technical account is thus mandatory (i.e. each ancillary system transfer will have to present the ancillary system technical account on either debit or credit side).

In order to achieve the simultaneous execution of debits and credits, the ancillary system settlement procedure B benefits from the usage of a dedicated settlement algorithm (please refer to chapter [Dissolution of the payment queue](#) [132]). During the optimisation algorithm, RTGS checks that there is sufficient liquidity to settle all debit and credit ancillary system transfers of an ancillary system simultaneously ("All or nothing"). If this check is successfully passed, all debit and credit ancillary system transfers are settled simultaneously. If the check fails, all linked Ancillary system transfers remain in the queue and the partial optimisation with ancillary system optimisation algorithm is triggered again.

In order to limit the negative impact of failed settlement, the ancillary system can make use of the guarantee fund mechanism. Due to the above mentioned optimisation, prior to the optional running of the guarantee fund mechanism it is needed to single out the failed ancillary system transfers. This is achieved by transforming all ancillary system transfers from ancillary system settlement procedure B into ancillary system settlement procedure A and settling those debits covered by the needed liquidity. This mechanism implies also that the ancillary system technical account used for ancillary system settlement procedure B cannot be used in any other procedure in the framework of ancillary system processing.

Optional connected mechanisms

The ancillary system settlement procedure B may include optional connected mechanisms:

- information period
- settlement period ("till")
- guarantee fund mechanism

For further details on the usage and functionalities offered by the optional connected mechanisms please refer to chapter [Optional connected mechanisms](#) [172].

Process description

The ancillary system settlement procedure B consists of the following steps

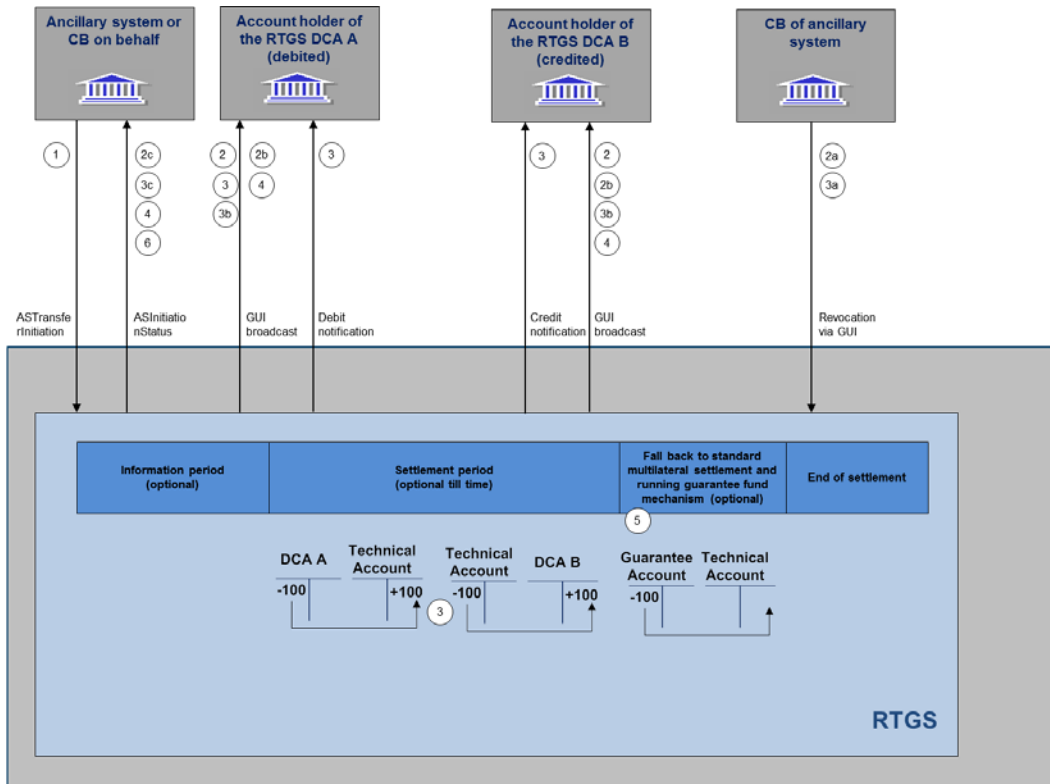


Figure 26 - Flow simultaneous multilateral settlement

Phase	Step	Processing in/between	Description
Initiation	1	Ancillary system via ESMIG to RTGS	The ancillary system (or the relevant CB on its behalf) sends an ancillary system batch message (ASTransferInitiation) with all multilateral balances to be debited and credited on the ancillary system settlement banks accounts.
Information period	2	RTGS	If the "Information Period" option is used, the involved ancillary system settlement banks receive via GUI the broadcast notification on the start of the information period. If no ancillary system settlement bank disagrees (the suitable communication means have to be agreed within the contractual relationship with the ancillary system) during the information period the processing will continue.
	2a	RTGS	If an ancillary system settlement bank disagrees, no settlement is triggered. The relevant CB revokes the ancillary system batch message via GUI.
	2b	RTGS via ESMIG to ancillary system	After disagreement all involved ancillary system settlement banks are informed via GUI broadcast about failure of settlement due to disagree-

Phase	Step	Processing in/between	Description
		settlement banks	ment.
	2c	RTGS via ESMIG to ancillary system	The ancillary system is informed about the settlement failure due to disagreement via ASInitiationStatus message. Processing stops.
Settlement	3	RTGS	In case no revocation due to disagreement applies, debits and credits are processed simultaneously for settlement using the optimisation algorithm. RTGS checks that there is sufficient liquidity to settle all debit and credit ancillary system transfers of an ancillary system simultaneously. If this check is successfully passed, all debit and credit Ancillary system transfers are booked simultaneously. If the check fails, all linked ancillary system transfers remain in the queue and the partial optimisation with ancillary system algorithm is triggered again. Via GUI it is possible to single out the RTGS DCAs not having enough liquidity.
	3a	RTGS	The CB of the ancillary system is allowed to revoke the ancillary system batch message as long as not final.
	3b	RTGS via ESMIG to ancillary system settlement banks	A broadcast is sent to all the involved ancillary system settlement banks informing about the settlement failure due to revocation.
	3c	RTGS via ESMIG to ancillary system	The ancillary system is informed about the settlement failure due to revocation via ASInitiationStatus message.
	4	RTGS	If the ancillary system (or the relevant CB on its behalf) has indicated a "Settlement Period ("till")", RTGS - if related payments are still unsettled - continuously checks whether the time limit is reached. If the time limit is exceeded, and guarantee fund mechanism is not set up, the settlement fails and the ancillary system batch message is rejected. The ancillary system is notified of the settlement failure with ASInitiationStatus message, the ancillary system settlement banks receive a GUI broadcast informing about the failed settlement.
	5	RTGS	<p>If the time limit is exceeded and the guarantee fund mechanism is set up, it can be activated according to the agreed procedures. For details such as the involved messages and notifications please refer to chapter Optional connected mechanisms [> 172].</p> <p>In order to identify the ancillary system transfers not covered by the needed liquidity, all ancillary system transfers will be transferred into ancillary system settlement procedure A and a single settlement attempt will be made (i.e. first all debits are executed, please refer to chapter Ancillary system settlement procedure A [> 144]). Only afterwards the</p>

Phase	Step	Processing in/between	Description
			guarantee fund mechanism is started. In such a scenario, it has to be kept in mind that the execution of debits and credits will not be simultaneous anymore. This behavior also implies that, in case the guarantee mechanism ends unsuccessfully (i.e. an error within the ancillary systems guarantee procedures), a reversal of the already settled debits is to be executed.
End of settlement	6	RTGS via ESMIG to ancillary system	After all ancillary system transfers have been settled the ancillary system (or the relevant CB on its behalf) receives a notification (ASInitiationStatus), confirming the settlement of the ancillary system batch message.

Table 61 - Process flow for standard multilateral settlement

At each step throughout the process information for ancillary system settlement banks and ancillary systems is available, please refer to chapter [Overview](#) [▶ 140].

Used messages

- | [ASTransferInitiation \(pain.998\)](#) [▶ 633], [ASInitiationStatus \(pain.998\)](#) [▶ 620]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522]
- | [ReceiptAcknowledgement \(admi.007\)](#) [▶ 391]

5.3.4 Settlement on dedicated liquidity accounts (ancillary system settlement procedure C and ancillary system settlement procedure D)

Basics

Ancillary systems which run settlement procedures based on the confidence of liquidity “fixed” amount (i.e. having continuous knowledge about the present and needed liquidity) can benefit from a pre-funding function that allows ancillary system settlement banks to set aside the needed liquidity in one or more separate sub-accounts dedicated to a specific ancillary system (ancillary system settlement procedure C) or the ancillary system technical account of a specific ancillary system (ancillary system settlement procedure D).

The settlement on dedicated liquidity accounts (ancillary system settlement procedure C and ancillary system settlement procedure D) must therefore be used to settle balances for an amount equal to or lower than the set aside liquidity.

Settlement on dedicated liquidity is a functionality to provide liquidity in batch-mode (i.e. more than one) either for ancillary system settlement procedure C or for ancillary system settlement procedure D.

Accounting

Following accounts are suitable for the described procedures:

Account type	Account holder	Description	Procedure
Sub-account	Ancillary system settlement bank	Used to set aside liquidity for exclusive settlement of a specific ancillary system and needs to be mapped to the RTGS DCA.	Ancillary system settlement procedure C
Ancillary system technical account	Ancillary system or the CB of the ancillary system	Used as intermediary account for the collection of debits and credits resulting from the settlement of ancillary system transactions and used to collect liquidity for settlement procedure D.	<ul style="list-style-type: none"> Ancillary system settlement procedure C Ancillary system settlement procedure D

Table 62 - Accounting

A sub-account is identified by the BIC of the related RTGS Account Holder in combination with an account number that is specific for the sub-account. Only RTGS Account Holders can hold such a sub-account.

The ancillary system settlement banks participating in ancillary systems using ancillary system settlement procedure C have to open one sub-account per ancillary system.

Procedures and cycles

Settlement with dedicated liquidity is a standardised procedure in the RTGS component. It is operated in so-called procedures and cycles. For the settlement with dedicated liquidity one mandatory procedure is used, which is automatically opened by the RTGS component at 19:30h on calendar day C (business day D) and automatically closed at 18:00h on calendar day C+1 (business day D). In addition there is an optional procedure (only to be used in ancillary system settlement procedure C), which the ancillary system can open and close as often as needed during the operational hours for ancillary system processing, after the mandatory procedure was closed beforehand by the ancillary system or the CB on behalf.

Within a procedure several cycles for settlement can run consecutively. The functionality can be used in order to block the liquidity on the ancillary system technical account (settlement procedure D) or the sub-accounts (settlement procedure C). Before a cycle is started, a certain period of time for liquidity transfer orders between the RTGS DCAs and sub-accounts or between the RTGS DCAs and ancillary system technical account shall be foreseen. The closing of the mandatory procedure (ancillary system settlement procedure C only) and the opening/closing of the optional procedure as well as the opening and closing of cycles can be done via A2A messages or GUI screen.

Set aside liquidity

To set aside liquidity for the settlement different options are offered by the RTGS. Please refer to chapter [Liquidity management features](#) [191].

5.3.4.1 Ancillary system settlement procedure C

As mentioned above, the ancillary system settlement procedure C is based on ancillary system transfers initiated by ancillary systems between the ancillary system settlement banks' sub-accounts and the ancillary system technical account held by the ancillary systems. For credits only, also the RTGS DCA of the ancillary system settlement bank can be addressed.

The ancillary system settlement banks dedicate liquidity to the settlement of ancillary systems by opening at least one sub-account per ancillary system they are settling with using ancillary system settlement procedure C. It is possible to open several sub-accounts for one ancillary system (e.g. to allow a segregation of business).

The ancillary system needs an ancillary system technical account which is to be used for the settlement. The settlement then takes place from sub-accounts towards ancillary system technical accounts (debits) and from ancillary system technical accounts towards sub-accounts or RTGS DCAs (credits).

During the whole process, the ancillary system will be notified of the amounts available on the sub-accounts. This happens whenever the liquidity on sub-accounts changes (by standing liquidity transfer orders or immediate liquidity transfer orders) or by providing the result of the settlement instructed by the ancillary systems (i.e. ASInitiationStatus). Thus the ancillary system is always in a position to know the liquidity set aside for their settlement. Once a settlement cycle is opened, the liquidity on the sub-accounts will be blocked. Settlement shall only be started once the liquidity needed is available on the sub-accounts. In turn, during the settlement cycle only on an exceptional basis (i.e. an error on ancillary system side) ancillary system transfers can be pending on sub-accounts due to missing liquidity.

Liquidity provision

Liquidity will be dedicated by the ancillary system settlement banks on the sub-accounts opened for the ancillary system settlement. The setting aside of liquidity in the framework of ancillary system settlement procedure C can be done by

- | Setting up standing liquidity transfer orders in reference data (to be executed with each start of procedure), for mandatory and optional settlement procedure different standing liquidity transfer orders can be stored. Standing liquidity transfer orders set up in reference data will only become effective with the next business day.
- | sending LiquidityCreditTransfer (camt.050) messages (immediate liquidity transfer order)
- | using the dedicated RTGS GUI liquidity transfer order screens (immediate liquidity transfer order)

the ancillary system sending ASTransferInitiation messages debiting the ancillary system settlement banks' RTGS DCA and crediting the same ancillary system settlement bank's sub-account (immediate liquidity transfer order)

Liquidity transfer orders will be executed in the following way:

Standing liquidity transfer orders are executed with each start of procedure (different amounts for mandatory and optional procedure can be specified).

Immediate liquidity transfer orders will be executed during an open procedure (mandatory or optional settlement procedure). They will be executed with immediate effect during an open procedure with no cycle running. In the opposite case, where a cycle is running, the immediate liquidity transfer order will be stored and executed only once the cycle has closed.

Effects on liquidity transfers in case of missing liquidity

Due to the peculiarities of the two different settlement procedures (mandatory and optional), the amounts taken into account for the execution of the different types of liquidity transfer orders are explained below.

Liquidity transfer type	Initiator	Mandatory procedure	Optional procedure
standing liquidity transfer order	Ancillary system settlement bank	If the total sum of all standing liquidity transfer orders of an ancillary system settlement bank is larger than the liquidity on its RTGS DCA, all standing liquidity transfer orders will be reduced in a pro-rata mode, i.e. the existing liquidity is divided by the total sum of standing liquidity transfer orders and the resulting factor will be used to reduce each standing liquidity transfer order of this RTGS Account Holder	<p>If a standing liquidity transfer order is not covered, it will be rejected.</p> <p>If several ancillary systems have launched their procedures the standing liquidity transfer orders are executed in the same order as of the incoming start of procedure messages from the different ancillary systems (FIFO principle)</p>
immediate liquidity transfer order	Ancillary system settlement bank	<p>Rejected if liquidity is not sufficient to execute the immediate liquidity transfer order amount requested.</p> <p>In case an urgent payment is pending in queue and has been submitted earlier than the immediate liquidity transfer order, the immediate liquidity transfer order will be rejected.</p>	<p>Rejected if liquidity is not sufficient to execute the immediate liquidity transfer order amount requested.</p> <p>In case an urgent payment is pending in queue and has been submitted earlier than the immediate liquidity transfer order, the immediate liquidity transfer order will be rejected.</p>
	Ancillary system (or CB on behalf)	<p>Partial execution. (i.e. up to the available liquidity on the RTGS DCA or on the sub-account concerned)</p> <p>In case an urgent payment is pending in queue of the ancillary system settlement bank and has been submitted earlier than the immediate liquidity transfer order, the immediate liquidity transfer order will be rejected.</p>	<p>Partial execution. (i.e. up to the available liquidity on the RTGS DCA or on the sub-account concerned)</p> <p>In case an urgent payment is pending in queue of the ancillary system settlement bank and has been submitted earlier than the immediate liquidity transfer order, the immediate liquidity transfer order will be rejected.</p>

Table 63 - Amounts taken into account**Mandatory procedure**

The mandatory procedure is opened by the RTGS component on the new business day (19:30h) in an automated way for all ancillary systems using ancillary system settlement procedure C. This procedure cannot be reopened by the ancillary system (or its CB on behalf). In case the mandatory procedure was closed by the ancillary system (or its CB on behalf), settlement can only take place by opening an optional settlement procedure (which may imply different amounts being set aside by the ancillary system settlement banks by using standing liquidity transfer orders). Closing the mandatory procedure will launch the sweeping out of liquidity dedicated to the ancillary system, i.e. the balances present on the sub-accounts will be retransferred to the linked RTGS DCAs.

Optional procedure

Any optional procedure requires the ancillary system (or CB on behalf) to close the mandatory procedure beforehand. The ancillary system can open and close the optional procedure as often as needed during the operational hours for ancillary system processing. With each opening of this procedure the linked standing liquidity transfer order will be executed, debiting the RTGS DCA and crediting the sub-accounts of the ancillary system settlement banks. With each closure of the procedure the remaining liquidity on the sub-accounts is swept back to the linked RTGS DCA.

Overview on the settlement process

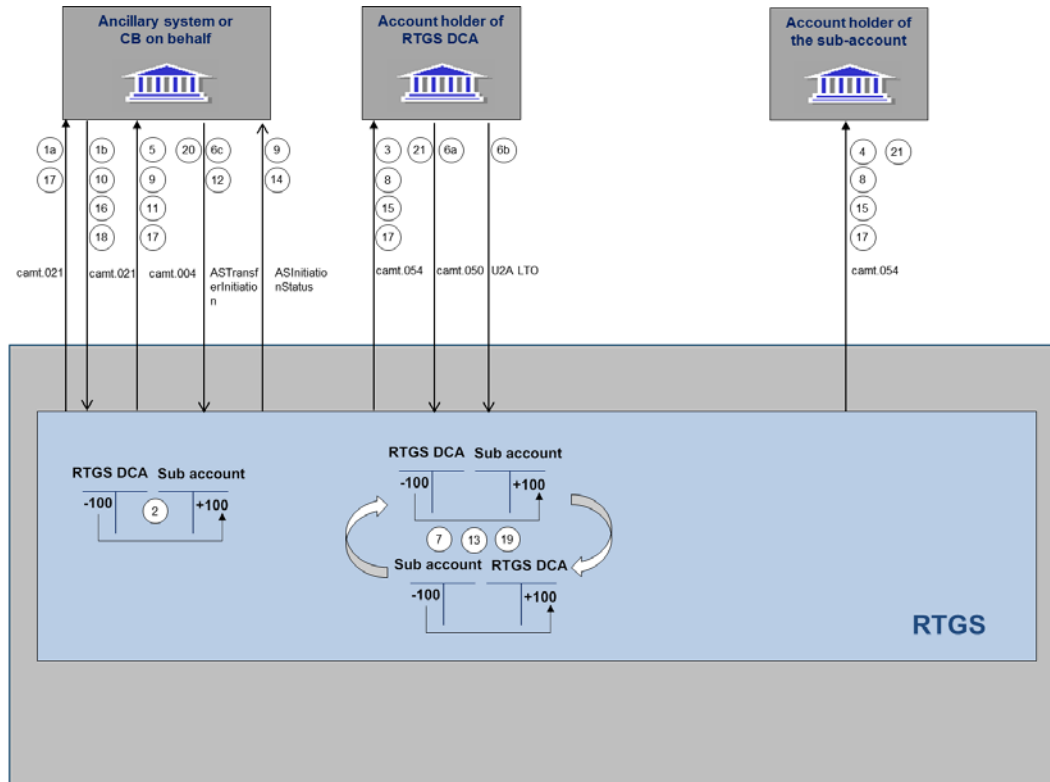


Figure 27 - Flow settlement on dedicated liquidity accounts (ancillary system settlement procedure C)

Phase	Step	Processing with	Description
Start of procedure	1a (mandatory procedure)	RTGS via ESMIG to ancillary system	Start of procedure message for mandatory procedure is automatically initiated by the RTGS component at 19:30h of new business day. The ancillary system will be notified of the event (camt.021-ReturnGeneralBusinessInformation)
	1b (optional procedure)	Ancillary system via ESMIG to RTGS	The ancillary system (or CB on behalf) sends a message (camt.021-ReturnGeneralBusinessInformation) indicating the start of the optional procedure. The closure of the mandatory procedure prior to this is mandatory. The procedure can also be opened using U2A GUI screen
standing liquidity transfer order execution	2	RTGS	The start of procedure triggers the execution of existing standing liquidity transfer orders debiting the ancillary system settlement banks' RTGS DCAs and crediting the pertaining sub-accounts

Phase	Step	Processing with	Description
	3	RTGS via ESMIG to ancillary system settlement banks	On an optional basis, the ancillary system settlement banks are notified of the debited amount on the RTGS DCA (camt.054 Debit notification)
	4	RTGS via ESMIG to ancillary system settlement banks	On an optional basis, the ancillary system settlement banks are notified of the credited amount on the sub-account (camt.054 Credit notification)
	5	RTGS via ESMIG to ancillary system	The ancillary system is notified of credit of the sub-account for the amounts actually booked (camt.004-ReturnAccount)
Liquidity adjustment	6a	Ancillary system Settlement banks via ESMIG to RTGS	Ancillary system settlement banks can adjust (increase or decrease) the liquidity on the sub-accounts by using immediate liquidity transfer orders (camt.050).
	6b	Ancillary system settlement banks via ESMIG to RTGS	Ancillary system settlement banks can adjust (increase or decrease) the liquidity on the sub-accounts by using immediate liquidity transfer orders (dedicated U2A GUI screens).
	6c	Ancillary system via ESMIG to RTGS	The ancillary system can take over the responsibility to manage the liquidity on the sub-account (e.g. based on standing liquidity transfer orders or immediate liquidity transfer orders stored within the ancillary system) by sending the liquidity transfer order (increase or decrease) via ASTransferInitiation to RTGS
	7	RTGS	The liquidity transfer orders are processed between the RTGS DCAs and sub-accounts.
	8	RTGS via ESMIG to ancillary system settlement banks	The ancillary system settlement banks are informed on an optional basis with camt.054 on the debits/credits executed on their RTGS DCAs and sub-accounts.
	9	RTGS via ESMIG to ancillary system	The ancillary system is notified:

Phase	Step	Processing with	Description
			<p>–with camt.004 ReturnAccount if the ancillary system settlement bank has issued the immediate liquidity transfer order.</p> <p>–with ASInitiationStatus if the ancillary system has issued the immediate liquidity transfer order</p>
Start of cycle	10	Ancillary system via ESMIG to RTGS	In order to block the liquidity set aside on the sub-accounts, the ancillary system can open a settlement cycle using camt.021-ReturnGeneralBusinessInformation message (or via dedicated U2A GUI screen).
Blocking of liquidity	11	RTGS via ESMIG to ancillary system	<p>Once the cycle is started, the liquidity on the sub-accounts is blocked as long as the cycle is open. Any immediate liquidity transfer order on the sub-account will be stored and executed only once the cycle has been closed.</p> <p>The ancillary system is notified of the liquidity blocked on all sub-accounts with camt.004-ReturnAccount message.</p>
Settlement	12	Ancillary system (or CB on behalf) via ESMIG to RTGS	The ancillary system instructs the settlement transactions with ASTransferInitiation.
	13	RTGS	<p>Settlement takes place debiting the sub-accounts and crediting ancillary system technical accounts and afterwards debiting the ancillary system technical account and crediting the sub-accounts (crediting can take place directly on the RTGS DCA if indicated by the ancillary system). In case (due to error on ancillary system side) one or more transactions are not covered by the needed liquidity, the transactions remain queued on the sub-account.</p> <p>At the end of the cycle all transactions debiting the same sub-account with insufficient liquidity for their settlement are re-</p>

Phase	Step	Processing with	Description
			jected even if only one transaction cannot be settled. The settlement can avail itself of the optimisation process (i.e. settlement algorithm, please refer to Dissolution of the payment queue [132])
	14	RTGS via ESMIG to ancillary system (or CB on behalf)	After the end of the settlement the ancillary system will receive one message as confirmation. The message will contain a list of the credits and debits settled (ASInitiation-Status). If some transactions are not settled until the end of cycle, the ASInitiation-Status will be sent at the end of the cycle with the individual status of each transaction.
	15	RTGS via ESMIG to ancillary system settlement banks	On an optional basis ancillary system settlement banks receive camt.054 notifications for the debits and credits on the sub-accounts respectively credits on RTGS DCAs
End of cycle	16	Ancillary system (or CB on behalf) via ESMIG to RTGS	Ancillary system (or CB on behalf) sends an XML "end of cycle" message to RTGS (camt.021-ReturnGeneralBusinessInformation (optional in U2A via GUI))
	17	RTGS via ESMIG to ancillary system (or CB on behalf)	<p>The remaining liquidity on the sub-accounts is released and the ancillary system is notified with camt.021-ReturnGeneralBusinessInformation.</p> <p>Stored liquidity transfer orders will now be executed and informed via camt.054 to ancillary system settlement banks and camt.004-ReturnAccount to the ancillary system.</p> <p>A new liquidity adjustment phase is now available. The ancillary system can also start a new cycle.</p>

Phase	Step	Processing with	Description
End of procedure	18	Ancillary system (or CB on behalf) via ESMIG to RTGS	Ancillary system (or CB on behalf) can send an end of procedure message (camt.021-ReturnGeneralBusinessInformation) or using the U2A GUI functionality to close the procedure.
	19	RTGS	Once the procedure was closed (automatically at 18:00h or as the procedure was closed manually), the remaining liquidity on sub-accounts is transferred back to the ancillary system settlement banks' RTGS DCAs
	20	RTGS via ESMIG to ancillary system (or CB on behalf)	The ancillary system is informed via camt.004-ReturnAccount on the back transfer of liquidity if ordered manually, i.e. it is not sent in case the procedure is closed by RTGS at 18:00
	21	RTGS via ESMIG to ancillary system settlement banks	On an optional basis the ancillary system settlement banks receive camt.054 notifications on the back transfer of liquidity.

Table 64 - Start of procedure and liquidity provision

Used messages

- | [ASTransferInitiation \(pain.998\) \[▶ 633\]](#), [ASInitiationStatus \(pain.998\) \[▶ 620\]](#)
- | [BankToCustomerDebitCreditNotification \(camt.054\) \[▶ 522\]](#)
- | [Receipt \(camt.025\) \[▶ 474\]](#)
- | [ReceiptAcknowledgement \(admi.007\) \[▶ 391\]](#)
- | [ReturnGeneralBusinessInformation \(camt.021\) \[▶ 458\]](#)
- | [ReturnAccount \(camt.004\) \[▶ 397\]](#)
- | [LiquidityCreditTransfer \(camt.050\) \[▶ 497\]](#)

5.3.4.2 Ancillary system settlement procedure D

As mentioned above, the ancillary system settlement procedure D is based on ancillary system transfers initiated by ancillary systems between the settlement banks' RTGS DCA and the ancillary system technical

account. The ancillary system includes this liquidity to the settlement bank's account held within the ancillary system.

For ancillary system settlement procedure D the settlement phase is an internal process of the ancillary system and therefore no details are provided here.

During the whole process, the ancillary system will be notified about the amounts available on the ancillary system technical account. This happens whenever the liquidity on this account changes (by standing liquidity transfer orders or immediate liquidity transfer orders) or by providing the result of the settlement instructed by the ancillary system (i.e. ASInitiationStatus). Thus, the ancillary system is always in a position to know the liquidity set aside for their settlement. When the procedure is closed at 18:00h the ancillary system technical account can have a non-zero-balance.

Liquidity provision

Liquidity will be dedicated by the settlement banks on the ancillary system technical account opened for the ancillary system. The setting aside of liquidity in the framework of ancillary system settlement procedure D can be done by

- | setting up standing liquidity transfer orders in reference data (to be executed with the start of mandatory procedure). Standing liquidity transfer orders set up in reference data will only become effective with the next business day.
- | sending pacs.009 SettlementBankTransferInitiation (SBTI) messages (immediate liquidity transfer order)
- | using the dedicated RTGS GUI liquidity transfer screens (immediate liquidity transfer order)
- | the ancillary system sending ASTransferInitiation messages debiting the settlement banks RTGS DCA and crediting the ancillary system technical account (immediate liquidity transfer order)

Liquidity transfer orders will be executed in the following way:

- | standing liquidity transfer orders are executed with each start of mandatory procedure.
- | immediate liquidity transfer orders will be executed during an open procedure. They will be executed with immediate effect during an open procedure with no cycle running. In the opposite case, where a cycle is running, the liquidity transfer will be stored and executed only once the cycle has closed.

Effects on liquidity transfers in case of missing liquidity

The amounts taken into account for the execution of the different types of liquidity transfer orders are explained below.

Liquidity transfer type	Initiator	Mandatory procedure
Standing liquidity transfer order	Settlement bank	If the total sum of all standing liquidity transfer orders of an ancillary system settlement bank is larger than the liquidity on its RTGS DCA, all standing liquidity transfer orders will be reduced in a pro-rata mode, i.e. the existing liquidity is divided by the total sum of standing liquidity transfer orders and the resulting factor will be used to reduce each standing liquidity transfer order of this account holder
Immediate liquidity transfer order	Settlement bank	Rejected if liquidity is not sufficient to execute the immediate liquidity transfer order amount requested. In case an urgent payment is pending in the queue and has been submitted earlier than the immediate liquidity transfer order, the immediate liquidity transfer order will be rejected
	Ancillary system (or CB on behalf)	Partial execution. (i.e. up to the available liquidity on the RTGS DCA concerned) In case an urgent payment is pending in the queue of the settlement bank and has been submitted earlier than the immediate liquidity transfer order, the immediate liquidity transfer order will be rejected

Table 65 - Amounts taken into account

Mandatory procedure

The mandatory procedure is opened by the RTGS component on the new business day (19:30h) in an automated way for all ancillary systems using ancillary system settlement procedure D. The mandatory procedure cannot be closed or reopened.

Overview on the settlement process

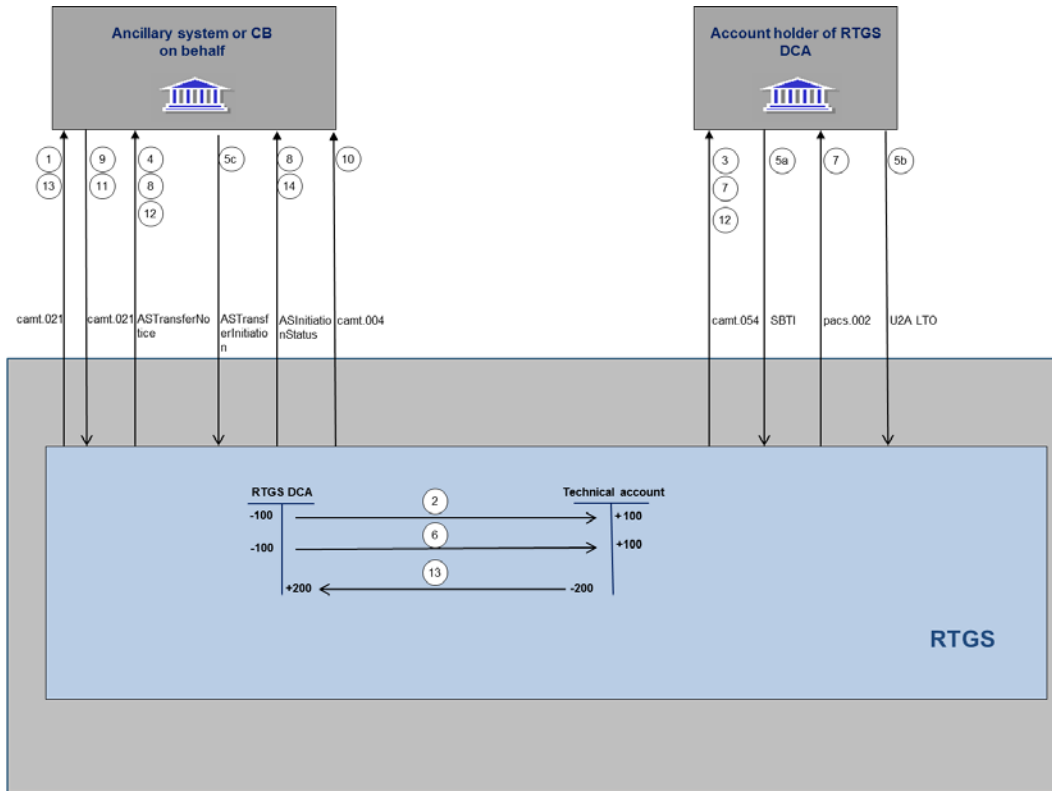


Figure 28 - Flow real-time – start of procedure

Phase	Step	Processing with	Description
Start of procedure	1	RTGS via ESMIG to ancillary system	Start of procedure message for mandatory procedure is automatically initiated by the RTGS component at 19:30h on the new business day. The ancillary system will be notified of the event (camt.021 ReturnGeneralBusiness-Information).
standing liquidity transfer order execution	2	RTGS	The start of procedure triggers the execution of existing standing liquidity transfer orders debiting the settlement bank's RTGS DCAs and crediting the ancillary system technical account.
	3	RTGS via ESMIG to settlement banks	On an optional basis, the settlement banks are notified of the debited amount on the RTGS DCA (camt.054 Debit notification)
	4	RTGS via ESMIG to ancillary system	The ancillary system is notified of the

Phase	Step	Processing with	Description
		tem	credit of the ancillary system technical account for the amounts actually settled (ASTransferNotice)
Liquidity adjustment	5a	Settlement banks via ESMIG to RTGS	Settlement banks can adjust (increase) the liquidity on the ancillary system technical account by using immediate liquidity transfer orders (pacs.009-SBTI).
	5b	Settlement banks via ESMIG to RTGS	Settlement banks can adjust (increase) the liquidity on the ancillary system technical account by using immediate liquidity transfer orders (dedicated U2A GUI screens).
	5c	Ancillary system via ESMIG to RTGS	The ancillary system can take over the responsibility to manage the liquidity on the ancillary system technical account based on immediate liquidity transfer orders stored within the ancillary system by sending the liquidity transfer order via ASTransferInitiation to RTGS. The ancillary system cannot set standing liquidity transfer orders on behalf of its settlement bank, to provide such a functionality the ancillary system has to store and manage its own procedure outside the RTGS and send them at the appropriate time as immediate liquidity transfer orders.
	6	RTGS	The liquidity transfers are processed between the RTGS DCAs and the ancillary system technical account.
	7	RTGS via ESMIG to settlement banks	The ancillary system settlement banks are informed on an optional basis with camt.054 or pacs.002 (if instructed via SBTI) on the debits executed on their RTGS DCAs.
	8	RTGS via ESMIG to ancillary system	Notified to the ancillary system:

Phase	Step	Processing with	Description
			<p>with ASTransferNotice when the settlement bank has issued the immediate liquidity transfer order.</p> <p>with ASInitiationStatus when the ancillary system has issued the immediate liquidity transfer order</p>
Start of cycle	9	Ancillary system via ESMIG to RTGS	<p>Ancillary system sends a “start of cycle messages” to RTGS (camt.021 ReturnGeneralBusinessInformation (optional in U2A via GUI)).</p> <p>The incoming immediate liquidity transfer orders will not be any longer immediately executed.</p>
	10	RTGS via ESMIG to ancillary system	<p>The ancillary system is notified about the global amount on the ancillary system technical account with a camt.004 ReturnAccount</p>
End of cycle	11	Ancillary system (or CB on behalf) via ESMIG to RTGS	<p>Ancillary system sends an XML “end of cycle” message to RTGS (camt.021 ReturnGeneralBusinessInformation (optional in U2A via GUI))</p>
	12	RTGS via ESMIG to ancillary system and settlement banks	<p>Possible immediate liquidity transfer orders received during the cycle will now be executed and informed via camt.054 to settlement banks and ASTransferNotice to the ancillary system.</p> <p>Reverse liquidity transfers issued by the ancillary system aiming at debiting ancillary system technical account and crediting the RTGS DCA are also possible to allow the reallocation of the liquidity in favor of other settling ancillary systems.</p>

Phase	Step	Processing with	Description
Liquidity release	13	RTGS via ESMIG to ancillary system	The RTGS releases the remaining liquidity and notifies the ancillary system about the closure of the cycle with a camt.021 ReturnGeneralBusinessInformation

Table 66 - Start of procedure and liquidity provision

Used messages

- | [ASTransferInitiation \(pain.998\) \[▶ 633\]](#), [ASInitiationStatus \(pain.998\) \[▶ 620\]](#), [ASTransferNotice \(pain.998\) \[▶ 610\]](#)
- | [BankToCustomerDebitCreditNotification \(camt.054\) \[▶ 522\]](#)
- | [Receipt \(camt.025\) \[▶ 474\]](#)
- | [ReceiptAcknowledgement \(admi.007\) \[▶ 391\]](#)
- | [ReturnGeneralBusinessInformation \(camt.021\) \[▶ 458\]](#)
- | [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\) \[▶ 585\]](#) used as SettlementBankTransferInitiation (pacs.009-SBTI)
- | [ReturnAccount \(camt.004\) \[▶ 397\]](#)

5.3.4.3 Cross-ancillary system settlement

Basics

In addition to the above described procedures for the settlement of ancillary system, there is also the possibility to send ancillary system transfers on a cross-ancillary system basis. As a precondition to use this feature a bilateral agreement between the sending and receiving ancillary systems needs to be in place. This agreement has to be put in the reference data by the relevant CBs on behalf of the ancillary systems. The functionality of cross-ancillary system settlement is independent from the procedure the sending and receiving ancillary systems are using (i.e. ancillary system settlement procedure D vs. ancillary system settlement procedure C). Such transfers are always instructed by the ancillary system (or its CB on behalf) as a single ancillary system transfer via an ASTransferInitiation. A prerequisite for the settlement of such ancillary system transfer is that the ancillary system transfer is sent during an open cycle of the sending ancillary system (only valid if sending ancillary system is using settlement procedure C) and an open procedure of the receiving ancillary system (regardless if mandatory or optional procedure is open). In such case the settlement is executed immediately whatever is the status of the cycle of the receiving ancillary system. Reverse transactions (i.e “pulling” liquidity from another ancillary system) are not allowed.

Ancillary system procedure D ancillary system to ancillary system procedure C ancillary system

With this ancillary system transfer the ancillary system technical account of the sending ancillary system on behalf of an ancillary system settlement bank is debited in order to credit the sub-account of one of the ancillary system settlement banks of the receiving ancillary system. The receiving ancillary system is notified with an ASTransferNotice about the incoming liquidity to the sub-account including the information of the resulting balance. The receiving ancillary system has the possibility to use this credit immediately. The sending ancillary system or its CB on behalf is notified with an ASInitiationStatus about the outcome of the request. On an optional basis the ancillary system settlement bank of the receiving ancillary system is notified with a camt.054 Credit notification. If the settlement request was sent by the CB on behalf of the sending ancillary system, then it is notified on its execution with a ReturnAccount message.

Ancillary system procedure D ancillary system to ancillary system procedure D ancillary system

With this ancillary system transfer the ancillary system technical account of the sending ancillary system on behalf of an ancillary system settlement bank is debited in order to credit the ancillary system technical account of the receiving ancillary system in favor of one of the ancillary system settlement banks. The receiving ancillary system is notified with an ASTransferNotice about the incoming liquidity including the information of the resulting balance. The receiving ancillary system has the possibility to use this credit immediately. The sending ancillary system or its CB on behalf is notified with an ASInitiationStatus about the outcome of the request. If the settlement request was sent by the CB on behalf of the sending ancillary system, then it is notified on its execution with a ReturnAccount message.

Ancillary system procedure C ancillary system to ancillary system procedure C ancillary system

With this ancillary system transfer the sub-account of an ancillary system settlement bank of the sending ancillary system is debited in order to credit the sub-account of one of the ancillary system settlement banks of the receiving ancillary system. The receiving ancillary system is notified with an ASTransferNotice about the incoming liquidity to the sub-account including the information of the resulting balance. The receiving ancillary system has the possibility to use this credit immediately. The sending ancillary system or its CB on behalf is notified with an ASInitiationStatus of the outcome of the request. On an optional basis the ancillary system settlement banks of the receiving and sending ancillary systems are notified with camt.054 Credit/Debit notifications. If the settlement request was sent by the CB on behalf of the sending ancillary system, then it is notified on its execution with a ReturnAccount message.

Ancillary system procedure C ancillary system to ancillary system procedure D ancillary system

With this ancillary system transfer the sub-account of an ancillary system settlement bank of the sending ancillary system is debited in order to credit the ancillary system technical account of the receiving ancillary system in favour of one of the ancillary system settlement banks. The receiving ancillary system is notified with an ASTransferNotice about the incoming liquidity including the information of the resulting balance. The receiving ancillary system has the possibility to use this credit immediately. In case the liquidity on the sub-account is insufficient, the ancillary system transfer is rejected. The sending ancillary system or its CB on

behalf is notified with an ASInitiationStatus of the outcome of the request. On an optional basis the ancillary system settlement bank of the sending ancillary system is notified with a camt.054.Debit notification. If the settlement request was sent by the CB on behalf of the sending ancillary system, then it is notified on its execution with a ReturnAccount message.

5.3.5 Processing of ancillary system transactions using payments

Basics

Besides the instruction of ancillary system specific transactions using proprietary messages (ASTransferInitiation) also the usage of payments (pacs.009) is possible. In order to send pacs.009 debiting a settlement bank, the ancillary system needs to be authorised in reference data to send pacs.009 for the given RTGS DCA by the account owner.

The usage of pacs.009 for ancillary systems entails some peculiarities compared to their usage by normal RTGS Account Holders.

Type	Peculiarity
Priority	Urgent for ancillary system transactions (high and normal for other RTGS Account Holders). (Please note: only Priority URGENT is allowed)
Execution Date	Warehouse functionality is not allowed. Only the current business day can be used.
Instruction	Ancillary system can debit the RTGS DCAs of settlement banks where they are authorised for, i.e. the sender of the pacs.009 is not the owner of the debited RTGS DCA.
Notification to settlement banks	As the ancillary system is sender of the pacs.009, the debited settlement bank is notified on an optional basis via camt.054. The credited settlement bank receives the pacs.009 as created by the ancillary system.

Type	Peculiarity
Bundling of ancillary system related payments	Differently to other RTGS Account Holder, ancillary system have the option to bundle several (up to 20,000) pacs.009 into a file. This bundling is only applicable on inbound side. Outbound messages are not bundled into one file.
Identification of ancillary system related payments	The pacs.009 sent by ancillary system (and in the end forwarded to the credited settlement bank) has to contain a specific code word ("ASTI"). Additionally the BIC of the ancillary system has to be used in field "DebtorAgent". With this two fields (and URGENT priority) the receiver of the camt.054 as well as the receiver of the pacs.009 will be able to identify the transaction as driven by the indicated ancillary system.
Monitoring of ancillary system related payments	The monitoring functionalities of the ancillary system is strongly connected to the access rights granted by the settlement banks to the ancillary system as sender of those pacs.009.

Table 67 - Peculiarities for pacs.009 sent by ancillary system

When using payments, it is only possible to debit and credit RTGS DCAs. The usage of other account types (ASTechnicalAccount, sub-account etc.) is not allowed. Whilst the debit leg of the payment requires the authorization by the RTGS DCA owner, this is not applicable for the credit leg. Thus ancillary systems using payments might credit also RTGS DCAs not being assigned to as a settlement bank of this ancillary system.

The processing of payments sent by the ancillary system via pacs.009 follows the principles as defined in [Processing of payments](#) [124]. Even though it is possible to technically bundle several pacs.009 into a file (please refer to [BusinessFileHeader \(head.002\)](#) [566]), all pacs.009 will be treated individually. The single payments have a bilateral character.

Time indicators

In fact no optional connected mechanisms (refer to [Optional connected mechanisms](#) [172]) can be used for ancillary system using payments. Anyway the ancillary system can make use of time indicators (please refer to [Definition of execution time](#) [81]) to reach a similar processing as regards the times for start and end of settlement.

Note: Although the effect on settlement is very similar to Information Period and Settlement Period ('till') option, there are also differences (refer to [Optional connected mechanisms](#) [172]), in particular the absence of GUI broadcasts to settlement banks pre-warning them on an upcoming ancillary system transaction and allowing them to provide the needed liquidity. For this reason it could be recommended to set up internal means within the ancillary system's community to inform settlement banks on upcoming payments and the needed liquidity.

Process description

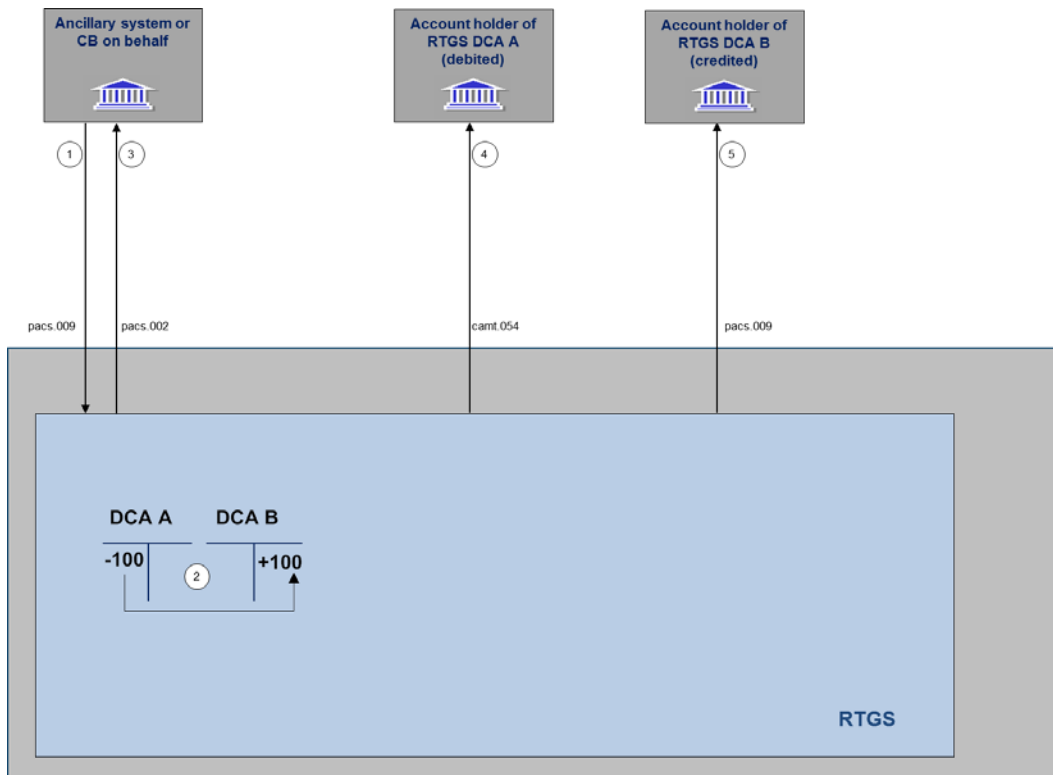


Figure 29 - Instruction using pacs.009

Phase	Step	Processing with	Description
Payment sending	1	Ancillary system via ESMIG to RTGS	The ancillary system sends pacs.009 (either one by one or optionally bundled into a file)
Payment processing	2	RTGS	<p>If the pacs.009 are bundled into a file, RTGS will unpack the single messages and forward the single pacs.009 for processing. The payments are processed according to the principles valid for all urgent payments (see Processing of payments [124]).</p> <p>If the RTGS DCA to be debited is not covered with the needed liquidity, the payment is queued until 18:00h or (if indicated) the “RejectTime” is reached. If, once the earliest of the above mentioned timestamps is reached, the payment is still queued, it will be rejected and no information will be sent to the settlement banks. As long as settlement is not final, the ancillary system can revoke a single payment via standard U2A function-</p>

Phase	Step	Processing with	Description
			<p>ality or using camt.056 (please refer to Revocation of payments [118]).</p> <p>Once the provided liquidity is available, the indicated RTGS DCA is debited and the indicated RTGS DCA on creditor side is credited</p>
Notification to ancillary system	3	RTGS via ESMIG to Ancillary System	<p>The sender of the pacs.009 is notified via pacs.002 on the settlement status as soon as a final status was reached. In case of settlement failure the pacs.002 is treated as mandatory, in case of successful execution the pacs.002 is optional.</p> <p>This process is valid for each single pacs.009 sent by the ancillary system, irrespective of whether it was sent individually or bundled into a file</p>
Notification to settlement banks	4	RTGS via ESMIG to debited settlement bank	<p>After execution of the payment the debited settlement bank is informed via optional camt.054. The information present in the reference fields of the initial pacs.009 is mapped into this camt.054.</p> <p>This process is valid for each single pacs.009 sent by the ancillary system, irrespective of whether it was sent individually or bundled into a file</p>
	5	RTGS via ESMIG to credited settlement bank	<p>Creation and forwarding of pacs.009 to the credited settlement bank generated by the RTGS component (mandatory).</p> <p>This process is valid for each single pacs.009 sent by the ancillary system, irrespective of whether it was sent individually or bundled into a file</p>

Table 68 - Process description for using plain payments

5.3.6 Optional connected mechanisms

General aspects

In connection with settlement of ancillary systems, a set of additional options is available which can be used for a more efficient liquidity management:

- Information Period

- Settlement Period (“till”)
- Guarantee fund mechanism

In order to use one or several of these optional mechanisms, the ancillary system either has to fill specific fields of the ASTransferInitiation or to rely on reference data (guarantee fund mechanism).

Only the ancillary system (or the CB acting on its behalf) is entitled to insert these parameters in the message. Once a message is sent the parameter can be updated in U2A mode by the ancillary system for optional mechanism “Settlement period” before the inserted “till”- time has been expired.

“Information Period” and “Guarantee fund mechanism” parameters can be updated neither by the ancillary systems nor by the settlement banks nor by CBs.

The table below summarizes which optional connected mechanism can be used with which ancillary system procedure.

Settlement Procedure	Information Period	Settlement Period (“till”)	Guarantee fund mechanism
Settlement procedure A	X	X	x
Settlement procedure B	X	X	X
Settlement procedure C			
Settlement procedure D			

Table 69 - Usability of optional connected mechanism per ancillary system processing procedure

Information Period

The Information Period option allows settlement banks a more efficient liquidity management giving the possibility of knowing in advance the liquidity needed for the settlement of a specific ancillary system transfer. This optional connected mechanism can be used for

- settlement procedure A
- settlement procedure B

The Information Period option can be used by indicating a specific end time (within the operational hours for ancillary system processing) or duration (the calculated end time as well has to be within the operational hours for ancillary system processing) within an ASTransferInitiation message. The start time of the information period is the time of reception after validations. The usage of this option will lead to:

- information about the needed liquidity and specified time to settlement banks
- possibility for settlement banks to disagree on the amount

Under certain circumstances settlement banks have the possibility to disagree on specific balances before settlement takes place. The business rules and regulations for disagreements need to be defined by the ancillary system and the relevant CB. Anyway, RTGS technically always allows the CB of the ancillary system to revoke the pertaining set of ancillary system transfers, i.e. there is no parameter controlling whether disagreement procedures are defined or not on the level of ancillary systems, their settlement banks and the ancillary system's CB.

Process flow

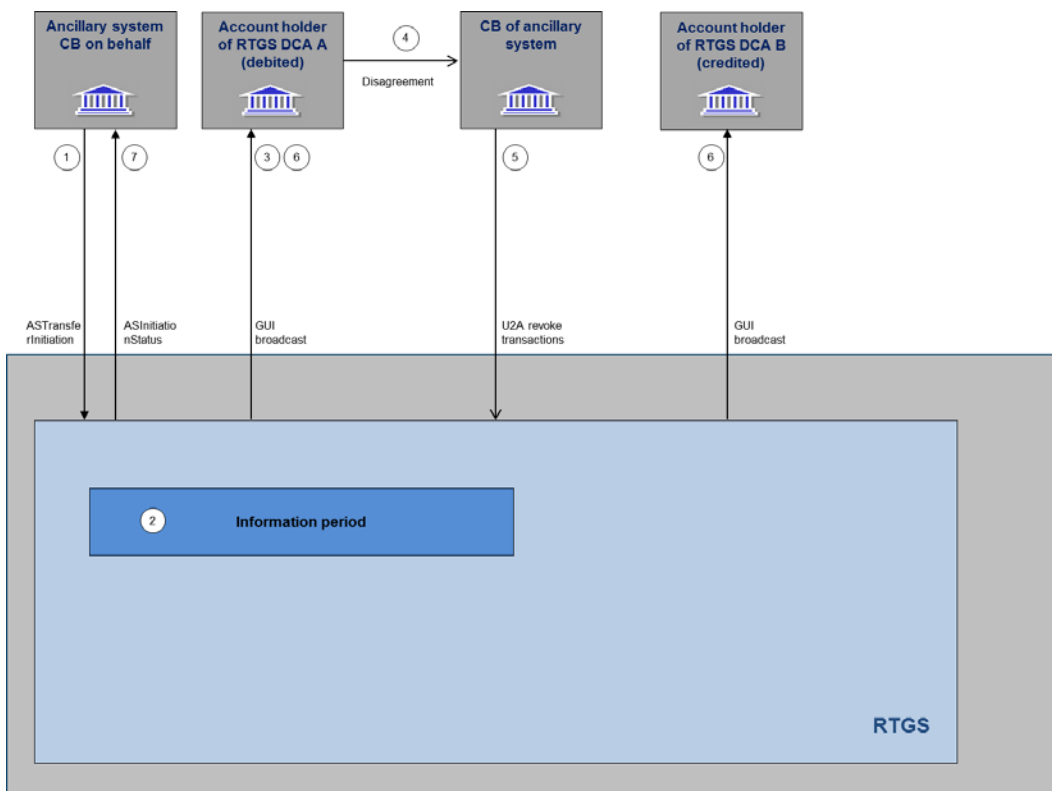


Figure 30 - Flow information period

Action	Step	Interaction	Description
Initiation	1	Ancillary system (or CB on behalf) via ESMIG to RTGS	The ancillary system or the CB on behalf of the ancillary system sends the ASTransferInitiation with the information period indicated in the GroupHeader. The time or duration indicated will be used for the start of the settlement period.
Information Period	2	RTGS	Immediately after reception and positive validations the information period starts.
	3	RTGS via ESMIG to settlement banks	With the start of the information period the settlement banks are informed via GUI broadcast on the indicated start of settlement period and the needed amount of liquidity.
	4	settlement bank to CB of	In case one or several settlement banks disagree on the

Action	Step	Interaction	Description
		the ancillary system	amount of the ancillary system transfers present in the pertaining set of transactions, it may contact the CB of the ancillary system. The procedure on if, when and how such disagreement is to be applied has to be set up internally within the ancillary system community. Also the way the settlement bank contacts the CB (directly or indirectly via the pertaining ancillary system) is out of scope of RTGS.
	5	CB via ESMIG to RTGS	The CB, via GUI revokes the disagreed set of transactions, leading to a rejection of all transactions and settlement is not triggered. The information period and all processing of the involved ancillary system transfers is stopped.
Notification in case of disagreement	6	RTGS via ESMIG to all settlement banks	All settlement banks are informed via broadcast on the rejection of the transactions due to disagreement
	7	RTGS via ESMIG to ancillary system	The ancillary system is informed via ASInitiationStatus message on the rejection due to disagreement
End of information period	8	RTGS	In case no disagreement was expressed during the information period, the indicated end of the information period will mark the start of the settlement period.

Table 70 - Process flow information period with disagreement

Used messages

ASTransferInitiation (pain.998) [▶ 633], ASInitiationStatus (pain.998) [▶ 620]

Settlement Period (“till”)

The settlement of an ancillary system may only take place during a pre-defined period of time. If the settlement is not completed successfully during this period of time the transactions are rejected or a guarantee fund mechanism is activated.

Similar to Information Period option, the settlement period (“till”) option has to be indicated per ASTransferInitiation in the GroupHeader of the message and is then valid for the whole set of transactions.

The ancillary system (or its responsible CB on behalf), according to rules established within the ancillary system’s community, can modify the end of the settlement period (“Change settlement period” in U2A mode) before it is expired.

Ancillary systems are expected to use the settlement period(“till”) option to avoid the extension of the arranged settlement timeframe. This option helps the ancillary system to control the execution time of their transactions but also helps the settlement banks to have a better control of their liquidity.

Please note that the start of the settlement period is always marked either with the end of information period (if it was indicated) or immediately after reception and positive validation of the ASTransferInitiation. The settlement period ("till") option only allows defining an end time or duration of the settlement period. In case no settlement period ("till") is used, the settlement period will end after final settlement or rejection of all transactions presented in the ASTransferInitiation message or, if one or several transactions are not executed due to missing liquidity, until the end of operational hours for ancillary system processing.

The usage of this option is a prerequisite for launching the optional guarantee fund mechanism.

Guarantee fund mechanism

The guarantee fund mechanism (if opted for by the ancillary system) could be used to provide the needed liquidity when a settlement failure occurs.

This optional mechanism can be used only:

- | in relation to ancillary system settlement procedures A and B
- | together with "Settlement Period("till")" time option

The guarantee fund mechanism is based on a guarantee account where the liquidity is collected to support the ancillary system settlement procedure - either continuously or arranged shortly before.

In order to use the guarantee fund mechanism, it has to be opted for by the ancillary system in its reference data. The usage of the guarantee fund mechanism is then valid whenever a settlement period (end time indicated with the Settlement Period ("till") option) ends unsuccessfully. In case no settlement period ("till") option was used, the underlying transaction processing will stop and rejection and reversal procedure (standard multilateral settlement) will be started.

Process description

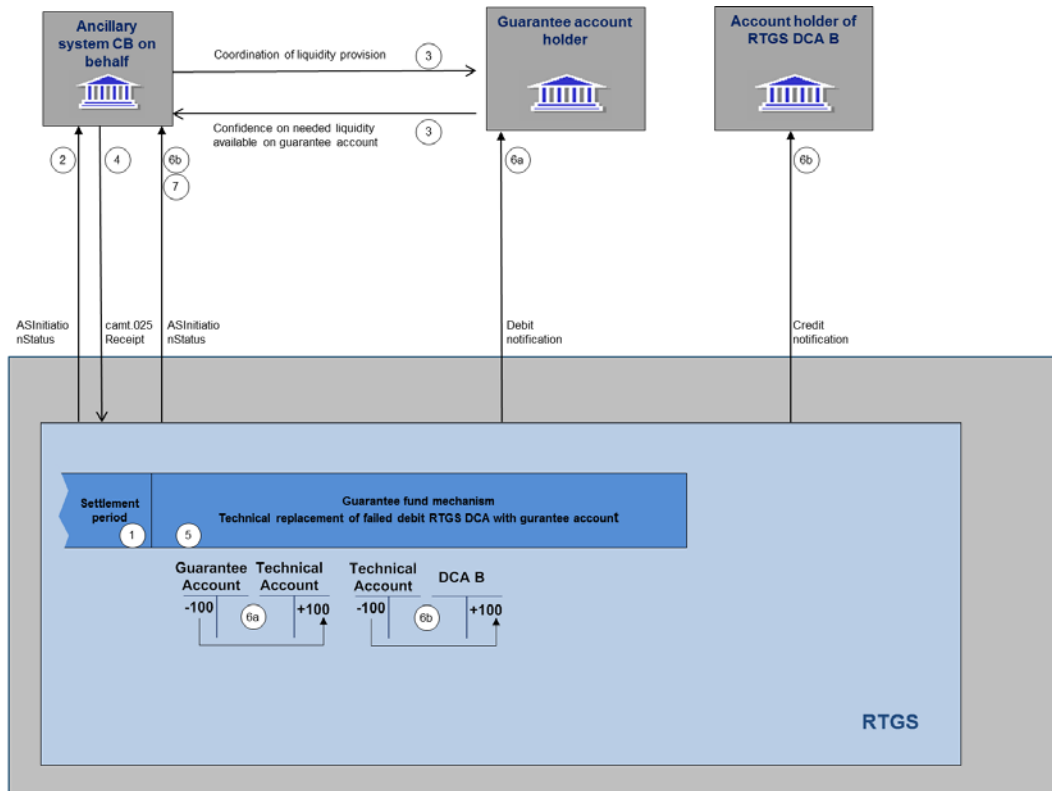


Figure 31 - Flow guarantee fund

Phase	Step	Description
Settlement failure	1	If the "settlement Period ("till") time is indicated and the settlement of either standard or simultaneous multilateral settlement is not yet achieved when the allotted time is exceeded the settlement fails. In case of simultaneous multilateral settlement, prior to the start of the guarantee fund mechanism a transformation of all transactions to standard multilateral settlement is performed and debit transactions covered by needed liquidity are executed. Please refer to Ancillary system settlement procedure B [148] for the flow of messages related to this scenario
Guarantee fund mechanism	2	If the guarantee fund mechanism has been set up (reference data), the ancillary system is notified on settlement failure with an ASInitiationStatus message containing the request to confirm the use of the guarantee fund mechanism by using the "decision indicator" flag within this message.
	3	Depending on the guarantee schema either the collection of the needed liquidity has been granted in advance by the ancillary system and its community (i.e. pre-funding) or the ancillary system has to co-ordinate the liquidity collection making it available on the specific "guarantee account". This can be done, depending on the specific set up of the guarantee account by using liquidity transfer orders (camt.050) or payments (pacs.009/pacs.010). The notifications to the guarantee account holder

Phase	Step	Description
		then depend on the messages used. In any case, before the guarantee fund mechanism starts the ancillary system has to assure the needed liquidity is provided on the guarantee account.
	4	The ancillary system sends an XML message (camt.025 Receipt) to give either a positive or a negative confirmation in order to proceed or not with a new settlement phase against the guarantee account.
New settlement phase	5	If the ancillary system confirms the actual use of the guarantee fund mechanism RTGS re-enters the transactions for which the liquidity was missing in order to be settled on the guarantee account by substituting the failed debtor's RTGS DCA with the guarantee account.
	6a	In case of sufficient liquidity the settlement of the debit from guarantee account to the ancillary system technical account will be executed. Depending on the message subscription also the guarantee account holder is notified with a camt.054 debit notification.
	6b	After successfully debiting the guarantee account, all credit bookings from the ancillary system technical account to the RTGS DCAs of the settlement banks will be executed. The ancillary system is notified about the completion of the whole settlement procedure. On an optional basis, the settlement banks of the creditor side are notified with a camt.054 credit notification.
	7	If the ancillary system sends a negative confirmation or there is a lack of liquidity on the guarantee account the "reversing procedure" is initiated in order to transfer back the already executed debits from the ancillary system technical account to the RTGS DCAs of the settlement banks. All involved settlement banks are notified with a GUI broadcast about failed settlement. The ancillary system will receive an ASIInitiationStatus informing on the failed settlement

Table 71 - Process description

Used messages

! [ASTransferInitiation \(pain.998\)](#) [▶ 633], [ASInitiationStatus \(pain.998\)](#) [▶ 620]

! [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522]

! [Receipt \(camt.025\)](#) [▶ 474]

! [ReceiptAcknowledgement \(admi.007\)](#) [▶ 391]

5.4 Liquidity management

5.4.1 Available liquidity

The RTGS DCA in the RTGS component is used for the settlement of real-time interbank and customer payments and payment instructions from ancillary systems. An RTGS DCA may either have a zero or a positive balance.

Generally, the positive balance on the RTGS DCA is available to settle payments and payment instructions from ancillary systems on the RTGS DCA of an RTGS Account Holder. The credit line – if available - is managed on the MCA in CLM.

Depending on the priority of a payment (see chapter [Payment priorities](#) [▶ 90]) and the liquidity management features used by the RTGS Account Holder, the actual liquidity available for settlement of a specific payment might be less than the balance on the RTGS DCA (see table below).

Effect	Urgent payment	High payment	Normal payment
Available liquidity	Balance on the RTGS DCA	Balance on the RTGS DCA ./ Urgent reservation	Balance on the RTGS DCA ./ Urgent reservation ./ High reservation

Table 72 - Effect of reservations on the available liquidity

In case the available liquidity on the RTGS DCA is not sufficient to settle a payment and depending on the configuration chosen by the RTGS Account Holder, inter-service liquidity transfers might be triggered. Further details can be found in the following chapters.

As CB accounts in the RTGS component can have a negative balance, the available liquidity for CBs is not limited.

5.4.2 Liquidity transfer

5.4.2.1 Overview

In general, liquidity transfers debiting a RTGS DCA are initiated by the RTGS Account Holder (either in A2A or U2A). In order to instruct the transfer of cash from one cash account to another cash account via A2A, the liquidity transfer order message ([LiquidityCreditTransfer \(camt.050\)](#) [▶ 497]) is used. Liquidity transfers are not classified as payments (i.e. pacs) but are cash transfers using camt messages. In order to transfer liquidity to the ancillary system dedicated liquidity account (real-time) in A2A, the settlement banks can also use the [ASTransferInitiation \(pain.998\)](#) [▶ 633] message.

Further details on the initiation of liquidity transfers via U2A are provided in the RTGS user handbook.

In general, liquidity transfer orders can be used to transfer liquidity

- | between two cash accounts within the RTGS component, i.e. RTGS DCA, sub-account for ancillary system, ancillary system technical account (intra-service)
- | from an RTGS DCA to a CLM MCA - or vice versa (inter-service)
- | from an RTGS DCA to a DCA of the TIPS or T2S Service - or vice versa (inter-service)

A liquidity transfer can be executed **within** the RTGS component only if

- | all involved RTGS DCAs belong to the same Liquidity Transfer Group or
- | a CB account is involved; or
- | it is a liquidity transfer between an RTGS DCA and the sub-accounts linked to this RTGS DCA; or
- | it is a liquidity transfer between an RTGS DCA and an ancillary system technical account.

In general liquidity transfers are never queued in the RTGS component, they are either

- | settled immediately (fully or partially) or
- | rejected.

Only under certain conditions an automatically generated liquidity transfer can be pending. This is only the case if a CLM MCA has insufficient liquidity for settling a CBO and there is not sufficient liquidity on the RTGS DCA to settle this automatic inter-service liquidity transfer. In such scenario any incoming liquidity (up to the required amount) on the RTGS DCA will be transferred stepwise to the MCA until the original amount of the automatic inter-service liquidity transfer due to pending CBO (i.e. the amount needed to settle the pending CBOs in CLM) is completely settled.

Note: Whenever such automatic inter-service liquidity transfer is pending, it is settled prior to any other payment and does not allow the settlement of any other payment.

Once a liquidity transfer is booked on the RTGS DCA, this booking is irrevocable and unconditional.

The following types of liquidity transfers can be initiated in the RTGS component.

Liquidity transfer type	Description
Immediate liquidity transfer order	Immediate transfer of a certain amount of liquidity initiated by the RTGS Account Holder or an authorised third party in U2A or A2A.
Rule-based liquidity transfer order	Liquidity transfers due to a floor/ceiling. Definition of amount by the system based on the target amount, the account balance and event by the RTGS Account Holder. Pending U-/H-payment configuration rule
Standing liquidity transfer order	Transfer of liquidity (a certain amount) regularly at certain events. Definition by the RTGS Account Holder.

Table 73 - Liquidity transfer types

For immediate liquidity transfer orders the process will be initiated by either the RTGS Account Holder itself or by another authorised actor of the RTGS component acting on behalf by sending the respective liquidity transfer to the RTGS component. For rule-based and standing liquidity transfer orders the process will be initiated by the RTGS itself whenever the respective event to trigger the liquidity transfer order is reached. The RTGS component will then process the liquidity transfer.

If the content of the immediate liquidity transfer order is either invalid or would result in checks to fail, it is rejected and a rejection notification is sent to the sender (depending on the channel a message in A2A mode or an error message on the screen in U2A mode). If the content of the liquidity transfer order is valid and certain checks have been passed, the RTGS component will try to transfer (part of) the liquidity amount requested to the relevant cash account referred to in the liquidity transfer order. Where the intra-RTGS liquidity transfer (partially) succeeds, RTGS will transfer (part of) the amount requested and RTGS will send a (partially) transfer success notification to the RTGS Account Holder/ancillary system involved (where the RTGS Account Holder opted for it).

In case of partial execution of a liquidity transfer order, the respective debit notification sent to the account owner of the debited RTGS DCA will contain the amount actually settled (which might differ from the instructed amount). In case there is no liquidity at all available in the non-reserved part of the MCA, the partial settlement takes place with the amount of zero. The RTGS Account Holder is informed accordingly via a [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522].

5.4.2.2 Initiation of liquidity transfers

Liquidity transfers in the RTGS component in A2A using a camt.050 message are initiated by either

- | the RTGS Account Holder itself
- | by another authorised actor (e.g. an ancillary system or another credit institution) or
- | by the RTGS component itself, based on information provided by CRDM.

Liquidity transfer orders can have the following type:

- | immediate liquidity transfer via A2A or U2A or
- | standing liquidity transfer order or
- | event-based liquidity transfer order.

As regards the execution of liquidity transfers which can be initiated in the RTGS component the following principles apply.

Liquidity transfer type	Initiator	Settlement
Immediate liquidity transfer	RTGS Account Holder	Only full settlement
	Ancillary system (on behalf)	Partial settlement possible; In case of partial settlement no further settlement attempt will be performed.
	CB (on behalf)	Only full settlement
Event-based liquidity transfer	Pre-configured by RTGS Account Holder	Partial settlement possible
Standing order liquidity transfer	Pre-configured by RTGS Account Holder	Partial settlement possible – in case several standing order liquidity transfers are triggered with the same event a pro rata execution applies. In case of partial settlement, no further settlement attempt will be performed.

Table 74 - Execution of liquidity transfers

5.4.2.3 Liquidity transfer process

5.4.2.3.1 Liquidity transfer between two DCAs of the RTGS component

Important preconditions

1. Both involved accounts exist and are active.
2. Respective access rights have been granted to the sender.
3. A Liquidity Transfer Group was set up by the responsible CB.

Message flow

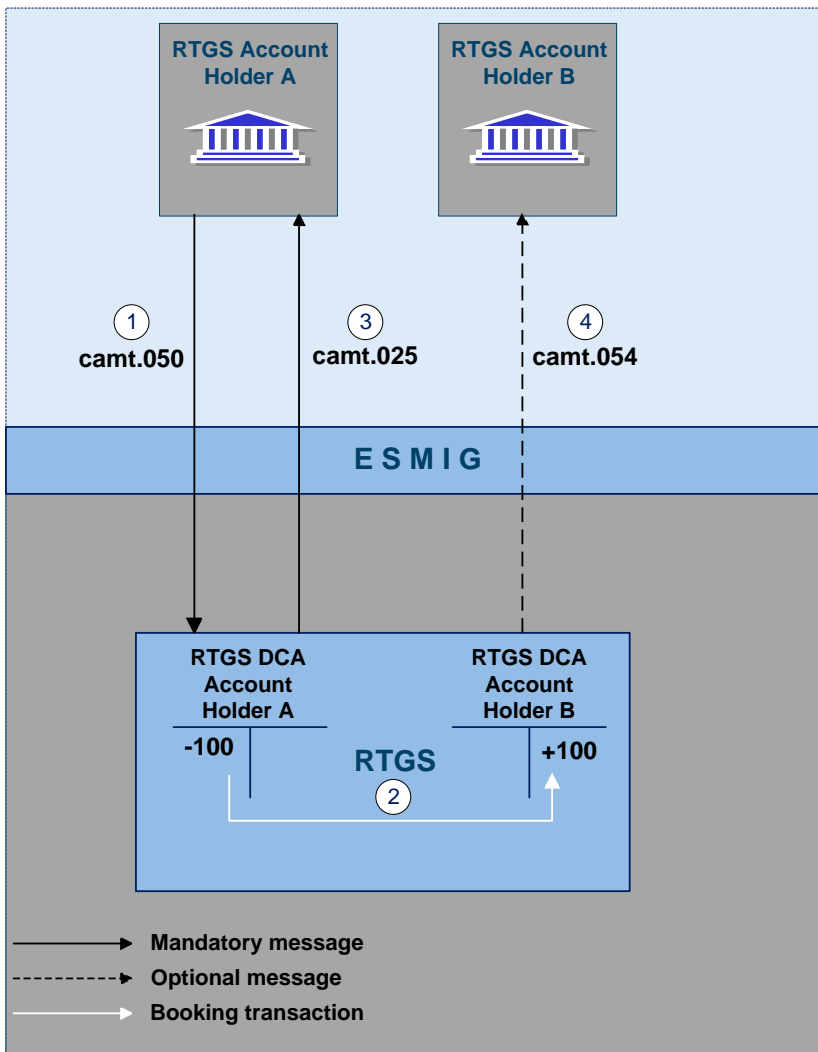


Figure 32 - Liquidity transfer order between two RTGS DCAs in the RTGS component

Process description

The liquidity transfer between two RTGS DCAs consists of the following process steps:

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	The RTGS Account Holder A sends a camt.050 via ESMIG to the RTGS component.
2	RTGS component	RTGS message check and validation in the RTGS component are positive. Simultaneous booking on the RTGS DCAs of RTGS Account Holders A and B
3	RTGS component via ESMIG to RTGS Account Holder A	Creation and forwarding of camt.025 (mandatory) to participant A
4	RTGS component via ESMIG to RTGS Account Holder B	Creation an forwarding of camt.054 (optional) to participant B

Table 75 - Process description

Used messages

- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 497]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522]
- | [Receipt \(camt.025\)](#) [▶ 474]

In addition to the classical intra-service liquidity transfer between two RTGS DCAs, the following business cases are also considered to be intra-RTGS liquidity transfers:

- | Liquidity transfer from an RTGS DCA to a linked sub-account dedicated to an ancillary system using the “interfaced” ancillary system procedure (and vice versa).
- | Liquidity transfer from an RTGS Participant’s RTGS DCA to the dedicated liquidity account related to an ancillary system using ancillary system procedure “real-time” (and vice-versa).
- | Liquidity transfer from one RTGS DCA to another RTGS DCA of the same party.

5.4.2.3.2 Liquidity transfer from DCA of the RTGS component to CLM MCA

Preconditions

1. Both RTGS DCA and MCA exist and are active.
2. Respective access rights have been granted to the sender.

Message flow

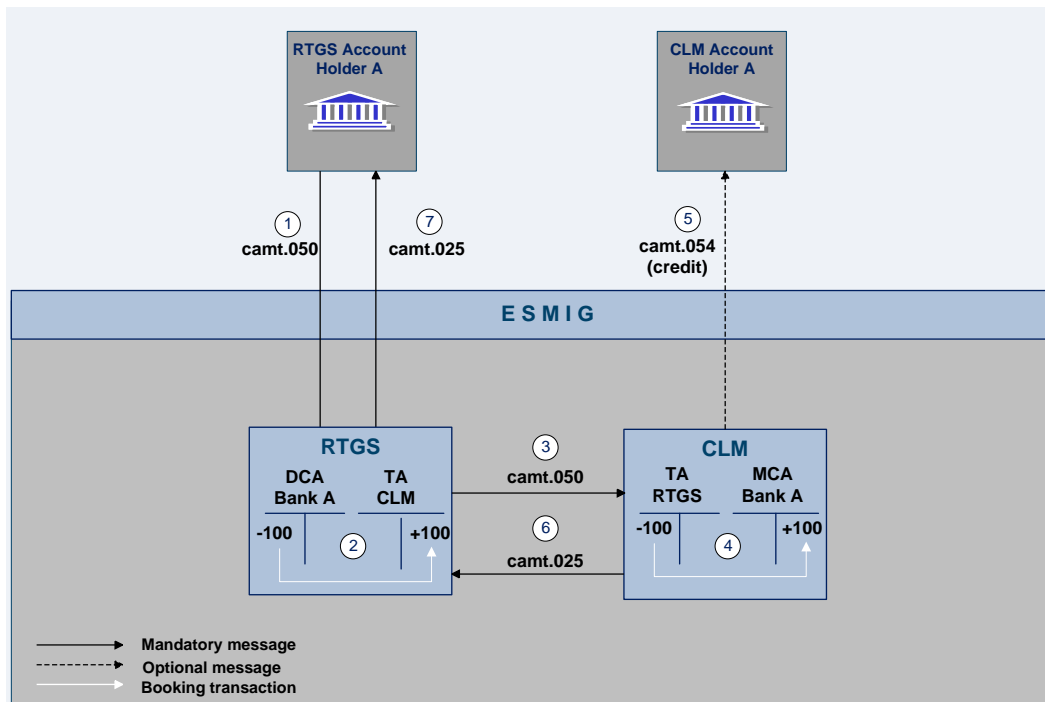


Figure 33 - Liquidity transfer from a RTGS DCA to a CLM MCA

Process description

The liquidity transfer order between an RTGS DCA and a MCA in CLM consists of the following process steps:

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	A camt.050 is sent from RTGS Account Holder A to the RTGS component
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCA of RTGS Account Holder A and the CLM transit account
3	RTGS component to CLM component	A camt.050 is forwarded to the CLM component
4	CLM component	Simultaneous booking on the RTGS transit account and the MCA of CLM Account Holder A (can be owned by a different party)

Step	Processing in/between	Description
5	CLM via ESMIG to the CLM Account Holder A	A camt.054 (credit) is sent by the CLM component via ESMIG to the CLM Account Holder A (optional)
6	CLM to RTGS component	A camt.025 is forwarded to the RTGS component
7	RTGS component via ESMIG to RTGS Account Holder A	Creation and forwarding of a camt.025 to RTGS Account Holder A (mandatory)

Table 76 - Process description

Used messages

- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 497]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522]
- | [Receipt \(camt.025\)](#) [▶ 474]

5.4.2.3.3 Liquidity transfer from DCA of the RTGS component to a DCA in different settlement services

Preconditions

1. Both RTGS DCA and DCA in other service exist.
2. Respective access rights have been granted to the sender.

Message flow

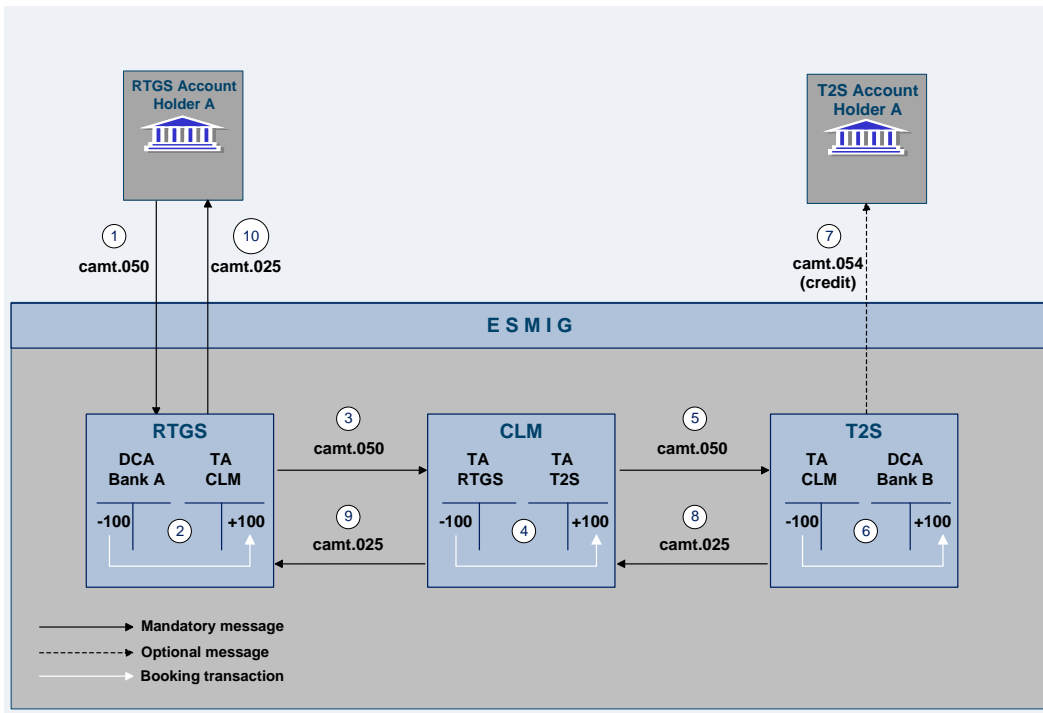


Figure 34 - Liquidity transfer from an RTGS DCA to a DCA in the T2S Service

Note: The detailed functionality of CLM and T2S is out of scope of this UDFS.

Process description

The liquidity transfer from the RTGS DCA to a DCA of a different service (T2S as example) consists of the following process steps:

Step	Processing in/between	Description
1	RTGS Account Holder A via ESMIG to the RTGS component	A camt.050 is sent from the RTGS Account Holder A to the RTGS component via ESMIG.
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS DCA of RTGS Account Holder A and the CLM transit account
3	RTGS component to CLM	A camt.050 is forwarded to the CLM component
4	CLM component	Simultaneous booking on the RTGS transit account and the T2S transit account
5	CLM component to T2S Service	A camt.050 is forwarded to the T2S Service
6	T2S Service	Simultaneous booking on the CLM transit account and the DCA of T2S Account Holder B

Step	Processing in/between	Description
7	T2S Service via ESMIG to the T2S Account Holder A	A camt.054 (credit) is sent by the T2S Service via ESMIG to the T2S Account Holder B (optional)
8	T2S Service to the CLM component	A camt.025 is forwarded to the CLM component
9	CLM to the RTGS component	A camt.025 is forwarded to the RTGS component
10	RTGS component via ESMIG to the RTGS Account Holder A	Creation and forwarding of a camt.025 to RTGS Account Holder A generated by the RTGS component (mandatory)

Table 77 - Process description

Used messages

- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 497]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522]
- | [Receipt \(camt.025\)](#) [▶ 474]

5.4.2.3.4 Liquidity transfer from DCA in different settlement service to a DCA of the RTGS component

Important preconditions

1. Both RTGS DCA and DCA in other service exist.
2. Respective privileges have been granted to the sender.

Message flow

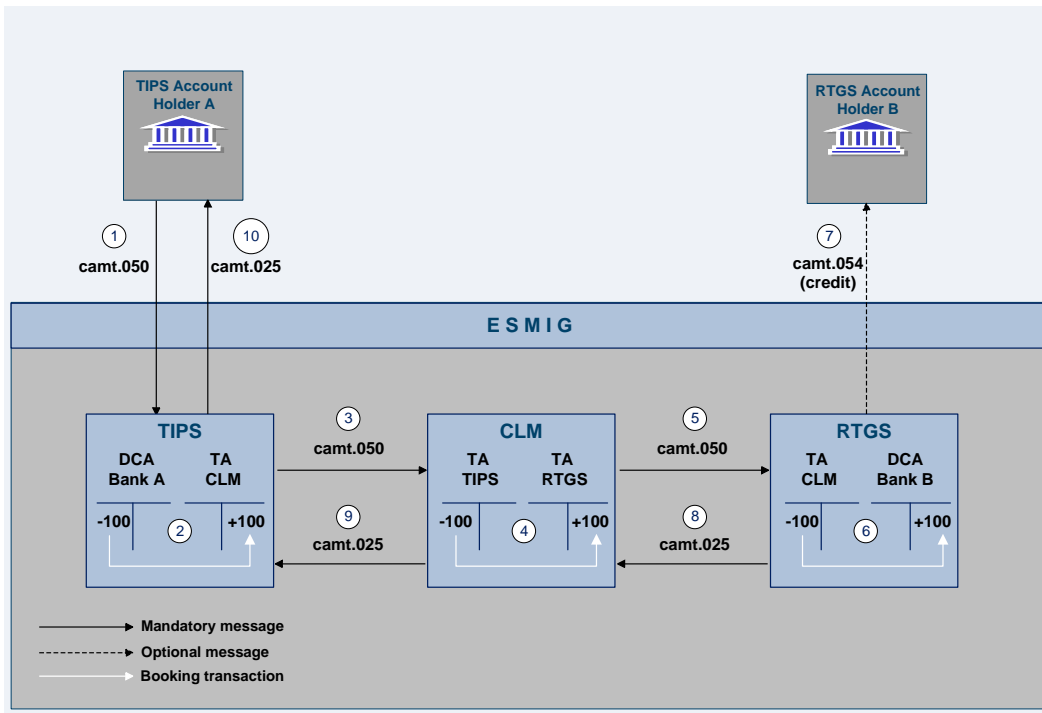


Figure 35 - Liquidity transfer from DCA of the TIPS Service to an RTGS DCA

Note: The detailed functionality of TIPS and CLM are out of scope of this UDFS.

Process description

The liquidity transfer from a different service (TIPS in this example) to the RTGS DCA consists of the following process steps:

Step	Processing in/between	Description
1	TIPS Account Holder via ESMIG to TIPS	A camt.050 is sent from the TIPS Account Holder A to TIPS via ESMIG
2	TIPS	Message check and validation in TIPS Service positive Simultaneous booking on the TIPS DCA of TIPS Account Holder A and the CLM transit account
3	TIPS to CLM component	A camt.050 is forwarded to CLM component
4	CLM component	Simultaneous booking on the TIPS transit account and the RTGS transit account
5	CLM component to RTGS component	A camt.050 is forwarded to the RTGS component
6	RTGS component	Simultaneous booking on the CLM transit account and the RTGS DCA of RTGS Account Holder B

Step	Processing in/between	Description
7	RTGS component via ESMIG to the RTGS Account Holder B	A camt.054 (credit) is sent by the RTGS component via ESMIG to the RTGS Account Holder B (optional)
8	RTGS component to CLM	A camt.025 is forwarded to the CLM component
9	CLM component to TIPS	A camt.025 is forwarded to TIPS
10	TIPS via ESMIG to the TIPS Account Holder A	Creation and forwarding of a camt.025 to TIPS Account Holder A generated by TIPS (mandatory)

Table 78 - Process description

Used messages

- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 497]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522]
- | [Receipt \(camt.025\)](#) [▶ 474]

5.4.2.4 Rejection of liquidity transfer orders

Liquidity transfer orders sent to the RTGS component have to pass several validations before the liquidity is booked on the accounts and effectively transferred. Validations performed include technical checks, format checks as well as business validations.

For different reasons, a liquidity transfer order can be rejected and a notification with the appropriate error code for rejection is returned to the sender. The validations are distinguished into technical validations and business validations.

In case the technical validation is not successful, a ReceiptAcknowledgement (admi.007) is sent to the sender indicating which error occurred. Rejections of camt.050 messages sent in A2A mode due to the business validations result in a receipt message (camt.025) being sent to the sending actor including the respective error code(s) according to chapter [Index of business rules and error codes](#) [▶ 670].

Note: The sending of a negative notifications is mandatory and not subject to message subscription.

In case of liquidity transfer orders initiated via U2A the failed validations are shown directly in the GUI.

5.4.2.4.1 Business validations

The validations described below will be performed in one step in order to capture all the possible breaches. Therefore, the checks must not stop after the first breach occurring as there could be further breaches in the subsequent checks. If the validation failed overall, a rejection notification with appropriate reason codes for

all breaches which occurred must be sent to the sender. The comprehensive list of business rules and error codes can be found in chapter [Index of business rules and error codes](#) [▶ 670].

The following business validations are inter alia performed in the RTGS component:

- | Liquidity Transfer Group check
- | duplicate check
- | process specific authorisation checks
- | settlement date check
- | field and reference data checks
- | account checks

5.4.3 Liquidity management features

5.4.3.1 Reservation

5.4.3.1.1 Overview

The RTGS component offers two different types of reservation:

- | urgent - with the usage of the urgent reservation facility, liquidity can be reserved for the execution of urgent payments.
- | high - with the usage of the high reservation facility, liquidity can be reserved for the execution of urgent and high payments.

The RTGS Account Holder decides which payment should have access to the reserved liquidity by determining the appropriate priority.

Reservation can be effected by RTGS Account Holders or other actors that have the appropriate access rights using U2A or A2A. Further details on the U2A functionality can be found in the RTGS user handbook.

In case of e.g. technical problems faced by an RTGS Account Holder, the responsible CB can act on behalf of this RTGS Account Holder.

RTGS Account Holders have the possibility to

- | create or to modify reservations with immediate effect during the current business day as a one-time reservation in the RTGS component. This includes
 - establishing a specific amount during the current day with immediate effect as a one-time reservation.
 - “resetting” to zero the liquidity reserved for the current business day only with immediate effect.

- modify the amount on demand during the day with immediate effect.
- | create, modify or delete a standing order reservation in CRDM valid from the following business day(s) (i.e. valid as of the next business day until next modification or the deletion of the standing order).

In case the available liquidity on the RTGS DCA is lower than the amount to be reserved, the part which can be reserved will be reserved and the remaining part of the reservation will be queued (i.e. the pending value) and the RTGS component will process it in an event-oriented manner. Consequently, in case of incoming credits, the RTGS component will decrease the pending value and increase the respective reservation accordingly.

The liquidity reservation (with immediate effect as well as standing order reservation) is possible throughout the whole business day with the exception of the EoD processing and the maintenance window.

Standing order reservation

Standing order reservations are created and managed in CRDM. The definition of standing order reservations is only possible for RTGS DCAs and not for sub-accounts.

The amount defined in the standing order for reservation will be valid at the SoD and can only be modified in CRDM. Modifications of standing orders during the business day will only be valid as of the following business day.

It is possible to have a standing order for the two types of reservations at the same time. Consequently, the RTGS Account Holder can have an urgent reserve and a high reserve in parallel. At the SoD, reservations are set according to the standing orders and up to the available balance on the RTGS DCA.

One-time reservation with immediate effect

One-time reservation are created and managed directly in the RTGS component. The definition of such reservations is only possible for RTGS DCAs and not for sub-accounts.

As outlined above it is possible to create a reservation for the current business day only. Moreover, it is possible to modify an existing reservation and to “reset to zero” the amount of the reservation with immediate effect for the current business day only. Owing to the asynchronous processing in the RTGS component, incoming liquidity might be blocked and used by a parallel booking process before the attempt to increase the reservation has been performed.

Upon receipt of

- | EoD notification,
- | a reservation revocation or
- | a new reservation order,

the RTGS component stops processing the original reservation order, i.e. the new reservation replaces the pending one or the EoD releases the reserved amount.

5.4.3.1.2 Liquidity reservation and management process

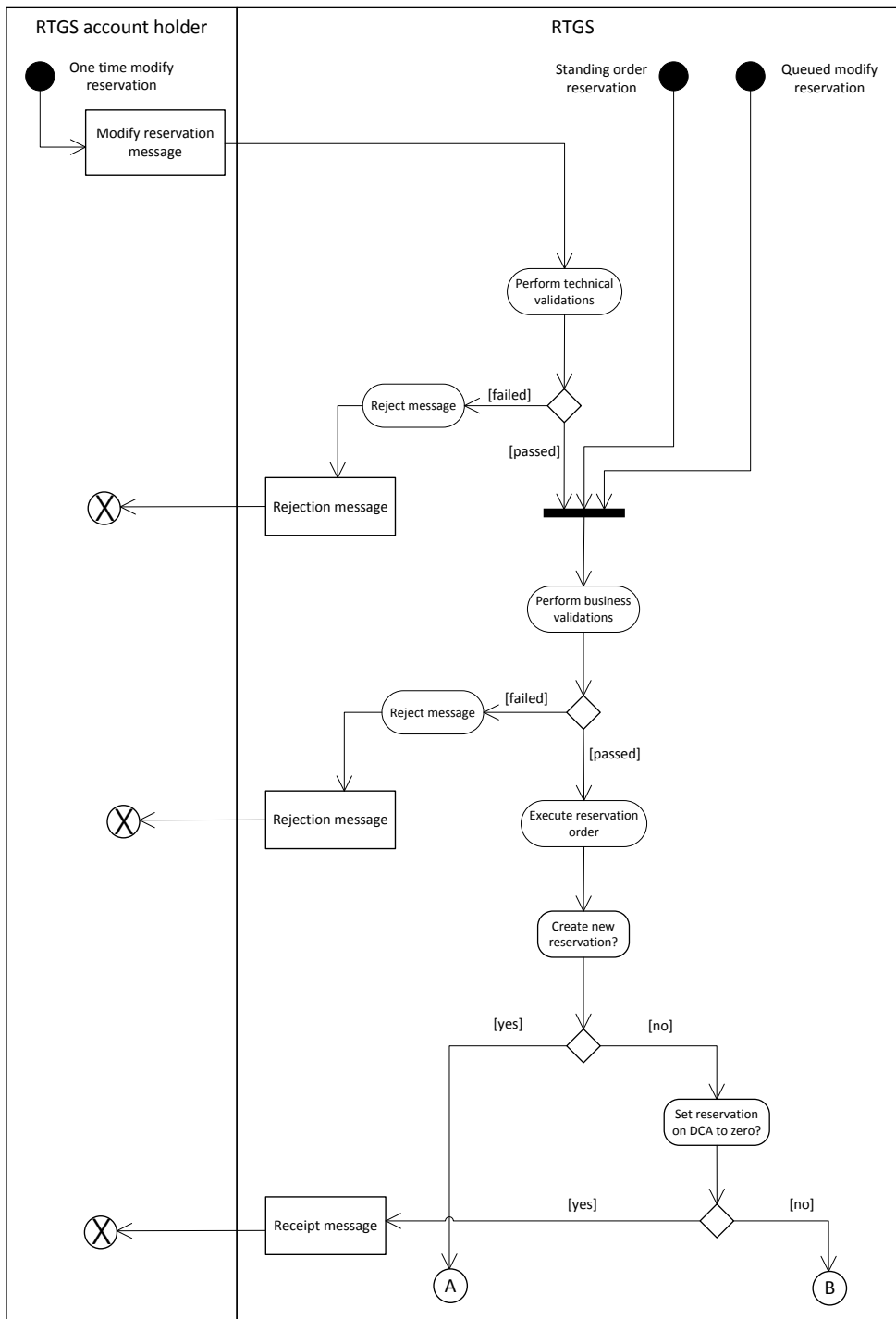
Reservation process – one time reservation with immediate effect

The following process flow illustrates the reservation creation ([ModifyReservation \(camt.048\)](#) [▶ 492]), the amendment ([ModifyReservation \(camt.048\)](#) [▶ 492]) and the “reset to zero” ([DeleteReservation \(camt.049\)](#) [▶ 494]) in the RTGS component.

In case an RTGS Account Holder wants to query a reservation, this can be done in A2A (see chapter [Query management for RTGS](#) [▶ 231]) as well as in U2A.

Note: The creation and the management of standing order reservations are done in CRDM.

Process flow



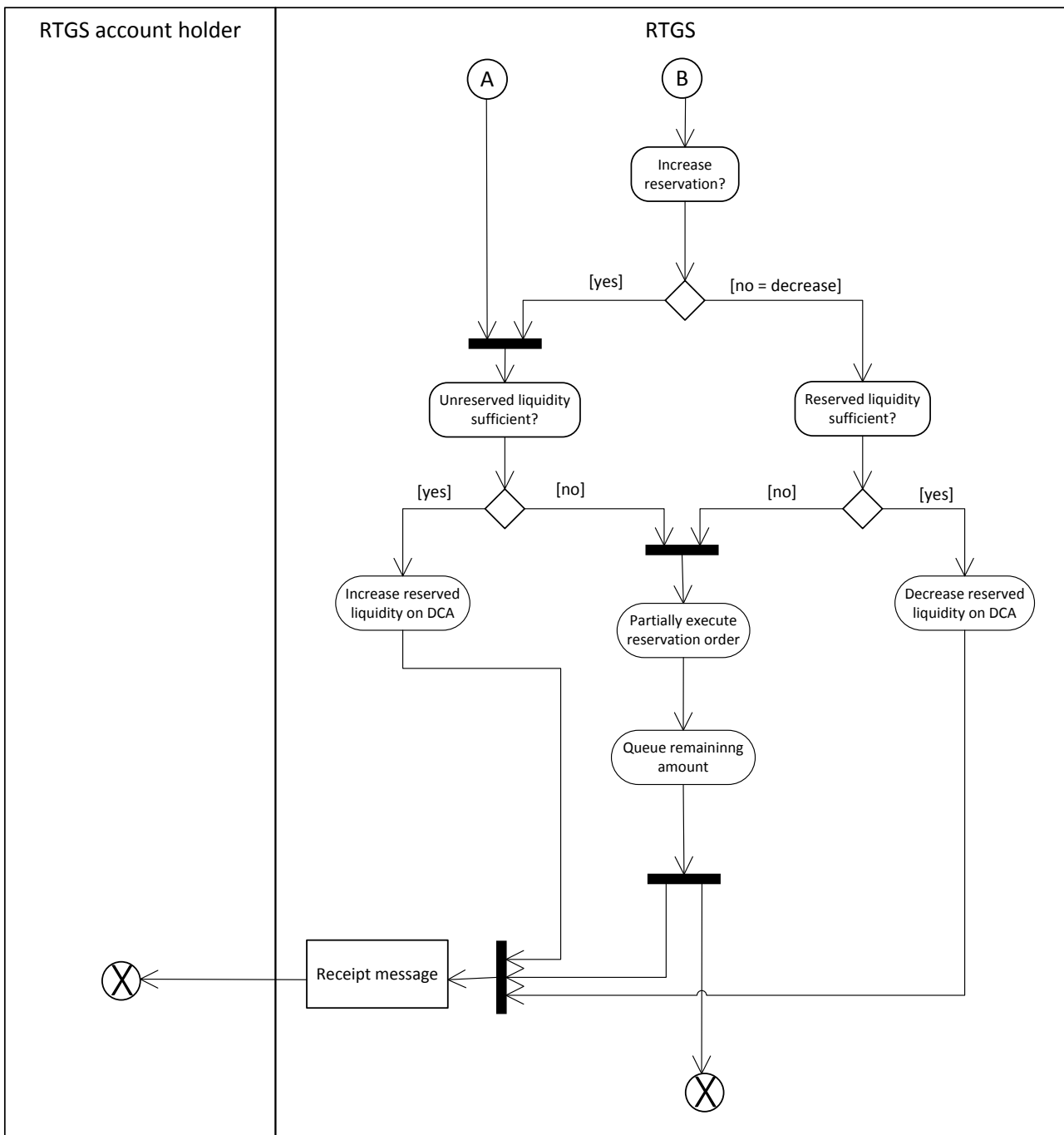


Figure 36 - Reservation management

Step	Processing in/between	Description
1	RTGS Account Holder via ESMIG to the RTGS component	RTGS Account Holder A sends a camt.048 via ESMIG to the RTGS component
2	RTGS component	RTGS message check and validation. In case of a negative technical business validation an admi.007 is sent. In case of negative business validation a camt.025 is sent. In case of a successful validation, execution of the one-time reser-

Step	Processing in/between	Description
		vation request.
3	RTGS component via ESMIG to RTGS Account Holder A	In case of (partial) execution of the reservation a camt.025 is created and sent via ESMIG to RTGS Account Holder A. Note: In case of an immediate reservation sent by an actor different from the account owner, also the sender of the camt.048 receives a camt.025.
4	RTGS component	The remaining reservation request (i.e. the pending value) will be queued and processed in an event-oriented way. In case of an increase of the available liquidity an asynchronous resolving process attempts to process the pending reservation order. New reservation requests related to the RTGS DCA replace already pending reservation requests. Note: Even if the increase of available liquidity is not sufficient for the complete processing, the pending reservation will be processed partly (the pending reservation is decreased and the existing reservation is increased).

Table 79 - Create one-time liquidity reservation with immediate effect

Used messages

- | [ModifyReservation \(camt.048\)](#) [▶ 492]
- | [Receipt \(camt.025\)](#) [▶ 474]
- | [ReceiptAcknowledgement \(admi.007\)](#) [▶ 391]

Process description

Step	Processing in/between	Description
1	RTGS Account Holder via ESMIG to RTGS component	RTGS Account Holder A sends a camt.048 via ESMIG to the RTGS component in order to modify the reservation with immediate effect
2	RTGS component	RTGS message check and validation. In case of a negative technical business validation an admi.007 is sent. In case of negative business validation a camt.025 is sent. In case of a successful validation, execution of modification request.
3	RTGS component via ESMIG to RTGS Account Holder A	In case of execution of the modification of the reservation a camt.025 is created and sent via ESMIG to RTGS Account Holder A. Note: In case of the increase of the reservation was sent by an entity different from the account owner, also the sender of the camt.048 receives a camt.025.
4	RTGS component	The remaining reservation request (i.e. increase) will be queued and processed in an event oriented way. In case of an increase of the available liquidity an asynchronous resolving process attempts to process the pending reservation order. New reservation requests related to the RTGS DCA replace pending reservation requests. Note: <ul style="list-style-type: none"> Even if the increase of available liquidity is not sufficient for the complete processing the pending reservation will be processed partly (the pending reservation is decreased and the existing reservation is increased). In case the RTGS Account Holder sends a deletion, this deletion is taken into account and replaces the pending reservation request.

Table 80 - Modify one-time liquidity reservations with immediate effect

Used messages

- | [ModifyReservation \(camt.048\)](#) [▶ 492]
- | [Receipt \(camt.025\)](#) [▶ 474]
- | [ReceiptAcknowledgement \(admi.007\)](#) [▶ 391]

“Resetting to zero”

Note: Owing to the fact that the messages used are the same for one-time reservation with immediate effect and standing order reservation, in principle the message flow applies for both cases.

Process description

Step	Processing in/between	Description
1	RTGS Account Holder via ESMIG to RTGS component	RTGS Account Holder A sends a camt.049 via ESMIG to the RTGS component in order to reset the reservation to zero.
2	RTGS component	<p>RTGS message check and validation. In case of a negative technical validation, an error message (admi.007) is sent. In case of negative business validation a camt.025 is sent.</p> <p>In case of a successful validation the reservation will be set to zero for the current business day and a camt.025 is sent to the RTGS Account Holder A.</p> <p>Note: In case the resetting to zero was sent by an actor different from the account holder, also the sender receives a camt.025.</p>

Table 81 - “Resetting to zero” of a reservation

Used messages

- | [DeleteReservation \(camt.049\)](#) [▶ 494]
- | [Receipt \(camt.025\)](#) [▶ 474]
- | [ReceiptAcknowledgement \(admi.007\)](#) [▶ 391]

5.4.3.1.3 Effect and tapping of liquidity reservation

The following tables explain the effect of the reservation functionality for the processing of payments in the RTGS component and provide a numeric example:

Effect	Urgent payment	High payment	Normal payment
Available liquidity for settlement of payments	Balance on RTGS DCA	Balance on RTGS DCA minus urgent reserve	Balance on RTGS DCA minus urgent reserve minus high reserve
Effect of outgoing payments	<ul style="list-style-type: none"> Reduction of balance on RTGS DCA Reduction of urgent reserve If the urgent reserve is not sufficient, the liquidity will be used as follows: <ul style="list-style-type: none"> – Available liquidity for normal payments. – Reduction of the high reserve. 	<ul style="list-style-type: none"> Reduction of balance on RTGS DCA Reduction of high reserve If the high reserve is not sufficient, the available liquidity for normal payments will be used. 	<ul style="list-style-type: none"> Reduction of balance on RTGS DCA
Effect of incoming payments	Increase of balance on RTGS DCA	Increase of balance on RTGS DCA	Increase of balance on RTGS DCA

Table 82 - Effect of reservations for payment procession

Note: Direct debits effect the reserved liquidity and the other way round.

Activity	Balance on RTGS DCA	Urgent reserve	High reserve	Available liquidity for normal payments
Start	1,000	100	200	700
Settlement of ancillary system = 50 (debit)	950 ↓	50 ↓	200 ↔	700 ↔
Submitting high payment to bank B = 200	750 ↓	50 ↔	0 ↓	700 ↔
Submitting normal payment to bank C = 20	730 ↓	50 ↔	0 ↔	680 ↓
Settlement of ancillary system = 100 (credit)	830 ↑	50 ↔	0 ↔	780 ↑
Incoming high payment from bank B = 50	880	50	0	830

Activity	Balance on RTGS DCA	Urgent reserve	High reserve	Available liquidity for normal payments
	↑	↔	↔	↑
Incoming normal payment from bank C = 30	910 ↑	50 ↔	0 ↔	860 ↑
Set a new high reservation with immediate effect = 500	910 ↔	50 ↔	500 ↑	360 ↓
Settlement of urgent payment in favour of CB = 450 (debit)	460 ↓	0 ↓	460 ↓	0 ↓

Table 83 - Usage of urgent and high reserve – numeric example

5.4.3.2 Limits

5.4.3.2.1 Overview

In general, limits determine the amount of liquidity an RTGS Account Holder is willing to accept as liquidity outflow for settling credit transfers with priority normal which are to be debited on his RTGS DCA.

The following types of limits can be used in the RTGS component:

- | bilateral limit
- | multilateral limit

The limits are debit limits and not credit limits, i.e. they define the amount an RTGS Account Holder is willing to pay

- | to another RTGS DCA in case of a bilateral limit or
- | to all other RTGS DCAs towards which no bilateral limit has been defined

without receiving any incoming payments (i.e. incoming credit transfers) first.

Limits can be defined and managed by RTGS Account Holders or other actors that have the appropriate access rights using U2A or A2A. Further details on the U2A functionality can be found in the RTGS user handbook. Limits are set up at account level, i.e. a bilateral/multilateral limit applies for payments processed on one specific RTGS DCA only.

At the SoD, limits are set according to the standing orders (so called defined limit) and are updated throughout the business day after each relevant credit and debit (so called free limit position). As a consequence, a normal payment will only be settled if it does not cause a breach of the free limit position. In case no limit is defined, the RTGS DCA's liquidity available for the respective priority is available for each payment.

In general, RTGS Account Holders have the possibility to

- | modify limits with immediate effect during the day trade settlement phase in the RTGS component. The modification of limits with immediate effect includes the increase, the decrease and the reduction to zero. If a limit is set to zero, it is not possible to increase it again on the same business day.
- | create, modify or delete a defined limit in CRDM valid from the following business day(s) (i.e. valid as of the next business day until next change).

The limitation process consists of the following elements:

- | definition of bilateral limits towards selected RTGS DCAs.
- | definition of a multilateral limit towards all RTGS DCAs towards which no bilateral limit is defined

Objectives for the use of limits

The setting of the limits enables the RTGS Account Holder

- | to ensure an early submission of normal payments with full control of the liquidity outflow at the same time
- | to avoid free-riding on the liquidity of one RTGS DCA by another RTGS DCA Holder's cash account
- | to synchronise the payment flow with other RTGS Account Holders and to promote its early submission

5.4.3.2.1.1 Bilateral limits

Bilateral position

The bilateral position from RTGS Account Holder A towards RTGS Account Holder B is defined as the sum of payments received from RTGS Account Holder B (i.e. credits for RTGS Account Holder A) minus the sum of payments made to RTGS Account Holder B (debits for RTGS Account Holder A). This means if the result is negative, the bilateral limit will be utilised with this amount.

Effect of bilateral limit

With the bilateral limit, the RTGS Account Holder restricts the use of liquidity when submitting payments for another RTGS Account Holder. Direct debits effect the bilateral position just the other way round as in case of direct debits outgoing payments are credits and incoming payments are debits.

Once a defined bilateral limit has been created in CRDM and is taken into account during the SoD for the current business day, the defined limit can be changed directly in RTGS with immediate effect throughout the business day.

5.4.3.2.1.2 Multilateral limits

Multilateral position

The multilateral position from RTGS DCA A is defined as the sum of payments (credits for RTGS DCA A) received from all RTGS DCAs towards which no bilateral limit has been defined, minus the sum of payments (debits for RTGS DCA A) made to these RTGS DCAs. This means if the result is negative, the multilateral limit is utilised with this amount.

Effect of multilateral limit

With the multilateral limit, the RTGS Account Holder restricts the use of liquidity, when submitting payments for any other RTGS account holder for which a bilateral limit has not been set.

Direct debits effect the multilateral position just the other way round because outgoing payments are credits and incoming payments are debits.

5.4.3.2.1.3 Rules for definition of limits

The creation of standing order limits is done in CRDM and the definition is done per RTGS DCA.

Changes and “resetting to zero” of bilateral and multilateral limits with immediate effect for the current business day are done in the RTGS component directly.

The following general rules apply:

- | The minimum amount of a limit is one million.
- | It is not possible to define a bilateral limit vis-à-vis CBs. For CB accounts it is not possible to define limits.
- | A bilateral or multilateral limit with an amount of zero is a limit which is considered as “not defined”.
- | A multilateral limit can be defined if at least one bilateral limit exists.
- | Any credits (related to payments with normal, high or urgent priority) from an RTGS DCA towards which a bilateral/multilateral limit is defined, increase the free limit position.

In order to take into account a defined limit (bilateral or multilateral) for the settlement of payments, the defined limit needs to be defined before the end of the previous business day. This means that a standing order above one million has to be defined at the latest before the end of the previous business day. One million is the minimum amount for a limit.

Once a defined multilateral limit has been created in CRDM and is taken into account during the SoD for the current business day, the defined limit can be changed directly in RTGS with immediate effect throughout the business day.

5.4.3.2.2 Process for the definition and management of limits

The creation, the amendment and the deletion of a standing order limit is managed in CRDM.

The following message flow illustrates the amendment (camt.011) or “resetting to zero” (camt.012) with immediate effect for the current business day in the RTGS component.

Case: limit amendment/deletion message with positive validation

Process description

Step	Processing in/between	Description
1	RTGS Account Holder via ESMIG to the RTGS component	RTGS Account Holder A sends a camt.011 / camt.012 via ESMIG to the RTGS component
2	RTGS component	RTGS message check and validation positive Execution of limit request (amendment or deletion)
3	RTGS via ESMIG to RTGS Account Holder A	Creation and forwarding of camt.025 by the RTGS component via ESMIG to RTGS Account Holder A

Table 84 - Limit management – positive validation

Used messages

- | [ModifyLimit \(camt.011\)](#) [▶ 445]
- | [DeleteLimit \(camt.012\)](#) [▶ 448]
- | [Receipt \(camt.025\)](#) [▶ 474]

Case: limit amendment/deletion message with negative validation

Process description

Step	Processing in/between	Description
1	RTGS Account Holder via ESMIG to the RTGS component	RTGS Account Holder A sends a camt.011 / camt.012 via ESMIG to the RTGS component
2	RTGS component	RTGS message check and validation negative
3	RTGS component via ESMIG to RTGS Account Holder A	Creation and forwarding of camt.025 by the RTGS component via ESMIG to RTGS Account Holder A (mandatory)

Table 85 - Limit management – negative validation

Used messages

- | [ModifyLimit \(camt.011\)](#) [▶ 445]
- | [DeleteLimit \(camt.012\)](#) [▶ 448]
- | [Receipt \(camt.025\)](#) [▶ 474]

Initiator of limit setting and changing

Limits are exclusively set by RTGS Account Holders. Only in the case of a technical problem on the RTGS DCA Holder’s side, the responsible CB can be authorised to adjust the amount of a limit with immediate effect for the next algorithm.

5.4.3.2.3 Effect of limits

General effect of limits

The following table explains the effects of limits on the processing and subsequent settlement of payments.

Normal payment	
Available liquidity for settlement of normal payments	Balance on RTGS DCA minus urgent reserve minus high reserve
Effect of outgoing payments (i.e. debits on the RTGS DCA ⁶)	<ul style="list-style-type: none"> Reduction of balance on RTGS DCA Reduction of bilateral or multilateral position (payments will be queued, if the amount of the normal payment is higher than the Free Limit Position)
Effect of incoming payments (i.e. credits on the RTGS DCA ⁷)	<ul style="list-style-type: none"> Increase of balance on RTGS DCA Increase of the Free Limit Position

Table 86 - Effects of limits

Bilateral limit

The processing of normal payments in case RTGS Account Holder A has set a bilateral limit for RTGS Account Holder B is illustrated in the following simplified example

6 Direct debits effect the bilateral/multilateral position just the other way round because outgoing payments are credits and incoming payments are debits. Debits related to payments with high or urgent priority do not have any effect on the free limit position.

7 Direct debits effect the bilateral/multilateral position just the other way round because outgoing payments are credits and incoming payments are debits. Debits related to payments with high or urgent priority do not have any effect on the free limit position.

Bilateral relation	Bilateral limit set	Submitted normal payments	Explanation
RTGS DCA A vis-à-vis RTGS DCA B	3 million EUR	10 million EUR	Up to a maximum of 3 million EUR of RTGS DCA A's liquidity will be used to settle normal payments between RTGS DCA A and RTGS DCA B.
RTGS DCA B vis-à-vis RTGS DCA A	Not relevant in this example	6 million EUR	<p>If RTGS DCA A has sufficient liquidity available, a maximum of 9 million EUR from RTGS DCA A and 6 million EUR from RTGS DCA B can be settled.</p> <p>1 remaining million EUR from bank A cannot be settled and are queued until</p> <ul style="list-style-type: none"> additional payments (high/normal) from RTGS DCA B will be settled or RTGS Account Holder A increases the bilateral limit to an amount of 4 million EUR or sets the bilateral limit to zero. <p>Otherwise the normal payments will not be settled and will be rejected by the end of the day.</p>

Table 87 - Processing in case of bilateral limit

Multilateral limit

The processing of normal payments in the case of bank A has set a multilateral limit is illustrated in a following simplified example (bank A has not defined bilateral limits vis-à-vis those banks).

Multilateral relation	Multilateral limit set	Submitted normal payments	Explanation
RTGS DCA A vis-à-vis RTGS DCAs C, D, E, ...	2 million EUR	20 million EUR	Up to a maximum of 2 million EUR of RTGS DCA A's liquidity will be used to settle payments between RTGS DCA A and RTGS DCAs C, D, E, ...
RTGS DCAs C, D, E, ... vis-à-vis RTGS DCA A	Not relevant in this example	15 million EUR	<p>If RTGS DCA A has sufficient liquidity available, a maximum of 17 million EUR from RTGS DCA A and 15 million EUR from RTGS DCAs C, D, E, ... can be settled.</p> <p>3 remaining million EUR from RTGS DCA A cannot be settled and are queued until</p> <ul style="list-style-type: none"> additional payments (high/normal) of RTGS DCAs C, D, E, ... will be settled or RTGS Account Holder A increases the multilateral limit to an amount of 5 million EUR or sets the limits to zero. <p>Otherwise the normal payments will not be settled and rejected by the end of the day.</p>

Table 88 - Processing in case of multilateral limits

5.4.3.3 Dedication of liquidity for ancillary system settlement

For the settlement of ancillary systems the RTGS Account Holder can “set aside” liquidity for this purpose only.

Depending on the settlement procedure the ancillary system is using, the liquidity needs to be provided on different accounts:

- | sub-account for the procedure “settlement on dedicated liquidity accounts (interfaced)”(account owner = RTGS Account Holder)
- | dedicated liquidity account for procedure “settlement on dedicated liquidity accounts (real-time)”(account owner = ancillary system or its CB)

Moreover, the RTGS Account Holder can open a dedicated RTGS DCA (account owner = RTGS Account Holder) which is used for ancillary system settlement only.

To transfer liquidity to the RTGS DCA Holder’s sub-account or to the dedicated liquidity account, the following possibilities can be used:

- | Setting up standing orders for liquidity transfers in CRDM. These will become effective with the next business day.

- | Immediate liquidity transfer orders using camt.050 LiquidityCreditTransfer messages or via dedicated RTGS GUI liquidity transfer screens
- | Immediate liquidity transfer orders initiated by the ancillary system using ASTransferInitiation messages debiting the settlement banks RTGS DCA and crediting the settlement bank's sub-account (procedure interface) or the dedicated liquidity account (procedure real-time)

Standing orders are executed with each start of a procedure (mandatory and optional). Different amounts for both procedures can be specified. Further details can be found in chapter [Settlement of ancillary systems](#) [▶ 140]. Immediate liquidity transfer orders will be executed with immediate effect during an open procedure with no cycle running. In the opposite case, where a cycle is running, the liquidity transfer will be stored and executed only once the cycle has closed.

In case the available liquidity on the RTGS DCA is not sufficient, the following shall apply:

- | if the total sum of all standing orders of a settlement bank is larger than the liquidity on its RTGS DCA, all standing orders will be reduced in a pro-rata mode, i.e. the existing liquidity is divided by the total sum of standing orders and the resulting factor will be used to reduce each standing order of this account holder (mandatory procedure). In optional procedure the standing order will be rejected
- | a current order initiated by the settlement bank will be rejected (mandatory and optional procedure)
- | a current order initiated by the ancillary system (or CB on behalf) will be partially settled up to the available liquidity on the RTGS DCA (mandatory and optional procedure)

5.4.3.4 Floor/ceiling

5.4.3.4.1 Definition of floor/ceiling threshold

The RTGS component can generate a floor/ceiling notification related to an RTGS DCA in case a floor/ceiling threshold has been defined in advance. In case such threshold has been defined, the sending of a floor/ceiling notification will be triggered by the RTGS component after the successful settlement of a payment or ancillary system related payment instruction whenever the amount on the RTGS DCA undercuts the floor amount or exceeds the ceiling amount.

Since this functionality is optional, it is up to the holder of the RTGS DCA (i.e. the RTGS Account Holder) to define a floor/ceiling threshold in CRDM.

The holder of the RTGS DCA can define a minimum ("floor") or maximum ("ceiling") amount for its RTGS DCA(s). The RTGS Account Holder has the option to choose what shall be done by the RTGS component once the balance is below the defined floor or above the defined ceiling amount.

Two options are available:

1. The RTGS component generates a notification to be sent to the RTGS Account Holder as the owner of the RTGS DCA informing about the floor/ceiling breach (upon which the RTGS Account Holder can take action); or
2. The RTGS component automatically generates an inter-service liquidity transfer to pull liquidity from the linked MCA in case the floor is breached on the RTGS DCA or the RTGS component pushes liquidity to the linked MCA in case the ceiling threshold was reached. When using this functionality, the RTGS Account Holder needs to define also a target floor amount and a target ceiling amount for its RTGS DCA.

The floor / ceiling functionality itself will only be triggered after the settlement of a payment or a payment instruction stemming from the settlement of ancillary systems. It is not triggered for liquidity transfers.

5.4.3.4.2 Breach of floor/ceiling threshold - notification

If the RTGS Account Holder chooses the first option, the RTGS component generates and sends out a notification with the information that the account balance is below the floor or that the account balance is above the ceiling respectively

- I in U2A (please refer to the respective part of the RTGS user handbook) or
- I in A2A mode ([ReturnAccount \(camt.004\)](#) [▶ 397], [Process floor and ceiling](#) [▶ 309])

The notification will be sent every time, the threshold is undercut (floor) or exceeded (ceiling). However, the RTGS component does not send the notification if, after trespassing the threshold, the balance of the RTGS DCA remains consistently below the floor or above the ceiling threshold defined.

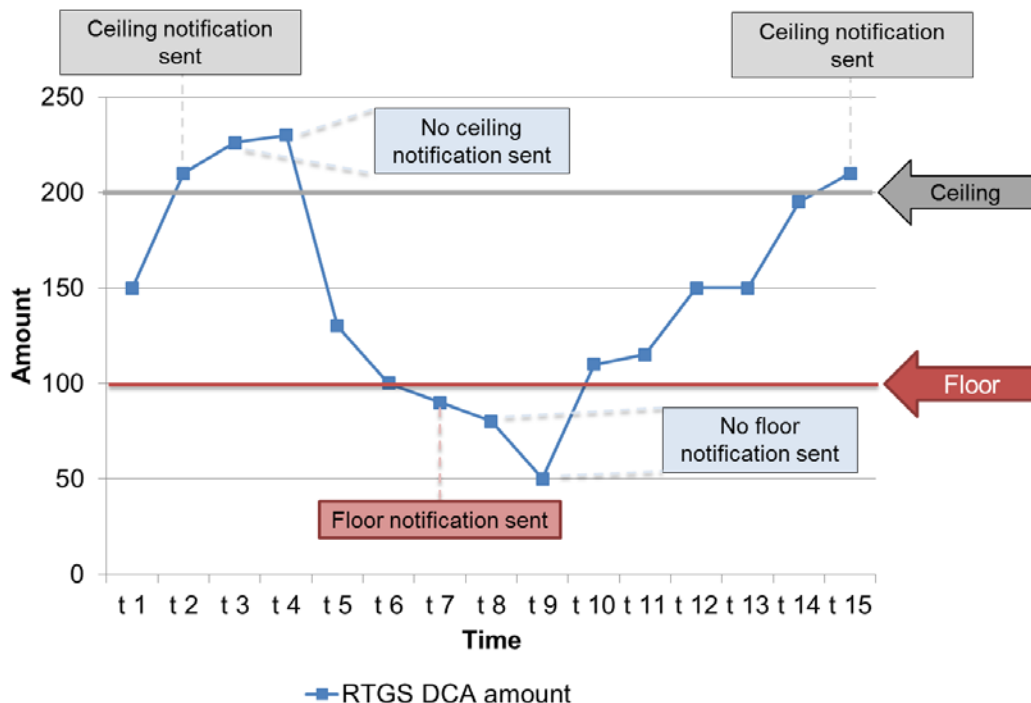


Figure 37 - Breach of floor/ceiling threshold - notification

5.4.3.4.3 Breach of floor/ceiling threshold - automatic liquidity transfer

If the RTGS Account Holder chooses the second option, the RTGS component creates and releases an inter-service liquidity transfer.

- | In case of a breach of the floor threshold the needed amount will be pulled from the MCA and credited on the RTGS DCA.
- | The used MCA will be the one linked to the RTGS DCA as defined in CRDM.
- | The amount to be transferred is the difference between the current balance on the RTGS DCA and the predefined target amount. The target floor amount could be different, but will in any case be equal or above the floor amount.
- | In case of a breach of the ceiling threshold the amount will be pushed to the MCA in CLM where it will be credited and the RTGS DCA will be debited.
- | The used RTGS DCA will be the same as for the floor threshold meaning it will be the one linked to the MCA as defined in CRDM.

- l The amount to be transferred to the MCA is the difference between the current balance and the predefined target ceiling amount. The target ceiling amount could be different but will be below the ceiling amount.
- l The target amount for ceiling will be a different one as the target amount of the floor threshold.

After the successful execution of the inter-service liquidity transfer, the amount on the RTGS DCA will be again within the boundaries of the floor or ceiling amount.

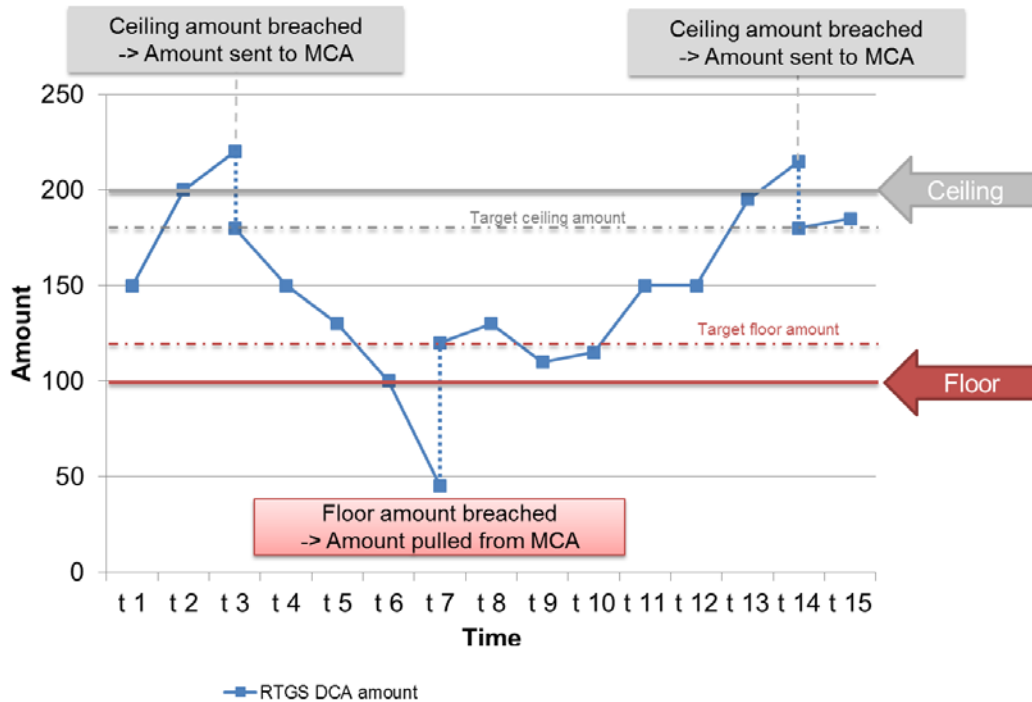


Figure 38 - Breach of floor/ceiling threshold – automated liquidity transfer

5.5 Information management for RTGS

5.5.1 RTGS status management

5.5.1.1 Concept

RTGS informs its RTGS Actors of the processing results. This information is provided to the RTGS Actors via a status reporting which is managed by the status management. The communication of status to RTGS Actors is complemented by the communication of reason codes in case of negative result of an RTGS process (e.g. validation failure notifications).

5.5.1.2 Overview

The status management process manages the status updates of the different instructions (e.g. payment, liquidity transfers, amendment instructions) existing in RTGS in order to communicate relevant status updates via status advice messages to the RTGS Actors throughout the lifecycle of the instruction. Some status notifications are mandatory, others are provided on optional basis. Status information on push basis is only available in A2A mode. Respective status advice messages are pushed via store-n-forward network service. For exceptional business cases notifications in U2A are foreseen.

The status management handling also provides the reason codes to be sent to RTGS Actors in case of negative result of an RTGS component process (e.g. to determine the reason why an instruction is unsuccessfully validated or settled).

The status of an instruction is indicated through a value, which is subject to change through the lifecycle of the instruction. This value provides RTGS Actors with information about the situation of the instruction with respect to a given RTGS process at a certain point in time.

Since each instruction in the RTGS can be submitted to several processes, each instruction in RTGS has several status. However, each of these status has one single value at a certain moment in time that indicates the instruction's situation at the considered moment. Depending on its instruction type, an instruction is submitted to different processes in RTGS. Consequently, the status featuring each instruction depends on the considered instruction type.

The following sections provide:

- l the generic principles for the communication of status and reason codes to RTGS Actors
- l the list of status featuring each instruction type as well as the possible values for each of these status

Reason codes are provided within the respective message documentation on MyStandards.

5.5.1.3 Status management process

Communication of status and reason codes to RTGS actors

RTGS Actors can query during the day the status values and reason codes of their instructions (e.g. payments, liquidity transfers, tasks, reference data updates).

The status can be classified into two different types, common to all types of instructions:

- l "Intermediate status". In general an instruction will have more than one status in its lifetime. If the status of an instruction is not a final status type, then the instruction is still being processed in RTGS. With each step in the process of the instruction the status will change until a final status is reached. Further status updates are communicated to the RTGS Actor if reached.

- “Final status”. This is the last status of an instruction (i.e. the status that an instruction has when processing for that instruction ends). At a point in time, any instruction in RTGS reaches a final status, all respective processes are completed.

For some specific status updates, the status management process informs the RTGS Actor of the status change by means of the sending of status advice messages (according to their message subscription configuration – please refer to [Messaging](#) [▶ 62]).

Status and status values in RTGS

As previously mentioned, the status of an instruction depend on the considered instruction type. The following paragraphs provide the list of status and status values. None of the status are stored for queries.

RTGS component status are:

- RTGS file status
- RTGS message status
- Ancillary system batch message status
- Cash transfer status
- Task queue status.

RTGS file status

Indicates the status of the file in RTGS and it can have the following status:

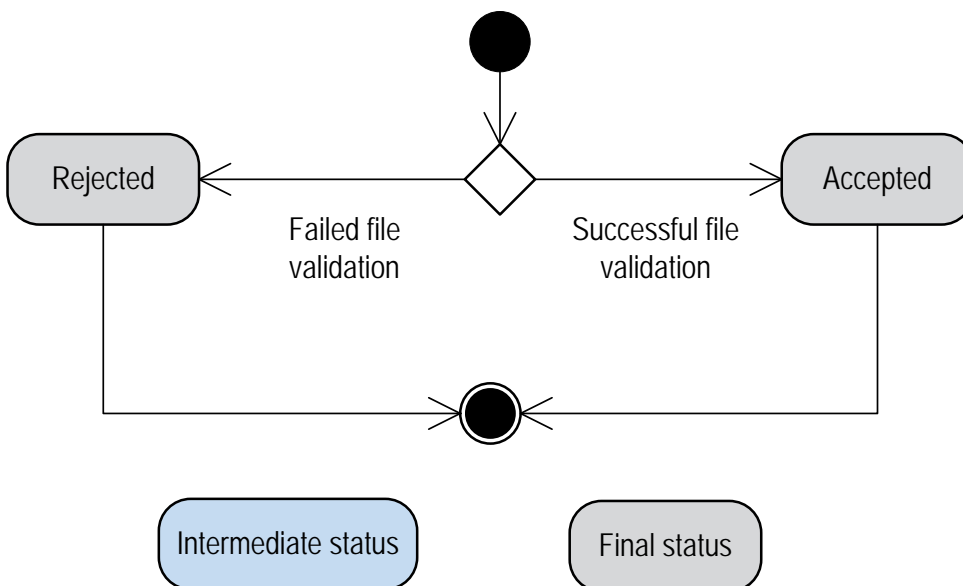


Figure 39 - File state diagram

Status value	Definition	Direction	Transition possible to status	Intermediate / final status	Reported via status notification to the sender
Accepted	File status if an incoming file is finally processed with positive validation result.	Inbound	-	Final	-
Rejected	File status if an incoming file is finally processed with negative validation result.	Inbound	-	Final	Mandatory

Table 89 - RTGS file status

RTGS message status

Indicates the status of the message in RTGS and it can have the following status:

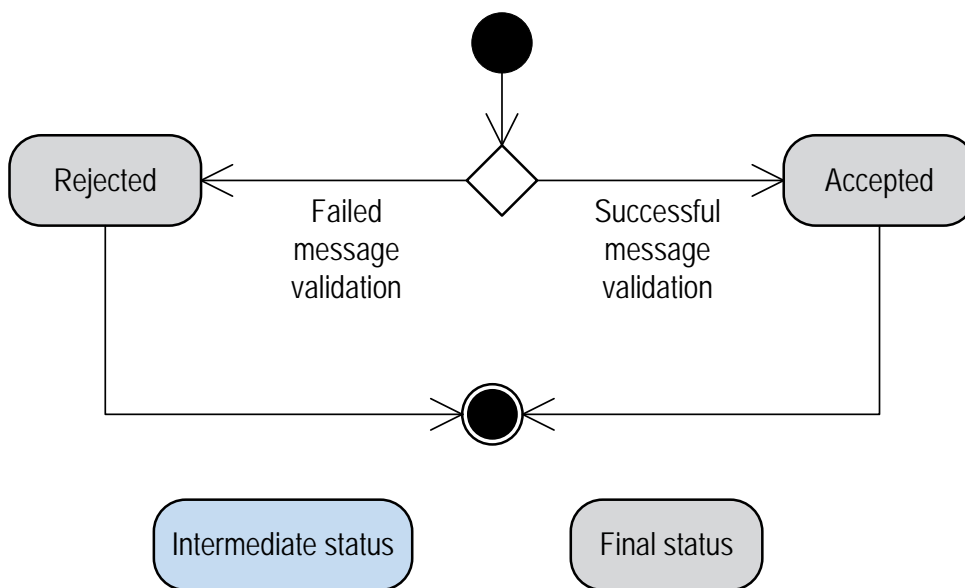


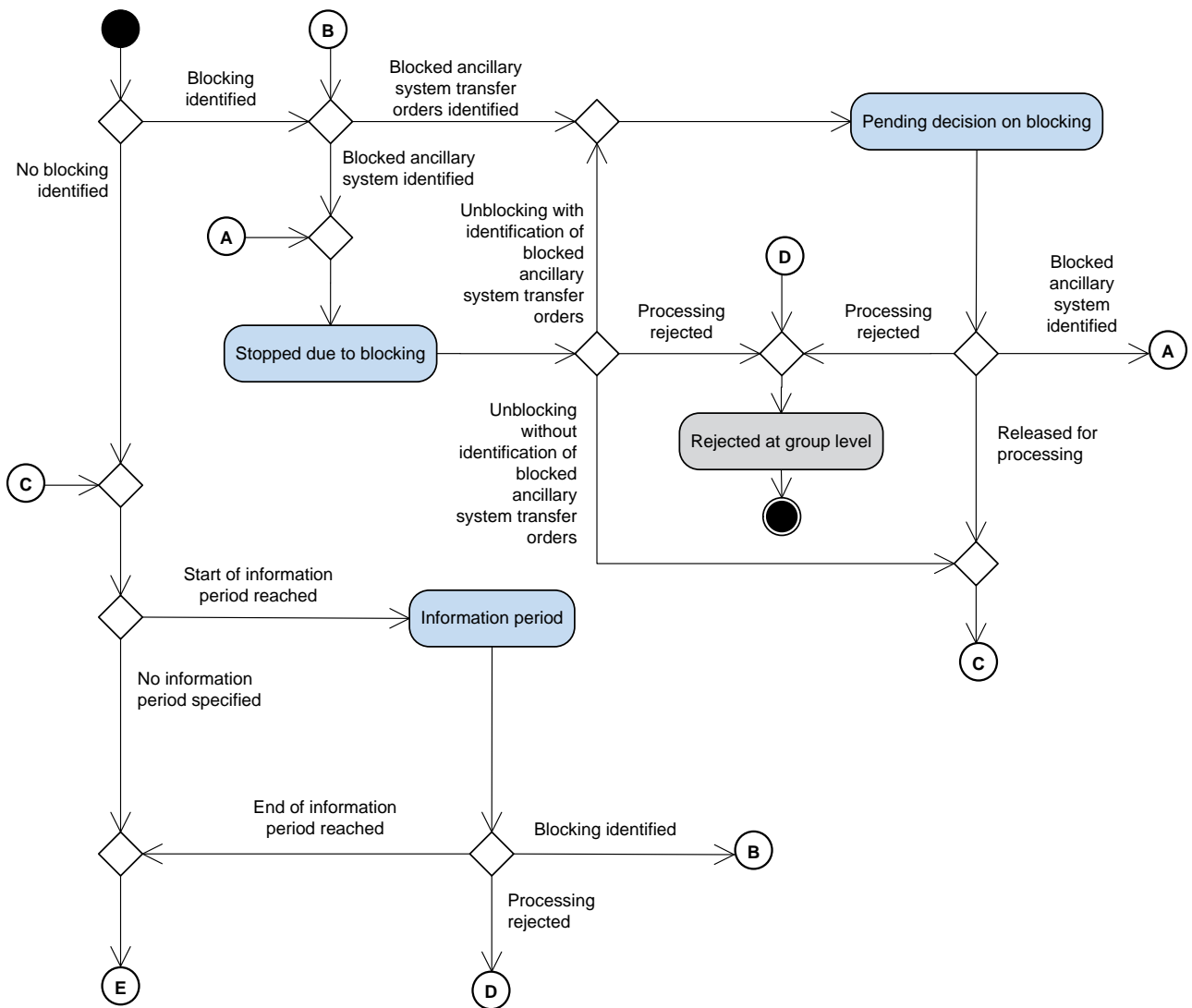
Figure 40 - Inbound RTGS message state diagram

Status value	Definition	Direction	Transition possible to status	Intermediate / final status	Reported via status notification to the sender
Accepted	Message status if an incoming message is finally processed with positive validation result.	Inbound	-	Final	-
Rejected	Message status if an incoming message is finally processed with negative validation result.	Inbound	-	Final	Mandatory
Provided	Status of an outgoing message sent to ESMIG.	Outbound	-	Final	-

Table 90 - RTGS message status

Ancillary system batch message status

Indicates the status of the ancillary system batch message in RTGS and it can have the following status:



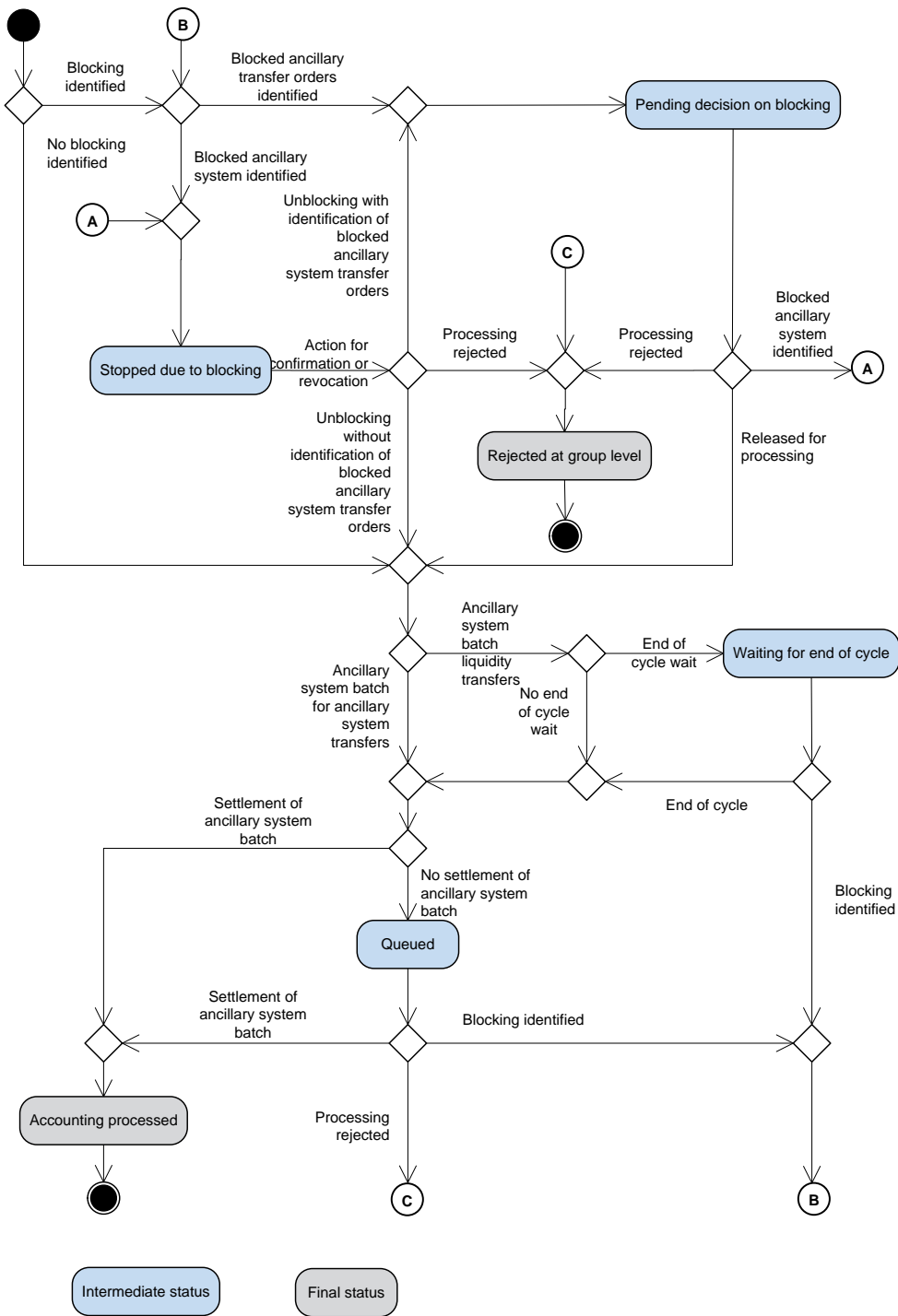


Figure 42 - Status transition diagram C

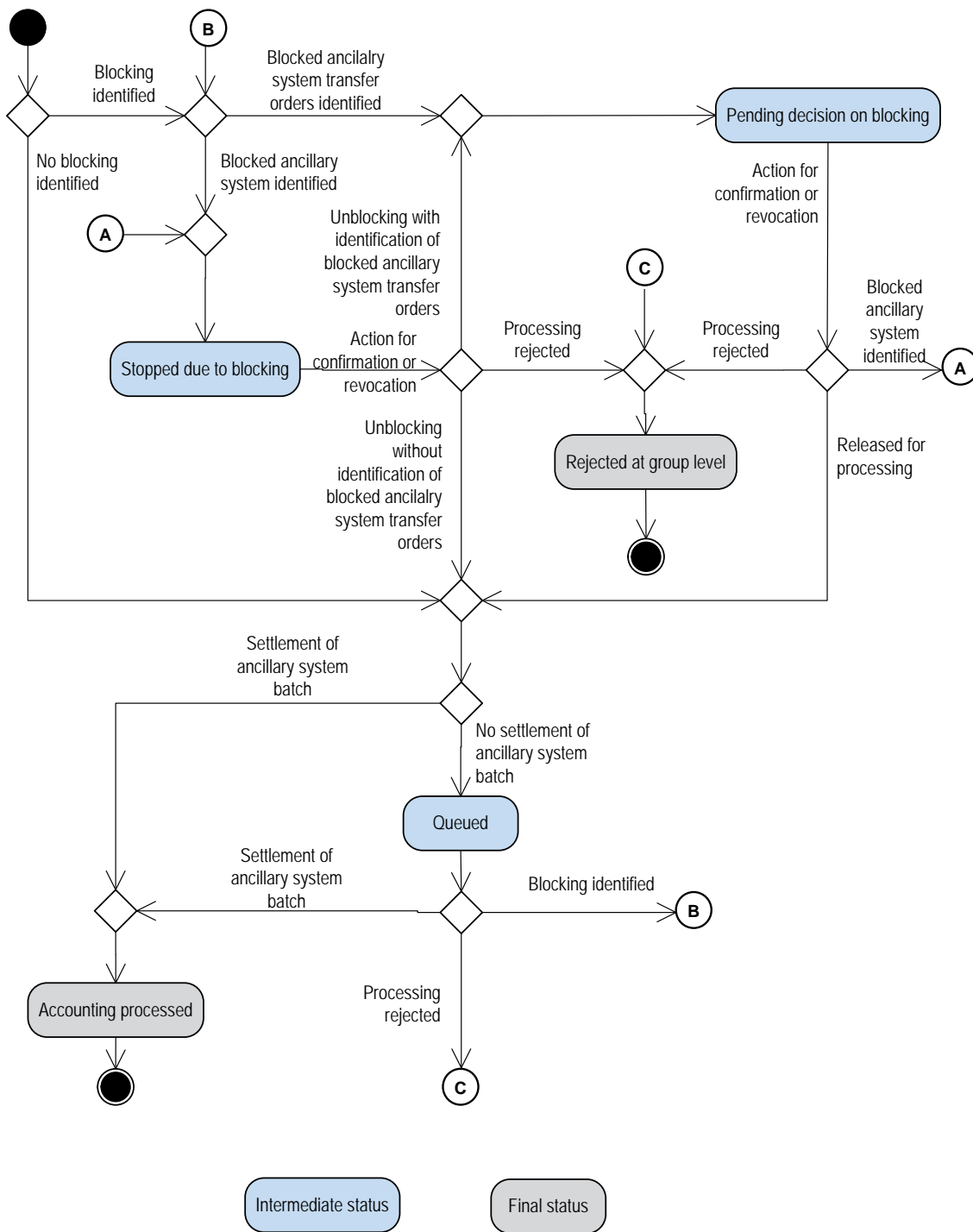


Figure 43 - Status transition diagram D

Status value	Definition	Transition possible to status	Intermediate / final status	Reported via status notification to the sender
Stopped due to blocking	This status is applied to an ancillary system batch message and to all ancillary system transfers within this ancillary system batch message in case of blocking of the ancillary system having submitted it The ancillary system batch message may either be agreed or disagreed by the CB.	Pending decision on blocking, Waiting for end of cycle, Information period, On guarantee mechanism, On settlement debit, Rejected at group level, Queued, Accounting processed	Intermediate	-
Pending decision on blocking	This status is applied to an ancillary system batch message in case of blocking of a RTGS Account Holder involved in another ancillary system Transfer in the same ancillary system batch message.	Queued	Intermediate	-
Waiting for end of cycle	Ancillary system liquidity transfer waiting for end of cycle.	Accounting processed, Rejected at group level	Intermediate	-
Information period	This status applies if the optional mechanism "information period" is selected for the relevant ancillary system batch message, between the start and the end of this information period.	Stopped due to blocking, Pending decision on blocking, On settlement debit, Queued, Accounting processed, Rejected at group level	Intermediate	Mandatory
On guarantee mechanism	If all ancillary system transfers are not settled (lack of liquidity by the end of settlement period) the ancillary system is asked on the use of the guarantee account.	Stopped due to blocking, Pending decision on blocking, On settlement debit, Queued, Accounting processed,	Intermediate	Mandatory

Status value	Definition	Transition possible to status	Intermediate / final status	Reported via status notification to the sender
		Rejected at group level		
On settlement debit	This status only applies to ancillary system batch messages, while the settlement attempts to settle the debit ancillary system transfers.	Queued, Stopped due to blocking, Pending decision on blocking, On guarantee mechanism, Rejected at group level	Intermediate	-
Queued	Status of an ancillary system batch message which is ready for settlement but the first settlement attempt was unsuccessful due to missing liquidity. Queued ancillary system batch messages are waiting for the next settlement booking attempt.	Stopped due to blocking, Pending decision on blocking, On guarantee mechanism, On settlement debit, Accounting processed, Rejected at group level	Intermediate	-
Accounting processed	The ancillary system batch message has been finally processed by the settlement.	-	Final	Mandatory
Rejected at group level	Ancillary system transfers already rejected at Group Level.	-	Final	Mandatory

Table 91 - Ancillary system batch message status

Some status values do not apply to all settlement procedures. Please find hereafter the list of possible values per ancillary system settlement procedure:

Status value	a	b	c	d
Stopped due to blocking	X	X	X	X
Pending decision on blocking	X	X	X	X
Waiting for end of cycle	-	-	X	-
Information period	X	X	-	-
On guarantee mechanism	X	X	-	-

Status value	a	b	c	d
On settlement debit	X	-	-	-
Queued	X	X	X	X
Settled	X	X	X	X
Rejected at group level	X	X	X	X

Table 92 - List of status values per ancillary system procedure

Cash transfer status

Indicates the status of the cash transfer in RTGS and it can have the following status:

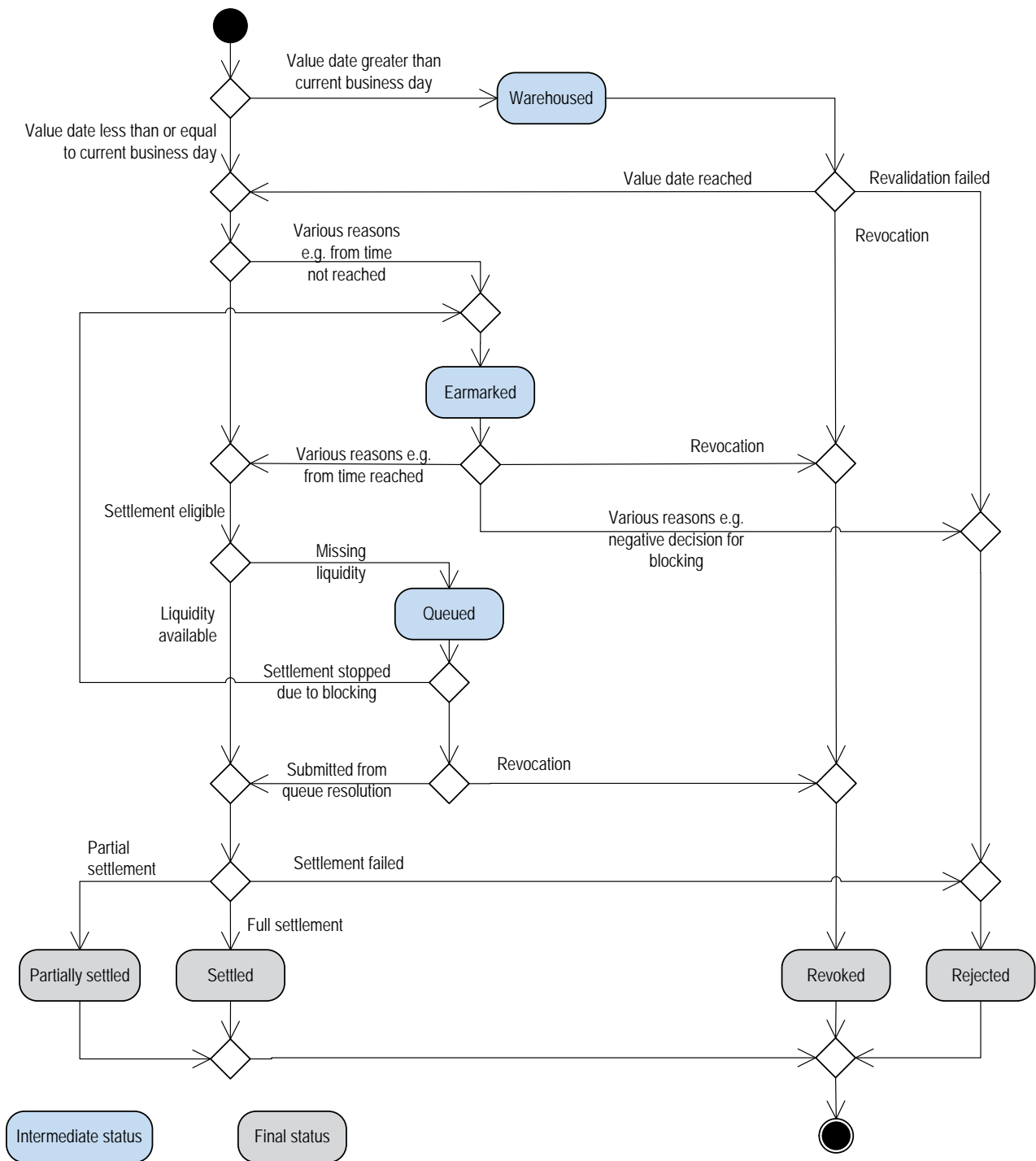


Figure 44 - Cash transfer state diagram

Status value	Definition	Transition possible to status	Intermediate / final status	Reported via status notification to the sender
Warehoused	Status of a cash transfer with a value date of a future business day and status of a cash transfer with the value date of the current business day until it will be forwarded to the processing at the start of the business day. From then on they will be processed normally. To this cash transfer status a time stamp is stored.	Earmarked, Partially settled, Queued, Revoked, Rejected, Settled	Intermediate	-
Earmarked	Status of a cash transfer which is ready for settlement but not taken into account for various reasons. The following scenarios are summarised in this status <ul style="list-style-type: none"> - pending start of settlement - accounting stopped due to earliest debit time indicator - ancillary system accounting not yet started due to active information period - accounting stopped due to blocking - pending decision on blocking - waiting for end of cycle - waiting for completion of debits - waiting for algorithm 4 (Settlement of queued normal payments [133]) 	Queued, Partially settled, Revoked, Rejected, Settled	Intermediate	-
Queued	Status of a cash transfer which is ready for settlement but the first settlement attempt was unsuccessful. Queued cash transfers are waiting for the next settlement booking attempt. To this cash transfer status a time stamp is stored.	Earmarked, Partially settled, Revoked, Rejected, Settled	Intermediate	-
Partially settled	Status of a cash transfer after settlement with an amount lower than ordered. For business cases where the remaining (unsettled) amount should be settled the component status creates a new cash transfer.	-	Final	Mandatory

Status value	Definition	Transition possible to status	Intermediate / final status	Reported via status notification to the sender
Revoked	Status of a cash transfer which is revoked by a system user i.e. by an action to prevent the settlement of a cash transfer order.	-	Final	Mandatory
Rejected	Status of a cash transfer which is rejected by the system i.e. by an action to refuse to continue processing (all cash transfers with error code, except error code for revoked).	-	Final	Mandatory
Settled	Status of a cash transfer after settlement. Final cash transfers cannot be revoked. To this cash transfer status a time stamp is added.	-	Final	Optional for payments, mandatory for liquidity transfers

Table 93 - Cash transfer status

Task queue order status

Indicates the status of the task queue order in RTGS and it can have the following status:

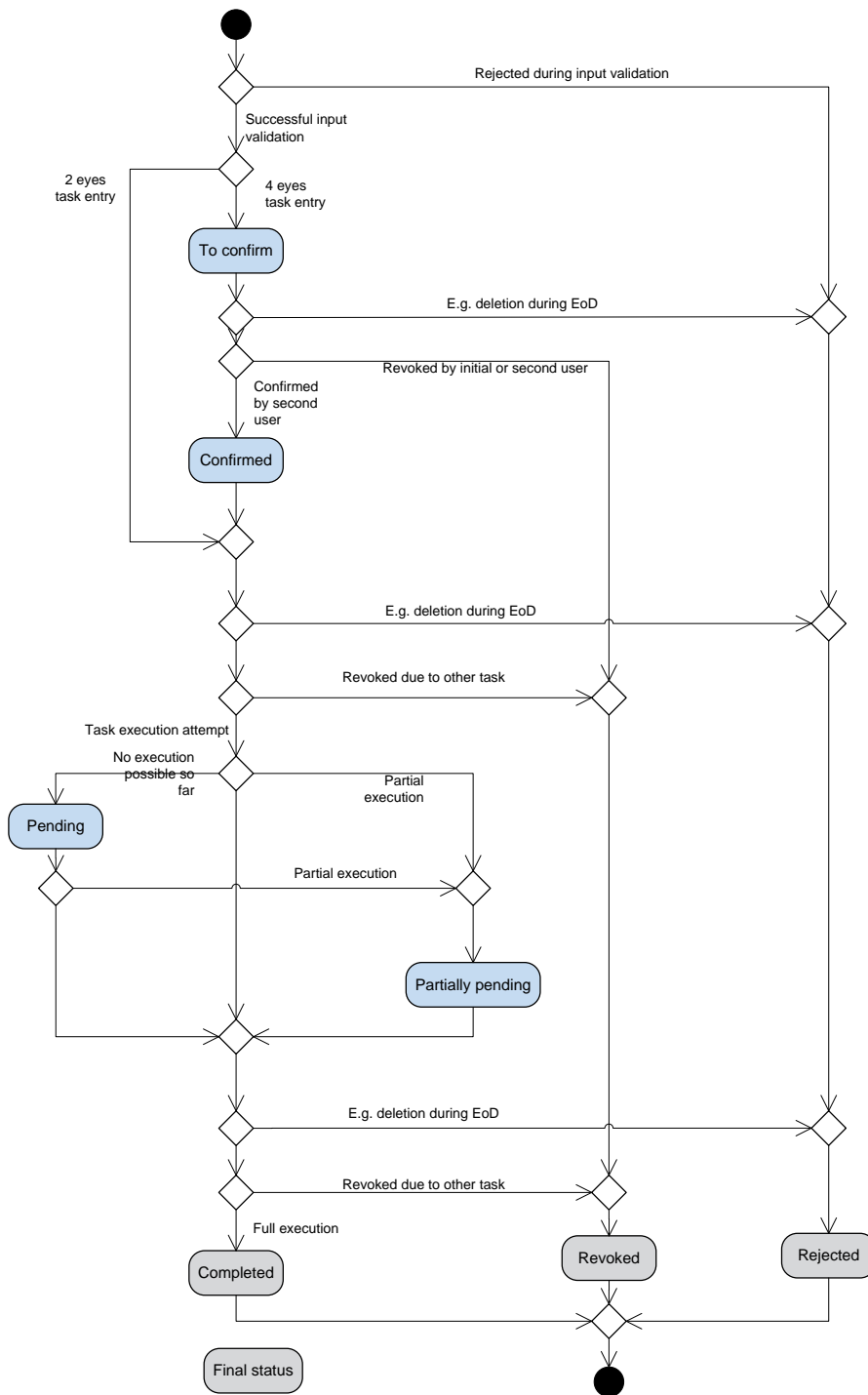


Figure 45 - Task queue order state diagram

Status value	Definition	Transition possible to status	Intermediate / final status	Reported via status notification to the sender
To confirm	The task must be confirmed by a second user and will not be processed. This status can only occur in U2A for four eyes principle. It is the only status in which a task revocation (and confirmation) is possible directly via respective screens.	waiting, revoked, rejected	Intermediate	-
Confirmed	The task is confirmed by a second user and is ready for further processing. This status can only occur in U2A for four eyes principle.	Pending, partially pending, completed, revoked, rejected	Intermediate	-
Pending	A task should be stored with status “pending”, if the task was already tried to process at least one time but it could not be finalised. The processing was interrupted after the storage of entries initiated by the task and before the final processing of these entries. The task will be updated and further processed, if the preconditions for the pending status (e.g. liquidity increase) are changed.	Partially pending, completed, revoked, rejected	Intermediate	-
Revoked	Status based on an action by the user to prevent the processing due to 4-eyes approval process.	-	Final	Mandatory
Rejected	Status based on an action by the system to refuse to continue processing.	-	Final	Mandatory
Completed	The task was processed successfully and the business case stemming from the task is final. The tasks changing an existing business case (like queue management) are completed, if the respective action is completely processed. The business case (managed payment) does not have to be final. To this task queue status a time stamp is added	-	Final	Mandatory

Table 94 - Task queue order status

Tasks with status “pending” can only be revoked via a new task.

5.5.2 RTGS report generation

5.5.2.1 Concept

RTGS provides the possibility to periodically create the predefined report “Statement of account”. RTGS triggers the generation of the “Statement of account” report based on the reference data configuration. It is only foreseen at business event “EoD”. The report is not created intraday. Depending on the RTGS Actor’s preferences the report is either sent out directly after creation or stored for later retrieval.

Report name	ISO message	ISO code
Statement of accounts	BankToCustomerStatement	BankToCustomerStatement (camt.053) [505]

Table 95 - Report “Statement of accounts”

The respective business process is described in chapter [Receive report](#) [▶ 351].

5.5.2.2 Overview

The report “Statement of account” includes information on one single RTGS cash account of an RTGS Actor. It is not possible to receive one combined “Statement of account” for more than one RTGS cash account. Furthermore it does not include information from other components, i.e. there is no report including combined information of CLM and RTGS.

The report provides information about all items that have been settled on the RTGS cash account and balance information of the current business day. This rule applies independent of the value date included in the payment, see transmission of unprocessed payments with original settlement date due to backup scenario in chapter [Subsequent delivery of single payments](#) [▶ 89].

It is provided as complete report i.e. no delta version is offered.

Report configuration and message subscription for notifications are different functionalities, i.e. no message subscription reference data is needed in case the report should be created and sent (later in case of push mode).

5.5.2.3 Report generation process

Preconditions for report creation

In order to avoid unnecessary processing and storage RTGS does not create reports automatically. So, to initiate the creation of a report, the report receiver has to configure the report in advance. The configuration of the report has to be done via the GUI for the reference data, which is described in the RTGS user handbook.

This configuration is stored as reference data and is valid until the “valid to” date stored within the report configuration is reached.

Moment of data extraction

The creation of a “Statement of account” report is always triggered at the EoD of the RTGS component after finalization of booking processes [business event “EoD”] – see [EoD](#) [▶ 76]. A new report configuration can be set-up at the earliest for the next business day. The respective component only creates those reports, for which the underlying report configuration is valid at the current business day.

Availability of the report in RTGS

A generated report is available for download until it is replaced by a new (next) version of it, i.e. a report that is created at the EoD of the current business day replaces the report that was created at the EoD of the previous business day. The replaced report is no longer available for download in RTGS. In A2A mode RTGS pushes the specific report, provided that the push preference for the report is stored for the respective recipient in reference data (i.e. report configuration). The message is sent out based on the routing information stored for the RTGS Actor. Otherwise the report is just stored after generation and can be downloaded in pull mode.

CRDM parameter synthesis

The following parameters are created and updated by the CRDM Actor (see Table 102 - [Report configuration](#) [▶ 242]) for the setup of a report.

Parameter	Mandatory/ optional	Possible values	Further information
Report type	Mandatory	Statement of accounts	
Concerned account	Mandatory	RTGS cash account	
Possible recipient of a report	Mandatory	RTGS Actor	

Parameter	Mandatory/ optional	Possible values	Further information
Communication channel	Mandatory	Push mode, pull mode	
Valid from	Mandatory	ISO-Date	
Valid to	Optional	ISO-Date	The field "valid to" is the only field that can be amended after the report configuration has been stored.

Table 96 - CRDM parameter synthesis

Concerned account

Each report provides information on a certain scope of data. The data scope is indicated by the RTGS cash account for which it is configured. The feature is available for all RTGS cash account types (including RTGS sub-accounts).

The concerned account has to be specified, when the report is configured for the first time. It is necessary to store one configuration per RTGS cash account and recipient for which the report should be created.

Possible recipients of a report

All reports can be received by the technical address of

- | concerned account owner;
- | another authorised party.

A created report can be received by one or several receivers. Each RTGS Actor can decide, if they wish to receive a report directly after its creation or rather query it ad-hoc.

If a recipient wishes to receive a report directly after its creation, this has to be stored in the reference data configuration of the report in CRDM (communication channel = push mode). In this case reports can be received by the technical address defined for the RTGS cash account or by the technical address defined for the other authorised party see chapter [Routing](#) [▶ 41].

If a recipient does not wish to receive a report directly after its creation but to request it afterwards, this RTGS behaviour has to be stored in the reference data configuration of the report as well (communication channel = pull mode).

Furthermore the recipient is stored as recipient of a report independent of the configuration with push or pull mode.

For information about the setup of a report configuration for a specific concerned report recipient, please see RTGS user handbook chapters related to report configuration setup.

5.5.3 Query management for RTGS

5.5.3.1 Concept for RTGS

Queries are provided by RTGS to the submitting actor as a means of satisfying his information needs on demand. The submitting actor can obtain information on different business items by submitting query requests to RTGS. These are answered on the basis of the latest data available.

For requests on RTGS queries using the specified (optional and mandatory) search and return criteria are available. Thus actors are not able to define these criteria by themselves.

The respective business process is described in chapter [Execute query](#) [▶ 348].

5.5.3.2 Overview for RTGS

RTGS provides a range of predefined query types, which the submitting actor can use to request information on business items. The offered queries are available for all authorised submitting RTGS Actors.

They can send query requests to RTGS in A2A mode or in U2A mode. Generally, all these query requests are processed in real-time. Exceptions occur during the maintenance window. During the maintenance window query management does not service any requests. In case ESMIG is available and the network interface is not closed, an A2A query request during business service maintenance window will be handled by using timeout management. As regards information on routing please see chapter [Routing](#) [▶ 41]. In case the network interface is closed, NSP informs the authorised submitting actor about the closure of the real-time channel.

5.5.3.3 Query management process for RTGS

Initiating queries for RTGS

In order to obtain the desired information the submitting actor needs to submit a query request to RTGS. For the communication with RTGS in A2A mode all query and response messages are set up as XML messages compliant with the ISO20022 standard. For the communication with RTGS in U2A mode a GUI based on a standard browser application is provided.

In general an authorised submitting actor can send each query request in A2A mode as well as in U2A mode. However, there are some queries which are only accessible via U2A mode. Query availability in the respective communication mode is shown in the table below. Query request and return criteria are described

in detail in RTGS user handbook for U2A mode and in chapter [List of messages](#) [381] with link to MyStandards for A2A mode.

Query type	Initiation via GUI (U2A mode)	Initiation via XML message (A2A mode)
Account balance query	X	X
Account statement query	X	X
Ancillary system batch message query	X	-
Audit trail for RTGS query	X	X
Broadcast query	X	-
Business case query	X	-
Cash transfer query	X	X
Current limits query	X	X
Current reservations query	X	X
Event query	X	X
File query	X	-
Message query	X	-
System time query	X	X
Task queue query	X	-

Table 97 - Initiating queries for RTGS

The different types of queries in RTGS are static regarding the set of selection parameters, which can be mandatory, optional or conditional.

Preconditions for successful processing of queries

RTGS validates the plausibility of search criteria that were specified by the submitting actor. In addition, RTGS ensures that the submitting actor of the query is allowed to initiate the query and to retrieve the requested information by checking, whether the submitting actor possesses all necessary privileges granted in advance (taking into account the validity dates) and ensuring the data scope.

Providing data for queries

If all checks performed by RTGS were successful, it extracts the requested business information from the production data. The submitting actor receives the latest available data.

If any plausibility or authorisation checks performed by RTGS fail, the submitting actor receives a response specifying the error(s) using the respective error code(s).

Retrieving the query response

In case the extraction of the query data is successful, the RTGS sends a query response containing the requested business information back to the requesting actor. In case the extraction of the query data returns a zero result, the submitting actor receives appropriate information. If a retrieval of the query result fails, then an error response is provided to the submitting actor.

If the submitting actor has sent the query via U2A mode, the response is given to the same submitting actor in U2A mode. The U2A dialogue is described more in detail in the RTGS user handbook.

If the submitting actor has sent the query via A2A mode, the response is given to the same component user in A2A mode. RTGS does not allow the routing of the query response to a dedicated technical address.

Parameter synthesis

No specific configuration from the submitting actor is needed.

6 Overview of used common components in RTGS component

6.1 CRDM features

6.1.1 Concept

The CRDM common component allows duly authorised users to create and maintain reference data objects. CRDM objects specify reference data for the configuration of parties, cash accounts and rules and parameters.

6.1.2 Overview

The CRDM common component is in charge of executing reference data maintenance instructions for the creation or the maintenance of reference data objects.

Duly authorised users belonging to CBs, payment banks and to the operator can trigger CRDM according to their own specific access rights, i.e. using the functions and maintaining the common reference data objects they have been granted.

Duly authorised users of the operator are responsible for system configuration tasks and for the management of common reference data for CBs. These users can also act on behalf of other CRDM Actors in order to perform some specific actions or within some pre-defined contingency scenarios.

The CRDM common component executes immediately all reference data maintenance instructions. The related reference data changes become effective in the relevant TARGET Service, common component(s) or back-office applications in a deferred way, by means of a daily reference data propagation process. The process takes place every business day and is scheduled in order to ensure a smooth and complete reference data propagation depending on the operational schedule of the relevant service.

All common reference data objects can be created and maintained in U2A mode, whereas only a sub-set of them can be created and maintained also through the data migration tool (DMT) (see chapter [Reference data maintenance types](#) [▶ 268]). All reference data changes performed in U2A mode can be executed either in two-eyes or in four-eyes mode. Duly authorised users can specify the applicable mode for the functions and the common reference data objects they manage (see chapter [Access rights](#) [▶ 235]).

Versioning facilities and validity periods allow the implementation of data revision and data history features, in order to keep track of all past data changes, to enter changes meant to become effective as of a future date and to define common reference data objects with limited or unlimited validity.

6.1.3 Access rights

This section provides information on access rights management in the CRDM. More into detail, chapter [Access rights concepts](#) [▶ 235] presents some basic concepts (e.g. user, privilege, role and data scope) related to access rights management. On this basis, chapter [Access rights configuration](#) [▶ 250] illustrates all the available options for the configuration of access rights. Finally, chapter [Access rights configuration process](#) [▶ 259] describes the access rights configuration process that each type of CRDM Actor has to put in place in order to set-up the appropriate assignment of roles and privileges for all its users. In order to clarify the differences in data scope per type of actor, this section uses the concepts of CB and payment bank, which are introduced in chapter [Common reference data objects and the hierarchical party model](#) [▶ 247], as well as the concept of system entity, which is introduced in chapter [Data scope](#) [▶ 248].

6.1.3.1 Access rights concepts

This chapter presents the main concepts related to access rights management in CRDM.

6.1.3.1.1 User function

DMT files, XML messages and GUI functions are the atomic elements users can trigger through the DMT and in A2A and U2A mode respectively to interact with CRDM as well as other services, common components or back-office applications. Based on these set of files, XML messages and GUI functions, it is possible to define the set of all user functions, i.e. of all the possible actions that a user can trigger in CRDM or other services, common components or back-office application services, either in the DMT or in A2A or U2A mode.

6.1.3.1.2 Privilege

A privilege identifies the capability of triggering one or several user functions and it is the basic element to assign access rights to users. This means that a user U_x owns the access right to trigger a given user function F_y if and only if U_x was previously granted with the privilege P_y identifying the capability to trigger F_y .

The following tables provide the exhaustive list of privileges covering all the user functions available:

- | table access rights management
- | table party data management
- | table cash account data management
- | table message subscription configuration
- | table report configuration
- | table reference data queries
- | table TIPS functions

I table other

Privilege	User function	Data scope
Administer party ⁸	n/a	n/a
Create certificate distinguish name	Certificate DN – new	Any certificate DN
Create DN-BIC routing	DN-BIC routing - new	DN-BIC routing data within own system entity (for CBs) or for DNs linked to own users and BICs authorised to own cash accounts (for payment banks).
Create role	Role – new	Roles within own system entity (for CBs).
Create user	User – new	Users within own system entity (for CBs) or own party (for payment banks).
Create user certificate distinguish name link	User certificate DN link – new	Links within own system entity (for CBs) or for own users (for payment banks).
Delete certificate distinguish name	Certificate DN – delete/restore	Any certificate DN
Delete DN-BIC routing	DN-BIC routing - delete/restore	DN-BIC routing data within own system entity (for CBs) or for DNs linked to own users and BICs authorised to own cash accounts (for payment banks).
Delete role	Role – delete/restore	Roles within own system entity (for CBs).
Delete user	User – delete/restore	Users within own system entity (for CBs) or own party (for payment banks).
Delete user certificate distinguish name link	User certificate DN link – delete/restore	Links within own system entity (for CBs) or for own users (for payment banks).
Grant privilege	Grant privilege	Privileges granted to parties, roles and users within own system entity (for CBs) or to own users (for payment banks)
Grant/revoke role	Grant/revoke role	Roles granted to parties and users within own system entity (for CBs) or to

⁸ This privilege enables a user to act as party administrator for their own party.

Privilege	User function	Data scope
		own users (for payment banks)
Revoke privilege	Revoke privilege	Privileges granted to parties, roles and users within own system entity (for centrals) or to own users (for payment banks)
Update DN-BIC routing	DN-BIC routing - edit	DN-BIC routing data within own system entity (for CBs) or for DNs linked to own users and BICs authorised to own cash accounts (for payment banks).
Update role	Role – edit	Roles within own system entity (for CBs)
Update user	User – edit	Users within own system entity (for CBs) or own party (for payment banks).

Table 98 - Access rights management

Privilege	User function	Data scope
Create Banking Group	Banking Group – new	Banking Groups within own system entity (for CBs)
Create MFI	MFI – new	MFI within own system entity (for CBs)
Create party	Party – new	Parties within own system entity (for CBs)
Create party-service link	Party-service link - new	Links within own system entity (for CBs)
Create technical address network service link	Technical address network service link - new	Links within own system entity (for CBs)
Delete Banking Group	Banking Group – delete/restore	Banking Groups within own system entity (for CBs)
Delete MFI	MFI – delete/restore	MFIs within own system entity (for CBs)
Delete party	Party – delete/restore	Parties within own system entity (for CBs) excluding own party
Delete party-service link	Party-service link - delete/restore	Links within own system entity (for CBs)
Delete technical address networks	Technical address network service link	Links within own system entity (for

Privilege	User function	Data scope
service link	- delete/restore	CBs)
Update Banking Group	Banking Group – edit	Banking Groups within own system entity (for CBs)
Update MFI	MFI – edit	MFIs within own system entity (for CBs)
Update party	Party – edit	Parties within own system entity (for CBs)
Update party-service link	Party-service link - edit	Links within own system entity (for CBs)

Table 99 - Party data management

Privilege	User function	Data scope
Create Account Monitoring Group	Account Monitoring Group – new	Account Monitoring Groups within own system entity (for CB) or for own cash accounts (for payment banks)
Create authorised account user	Authorised account user - new	Links within own system entity (for CB) or for own cash accounts (for payment bank).
Create cash account	Cash account – new	Cash accounts within own system entity (for CB) or credit memorandum balances (CMBs) linked to cash accounts owned by own party (for payment bank)
Create direct debit mandate	Direct debit mandate - new	Direct debit mandates on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Create limit	Limit – new	Limits on CMBs defined on cash accounts within own system entity (for CB) or linked to cash accounts owned by own party (for payment bank)
Create liquidity transfer order	Liquidity transfer order – new	Liquidity transfer orders on cash accounts within own system entity (for CB) or owned by own party (for payment bank)

Privilege	User function	Data scope
Create Liquidity Transfer Group	Liquidity Transfer Group – new	Liquidity Transfer Groups containing liquidity transfer orders on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Create standing order for limit	Standing order for limit – new	Standing orders for limit on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Create standing order for reservation	Standing order for reservation – new	Standing orders for reservation on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Delete Account Monitoring Group	Account Monitoring Group – delete/restore	Account Monitoring Groups within own system entity (for CB) or for own cash accounts (for payment bank)
Delete authorised account user	Authorised account user - delete/restore	Links within own system entity (for CB) or for own cash accounts (for payment bank)
Delete cash account	Cash account – delete/restore	Cash accounts within own system entity (for CB) or CMBs linked to cash accounts owned by own party (for payment bank)
Delete direct debit mandate	Direct debit mandate – delete/restore	Direct debit mandates on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Delete limit	Limit – delete/restore	Limits on CMBs defined on cash accounts within own system entity (for CB) or linked to cash accounts owned by own party (for payment bank)
Delete liquidity transfer order	Liquidity transfer order – delete/restore	Liquidity transfer orders on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Delete Liquidity Transfer Group	Liquidity Transfer Group – de-	Liquidity Transfer Groups containing

Privilege	User function	Data scope
	lete/restore	liquidity transfer orders on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Delete standing order for limit	Standing order for limit – delete/restore	Standing orders for limit on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Delete standing order for reservation	Standing order for reservation – delete/restore	Standing orders for reservation on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Update Account Monitoring Group	Account Monitoring Group – edit	Account Monitoring Groups within own system entity (for CB) or for own cash accounts (for payment bank).
Update authorised account user	Authorised account user - edit	Links within own system entity (for CB) or for own cash accounts (for payment bank).
Update cash account	Cash account – edit	Cash accounts within own system entity (for CBs) or CMBs linked to cash accounts owned by own party (for payment bank)
Update direct debit mandate	Direct debit mandate – edit	Direct debit mandates on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Update limit	Limit – edit	Limits on CMBs defined on cash accounts within own system entity (for CB) or linked to cash accounts owned by own party (for payment bank)
Update liquidity transfer order	Liquidity transfer order – edit	Liquidity transfer orders on cash accounts within own system entity (for CB) or owned by own party (for payment bank)

Privilege	User function	Data scope
Update Liquidity Transfer Group	Liquidity Transfer Group – edit	Liquidity Transfer Groups containing liquidity transfer orders on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Update standing order for limit	Standing order for limit – edit	Standing orders for limits on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Update standing order for reservation	Standing order for reservation – edit	Standing orders for reservation on cash accounts within own system entity (for CB) or owned by own party (for payment bank)

Table 100 - Cash account data management

Privilege	User function	Data scope
Create message subscription rule	Message subscription rule – new	Message subscription rules within own system entity (for CBs) or for own party (for payment banks)
Create message subscription rule set	Message subscription rule set – new	Message subscription rule sets within own system entity (for CBs) or for own party (for payment banks)
Delete message subscription rule	Message subscription rule – delete/restore	Message subscription rules within own system entity (for CBs) or for own party (for payment banks)
Delete message subscription rule set	Message subscription rule set – delete/restore	Message subscription rule Sets within own system entity (for CBs) or for own party (for payment banks)
Update message subscription rule	Message subscription rule – edit	Message subscription rules within own system entity (for CBs) or for own party (for payment banks)
Update message subscription rule set	Message subscription rule set – edit	Message subscription rule sets within own system entity (for CBs) or for own party (for payment banks)

Table 101 - Message subscription configuration

Privilege	User function	Data scope
Create report configuration	Report configuration – new	Report configurations within own system entity (for CBs) or for own party (for payment banks)
Delete report configuration	Report configuration – delete/restore	Report configurations within own system entity (for CBs) or for own party (for payment banks)
Update report configuration	Report configuration – edit	Report configurations within own system entity (for CBs) or for own party (for payment banks)

Table 102 - Report configuration

Privilege	User function	Data scope
Account Monitoring Group query	Account Monitoring Group – list	Account Monitoring Group
Authorised account user query	Authorised account user – list	Links within own system entity (for CBs) or for own cash accounts (for payment banks).
Banking Group query	Banking Group – list	Any Banking Group
BIC query	BIC query	Any BIC
Cash account audit trail query	Revisions - selection criteria + list	Data within own system entity (for CB) or linked to own party (for payment bank)
Cash account list query	Cash account list query	Cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Cash account reference data query	Cash account reference data query	Cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Certificate query	Certificate query	Any certificate DN
Country query	Countries – select + list	Any country
Currency query	Currencies – select + list	Any currency
Data changes of a business object details query	Data changes of a business object details query	Data within own system entity (for CBs) or linked to own party (for payment banks)

Privilege	User function	Data scope
Data changes of a business object list query	Data changes of a business object list query	Data within own system entity (for CBs) or linked to own party (for payment banks)
Direct debit mandate details query	Direct debit mandate – details	Direct debit mandates on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Direct debit mandate List query	Direct debit mandate – list	Direct debit mandates on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Granted roles list query	Granted roles – search	Roles granted to parties and users within own system entity (for CBs) or to own users (for payment banks)
Granted roles list query	Grant/revoke role – details	Roles granted to parties and users within own system entity (for CBs) or to own users (for payment banks)
Granted system privileges list query	Grant/revoke system privileges list query	Privileges granted to parties, roles and users within own system entity (for CBs) or to own users (for payment banks)
Limit query	Limit query	Limits on CMB defined on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Liquidity transfer order details query	Liquidity transfer order – details	Liquidity transfer orders on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Liquidity transfer order list query	Liquidity transfer order – list	Liquidity transfer orders on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Liquidity Transfer Group query	Liquidity Transfer Group – list	Liquidity Transfer Groups within own system entity (for CB) or containing cash accounts owned by own party (for

Privilege	User function	Data scope
		payment bank)
Market-specific restriction list query	Market-specific restriction list query	Restrictions defined by the operator
Market-specific restriction type rule detail query	Market-specific restriction type rule – detail query	Restrictions defined by the operator
Market-specific restriction type rule parameter details query	Market-specific restriction type rule parameter details query	Restrictions defined by the operator
Market-specific restriction type rule set list query	Market-specific restriction type Rule set list query	Restrictions defined by the operator
Message subscription rule list query	Message subscription rule list query	Message subscriptions within own system entity (for CBs) or for own party (for payment banks)
Message subscription rule set details query	Message subscription rule sets details query	Message subscriptions within own system entity (for CBs) or for own party (for payment banks)
Message subscription rule set list query	Message subscription rule set list query	Message subscriptions within own system entity (for CBs) or for own party (for payment banks)
MFI query	MFI – list	Any MFI
Network service list query	Network service list query	Any network service
Party audit trail query	Static data audit trail query	Data within own system entity (for CB) or linked to own party (for payment bank)
Party list query	Party list query	Parties within own system entity (for CB) or own party (for payment bank)
Party reference data query	Party reference data query	Parties within own system entity (for CB) or own party (for payment bank)
Party-service link list query	Party-service link list query	Links within own system entity (for CBs) or linked to own party (for payment banks)
Party-service link query	Party-service link query	Links within own system entity (for CBs) or linked to own party (for payment banks)
Privilege query	Privilege – selection criteria + list	Any privilege

Privilege	User function	Data scope
Queued data changes query	Queued data changes – select + list	Data within own system entity (for CBs) or linked to own party (for payment banks)
Report configuration details query	Report configuration details query	Report configurations within own system entity (for CBs) or for own party (for payment banks)
Report configuration list query	Report configuration list query	Report configurations within own system entity (for CBs) or for own party (for payment banks)
Residual static data audit trail query	Static data audit trail query	Data within own system entity (for CBs) or linked to own party (for payment banks)
Role list query	Role list query	Roles created or granted to parties and users within own system entity (for CBs) or to own users (for payment banks)
Service list query	Service list query	Any service
Standing order for limit details query	Standing order for limit – details	Standing orders for limit on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Standing order for limit list query	Standing order for limit – list	Standing orders for limit on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Standing order for reservation details query	Standing order for reservation – details	Standing orders for reservation on cash accounts within own system entity (for CB) or owned by own party (for payment bank)
Standing order for reservation list query	Standing order for reservation – list	Standing orders for reservation on cash accounts within own system entity (for CB) or owned by own party (for payment bank)

Privilege	User function	Data scope
System entity query	System entities – select + list	Own system entity (for CBs)
System user link query	System user link query	Links within own system entity (for CBs) or linked to own users (for payment banks)
Technical address network service link details query	Technical address network service link details query	Links within own system entity (for CBs) or linked to own party (for payment banks)

Table 103 - Reference data queries

Privilege	User function	Data scope
Adjust CMB limit	Adjust CMB limit	Data within own system entity (for CB) or linked to own party (for payment bank)
Instruct instant payment	Initiate instant payment Confirm/reject instant payment Request instant payment recall Confirm instant payment recall Reject instant payment recall Instant payment status investigation	Data related to accounts within own system entity (for CB) or for which own party is set as authorised user (for payment bank)
Instruct liquidity transfer	Initiate outbound liquidity transfer	Accounts within own system entity (for CB) or owned by own party (for payment bank)
Modify all blocking status	Block/unblock participant Block/unblock account Block/unblock CMB	Data within own system entity (for CB) or linked to own party (for payment bank)

Privilege	User function	Data scope
Modify CMB blocking status	Block/unblock CMB	Data within own system entity (for CB) or linked to own party (for payment bank)
Query all	Query account balance and status Query CMB limit and status Query instant payment transaction	Data related to accounts within own system entity (for CB) or owned by own party (for payment bank)
Query as reachable party	Query CMB limit and status Query instant payment transaction	Data related to accounts within own system entity (for CB) or for which own party is set as authorised user (for payment bank)

Table 104 - TIPS functions

Privilege	User function	Data scope
DMT access	n/a	n/a

Table 105 - Other

See chapter [Configuration of privileges](#) [▶ 250] for information on the configuration of privileges.

6.1.3.1.3 Role

A role is a set of privileges. See chapter [Configuration of roles](#) [▶ 257] for information on the configuration of roles.

6.1.3.1.4 User

A user is an individual or application that interacts with CRDM triggering the available CRDM user functions. See chapter [Configuration of users](#) [▶ 250] for information on the configuration of users.

6.1.3.1.5 Common reference data objects and the hierarchical party model

All parties in the CRDM are linked to each other according to a hierarchical model. As shown in the following diagram and on the basis of this hierarchical party model, the operator is the only party at level 1, all the CBs are level 2 parties, all payment banks are level 3 parties ⁹. All the other reference data objects are linked to a party. For example:

⁹ Participation types may be further detailed with information specific to each individual service, if the service foresees this possibility.

- | a cash account is linked to its CB or payment bank.
- | a restriction type is linked to the operator.

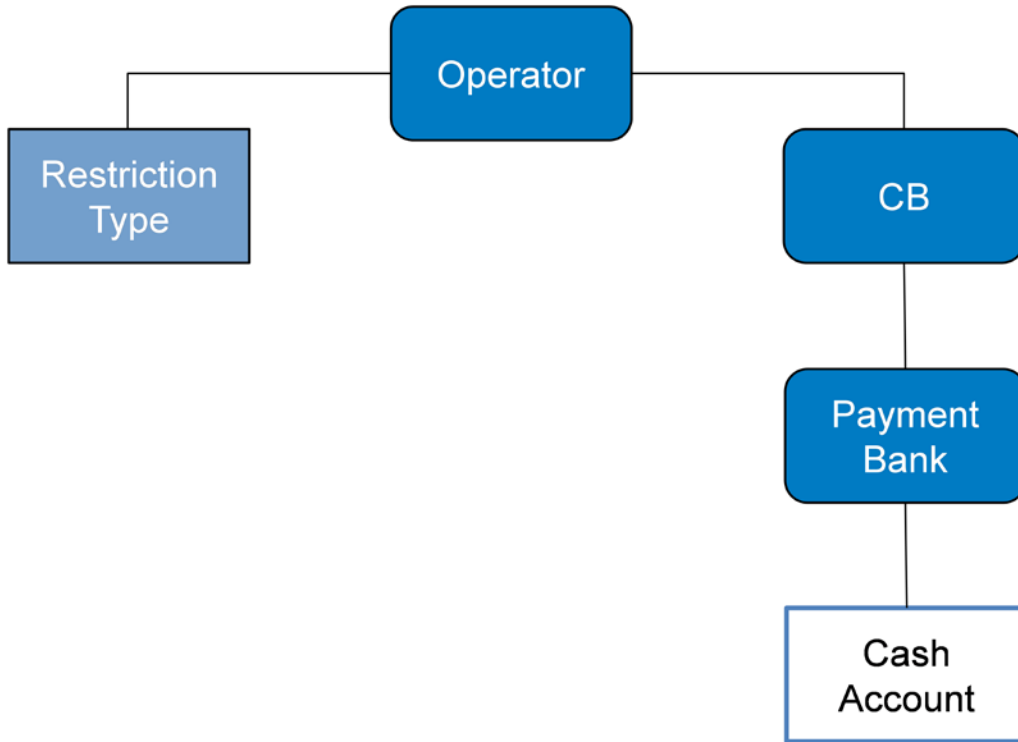


Figure 46 - Common reference data objects and the hierarchical party model

6.1.3.1.6 Data scope

For each privilege, the hierarchical party model determines the data scope of the grantee, i.e. the set of reference data objects on which the grantee can trigger the relevant user function. More precisely:

- | Users of the operator have visibility on all reference data objects and can act on objects belonging to participants only in exceptional circumstances, following a specific agreement
- | Users of the CBs have visibility on all reference data objects belonging to the same system entity ¹⁰
- | Users of the payment banks have visibility on reference data objects that are (directly or indirectly) linked to the same party

The following example describes the concept of data scope ¹¹.

10 A system entity in CRDM corresponds to a partition of data equating to the scope of a CB or of the operator. For example, the system entity of a CB includes all the data related to its payment banks.

11 The following example presents only the configuration data that are relevant for the example. All the possible configuration options are defined in the following sections.

Example – data scope

Three users, X, Y and Z, belonging to a payment bank, to a CB and to the operator respectively, are granted with the same privilege to query cash accounts.

User	Privilege
X	Cash account reference data query
Y	Cash account reference data query
Z	Cash account reference data query

Table 106 - User privileges (data scope)

The following diagram shows the data scopes stemming from this access rights configuration for the three users.

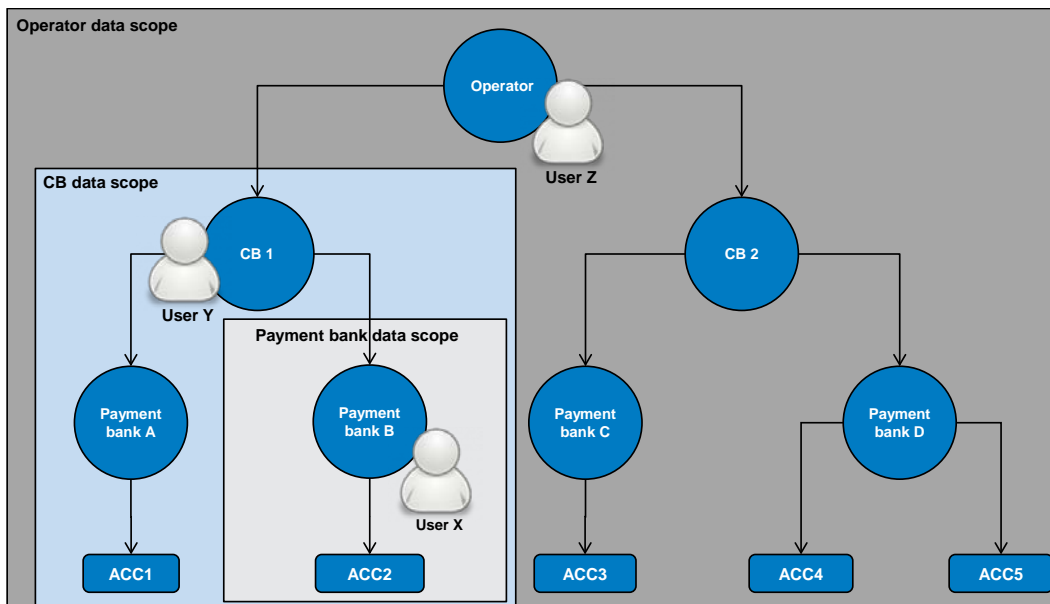


Figure 47 - Data scopes

The diagram shows that users X, Y and Z are given different data scopes, owing to the fact that they belong to different parties located at different levels of the hierarchical party model. More precisely:

- l User X of payment bank B gets a data scope including the cash account ACC2 only, as ACC2 is the only account of payment bank B. User X cannot query any other cash account in CRDM.
- l User Y of CB 1 gets a data scope including cash accounts ACC1 and ACC2, as these accounts belong to payment banks of CB 1. User Y cannot query any other cash account in CRDM, i.e. any cash account falling under the data scope of any other CB.
- l User Z of the operator gets a data scope including all cash accounts in CRDM, as the operator is at the top level of the hierarchical party model.

6.1.3.2 Access rights configuration

This chapter presents how roles and privileges can be configured in CRDM in order to grant each user with the appropriate set of access rights.

6.1.3.2.1 Configuration of users

Links between users and parties

Each new user is linked to the same party which the creator user belongs to. An exception takes place when creating the first user of a party, i.e.:

- | when a operator system administrator creates a new system administrator for a CB
- | when a CB system administrator creates a new system administrator for one of its payment banks

In all these cases the created user is linked to the party this user is going to administer.

Through the link with the relevant party, each user inherits a data scope (see chapter [Data scope](#) [► 248]). The link between a user and a party cannot be changed, i.e. a user is always linked to the same party.

Party administrators

Each party must have at least one party administrator, i.e. a user being granted specific system privileges that allow its grantee to grant any roles and privileges previously granted to the grantee's party.

6.1.3.2.2 Configuration of privileges

Availability of privileges

Each privilege, just after its creation, is available to the party administrator(s) of the operator only. This means that party administrators of all the other parties cannot grant this privilege to their users.

A privilege becomes available to a party administrator of a party different from the operator only after this privilege has been granted to this party. From this moment on, the party administrator can grant this privilege, according to the rules defined in the following sections.

This implies that a two-step process is required in order to grant a specific privilege to a user belonging to a party different from the operator. In the first step, the privilege is granted to the relevant party (so that it becomes available to the party administrator(s) of this party). With the second step, one of the party administrators grants the privilege to the relevant user.

The following diagram illustrates the access rights configuration steps needed to grant a user Z of a party B a given privilege P that is already available to the party administrator X of another party A. ¹²

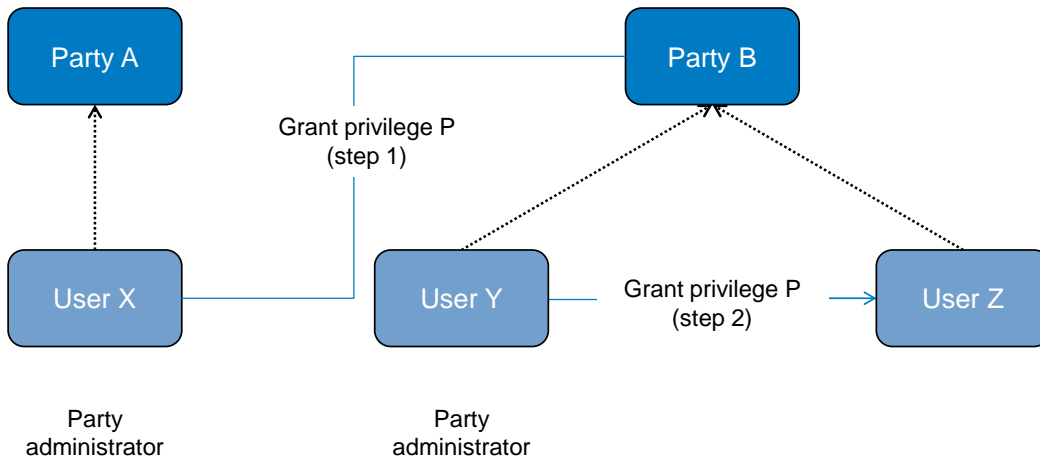


Figure 48 - Access rights configuration steps

The two configuration steps are as follows.

- | User X, as a party administrator of party A, grants privilege P to party B. From this moment on, privilege P becomes available to the party administrator Y of party B.
- | User Y, as a party administrator of party B, grants privilege P to user Z. From this moment on, user Z can trigger the user functions linked to privilege P.

At party level, access rights are propagated following the hierarchical party model, i.e. the operator propagates access rights to CBs which in turn propagate them to their payment banks. If necessary, the operator can act on behalf of a CB following a specific request to propagate access rights directly to its payment banks.

While the features described above apply to all privileges related to CRDM functions, it should be noted that TIPS privileges cannot be granted directly to parties or users, but can only be granted to roles, which can in turn be granted to parties and users. This implies that the above described configuration steps remain valid for TIPS as well, but in this case privileges have to be granted to roles in the first place and then roles can be granted to parties and users. For details on the configuration of roles see chapter [Configuration of roles](#) [▶ 257].

Granting privileges

Most privileges can be granted to roles, users and parties, with the exception of TIPS privileges that can be granted to roles only. When granting a privilege, the grantor specifies appropriate values for the three following assignment options: deny option, administration option and four-eyes option.

¹² Party A may be the operator or any other party which was previously granted privilege P.

Option	Description
Deny	This option specifies whether the associated user function is allowed (deny is false) or explicitly denied (deny is true).
Administration	<p>If the grantee of the privilege is a user or a role, this option specifies whether the grantee is allowed to grant the same privilege to another user or role of the same party (administrator is true) or not (administrator is false).</p> <p>If the grantee of the privilege is a party, this option specifies whether the party administrators of the grantee party is allowed to grant the same privilege only to users and roles of the same party (administrator is false) or also to other parties (administrator is true).</p>
Four-eyes	<p>This option specifies whether the grantee of the privilege is allowed to use the function associated to the privilege according to the two-eyes (four-eyes is false) or four-eyes (four-eyes is true) principles.</p> <p>This option is relevant only when the deny option is set to false and it is always not relevant for privileges related to queries.</p>

Table 107 - Privilege assignment options

Example - assignment of privileges to roles

The following table shows some examples of assignment of privileges to roles:

Row	Role	Privilege	Deny	Admin	Four-eyes
1	Cash account management	Cash account reference data query	False	False	Not relevant
2	Cash account administration	Cash account reference data query	True	True	Not relevant
3	Party management	Create party	False	False	True
4	Party management	Update party	False	False	True
5	Party management	Delete party	False	False	True
6	Party management	Party reference data query	False	True	Not relevant

Table 108 - Assignment of privileges to roles

For each assignment of a privilege to a role, three additional attributes define the features of such assignment.

For example, according to row 1, the privilege to query cash account data is assigned to the cash account management role:

- | without deny, i.e. users linked to the cash account management role can query cash account data ¹³;
- | without admin, i.e. users linked to the cash account management role cannot grant the privilege to query cash account data to other roles and users

According to row 2, the privilege to query cash account data is assigned to the cash account administration role:

- | with deny, i.e. users linked to the cash account administration role cannot query cash account data
- | with admin, i.e. users linked to the cash account administration role can grant the privilege to query cash account data to other roles and users of the same party

As a whole, rows 1 and 2 result in a segregation of duties between business users and access rights administrators. In fact, users linked to the cash account management role can query accounts, but they cannot configure the same access rights for any other user. On the contrary, users linked to the cash account administration role cannot query accounts, but they can configure these access rights for other users.

According to row 3, the privilege to create parties is assigned to the party management role:

- | without deny and with four-eyes set to true, i.e. users linked to the party management role can create parties according to the four-eyes principle only
- | without admin, i.e. users linked to the party management role cannot grant the privilege to create parties to other roles and users

As per rows 4 and 5, the privileges to maintain and delete parties are assigned to the party management role with the same assignment options.

Finally, according to row 6, the privilege to query parties is assigned to the party management role:

- | without deny, i.e. users linked to the party management role can query parties
- | with admin, i.e. users linked to the party management role can grant the privilege to query parties to other roles and users of the same party

As a whole, rows from 3 to 6 only result in a partial segregation of duties between business users and access rights administrators. In fact:

- | business users linked to the party management role can create, maintain, delete and query parties, they can only configure the same access rights for any other user limited to the query privilege
- | on the contrary, access rights administrators linked to the party management role, and whose party is also linked to the same role, can create, maintain, delete and query parties and they can also grant the

¹³ In this case the setting for the four eyes assignment option is not applicable, as the privilege refers to a query.

same privilege to other users of the same party; in addition, they can also grant the query privilege to other parties.

Example - assignment of privileges to users

The following table shows two examples of assignment of privileges to users:

Row	Privilege	User	Deny	Admin	Four-eyes
1	Create cash account	U _x	False	False	False
2	Create cash account	U _y	True	True	False

Table 109 - Assignment of privileges to users

For each assignment of a privilege to a user, three additional attributes define the features of such assignment.

According to row 1, the privilege to create cash accounts is assigned to user U_x:

- | without deny, i.e. user U_x can create cash accounts according to the two-eyes principle (as the privilege is assigned without four-eyes)
- | without admin, i.e. user U_x cannot grant the privilege to create cash accounts to other roles and users

Similarly, row 2 stipulates that the privilege to create cash accounts is assigned to user U_y:

- | with deny, i.e. user U_y cannot create cash accounts
- | with admin, i.e. user U_y can grant the privilege to create cash accounts to other roles and users of the same party, according to the two-eyes principle or to the four-eyes principle (as the privilege is assigned without four-eyes)

As a whole, this configuration results in a full segregation of duties between business users and access rights administrators. In fact, user U_x can create cash accounts, but without having the possibility to grant the same privilege to any other user. Vice versa, user U_y can configure this privilege for other users, but without having the possibility to use it.

Example - assignment of privileges to parties

The following table shows one example of assignment of a privilege to a party:

Privilege	Party	Deny	Admin	Four-eyes
Cash account reference data query	Payment bank A	False	True	False

Table 110 - Assignment of privileges to parties

For each assignment of a privilege to a party, three additional attributes define the features of such assignment. In this example, the privilege to query cash accounts is assigned to the payment bank A:

- | without deny, i.e. party administrators of the payment bank A can grant the privilege to query cash accounts to other roles and users of the same party
- | with admin, i.e. party administrators of the payment bank A can grant the privilege to query cash accounts to other parties

The four-eyes attribute is set to false but it is not relevant for this example, as the privilege refers to a query.

Revoking privileges

Privileges can be revoked from roles, users and parties. When revoking a privilege from the user, this just results in the removal of the privilege from the list of privileges linked to the user. When revoking a privilege from a role, this results in the removal of the privilege from the list of privileges linked to the role. Consequently, all the users and parties linked to the role are not linked anymore to the privilege, with immediate effect. When revoking a privilege from a party, CRDM applies a cascade effect. This results in the removal of the privilege

- | from the list of privileges linked to the party and
- | from the list of privileges linked to all the roles and users of the party.

The following table shows all the possible scenarios for revoking privileges that are allowed in CRDM, their link with the cascade process and how party administrators of CBs can ensure that all the privileges revoked from one of their parties are revoked also from all the users of the same party.

Function	From	Cascade	Propagation to user
Revoke privilege	User	n/a	As the grantee is already a user, there is no need to trigger any cascade process.
Revoke privilege	Role	n/a	<p>If the party administrator of the payment bank granted a privilege included in the role directly to other users of the payment bank, then the removal of this privilege from the role would not revoke the same privilege from these users.</p> <p>In fact, when revoking a privilege from a role, CRDM does not trigger the cascade process as this may result in unintended removal of privileges from the users of the payment bank. For example, even a simple movement of a privilege between two roles assigned to the same payment bank (i.e. revoking the privilege from the first role and granting it to the latter) would imply the removal of the same privilege from all the users of this payment bank and this would oblige the party administrator of the payment bank to grant again this privileges to all the impacted users.</p> <p>In order to ensure that the relevant privilege is revoked also from the users of the payment bank (if this is the intended goal), the party administrator of the CB should grant directly this privilege to the payment bank and then revoke it, as this triggers the cascade process related to the revoke privilege function from party (see next row of this table).</p>
Revoke privilege	Party	Yes	CRDM triggers automatically the cascade process, which ensures that privileges revoked from a party are also revoked from all the users and roles of the same party.

Table 111 - Cascade process when revoking privileges

The cascade process is automatically triggered in a deferred mode one time per business day. However, in case the party administrator needs the cascade process to take place immediately, this can be achieved by contacting the operator, as the operator can trigger this process on demand also intraday.

Example – revoke privilege cascade effect

The following table shows one example of assignment of the same privilege to a party and its users:

Privilege	Grantee	Deny	Admin	Four-eyes
Cash account reference data query	Payment bank A	False	True	False
Cash account reference data query	User A1	False	True	False
Cash account reference data query	User A2	False	False	False

Table 112 - Assignment of privilege to party and users

Users A1 and A2 belong to payment bank A. If payment bank A’s CB wants to revoke the privilege “cash account reference data query” from all users of payment bank A, it just needs to revoke it from payment bank A at party level. The cascade process then automatically revokes it from users A1 and A2.

6.1.3.2.3 Configuration of roles

Links between roles

CRDM supports a role-based access control (RBAC) model. This results in the possibility to inherit privileges from one or more roles.

Granting roles

Roles can be granted to users and parties. When granting a role to a user, the grantee user immediately inherits all the privileges of the granted role, i.e. all the privileges linked to the granted role. When granting a role to a party, the grantee party immediately inherits all the privileges of the granted role, i.e. all the privileges linked to the granted role.

Revoking roles

Roles can be revoked from users and parties. When revoking a role from a user, this user immediately loses all the privileges of the revoked role, i.e. all the privileges linked to the revoked role. When revoking a role from a party, this party immediately loses all the privileges of the revoked role, i.e. all the privileges linked to the revoked role. Both when revoking roles from users and from parties, CRDM does not apply a cascade effect. The following table shows all the possible scenarios for revoking roles that are allowed in CRDM, their link with the cascade process and how party administrators of CBs can ensure that all the roles revoked from one of their parties (and all the privileges included in these roles) are revoked also from all the users of the same party.

Function	From	Cascade	Propagation to user
Revoke role	User	n/a	As the grantee is already a user, there is no need to trigger any cascade process.
Revoke role	Party	n/a	<p>If the party administrator of the payment bank granted the role (or a privilege included in the role) directly to other users of the payment bank, then the removal of this role from the party would not revoke the same role (or the privilege included in the role) from these users.</p> <p>In fact, when revoking a role from a party, CRDM does not trigger the cascade process as this may result in unintended removal of roles (or privileges) from the users of the payment bank.</p> <p>In order to ensure that the relevant role is revoked also from the users of the payment bank, the party administrator of the CB should revoke all the privileges included in the role from the role itself and then delete the role. It should be noted that this approach can be applied without unintended side effects on other payment banks only if the role was specifically created for (and assigned to) the relevant payment bank only, otherwise the procedure just described would also have an effect on all payment banks (and on all their users) being granted with the same role.</p> <p>Furthermore, in order to ensure that any privilege belonging to the role and that was granted directly to users of the payment bank is also revoked from these users, the party administrator of the CB should grant directly this privilege to the payment bank and then revoke it, as this triggers the cascade process related to the revoke privilege function from party (see table Table 111 - Cascade process when revoking privileges [256]).</p>

Table 113 - Cascade process when revoking roles

Example – procedure to revoke role from all users of a party

The following table shows one example of assignment of the privileges to a role, of the role to a User and of one of the privileges it contains directly to another user:

Row	Role	Privilege	Deny	Admin	Four-eyes
1	Party management	Create party	False	True	True
2	Party management	Update party	False	True	True
3	Party management	Delete party	False	True	True
4	Party management	Party reference data query	False	True	Not relevant

Table 114 - Assignment of privileges to roles

Row	Role	Privilege	Deny	Admin	Four-eyes
1	Party management	User A1	False	True	True

Table 115 - Assignment of roles to users

Row	Role	Privilege	Deny	Admin	Four-eyes
1	Update party	User A2	False	True	True

Table 116 - Assignment of privileges to users

Assuming users A1 and A2 belong to the same payment bank party and the responsible CB wants to make sure they both do not use any of the privileges included in role party management, the CB administrator should

- | revoke all privileges from the role, then delete the role. This renders the role useless and prevents other party administrators from granting privileges to it again for any reason. As a consequence, user A1 can no longer use the privileges contained in the role.
- | grant the “update party” privilege to the payment bank to which users A1 and A2 belong, then revoke it. This triggers the cascade process for revoking privileges, which results in privilege “update party” being revoked automatically from user A2, who had it granted directly.

6.1.3.3 Access rights configuration process

As described in chapter [Configuration of privileges](#) [250], before the party administrator of a given party can grant a privilege to a user of the same party, the same privilege has to be granted to the same party, so that it becomes available to the party administrator(s) of the party.

On this basis, the following diagram illustrates the steps needed for granting a given privilege P to the users of a CB (identified as party A in the diagram).

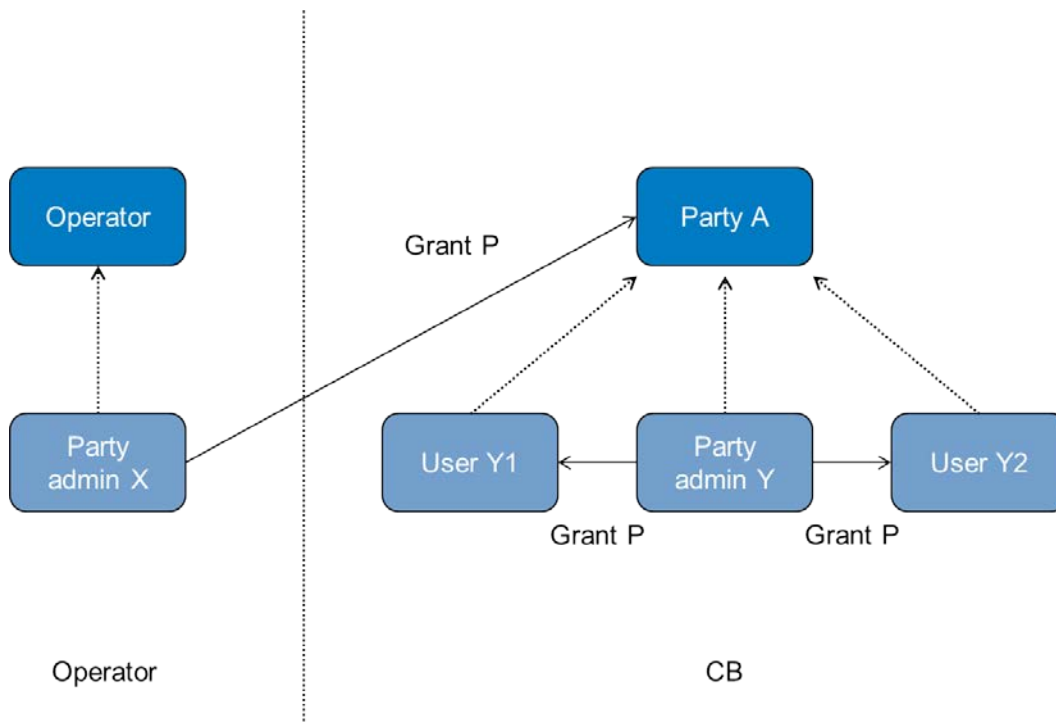


Figure 49 - Access rights configuration process (A)

The diagram shows that the two required steps are as follows.

- | User X, as a party administrator of the operator, grants the privilege P to the party A;
- | User Y, as a party administrator of the party A, grants the privilege P to all the relevant users (in this case, users Y1 and Y2).

The same process applies when a CB needs to configure access rights for their payment banks. The following diagram illustrates all the steps needed for granting a given privilege P to the users of a payment bank (party B in the diagram), via the relevant CB (party A in the diagram).

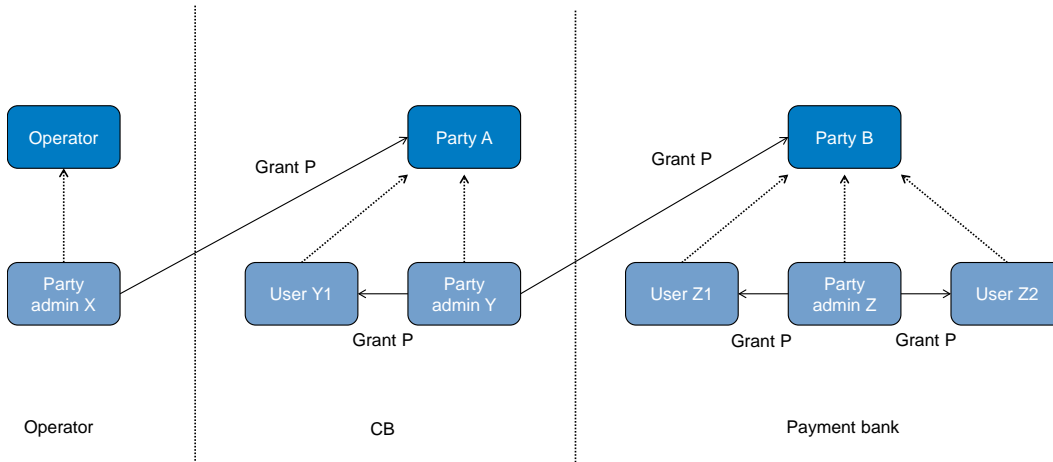


Figure 50 - Access rights configuration process (B)

The diagram shows that the three required steps are as follows.

- | User X, as a party administrator of the operator, grants the privilege P to the party A (i.e. to a CB).
- | User Y, as a party administrator of the party A, grants the privilege P to the party B (i.e. to a payment bank).
- | User Z, as a party administrator of the party B, grants the privilege P to the relevant users (in this case users Z1 and Z2).

In addition, the diagram shows that user Y, as a party administrator of the party A, can also grant the privilege P to the user Y1, as this user belongs to the same party.

These two examples illustrate that the access rights configuration process in the CRDM consists in two main tasks:

- | configuration of access rights at party level
- | configuration of access rights at user level

As stated in chapter [Configuration of privileges](#) [▶ 250], the above process is not directly applicable for TIPS privileges; in this case privileges have to be granted to roles in the first place and then roles can be granted to parties and users. For details on the configuration of roles see chapter [Configuration of roles](#) [▶ 257].

6.1.3.3.1 Configuration of access rights at party level

This task consists of the assignment of the relevant set of roles and privileges to a given party in CRDM. A party administrator of the operator performs this task for the configuration of access rights of CBs.

The following diagram shows an example in which the party administrator of the operator grants to all the CBs the same set of roles and privileges. This set includes all the privileges needed by the CBs and all the privileges needed by the payment banks.

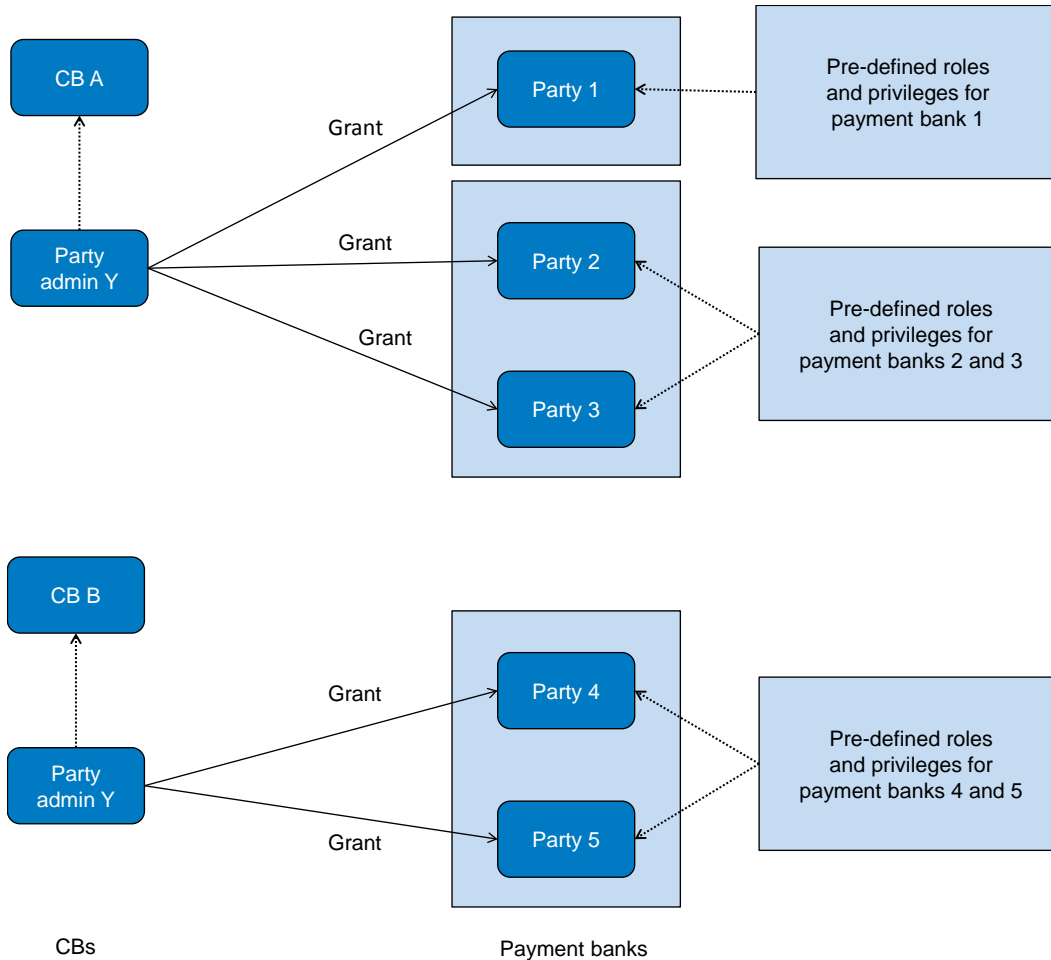


Figure 51 - Example - configuration of access rights at party level by the operator

A party administrator of each CB assigns the relevant set of roles ¹⁴ and privileges to all its payment banks. In this example the party administrator of a CB A configures the relevant access rights for three payment banks party 1, party 2 and party 3. This results in two different set of roles and privileges, the first one being granted to the payment bank party 1 only, the latter being assigned to both payment banks party 2 and party 3. Similarly, the party administrator of a CB B assigns the relevant access rights to two payment banks party 4 and party 5, this task resulting in the configuration of the same set of access rights for both payment banks party 4 and party 5.

¹⁴ New roles can only be created and maintained by the operator and CB parties. Payment banks can only grant/ revoke roles that have previously been granted to them by their CBs.

6.1.3.3.2 Configuration of access rights at user level

After the configuration of access rights at party level has been set-up for a given party, its party administrator(s) can perform the configuration of access rights at user level, in order to assign the appropriate roles and privileges to all the users of the given party.

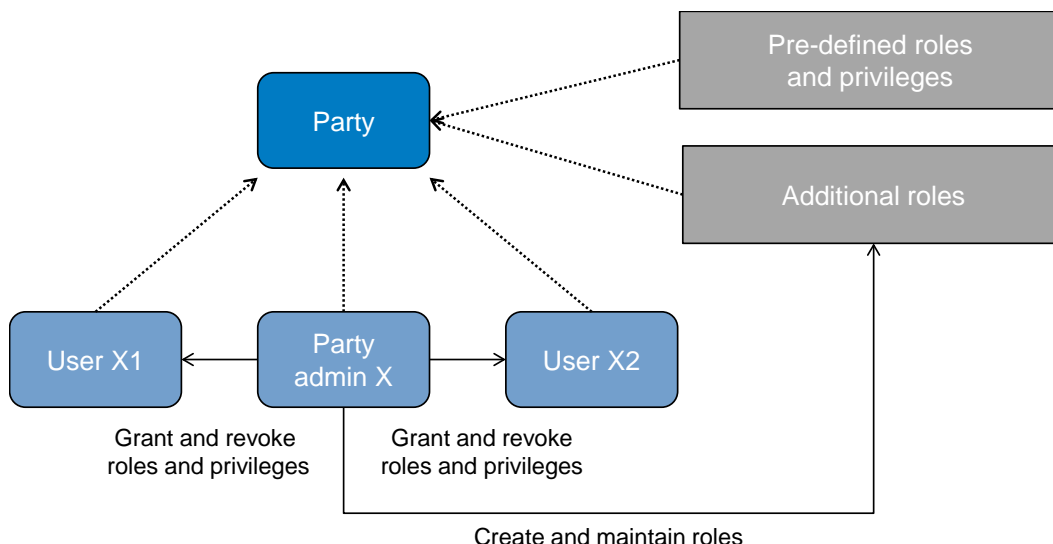


Figure 52 - Configuration of access rights at user level

The above diagram shows that the party administrator(s) can set-up the appropriate access rights configuration for the users of the same party

- | by possibly creating and maintaining ¹⁵ additional roles, besides the ones previously granted at party level ¹⁶.
- | by granting (and revoking) the (default and additional) roles and the (default) privileges to the users of the same party.

6.1.4 Message subscription

6.1.4.1 Message subscription configuration

CBs can configure, for payment banks they are responsible for, the specific set of messages they want to receive from individual services and components.

Each message subscription rule set is defined by the following elements:

¹⁵ New roles can only be created and maintained by the operator and CB parties. Payment banks can only grant/revoke roles that have previously been granted to them by their CBs.

¹⁶ These additional roles can only be granted with available privileges, i.e. privileges previously granted at party level.

- | the name and the description of the message subscription rule set
- | a validity period, specified by a mandatory initial date of validity and an optional final date of validity
- | a set of subscribing parties to which the relevant service or component sends all the messages matching the rule set
- | a set of rules defining the criteria according to which the relevant service checks whether a message has to be sent or not.

These criteria are expressed on the basis of a pre-defined set of parameter types. Each rule is assigned a validity period, specified by a mandatory initial date of validity and an optional final date of validity. The validity period of a rule cannot exceed the validity period of the message subscription rule set it belongs to, i.e. the validity period of a rule cannot start before or end after the validity period of the relevant message subscription rule set.

If deemed necessary, CBs can decide to hand over the control to their payment banks by granting them the privilege for message subscription configuration (for more information on privilege granting see chapter [Access rights \[235\]](#)).

6.1.4.2 Message subscription parameter types

The table below describes the exhaustive list of parameter types that CBs can use for configuring their message subscription rule sets.

Parameter type	Description
Message type	<p>It specifies the type of message, depending on the service. Possible values depend on the specific service for which messages are being subscribed and are listed below.</p> <p>For TIPS:</p> <ul style="list-style-type: none"> BankToCustomerDebitCreditNotification <p>For RTGS:</p> <ul style="list-style-type: none"> ResolutionOfInvestigation BankToCustomerDebitCreditNotification PaymentStatusReport
Cash account	It specifies the cash account for which relevant messages shall be sent.

Table 117 - Message subscription parameter types

6.1.4.3 Message subscription examples

The above described message subscription configuration is illustrated below.

Example - subscribing for liquidity transfer credit notification

This example is about a message subscription configuration which allows a payment bank A to receive credit notifications related to settlement of liquidity transfers.

This message subscription configuration must be valid as of 1 July 2019. The general features of the new message subscription rule set for the payment bank A, i.e. the rule set name, the starting validity date and the relevant interested party can be specified as follows.

Message subscription rule set	
Name	CREDIT_NOTIFY_ACCOUNT_A
Description	Receive credit notifications for account A
Interested party	Payment Bank A
Valid from	1-July-2019
Valid to	-
Positive/negative	Positive

Table 118 - Definition of a new message subscription rule set

The rule that the payment bank A needs to specify for itself in order to fulfil the requirements described before is as follows:

Rule set	Valid from	Valid to	Message type	TIPS account
Rule 1	2019-07-01	-	BankToCustomerDebitCreditNotification	Account A

Table 119 - Definition of the rules for a new message subscription rule set

6.1.5 Common reference data maintenance process

6.1.5.1 Common reference data objects

Duly authorised users manage common reference data by creating and maintaining common reference data objects. A common reference data object is a set of logically related, self-consistent information. Parties and cash accounts are examples of common reference data objects. The following table provides the exhaustive list of common reference data objects defined in CRDM and the CRDM Actors that are responsible for their management, i.e. for creating and maintaining them.

Area	Object	Responsible CRDM Actors ^{17 18}
Party	Party	Operator, CB
	Party service link	Operator, CB
	Banking Group	CB
	MFI	CB
Cash account	Cash account	All ¹⁹
	Limit	Payment bank
	Authorised account user	Payment bank
	Account Monitoring Group	Payment bank
	Standing liquidity transfer order	Payment bank
	Liquidity Transfer Group	Payment bank
	Direct debit mandate	Payment bank
	Standing order for limit	Payment bank
	Standing order for reservation	
	Access rights management	User
Role		Operator, CB
Privilege		Operator
Certificate DN		All
User-certificate DN link		All
Role user ²⁰		All
Role party ²¹		Operator, CB
Grantee privilege ²²		Operator, CB, payment bank
Message sub- scription configu-	Message subscription rule	CB, payment bank
	Message subscription rule set	CB, payment bank

17 "All" indicates that all types of CRDM actors (operator, CBs, payment banks) have the ability to manage the object type.

18 The actor types listed for each function refer to the default responsible actor in normal operating conditions. However, it is possible for the operator to act on behalf of CBs (and of payment banks, upon request of the relevant CB) and for the CBs to act on-behalf of their payment banks, under well-defined contingency scenarios.

19 The cash account object includes both TIPS accounts and TIPS CMBs. In this respect, payment banks may only create and maintain TIPS CMBs, whereas CBs create and maintain TIPS accounts and may create and maintain TIPS CMBs on behalf of their payment banks.

20 This object is related to the granting/revoking of roles to/from users.

21 This object is related to the granting/revoking of roles to/from parties.

22 This object is related to the granting/revoking of privileges to/from roles, parties and users.

Area	Object	Responsible CRDM Actors ^{17 18}
ration		
Network configuration	DN BIC routing	Payment bank
	Network service	Operator
	Technical address network service link	Operator, CB
Report configuration	Report configuration	Payment bank
Restriction type management	Restriction type	Operator
Billing configuration	Service item	Operator
Configuration parameters	Country	Operator
	Currency	Operator
	Currency service link	Operator
	System entity	Operator
	BIC directory	Operator
	Service	Operator

Table 120 - Common reference data objects

A common reference data object consists of one or more classes of information. For example, a party is a common reference data object, consisting of the following classes of information.

- | party
- | party code
- | party name
- | party address
- | party technical address

Each class of information includes a defined set of attributes. For example, the class of information party name of the common reference data object party includes the following attributes.

- | the long name of the party
- | the short name of the party
- | the starting validity date of the party name

The CRDM common component provides functions to maintain all common reference data objects (see chapter [Reference data maintenance types](#) [268]). Each maintenance operation on a common reference data object results in a new version of the same object. Each version of a common reference data object is

called a revision of the object. Consequently, at any point in time, CRDM stores one or many revisions of each common reference data object, more precisely only one revision for newly created objects that were never maintained after their creation and N revisions for objects that were maintained N-1 times after they were created. The first revision of each common reference data object includes all the attribute values provided at creation time. After that, each maintenance request successfully processed creates a new revision for the object. This means that each revision may entail changes of many attributes of the same common reference data object at the same time. A new revision is also created when deleting and restoring a common reference data object.

Some classes of information are subject to data history, i.e. classes of information having multiple occurrences with continuous and non-overlapping validity periods. For example, the classes of information party name and party code of the common reference data object party can be subject to data history. In fact, they include a valid from attribute which determines the valid value of these classes of information at any given point in time.

6.1.5.2 Reference data maintenance types

CRDM allows a duly authorised user to perform the following types of reference data maintenance operations on common reference data objects.

- l **create:** creates a new common reference data object.
- l **update:** updates an already existing common reference data object. It is possible, with a single update, to create, update or delete one or many classes of information of a common reference data object at the same time.
- l **delete:** it deletes an already existing common reference data object. Deletion is always logical and not physical. Physical deletion is performed automatically by CRDM when performing the purge process following the archiving process (see chapter [Common reference data archiving and purging \[273\]](#)).
- l **restore**²³: it reactivates a previously deleted common reference data object, i.e. it updates the status of this object from deleted to active.

Besides these operations, CRDM provides some specific types of reference data maintenance operations for the configuration of access rights (see chapter [Access rights \[235\]](#) for a detailed description of these operations).

CRDM allows all reference data maintenance types on all reference data objects in U2A mode, whereas it allows them only on a subset of reference data objects through the DMT and A2A mode respectively. The following tables show the exhaustive list of all the available reference data maintenance types that are possible in the DMT and in A2A mode.

23 This function is available in U2A mode only and it is granted, for each object, with the system privilege that allows deleting the same object as well.

Area	Object	DMT function
Party data management	Party	Create
	Technical address network service link	Create
Cash account data management	Cash account	Create
	Authorised account user	Create
	Limit	Create
Access rights management	User	Create
	Role	Create, grant
	Privilege	Grant
	Certificate DN	Create
	User-certificate DN link	Create
Message subscription configuration	Message subscription rule set	Create
	Message subscription rule	Create
Report configuration	Report configuration	Create

Table 121 - Management of reference data objects in DMT

Area	Object	DMT function
Party data management	Party	Create, update, delete
Cash account data management	Cash account	Create, update, delete
	Liquidity transfer order	Update, delete
	Limit	Update, delete

Table 122 - Management of reference data objects in A2A mode

6.1.5.3 Validity of common reference data objects

Some common reference data objects include attributes limiting the validity period of these objects. For example, each party service link, which defines the participation of a given payment bank in a specific service, common component or back-office application, includes two attributes specifying the date from which and the date to which the link is valid, i.e. the period in which said payment bank can operate in that service, common component or back-office application. Between the creation date and the deletion date of the link, but outside the validity period just defined, the payment bank is not allowed to operate in the service, even though it is active in CRDM repository and it can be queried and maintained by a duly authorised user.

CRDM common component makes a distinction between the following two categories of common reference data objects.

- common reference data objects with unlimited validity period
- common reference data objects with limited validity period

The following table shows the exhaustive list of all the common reference data objects with unlimited validity period.

Area	Object
Party	Banking Group
	MFI
Cash account	Account Monitoring Group
	Liquidity Transfer Group
Access rights management	User
	Role
	Privilege
	Certificate DN
	User-Certificate DN link
	Role user link
	Role party link
Network configuration	Privilege role link
	Network service
Configuration parameters	Technical address network service link
	Country
	Currency
	Currency service link
	System entity
	Service
	Currency service link

Table 123 - Common reference data objects with unlimited validity period

This type of common reference data object starts being valid in CRDM immediately after it has been created. Similarly, a common reference data object with unlimited validity period may be immediately updated or deleted by a duly authorised user. However, in both cases the reference data change, i.e. the creation of a new

object or the update or deletion of an already existing object is made effective in the relevant component or service only by means of the daily reference data propagation process.

Regardless of the way common reference data object with limited validity period are propagated to the relevant component or service, between the creation date and the deletion date of this object, it is active in the CRDM common component and it can be queried and maintained by a duly authorised user.

Common reference data objects with limited validity period can be updated either intraday, i.e. while they are in their validity period or as of a future date, i.e. before they become valid.

The following table shows the exhaustive list of all the common reference data objects with limited validity period, with the columns on the right specifying the possible maintenance operations depending on the validity period.

Area	Object	Creation	Update	Deletion
Party	Party	Validity date may take the value of the current date	May take effect on the current date ²⁴	May be performed only on objects that are not valid on the current date
	Party service link	Validity date may take the value of the current date	May take effect on the current date	May be performed only on objects that are not valid on the current date
Cash account	Cash account	Validity date may take the value of the current date	May take effect on the current date	May be performed only on objects that are not valid on the current date
	Standing liquidity transfer order	Validity date may take the value of the current date	May take effect on the current date	May be performed only on objects that are not valid on the current date
	Standing order for reservation	Validity date may take the value of the current date	May take effect on the current date	May be performed only on objects that are not valid on the current date
	Direct debit mandate	Validity date may take the value of the current date	May take effect on the current date	May be performed only on objects that are not valid on the current date

24 This is not applicable to the party code, which cannot be updated if it is currently active.

Area	Object	Creation	Update	Deletion
				date
	Authorised account user	Validity date may take the value of the current date	May take effect on the current date	May be performed only on objects that are not valid on the current date
	Floor/ceiling	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.
Message subscription	Message subscription rule set	Validity date may take value of the next business day at the earliest	May take effect only as of a future date	May be performed only on objects that are not valid on the current date
	Message subscription rule	Validity date may take value of the next business day at the earliest	May take effect only as of a future date	May be performed only on objects that are not valid on the current date
Report configuration	Report configuration	Validity date may take value of the next business day at the earliest	May take effect only as of a future date	May be performed only on objects that are not valid on the current date
Restriction type management	Restriction type	Validity date may take value of the next business day at the earliest	May take effect only as of a future date	May be performed only on objects that are not valid on the current date
Network configuration	DN-BIC routing	Validity date may take the value of the current date	May take effect on the current date	May be performed only on objects that are not valid on the current date
Configuration parameters	BIC directory	Validity date may take the value of the current date	May take effect on the current date	May be performed only on objects that are not valid on the current date

Table 124 - Common reference data objects with limited validity period ²⁵

For parties and cash accounts the validity period is defined by an opening date and a closing date attribute. Between these two dates the common reference data object, i.e. the party or the cash account, is valid, meaning that components or services can use it for processing (e.g. for settlement purposes). Outside this period, the common reference data object can only be queried or maintained in the CRDM common component by a duly authorised user.

6.1.5.4 Common reference data archiving and purging

CRDM archives new reference data and their changes three calendar months after they were created or changed. CRDM purges, i.e. physically deletes reference data from the production data base three calendar months after they were deleted. For example, a party has to be deleted before CRDM can purge it. This implies that a party is never purged, unless a duly authorised user makes the decision to delete it.

The following example illustrates how CRDM archives and purges the different revisions of a generic common reference data object.

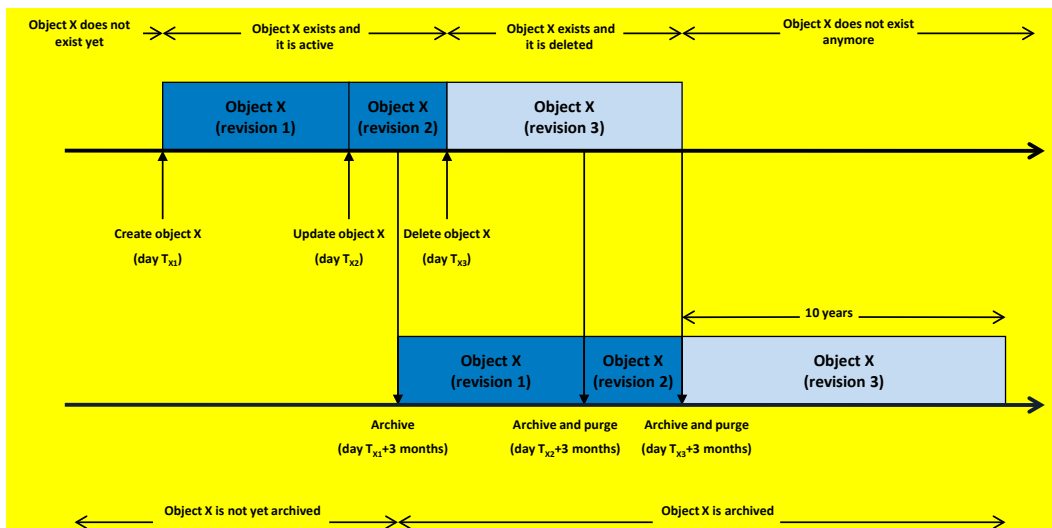


Figure 53 - Example - archiving and purging after deletion of a common reference data object

In this example, a duly authorised user creates intra-day, on business day T_{X1} , a common reference data object X. This results in the creation of the first revision of the object X.

²⁵ In the following table, the columns 'creation/update/deletion' clarify whether it is possible to perform a given maintenance operation on each object with immediate effect on CRDM. For example, if a user updates an object on which updates "may take effect on the current date", they are able, should they wish to do so, to perform changes that become immediately valid in CRDM. On the contrary, if the update "may take effect only as of a future date" then it is not possible to perform intraday changes on the object. The possibilities described in the table represent the level of flexibility offered to the user. Within these limitations, the user decides exactly when a specific modification should take effect.

During business day T_{x2} (with $T_{x2} < T_{x1} + \text{three calendar months}$) a duly authorised user updates the common reference data object X changing one (or many) of its attribute(s). This results in the creation of a new revision (2) for X.

On business day $T_{x1} + \text{three calendar months}$, the archiving process copies the first revision of the common reference data object X into the archiving data base. It is worth mentioning that

- CRDM does not purge the archived revision, as it still refers to a period of time that expired on T_{x2} , i.e. since less than three calendar months,
- CRDM does not archive the second revision of the common reference data object X, as it was created on T_{x2} , i.e. since less than the duration of the retention period.

During business day T_{x3} (with $T_{x3} < T_{x2} + \text{three calendar months}$), a duly authorised user deletes the common reference data object X. This results in the creation of a new revision (3) for the same object.

On business day $T_{x2} + \text{three calendar months}$, the archiving process copies the second revision of the common reference data object X into the archiving data base. In this case

- CRDM does not purge this second revision, as it still refers to a period of time that expired on T_{x3} , i.e. since less than three calendar months,
- CRDM does not archive the third revision of the common reference data object X, as it was created on T_{x3} , i.e. since less than three calendar months,
- CRDM purges the first revision of the common reference data object X, as it refers to a period of time that expired exactly since three calendar months.

Finally, on business day $T_{x3} + \text{three calendar months}$, the archiving process copies the third and final revision of the common reference data object X into the archiving data base. On the same day, just after the archiving process is successfully performed, CRDM purges the common reference data object X, by physically deleting the last two revisions of the object X that are still present in the production data base.

From this moment on, all revisions of the common reference data object X are available only in the archiving data base, where the archiving common component keeps them for a period of ten years.

6.1.5.5 Lifecycle of common reference data objects

This section puts together all the concepts described so far and provides a general description of the lifecycle of common reference data objects.

Lifecycle of common reference data objects with unlimited validity period

The following diagram illustrates the lifecycle of a common reference data object with unlimited validity period both in the production data base and in the archiving data base.

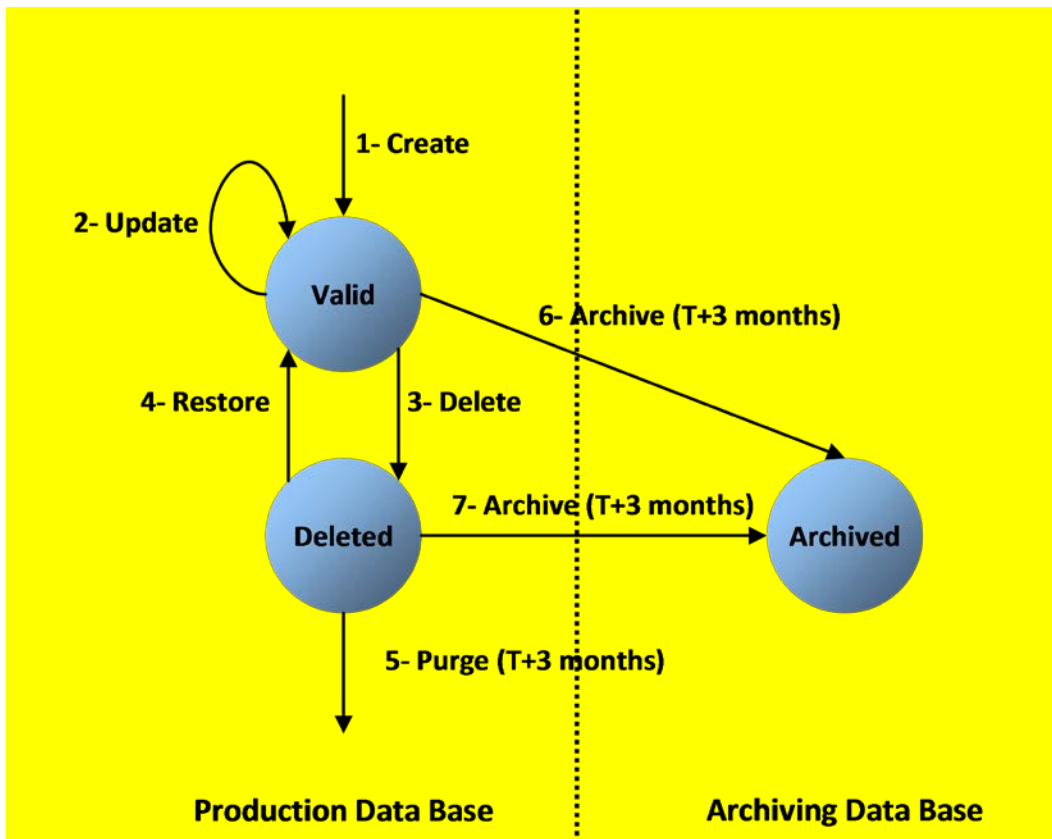


Figure 54 - Lifecycle of common reference data objects with unlimited validity period

When a duly authorised user submits a reference data maintenance instruction to CRDM to create a common reference data object with unlimited validity period, CRDM processes it and, in case of successful processing, it creates the relevant object. This object is valid and it exists in the production data base only (transition 1).

From this moment on, a duly authorised user may submit to CRDM one or many reference data maintenance instructions to update the common reference data object. Regardless of the result of CRDM processing, i.e. whether the reference data maintenance instruction is successfully or unsuccessfully processed, the common reference data object remains valid (transition 2).

When a duly authorised user submits to the CRDM reference data maintenance instruction to delete a common reference data object, the CRDM processes it and, in case of successful processing, it deletes the relevant object. This object is logically deleted (transition 3), even if it is still physically present in the production data base.

From this moment on and within a period of three calendar months, if a duly authorised user submits to CRDM a reference data maintenance instruction to restore a previously deleted common reference data

object, CRDM processes it and, in case of successful processing, it restores the relevant object. As a result, the object becomes valid again (transition 4).

Three calendar months after a common reference data object is deleted, CRDM physically deletes it from the production data base. This results in the object being purged by the production data base (transition 5), i.e. it exists only in the archiving data base.

Three calendar months after a common reference data object is created, updated or deleted, CRDM copies the revision of the common reference data object resulting from this reference data maintenance instruction from the production data base to the archiving data base. As a result the common reference data object is both in the production data base and archived in the archiving data base, in case it was created or updated, or only in the archiving data base, in case it was deleted (transitions 6 and 7).

Lifecycle of common reference data objects with limited validity period

The following diagram illustrates the lifecycle of a common reference data object with limited validity period both in the production data base and in the archiving data base.

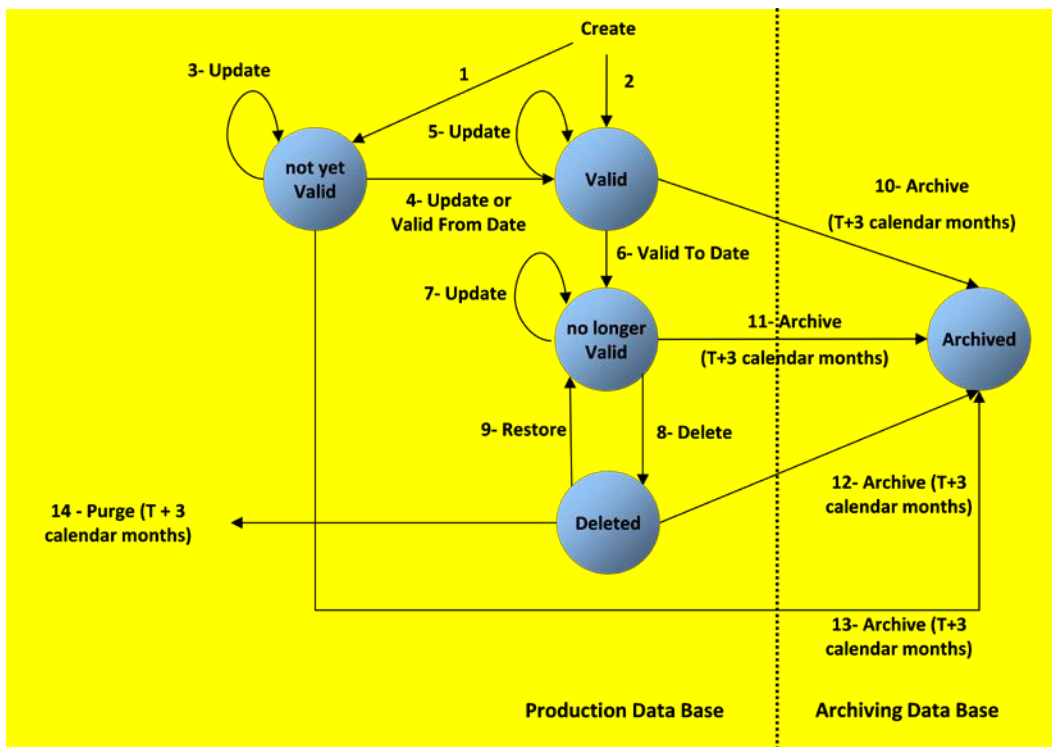


Figure 55 - Lifecycle of common reference data objects with limited validity period

When a duly authorised user submits to CRDM a reference data maintenance instruction to create a common reference data object with limited validity period, CRDM processes it and, in case of successful processing, it creates the relevant object. This object is either valid or not yet valid, depending on the starting date of its validity period, and it exists in the production data base only (transitions 1 and 2).

From this moment on, a duly authorised user may submit to the CRDM one or many reference data maintenance instructions to update the common reference data object. If the object is valid, then it remains valid, regardless of the result of CRDM processing, i.e. whether the reference data maintenance instruction is successfully or unsuccessfully processed (transition 5). If the object is not yet valid, two sub-cases are possible.

If the reference data maintenance instruction also updates the starting date of the validity period to the current business date and it is successfully processed, then the common reference data object becomes valid (transition 4).

In all other cases, whether the reference data maintenance instruction is successfully or unsuccessfully processed, the common reference data object remains not yet valid (transition 3).

A common reference data object becomes valid from the starting business date of the validity period (transition 4).

A common reference data object is valid until the EoD of the final date of the validity period (transition 6). As far as TIPS is concerned, this implies that the object is valid until TIPS receives from the RTGS system the message notifying the first business day greater than the final date of the validity period.

When a duly authorised user submits to CRDM a reference data maintenance instruction to delete a common reference data object, CRDM processes it and, in case of successful processing, it deletes the relevant object. This object is logically deleted (transition 8), even if it is still physically present in the production data base.

From this moment on and within a period of three calendar months, if a duly authorised user submits to the CRDM a reference data maintenance instruction to restore a previously deleted common reference data object, CRDM processes it and, in case of successful processing, it restores the relevant object. As a result, the object becomes no longer valid again (transition 9).

Three calendar months after a common reference data object has been deleted, CRDM physically deletes it from the production data base. This results in the object being purged by the production data base (transition 14), i.e. it exists only in the archiving data base.

Three calendar months after a common reference data object is created, updated or deleted, CRDM copies the revision of the common reference data object resulting from this reference data maintenance instruction from the production data base to the archiving data base. As a result the object is both in the production data base (as a not yet valid, valid, no longer valid or deleted object) and in the archiving data base archived, in case it was created or updated, or only in the archiving data base, in case it was deleted (transitions 10, 11, 12 and 13).

6.1.5.6 Common reference data propagation

CRDM allows users to configure reference data to be used in the local reference data management (LRDM) of other TARGET Services or components (e.g. TIPS, CLM and RTGS).

Data set-up in CRDM is propagated to other services, common components or back-office applications on a regular basis, typically once a day, at a present time before the change of business date. If needed, participants can request an ad-hoc propagation to be run at different times of day for a specific service, common component or back-office application. There is no technical limit on the number of times a data propagation can run during a given business date.

No data propagation flow exists from TIPS, CLM and RTGS to CRDM. Since CRDM contains data belonging to different services, common components or back-office applications, specific segregation principles are put in place to make sure that relevant data is made available in each service, common component or back-office application depending on the individual needs. In this respect certain objects (e.g. country, currency) are fully shared – they are made available to every service, common component or back-office application without distinction. Other objects are service-specific, and are made available in full to a single service (example includes Banking Group for CLM). Finally, certain objects are shared among multiple services, but the data is segregated and made available in a given service based on the values of specific attributes that link each instance to a specific service, either directly or indirectly. Examples of this type of objects include party and cash account.

The following table lists the possible CRDM reference data objects and their relevance for each service, as well as the data segregation principles defining which instances are propagated to which service.

Area	Object	CLM	RTGS	T2S	TIPS	Segregation principles
Party	Party	X	X	X	x	All data is available in T2S. Parties with a party service link to CLM, RTGS or TIPS are available in that service/component.
	Party service link					Only relevant for CRDM; defines the availability of party data for a given service.
	Banking Group	X				All data is available in CLM.
	MFI		X			All data is available in CLM.
Cash account	Cash account	X	X	X	X	Data is available in different services depending on the cash account type attribute; each possible value of this attribute identifies a type of cash account used by a single service.
	Authorised account user				X	All data is available in TIPS.
	Account Monitoring	X	X			All data is available in CLM.

Area	Object	CLM	RTGS	T2S	TIPS	Segregation principles
	Group					
	Standing liquidity transfer order	X	X	X		Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Liquidity Transfer Group	X	X			Data is available in different services depending on the cash account type attribute of the cash accounts it refers to.
	Limit	X	X	X	X	Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Direct debit mandate	X	X			Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Standing order for limit		X			All data is available in RTGS.
	Standing order for reservation	X	X			Data is available in different services depending on the cash account type attribute of the cash accounts it refers to.
Access rights management	User	X	X	X		All data is available in T2S. Data related to parties with a party service link to CLM or RTGS is available in that service.
	Role	X	X	X	X	All data is available in T2S. Data containing privileges related to CLM, RTGS or TIPS is available in that service.
	Privilege			X		All data is available in T2S. It is not available in other services, but it is used by CRDM to determine the availability of other access rights data in those services. Each privilege includes a link to a

Area	Object	CLM	RTGS	T2S	TIPS	Segregation principles
						single service which defines the service that contains the user function activated by the privilege.
	Certificate DN	X	X	X	X	All data is available in T2S. Data linked to users flagged as main users for TIPS is available in TIPS. Data linked to users under parties with a party service link to CLM or RTGS is available in that service.
	User-certificate DN link	X	X	X	X	All data is available in T2S. Data linked to users flagged as main users for TIPS is available in TIPS. Data linked to users under parties with a party service link to CLM or RTGS is available in that service.
	Role user	X	X	X	X	Data is available in different services depending on the service the privileges contained in the role refer to.
	Role party	X	X	X	X	Data is available in different services depending on the service the privileges contained in the role refer to.
	Grantee privilege	X	X	X	X	Data is available in different services depending on the service the privilege refers to.
Message subscription configuration	Message subscription rule set			X	X	All data is available in T2S. Data containing message subscription rules that reference data from CLM, RTGS or TIPS is available in those services.
	Message subscription rule			X	X	Data is available in different services depending on the underlying reference data objects the rule

Area	Object	CLM	RTGS	T2S	TIPS	Segregation principles
						refers to.
Network configuration	Network service	X	X	X	X	Data is available in different Services based on an attribute that defines a direct reference to a single Service.
	Technical address network service link	X	X	X	X	Data is available in different services depending on the service the related network service refers to.
	DN BIC routing				X	All data is available in TIPS.
Report configuration	Report configuration	X	X	X	X	Data is available in different services depending on the specific type of report being subscribed.
Restriction type management	Restriction type		X	X	X	Data is available in different services based on an attribute that defines a direct reference to a single service.
Billing configuration	Service item					Only relevant for CRDM and Billing.
Configuration parameters	Country	X	X	X	X	All data is available in all services.
	Currency	X	X	X	X	All data is available in all services.
	Currency service link	X	X	X	X	Data is available in different services depending on the service the link refers to.
	System entity	X	X	X	X	All data is available in all services.
	BIC directory	X	X	X	X	All data is available in all services.
	Service					Only relevant for CRDM.

Table 125 - CRDM data segregation per service/component

6.2 DWH

Will be completed in v2.0.

6.2.1 Introduction

6.2.2 Scope of DWH

6.2.3 Access

6.2.3.1 Connectivity

6.2.3.2 Authentication and authorisation

6.2.4 User roles and access rights

6.2.4.1 Overview

6.2.4.2 User rights

6.2.4.3 User profiles

6.2.5 Data warehouse queries and reports

6.2.5.1 Overview

6.2.5.2 Types of queries and reports

6.2.5.3 Predefined queries and reports

6.3 Billing

Will be completed in v2.0.

6.4 Legal archiving

Will be completed in v2.0.

6.5 ESMIG features

6.5.1 ESMIG features overview

The ESMIG infrastructure provides a set of features shared among all the TARGET Services, common components and back-office applications beyond representing a single point of contact with the external networks.

These features, detailed below, belong to two main areas and can be provided by either the network service providers (NSPs) or by the ESMIG component.

- | security, e.g. authentication of the sender and authorisation against a closed group of users
- | message management, e.g. message technical validation and forwarding

6.5.1.1 Authentication of the message sender

The authentication of the message sender is performed by the NSP both at the entry point of the network (by providing to the actors digital certificates needed to access the A2A and U2A messaging services) and at the interface with the TARGET Services, common components and back-office applications through the relevant services provided by the NSP.

The NSP identifies the actor and the TARGET Services, common components and back-office applications every time they open a new session with the NSP's network gateway for A2A traffic. There is no end-to-end session. The NSP transfers the identity of the sender to the receiver, including this information in the network envelope provided to the receiver together with the message. Moreover, the NSP authenticates the actor and the TARGET Services, common components and back-office applications as local message partner every time they open a new session with the NSP's network gateway for A2A traffic exchange.

6.5.1.2 Participation to the Closed Group of Users

Will be completed in v2.0.

6.5.1.3 Validation of the received messages

Will be completed in v2.0.

6.5.1.4 Message forwarding

ESMIG is responsible for forwarding inbound/outbound communication to the right service/NSP. For the inbound path all the messages are passed to the TARGET Services, common components and back-office applications in charge to manage inbound messages. For the outbound path, ESMIG addresses the correct NSP interface among the available ones based on the information available in CRDM database. The reader can refer to the CRDM UDFS for any related additional information.

6.5.2 Access to ESMIG

Will be completed in v2.0.

6.5.2.1 Single access point for the external communication

6.5.2.2 Network agnostic communication

6.5.3 ESMIG Portal

Users of TARGET Services and back-office applications belonging to the appropriate closed group of users, defined and enforced at NSP level, can communicate in U2A mode via a web-based GUI.

Those users are directed to an initial page named ESMIG Portal that ensures proper routing to the web applications according to the access rights profiles.

In particular, the ESMIG Portal shows to the user all the applications he is authorised to access. These applications are linked one-to-one to special system privileges (stored in CRDM) the user has been previously granted with and that are specifically dedicated to those web applications.

When accessing the ESMIG Portal without any authentication, the user is redirected to the IAM page that asks user to authenticate the access validating his distinguished name (DN). Thus, the authentication process, at IAM level, securely associates the DN to the person accessing the system.

After authentication, the person must choose the logical “user” he wants to impersonate, selecting it among a set of user-IDs that have been previously linked to his DN. This selection is done in the ESMIG Portal.

So, the ESMIG portal allows and guides the person accessing the system to:

- I **choose the application** among the authorised applications accessible by at least one user-ID linked to the DN of the user
- I **choose the user** to impersonate when accessing such an application

After this process, the ESMIG Portal redirects to the homepage of the application selected (e.g. CRDM, DMT, etc.).

7 Contingency services

Will be completed in v2.0.

8 Operations and support

Will be completed in v2.0.

8.1 Business application configuration

8.2 Calendar management

8.3 Business day management

8.4 Business and operations monitoring

8.5 Possible actions of operator service desk in ESMIG

8.5.1 Technical monitoring

8.6 Archiving management

8.7 Trouble management

9 Additional information for CBs

9.1 Role of CBs in the RTGS component

General

The processes with RTGS – as a component of T2 Service - take place on a centralised technical platform. Nevertheless, the decentralised nature of the relationship between the CBs and their national banking communities remains unchanged. Indeed, the principle of a centralised platform enables the CBs to provide improved, harmonised and cost-efficient services to their counterparties.

Responsibilities of the CBs

Each CB remains fully responsible for the business relations with its national RTGS Account Holder. Therefore, the system is designed in a "client-based" way in order to meet the administrative and monitoring requirements of the participating CBs.

Tasks of the CBs

In the context of RTGS, the CBs have the following responsibilities:

Administrative tasks	Operational tasks
All contacts and provision of any kind of support to their banking community (credit institutions, ancillary systems)	Inclusion and exclusion of participants of their banking community
	Monitoring of the activities of their banking community
	Initiating payments on behalf of their own or on behalf of their banking community
	Handling of local contingency

Table 126 - Tasks of the CBs

Any payment can be submitted via U2A and A2A. A2A submitting can be done via individual messages or in file format.

CBs as participants

Each CB has also the status of a RTGS account holder. In practical terms, this means that each CB must be:

- | directly addressable in RTGS in order to receive payments from other participants
- | able to submit payments on its own or on behalf of its customers in RTGS

9.2 Settlement of payments - specific functions for CBs

CBs have, apart from the possibility of being a conventional account holder, other more specific functions, only applicable to CBs. This includes queries about activities and balances of the participants of their banking community (please see chapter [Query management - specific functions for CBs](#) [▶ 290]), actions in connection with blocked account holders and the involvement in ancillary system activities. In connection with ancillary systems, the CB can be the holder of several types of accounts including dedicated liquidity accounts, guarantee funds accounts and technical accounts (please see chapter [Settlement of ancillary systems](#) [▶ 140]).

The table below shows a summary of queries and actions in U2A and A2A mode applicable for CBs in RTGS.

Query/ Action	U2A	A2A
Query payments per status for their banking community	x	
Query balances of RTGS DCAs for their banking community	x	
Release/reject debit payment for/from blocked participant	x	
Release/reject credit payment for/from blocked participant	x	
Revoke ancillary system payment instructions in case of disagreement of standard multi-lateral settlement.	x	
Allow backup payments	x	
Allow back value payments	x	
Create broadcasts	x	
Act on behalf of the participants of their banking community in terms of cash transfer processing	x	x
Act on behalf of their capacity as RTGS Account Holder	x	x

Table 127 - Summary of queries and actions in U2A and A2A mode for CBs in RTGS

9.3 RTGS General Ledger

9.3.1 RTGS general ledgers production

During EOD process (see chapter [EoD](#) [▶ 76]), the pending inter-service liquidity transfer processing is finalised. Afterwards RTGS sends one camt.053 message to the CLM containing the “EoD account balance” information related to the business day that just elapsed for all RTGS accounts (see below) of all participants.

The final build and generation of the general ledger files made available to the CBs takes place inside the CLM component. The description of the utilised message type (camt.053) can be found in chapter [BankToCustomerStatement \(camt.053\)](#) [▶ 505].

9.3.2 RTGS general ledgers content

The general ledger file contains all cash accounts and the dedicated transit account held in the RTGS component. For a comprehensive description of accounts, see chapter [Accounts structure and functionalities](#) [▶ 53]. It includes the

- SoD and EoD balances
- Sum of credits and sum of debits

for all included RTGS cash accounts.

In case of CB accounts RTGS provides the total debits and credits on national level and per defined CB (i.e. cross border level). Please see chapter 4.5 “Calculating the positions of CBs vis-à-vis other CBs” in the CLM UDFS for the grouping of data and detailed numeric examples.

RTGS delivers general ledger data that fulfil the following consistency conditions:

- Single balance checks per account: each EoD balance is checked by adding the account turnovers to the SoD balance.
- The sum of all balances of the RTGS accounts (excluding the dedicated transit account itself) must be equal to the balance on the dedicated transit account in absolute value.

9.4 Query management - specific functions for CBs

Dedicated queries are provided to CBs in order to satisfy their specific information needs. Nonetheless the same processing applies to all queries independent of their availability for all parties or limitation to specific parties according to their access rights. Please see chapter [Query management for RTGS](#) [▶ 231]. As regards the processing the description in chapter [Execute query](#) [▶ 348] also applies for all queries irrespective of their access limitations.

Query type	Initiation via GUI (U2A mode)	Initiation via XML message (A2A mode)
Balances of RTGS DCAs for the whole banking community query	X	-
Payments per status for the whole banking community query	X	-

Table 128 - List of CB specific queries

Since these queries are only available in U2A, please refer to the RTGS user handbook for further details about the search parameters and query results.

9.5 Billing - specific functions for CBs

Will be completed in v2.0.

9.6 Contingency - specific functions for CBs

Will be completed in v2.0.

Part II - Dialogue with the RTGS Account Holder

10 Processes with RTGS

The purpose of Part II of this UDFS is to describe the messages that RTGS and the business application of a RTGS Actor exchange for a given business scenario (use case). It provides a formalised description of the A2A interfaces in order to enable RTGS Actors to adapt their business applications to interact with RTGS. Part II of this UDFS does not enter into any description regarding the required behaviour of the business application(s) of RTGS Actors, as this determination remains in the remit of the respective RTGS Actor.

This chapter uses activity diagrams according to Unified Modelling Language conventions to present the processes and actions in RTGS that result in message exchanges with the RTGS Actor(s). This chapter describes the behaviour of RTGS from the perspective of a technically directly connected RTGS Actor. The descriptions in this chapter document only the RTGS activities that process an inbound communication or trigger a possible outgoing communication to a RTGS Actor. The chapter does not document internal processing steps when those processing steps that do not lead to the disclosure of information (sending of messages) to users.

Conventions used

UML activity diagrams describe the interaction between RTGS and the relevant RTGS Actor(s) for every use case, as per examples in diagrams [insert reference here]:

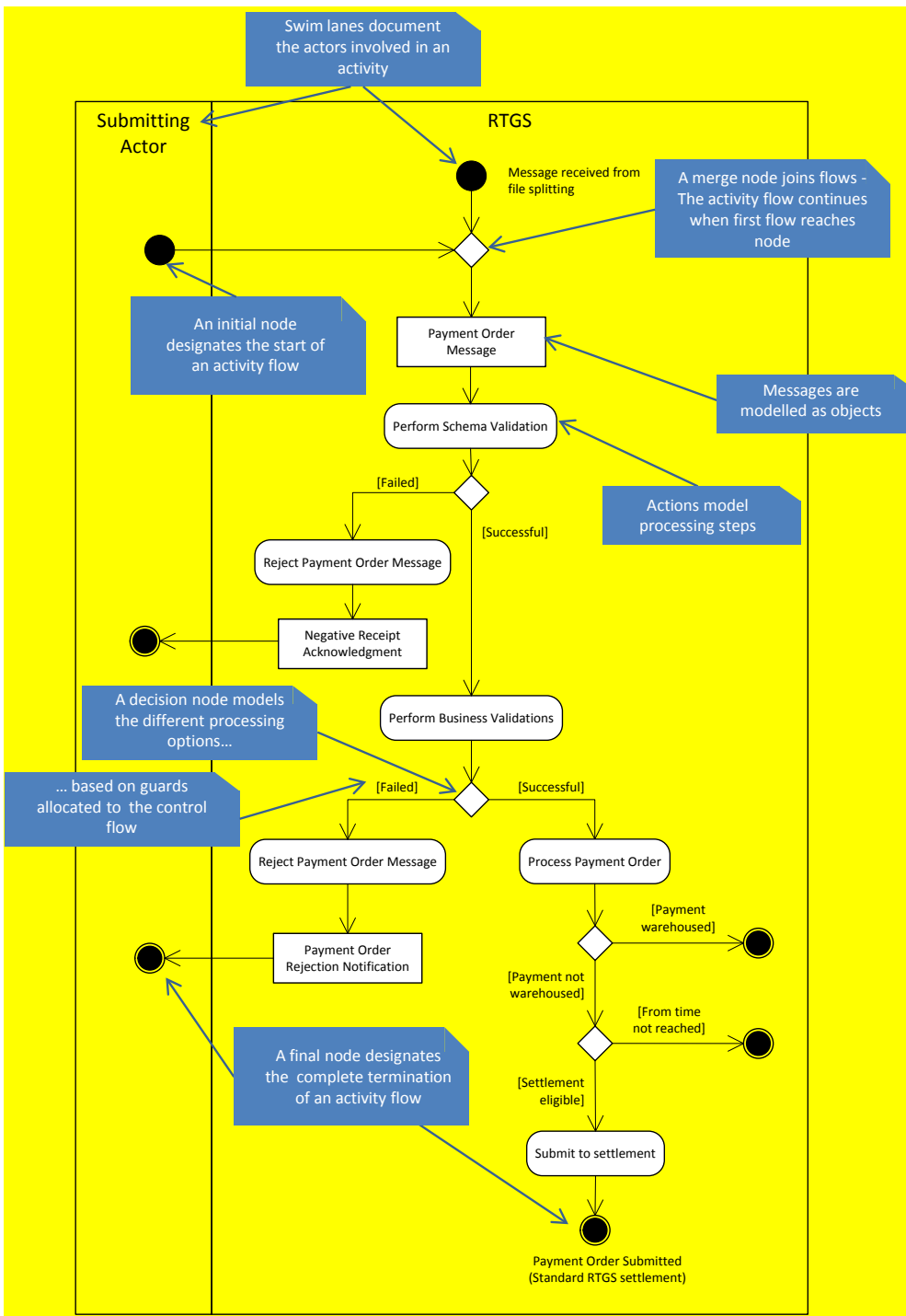


Figure 56 - UML conventions – example I

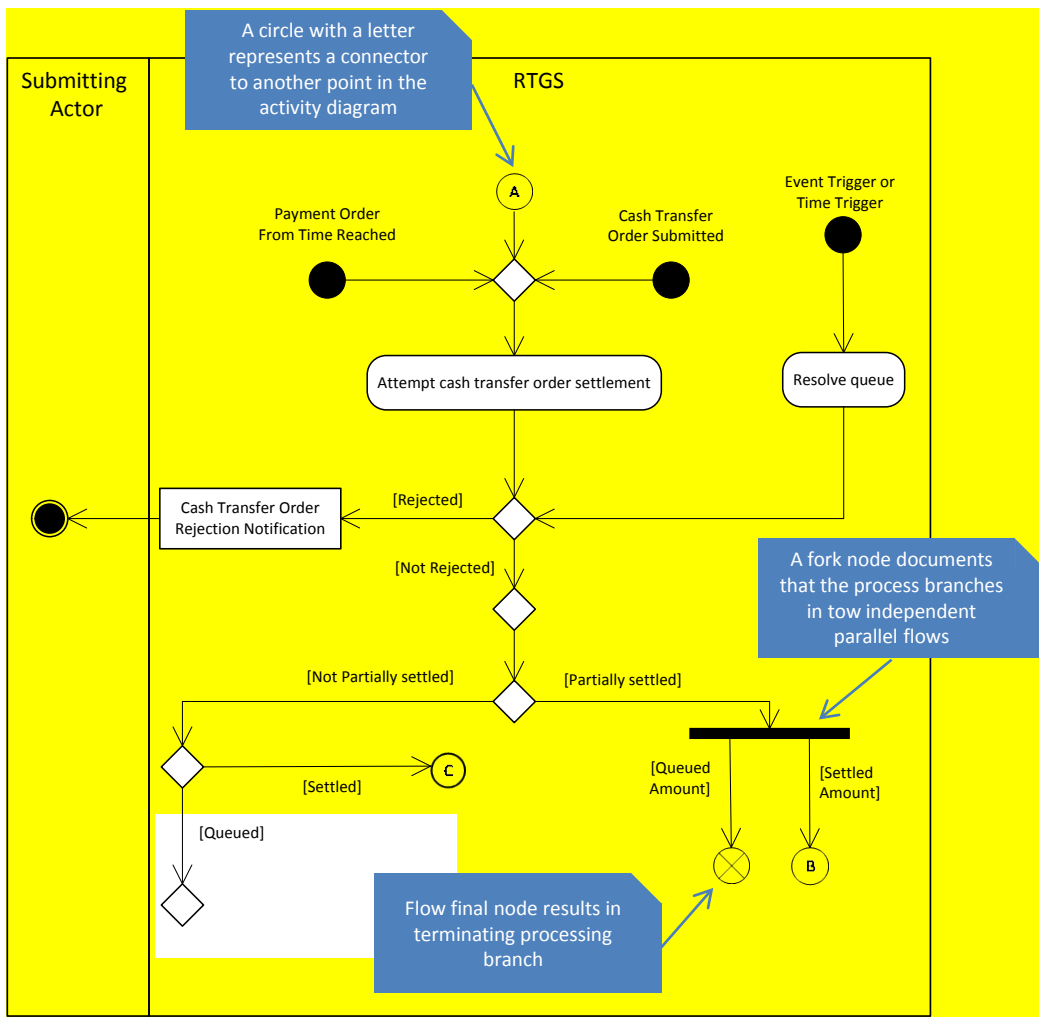


Figure 57 - UML conventions- example II

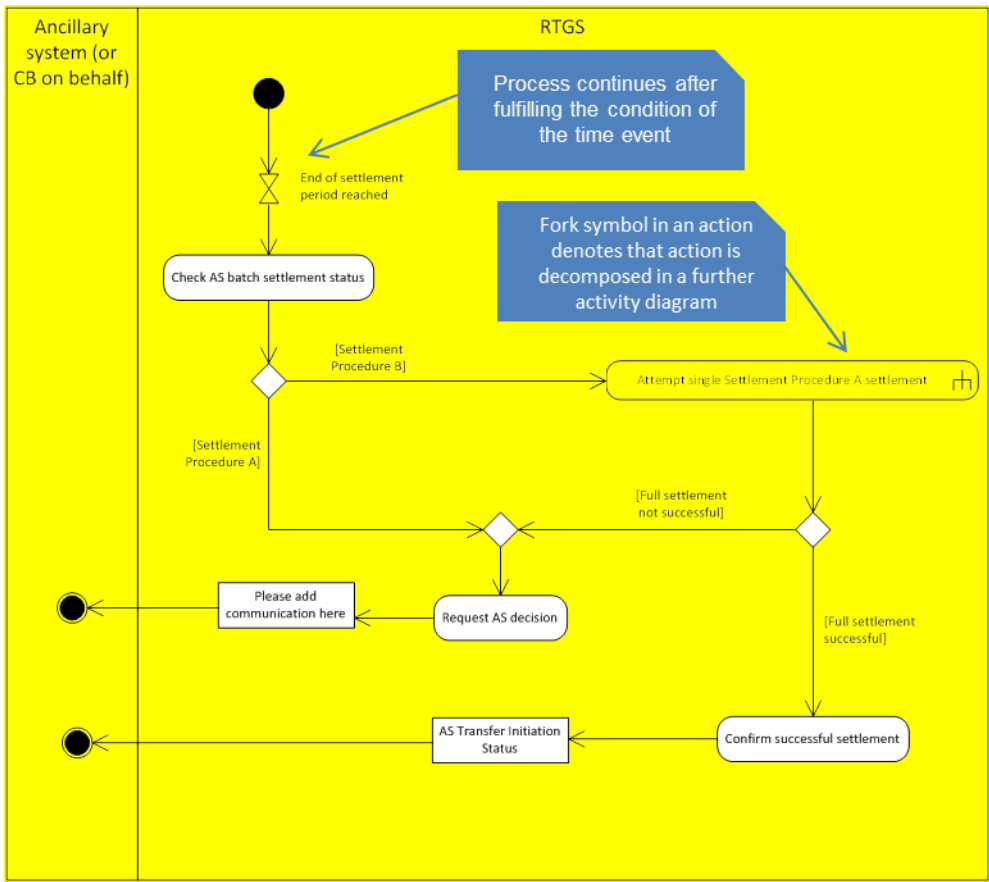


Figure 58 - UML conventions - example III

Each use case generally consists of one diagram. However, this approach can lead to very complex diagrams when a given use case covers many possible process variations. In order to reduce this complexity to ensure readability, a use case may be

- ! decomposed to provide diagrams on the level of its sub-processes;
- ! provided as a universal diagram to cover several use cases of the same type (e.g. a generic send query use case instead of a use case for each query).

10.1 Send file

This is a general process description for executing the A2A file processing, which is similar in CLM and RTGS component. The submitting actor sends a file including business file header and several messages including instructions via ESMIG to the relevant component.

The following activity diagram provides respective processes in the context of the component:

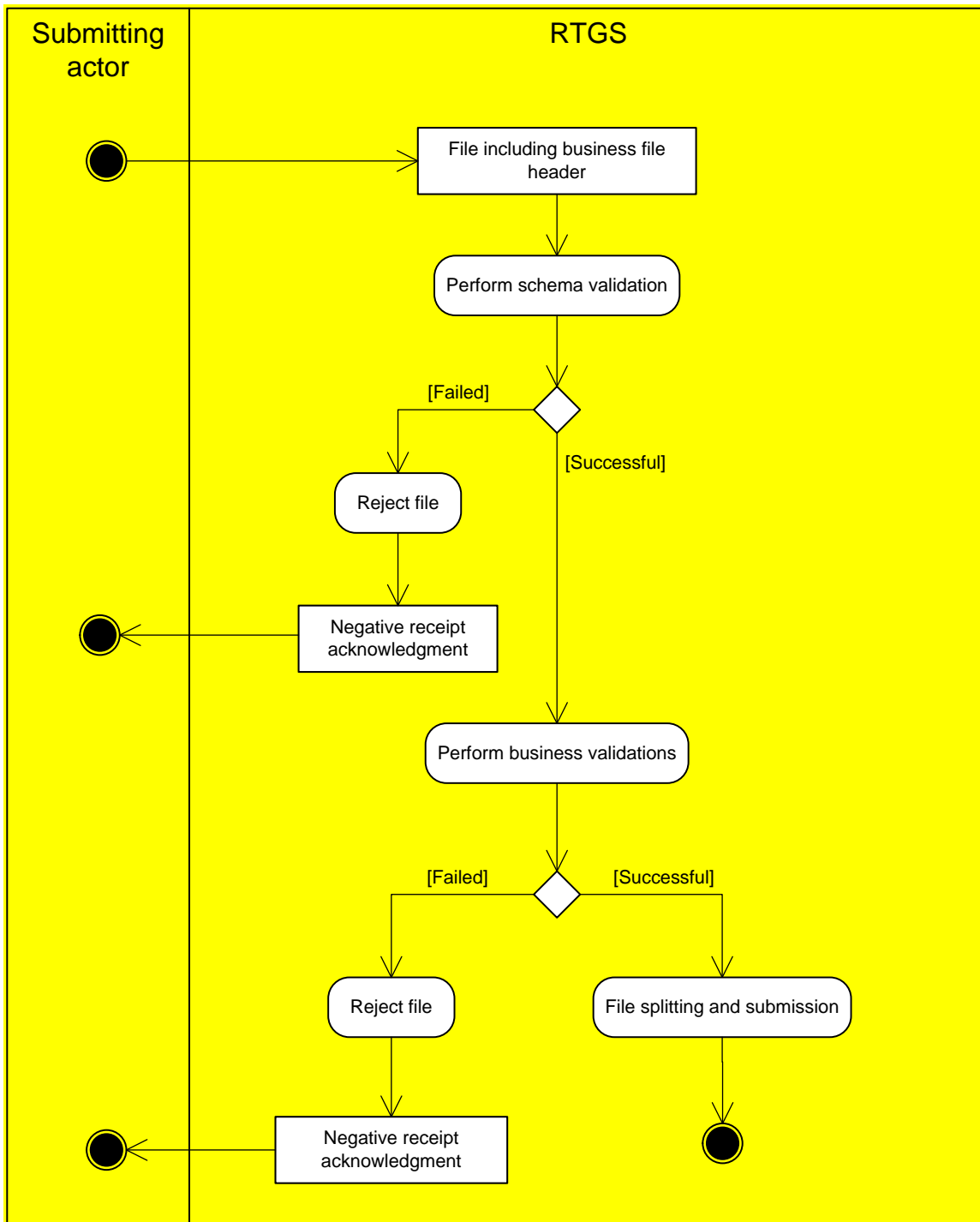


Figure 59 - Send file

Schema validation

As a first step within the respective component, the process “Perform schema validation” performs the schema validation of the business file header included in the [File including business file header]. It validates the file structure (i.e. file header as first record in a file and file trailer as last record in a file are not identifiable or individual messages are not recognisable).

[Failed] The process “Reject file” sends a [ReceiptAcknowledgement \(admi.007\)](#) [391] to the submitting actor including all information regarding the reasons for failed validation.

[Successful] The process triggers the business validation.

Business validation

In a second step (i.e. after successful schema validation), the component performs the business validations (all business rules which are relevant on file level). The validation procedure continues with business validations to the extent possible even after the business validation identifies one or more errors. It reports all identified validation errors.

[Failed] The process “Reject file” sends a [ReceiptAcknowledgement \(admi.007\)](#) [391] that includes the reasons for failing [Negative receipt acknowledgment] to the submitting actor.

[Successful] The process “File splitting and submission” starts. It splits the file into single messages and forwards them to the next process – see dedicated processes for single messages sent by the submitting actors.

10.2 Process cash transfer instruction

10.2.1 Send cash transfer order

This process starts

when the submitting actor sends one of the following messages via ESMIG to the RTGS component:

Message	Message name
PaymentReturn (pacs.004) [571]	PaymentReturn
CustomerCreditTransfer (pacs.008) [577]	CustomerCreditTransfer
FinancialInstitution-CreditTransfer (GEN and COV) (pacs.009) [585]	FinancialInstitutionCreditTransfer
FinancialInstitutionDirectDebit (pacs.010) [603]	FinancialInstitutionDirectDebit
LiquidityCreditTransfer (camt.050) [497]	LiquidityCreditTransfer

Table 129 - Messages sent by the submitting actor to RTGS component

when the RTGS component receives a message from the file splitting process (refer to interface process [Send file](#) [295]).

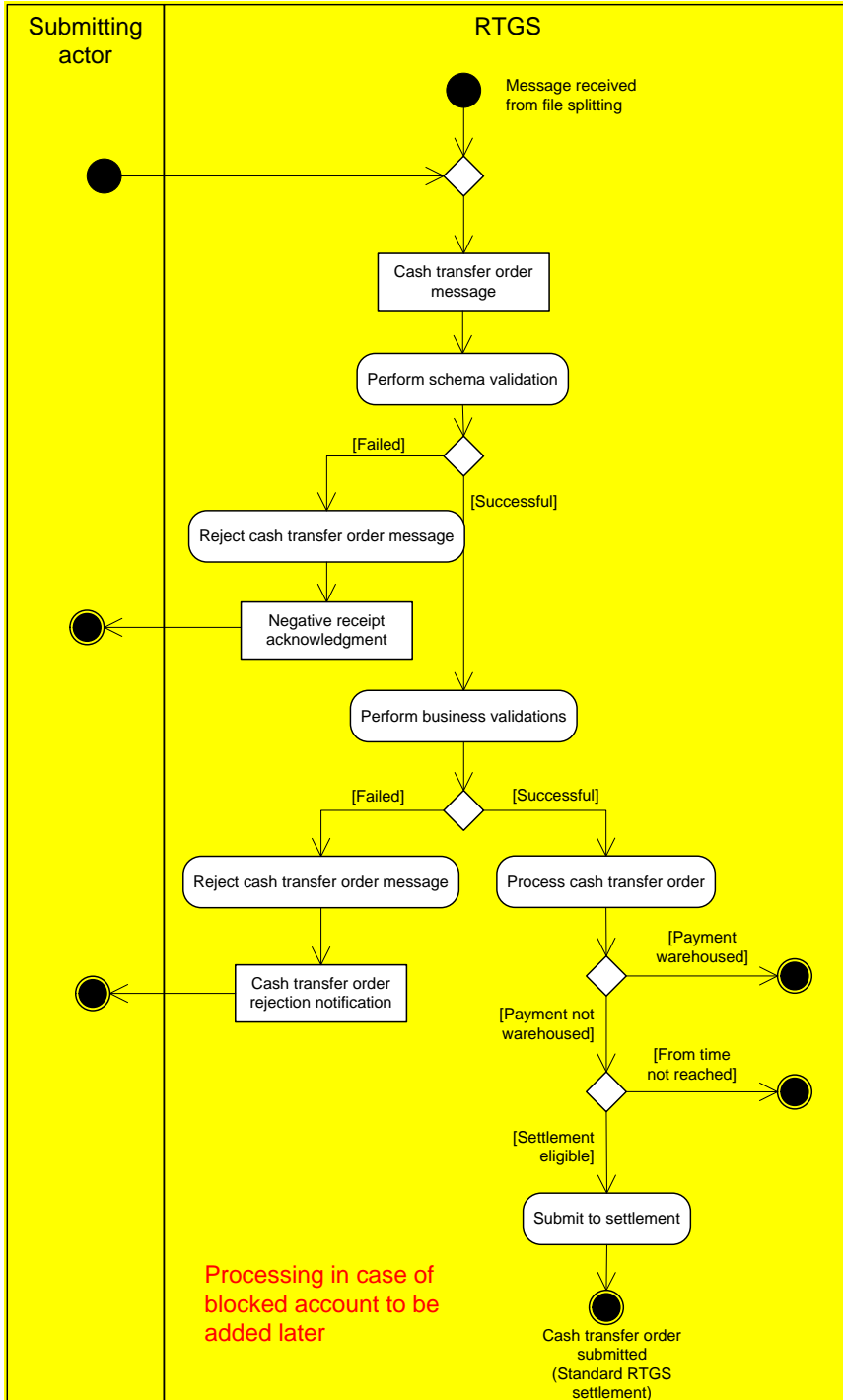


Figure 60 - Send RTGS cash transfer order

Schema validation:

In the first step, the RTGS component performs the schema validation of the cash transfer order message.

[Failed] In case the schema validation fails, the RTGS component rejects the cash transfer order message and the submitting actor receives a “*Negative receipt acknowledgement*” [ReceiptAcknowledgement \(admi.007\)](#) [391].

Note: RTGS identifies all possible schema validation errors and does not stop the schema validation after the first error is found.

[Successful] In case of a successful schema validation, the RTGS component continues with the business validation.

Business validation:

In the second step, RTGS performs the business validation with possible outcomes being:

[Failed] In case the business validation fails, the RTGS component rejects the cash transfer order message and the submitting actor receives a “*Cash transfer order Rejection Notification*” [PaymentStatusReport \(pacs.002\)](#) [568] or [Receipt \(camt.025\)](#) [474].

Note: The RTGS component continues with all possible business validations even after the business validation identifies one or more errors. It does not stop after identifying the first business validation error. Consequently, the rejection notification includes all relevant error codes.

[Successful] In case the business validation is successful, RTGS continues with the processing of the cash transfer order.

As part of this processing step, the RTGS component determines

- whether the cash transfer order is a warehoused payment;
- whether the defined “FromTime” when specified in the payment has not been reached;
- whether the payment is directly eligible for the settlement.

The processing submits the cash transfer order directly to the [Standard RTGS settlement](#) [305] process when it is directly eligible for settlement.

10.2.2 Revoke/cancel payment

A submitted payment can be revoked using a PaymentRevocationRequest (A2A or U2A). A revocation of a payment is only possible as long as the payment is not settled on the RTGS DCA. It is also possible to revoke warehoused payments.

For settled payments the PaymentRevocationRequest is forwarded to the RTGS counterpart for further processing. The counterpart decides whether or not the payment will be returned.

In the process flow below, the three possible cases are illustrated:

1. cancellation of unsettled payment
2. counterpart agrees to return the settled payment

3. counterpart disagrees to return the settled payment

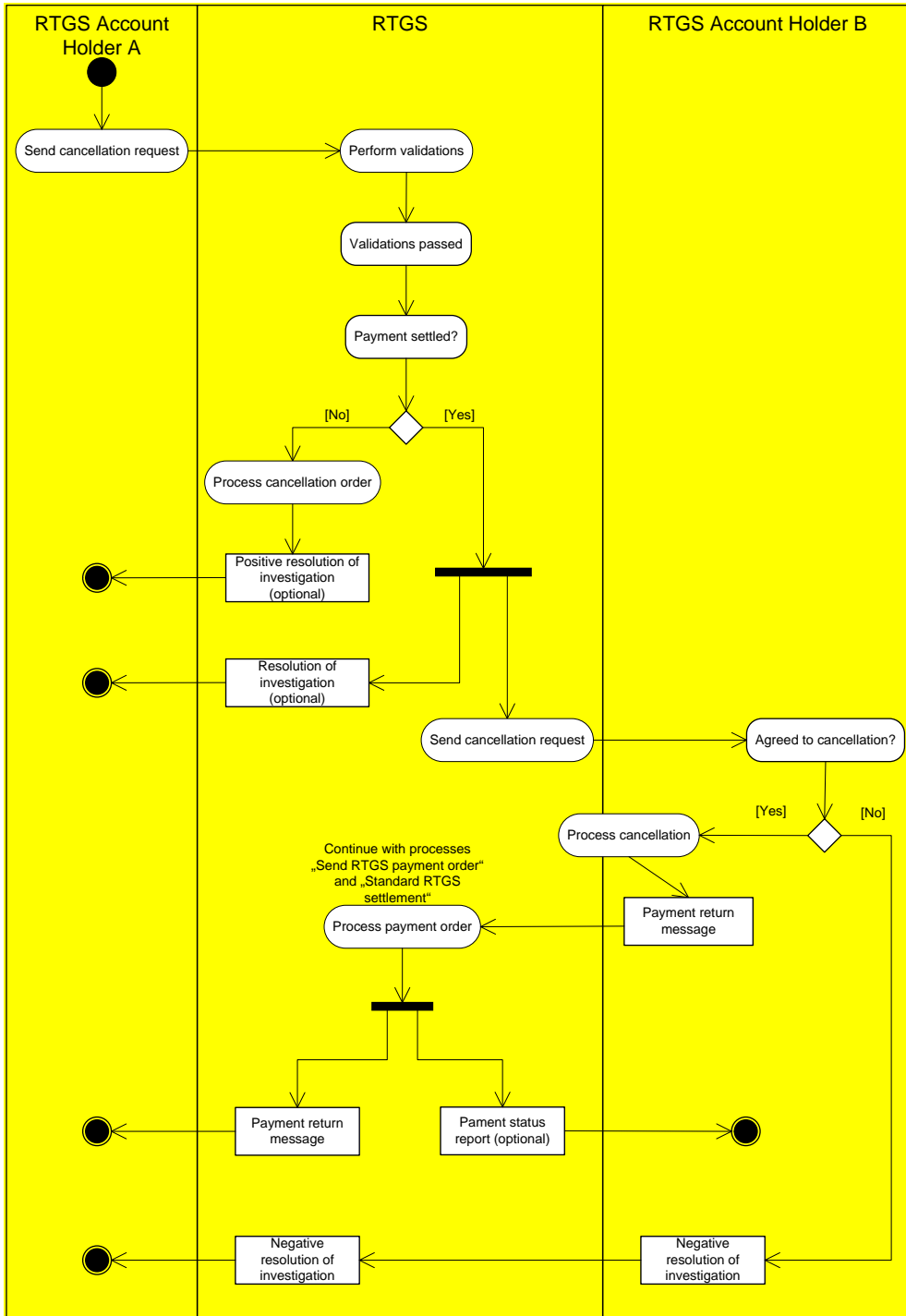


Figure 61 - Revoke/cancel payment

Step	Actor	Description
1	RTGS Account Holder A	The RTGS Account Holder sends a request to cancel an earlier payment to RTGS.
2	RTGS	RTGS carries out the technical and business validations.
3	RTGS	When validations are passed, RTGS checks if the requested payment is settled.

Step	Actor	Description
4	RTGS	In case the payment is not settled yet, RTGS processes the cancellation order and sends a positive resolution of investigation to the account holder – process case ends here.
5	RTGS	In case the requested payment is already settled, RTGS sends a cancellation request to the counterparty account holder of the payment.
6	RTGS Account Holder B	In case the counterpart agrees to the cancellation request, the payment is returned to RTGS.
7	RTGS	RTGS processes the payment return (see processes Send cash transfer order [297] and Standard RTGS settlement [305])
8	RTGS	RTGS send payment return message to account holder A as well as an optional payment status report to account holder B – Process case ends here.
9	RTGS Account Holder B	In case the counterpart does not agree to the cancellation request, a negative resolution of investigation is send to RTGS.
10	RTGS	RTGS forwards this negative resolution of investigation to account holder A – process case ends here.

Table 130 - Process description revoke/cancel payment

For further details about revocation / cancellation of payments please see chapter [Revocation of payments](#) [118].

10.2.3 Amend payment

A submitted payment (including warehoused payments) can be amended as long as the payment is not settled yet. Amendments can be submitted via A2A or U2A.

Process flow

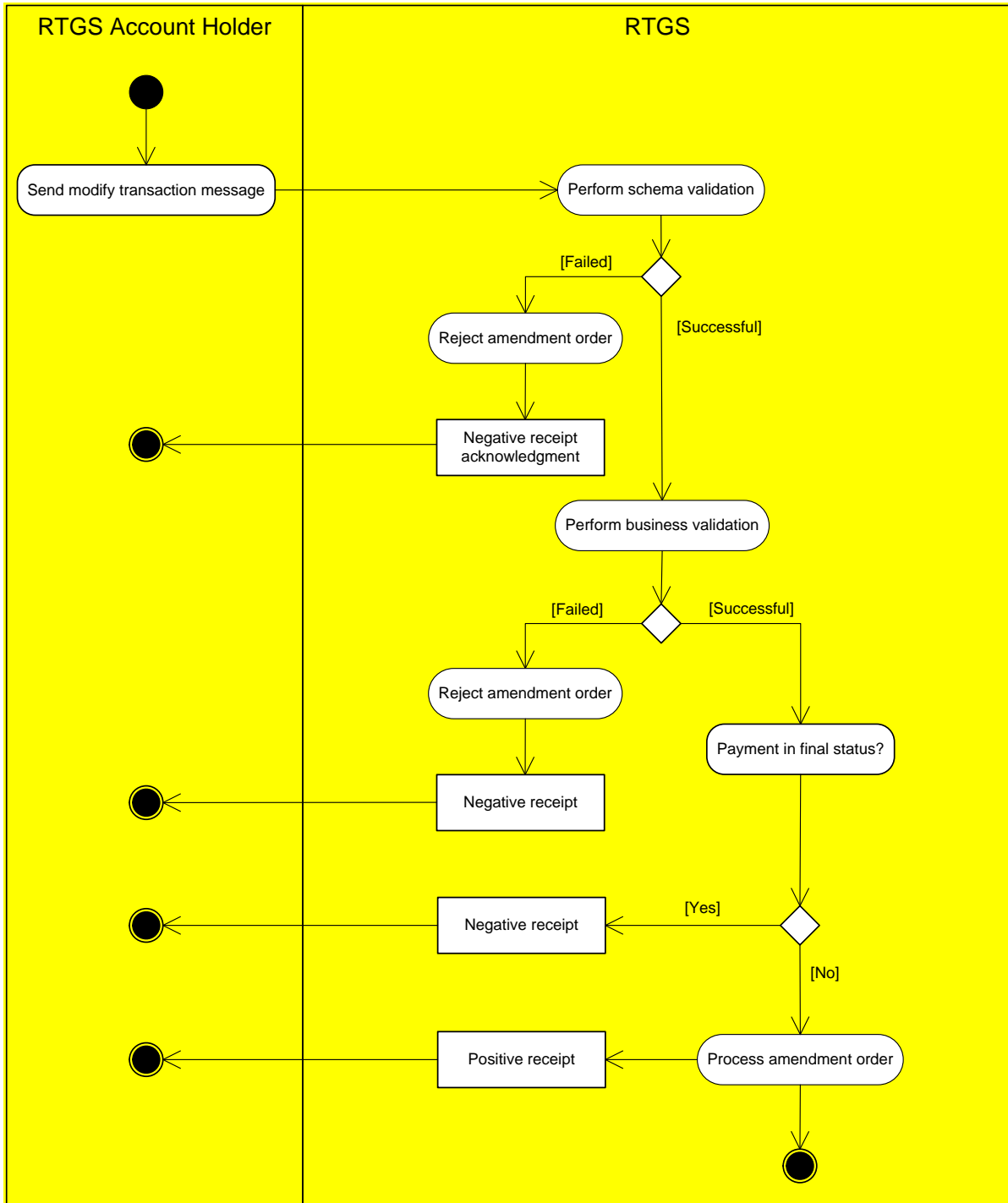


Figure 62 - Amend payment

Step	Actor	Description
1	RTGS Account Holder	RTGS Account Holder sends a request to RTGS to amend a payment.
2	RTGS	RTGS carries out the technical and business validations.
3	RTGS	When validations are passed, RTGS checks if the requested payment is settled.

Step	Actor	Description
4	RTGS	In case the payment is already settled, RTGS send a negative receipt to the RTGS Account Holder.
5	RTGS	In case the payment is not settled yet, RTGS processes the amendment order.
6	RTGS	RTGS send a positive receipt to the RTGS Account Holder.

Table 131 - Process description amend payment

For further details about the amendment of payments please see chapter [Amendment of payments](#) [▶ 113].

10.2.4 Execute RTGS standing order

RTGS standing orders are instructions of a settlement bank to transfer regularly a fixed amount from its RTGS DCA to

- the CLM main account
- another RTGS DCA
- another service
- an ancillary system sub-account (ancillary system type C, interfaced model) or
- an ancillary system technical account (ancillary system type D, real-time model).

Only the settlement bank can set standing orders; the ancillary system cannot instruct standing orders to credit the technical account - procedure 6 real-time or sub-accounts related to it.

Standing orders are set up and maintained in CRDM.

Process flow

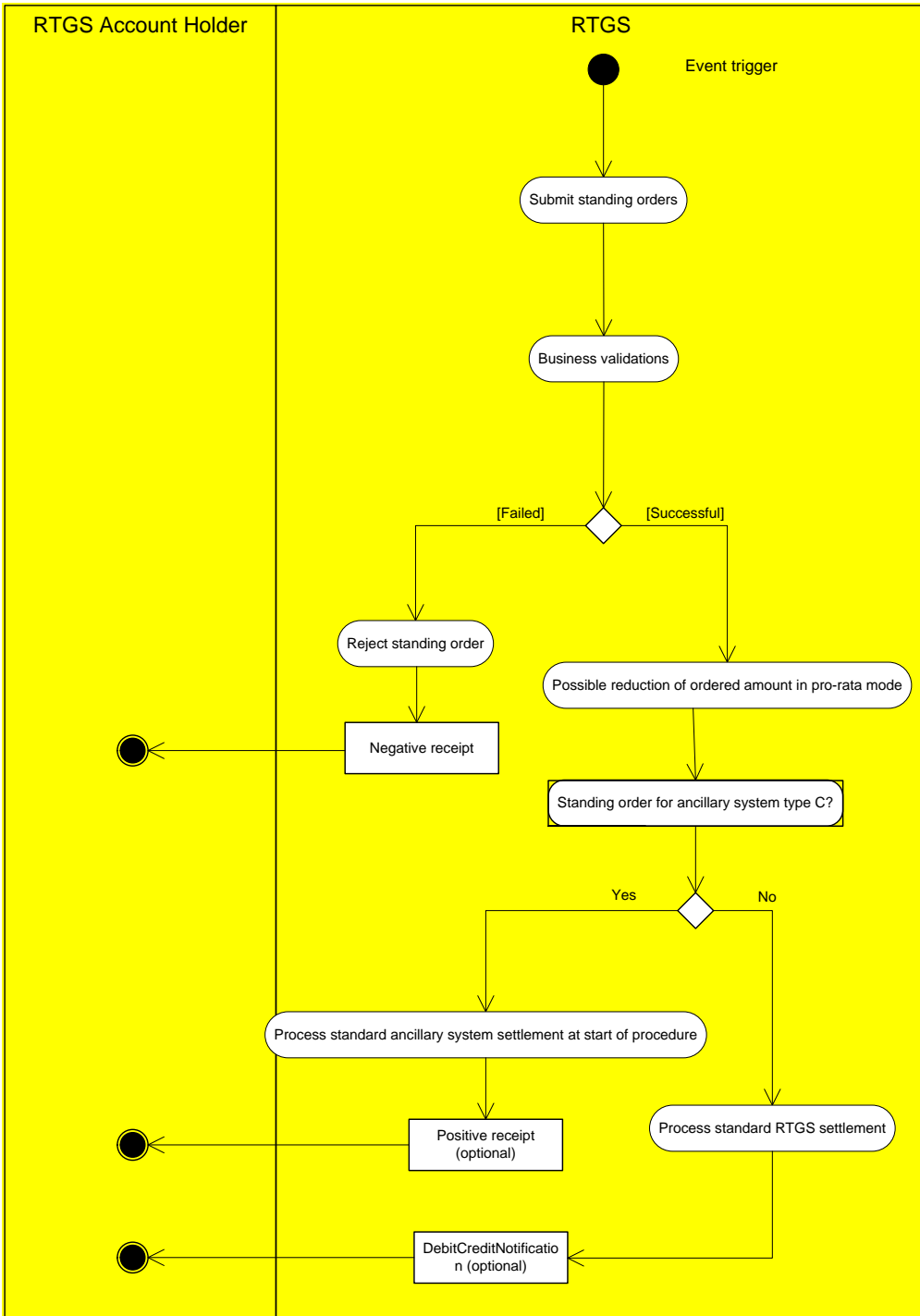


Figure 63 - Execute RTGS standing order

Step	Actor	Description
1	BD	Business day event "Processing time window for ancillary system settlement" initiates the process.
2	RTGS	RTGS carries out the business validations.

Step	Actor	Description
3	RTGS	In case of positive validation: RTGS carries out the settlement of the standing order instructions.
4	RTGS	RTGS sends a CreditDebitNotification to the debited and/or credited RTGS Account Holder if he subscribed for it.

Table 132 - Process description execute RTGS standing order

Business validation rules:

- | Is the order eligible to be booked on the settlement bank's debit account in terms of liquidity position and existing reservations?
- | Check against blocking of the settlement bank or the sub-account:
 - If yes the order is directly rejected.
- | Check against blocking of the related ancillary system:
 - Control messages sent by the ancillary system are directly rejected.
 - Control messages sent by the CB are accepted.
- | **[Failed]** A notification (return account resp. transfer notification) is sent to the initiator of the standing order with a specific error code.
- | **[Successful]** In case the liquidity/reservation is not sufficient a partial execution of the liquidity transfer is envisaged. That means: the ordered amount of all standing orders is reduced pro rata to the amount which can be settled. And no further settlement attempt is done for the remaining part.

Settlement Model C: The ancillary system payment instructions are executed at SoD procedure.

All other standing orders: The processing submits the payment directly to the standard RTGS settlement (see chapter [Standard RTGS settlement](#) [305]) process for settlement.

In case of ancillary system type C a notification (*ReturnAccount*) is sent to the ancillary system.

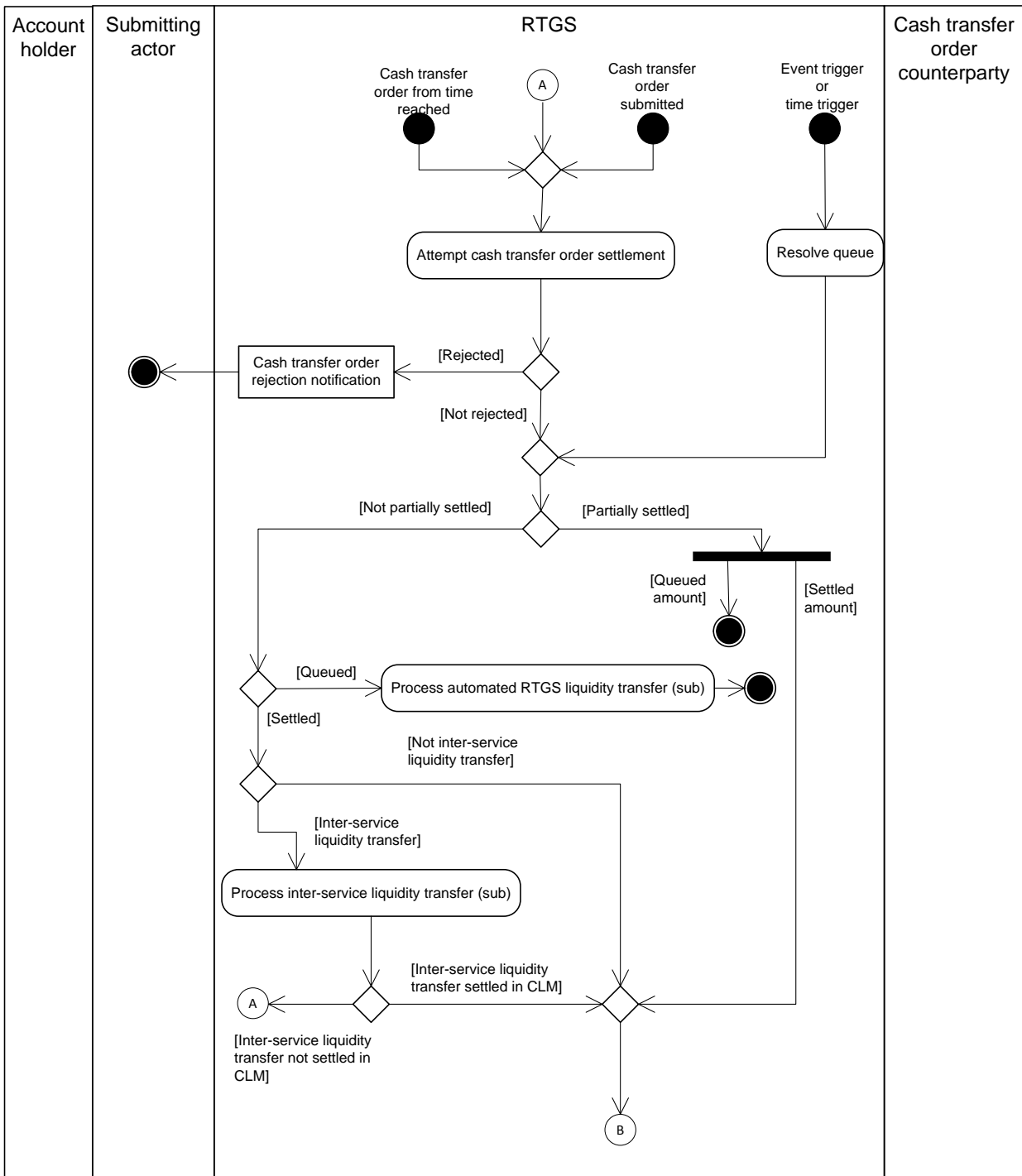
10.3 Settle RTGS cash transfer orders

10.3.1 Standard RTGS settlement

The process "*Attempt cash transfer order settlement*" starts

- | after receiving a successfully validated cash transfer order [Cash transfer order submitted],
- | in case of an inter-service liquidity transfer initiated in the RTGS component could not be successfully booked in the other service and the amount needs to be credited back to the RTGS DCA [A] and

- I for a successfully validated cash transfer order that specifies "From time" and the "From time" has been reached [Cash transfer order from time reached].



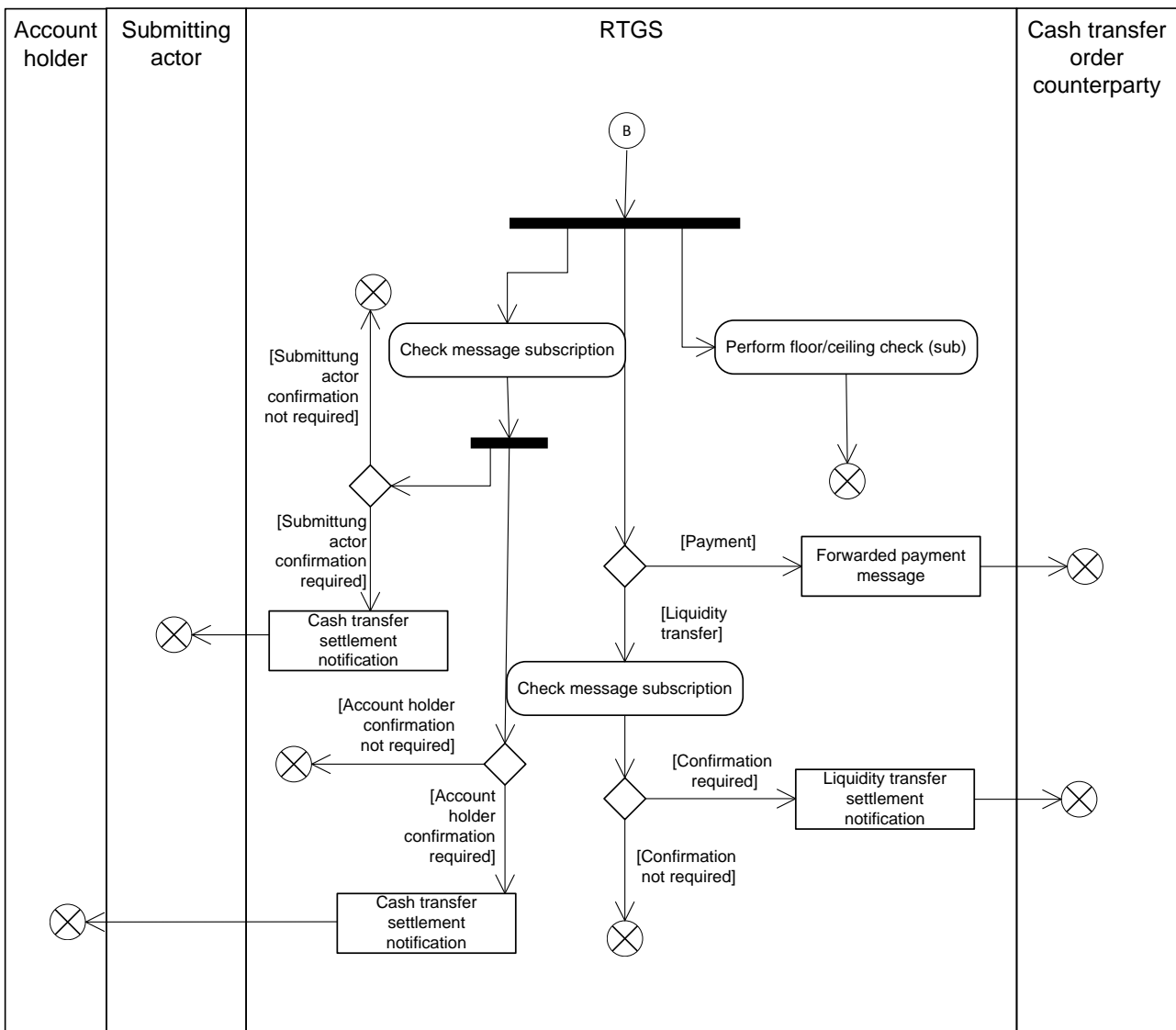


Figure 64 - Standard RTGS settlement

In the first step, the process “*Attempt cash transfer order settlement*” tries to settle the submitted payment order, resulting in one of the following outcomes:

- | **[Rejected]** In case settlement of the liquidity transfer is not possible due to insufficient liquidity, the process rejects the liquidity transfer and sends a “Cash transfer order rejection notification” [Receipt \(camt.025\)](#) [▶ 474] to the submitter of the original incoming camt.050.

Note: This is not valid for automated inter-service liquidity transfers from CLM due to pending CBOs.

- | **[Not rejected]**

 - The payments settle or queue.
 - The automated inter-service liquidity transfers from CLM due to pending CBOs could be settled, partially settled or queued.
 - The liquidity transfers sent by the account holder settle.
 - The liquidity transfers sent by a submitting actor not being the account holder settle or partially settle.

In the second step

- | for all accepted (not rejected) cash transfer orders
- | as well as for all queued payments forwarded to the process “*Resolve queue*” in case of an event or time trigger

the result of the process can be:

- | **[Partially settled]** The only scenarios, in which a liquidity transfer is partially settled are
 - an automated inter-service liquidity transfer from CLM due to pending CBOs and insufficient liquidity on the RTGS DCA.
 - those transmitted by a submitting actor not being the account holder.

Note: Payments are never partially settled in the RTGS component.

For the partially settled amount the same messages are sent to the involved parties as for fully settled liquidity transfers.

- | **[Queued]** Payments which cannot settle are queued. As a consequence the sub-process “*Automated RTGS liquidity transfer*” is triggered.

Liquidity transfers are queued when it is an automated inter-service liquidity transfer from CLM due to pending CBOs which cannot settle in the RTGS component. In case of partial settlement of these liquidity transfers, the remaining part is queued by the RTGS component.

Note: In case of a new automated inter-service liquidity transfer from CLM due to pending CBOs the RTGS component uses a “cancel and replace logic”, i.e. the already pending automated inter-service liquidity transfer is cancelled and the new one is taken into account for further processing.

- | **[Settled]** After successful settlement the “*Cash transfer order counterparty*” receives in case of
 - payments, one of the following messages:

Message	Message name
PaymentReturn (pacs.004) [▶ 571]	PaymentReturn
CustomerCreditTransfer (pacs.008) [▶ 577]	CustomerCreditTransfer
FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▶ 585]	FinancialInstitutionCreditTransfer /FinancialInstitutionCreditTransferCOV
FinancialInstitutionDirectDebit (pacs.010) [▶ 603]	FinancialInstitutionDirectDebit

Table 133 - Message sent after settlement

- liquidity transfers:

a “*Liquidity transfer settlement notification*” [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522] provided that a respective message subscription configuration has been set up in advance.

Note: RTGS treats inter-service liquidity transfers that another service/component initiates as any other intra-service liquidity transfer.

I the *submitting actor* receives in case of

– intra-service liquidity transfers initiated via camt.050:

a “*Cash transfer settlement notification*” [Receipt \(camt.025\)](#) [▶ 474] provided that a respective message subscription configuration has been set up in advance;

– inter-service liquidity transfers initiated via camt.050 in RTGS:

a “*Cash transfer settlement notification*” [Receipt \(camt.025\)](#) [▶ 474] only after successful settlement in the other service or component provided that a respective message subscription configuration has been set up in advance.

– payments:

a “*Cash transfer settlement notification*” [PaymentStatusReport \(pacs.002\)](#) [▶ 568] provided that a respective message subscription configuration has been set up in advance.

I the *account holder* receives the following messages provided that the submitting actor and the account holder differ in case of

– intra-service liquidity transfers initiated via camt.050 :

a “*Booking notification*” [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522] provided that a respective message subscription configuration has been set up in advance.

– inter-service liquidity transfers initiated via camt.050 in RTGS:

a “*Booking notification*” [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522] only after successful settlement in the other service provided that a respective message subscription configuration has been set up in advance.

– payments:

a “*Booking notification*” [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522] provided that a respective message subscription configuration has been set up in advance.

10.3.1.1 Process floor and ceiling

This process starts after settlement of a payment (i.e. pacs.004/pacs.008/pacs.009/pacs.009COV/pacs.010) or an ancillary system payment instruction on the RTGS DCA.

Note: The settlement of liquidity transfers on RTGS DCAs trigger no floor/ceiling processing.

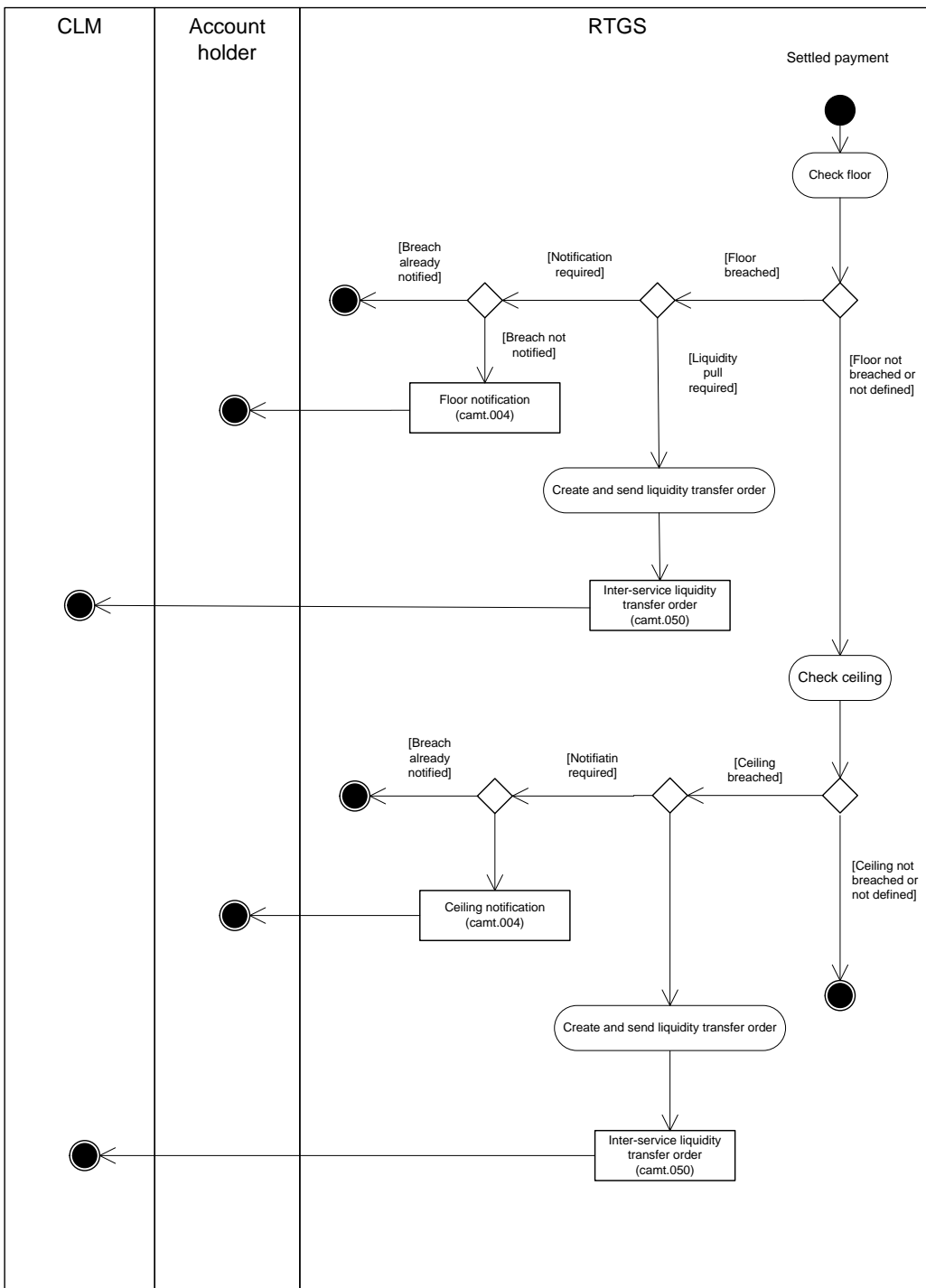


Figure 65 - Floor and ceiling processing

Floor processing:

- | In case
 - of a breach of a previously defined floor,
 - the configuration to receive a floor notification has been set up in advance and
 - no prior notification of the breach to the account holder,

the RTGS DCA holder receives a “*Floor notification*” [ReturnAccount \(camt.004\)](#) [▶ 397].

- | In case
 - Of a breach of a previously defined floor and
 - the configuration to trigger an inter-service liquidity transfer to pull liquidity from the linked MCA has been set up in advance

RTGS sends to the CLM component an inter-service liquidity transfer order, [LiquidityCreditTransfer \(camt.050\)](#) [▶ 497], in order to pull liquidity up to the targeted floor amount.

Ceiling processing:

- | In case
 - of a breach of a previously defined ceiling,
 - the configuration to receive a ceiling notification has been set up in advance and
 - no prior notification of the breach to the account holder,

the RTGS DCA holder receives a “*Ceiling notification*” [ReturnAccount \(camt.004\)](#) [▶ 397].

- | In case
 - of a breach of a previously defined ceiling and
 - the configuration to trigger an inter-service liquidity transfer to push liquidity to the linked MCA has been set up in advance

RTGS sends to the CLM component an inter-service liquidity transfer order as [LiquidityCreditTransfer \(camt.050\)](#) [▶ 497] in order to push liquidity to reach the predefined target ceiling amount on the RTGS DCA in the RTGS component.

10.3.1.2 Process automated liquidity transfer

This optional process starts when a payment with priority urgent or high does not settle and, therefore, is queued. In addition, it is necessary that the RTGS Account Holder has defined in advance that in such case liquidity shall be pulled from the linked MCA.

Note: This functionality can be used independently from the definition of a floor/ceiling. Details on the inter-service liquidity transfers due to a floor/ceiling configuration can be found in chapter [Process floor and ceiling](#) [▶ 309].

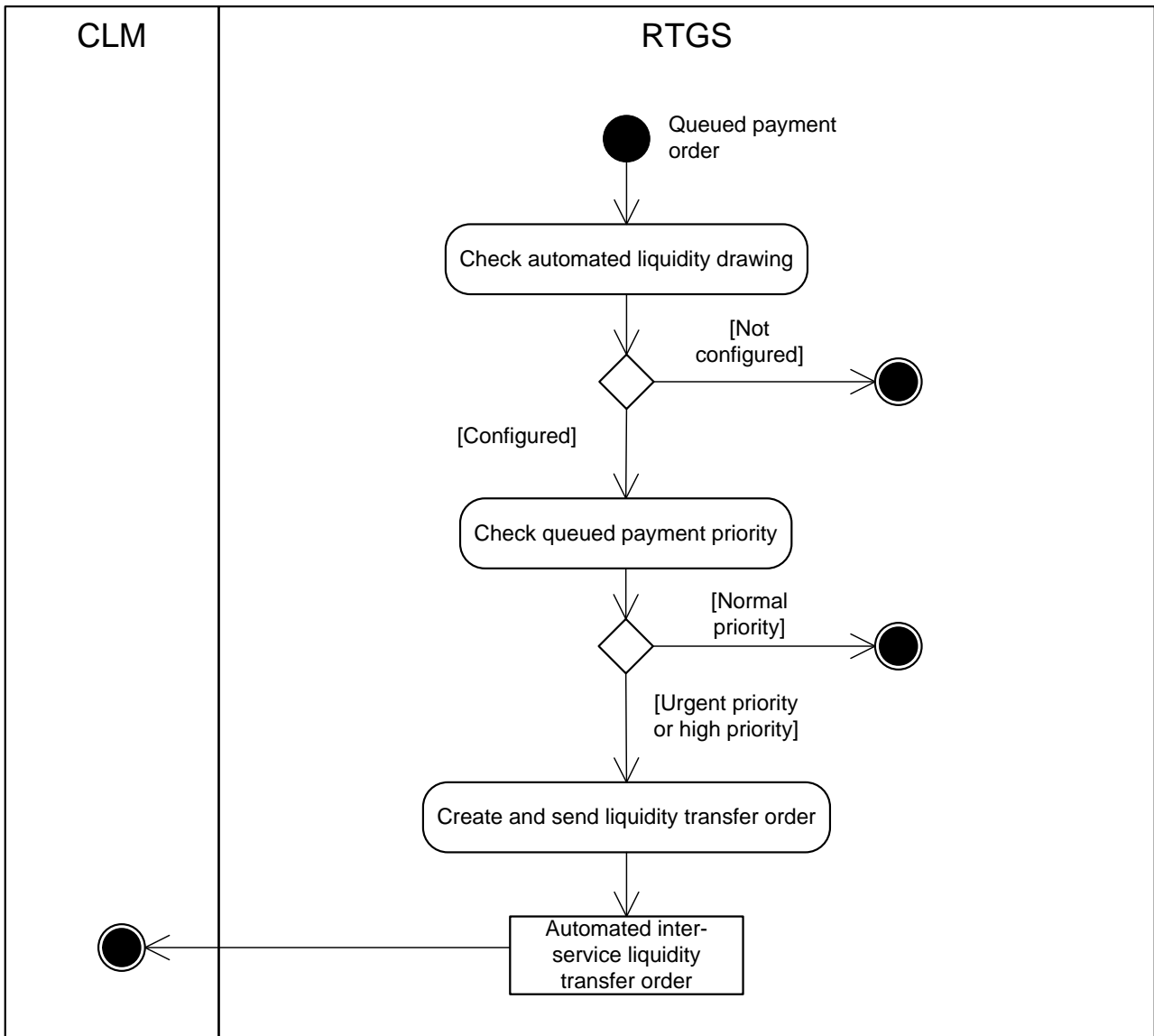


Figure 66 - Process automated RTGS liquidity transfer order

The RTGS component automatically creates a new inter-service liquidity transfer order and sends a [LiquidityCreditTransfer \(camt.050\) \[▶ 497\]](#) to CLM in order to pull the liquidity needed from CLM in order to settle the queued payment in the RTGS component.

Note: There is no earmarking and in case new payments with a higher priority than the queued payment are submitted, the liquidity might be used to settle payments with a higher priority.

10.3.2 Process RTGS till/reject time instructions

Payments can include a “latest debit time” indicator, to determine up to which point in time the payment has to be settled. The following process shows the flow in case of a given “latest debit time” indicator.

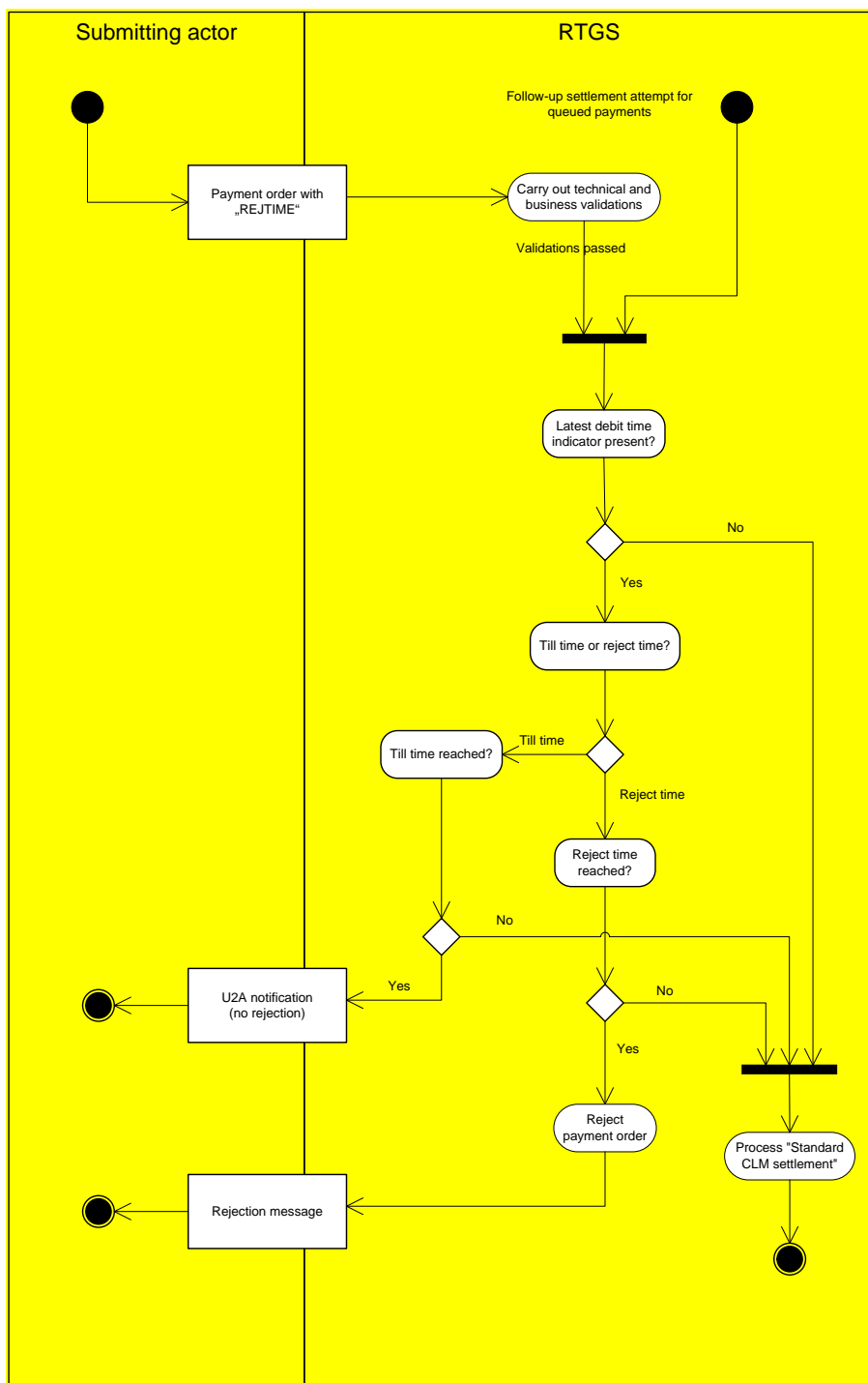


Figure 67 - Process till/reject time instructions

The process starts after two entry points:

- The submission of a new payment (pacs.008, pacs009, pacs.010) or
- New settlement attempt of a queued payment

After passing the technical and business validations, RTGS checks if a reject time is given

- In case a latest debit time is not present, the payment is further processed in standard CLM settlement (please refer to chapter [Standard RTGS settlement](#) [305])

In case a latest debit time indicator is present, CLM checks whether it is the till time or reject time indicator

- If the latest debit time indicator is “TILTIME” and the time has been reached, settlement attempts are stopped and the submitting actor receives a U2A notification.
- If the reject time has been reached, RTGS rejects the payment and sends a rejection message (pacs.002) to the submitting actor.
- If the latest debit time indicator is “TILTIME” has not been reached yet, the payment is further processed in standard CLM settlement (please refer to chapter [Standard RTGS settlement](#) [▶ 305])

10.3.3 Ancillary system payment settlement

10.3.3.1 Send ancillary system transfer initiation

This process is triggered whenever a proprietary message of type *ASTransferInitiation* is send by an ancillary system or CB via ESMIG to the RTGS component.

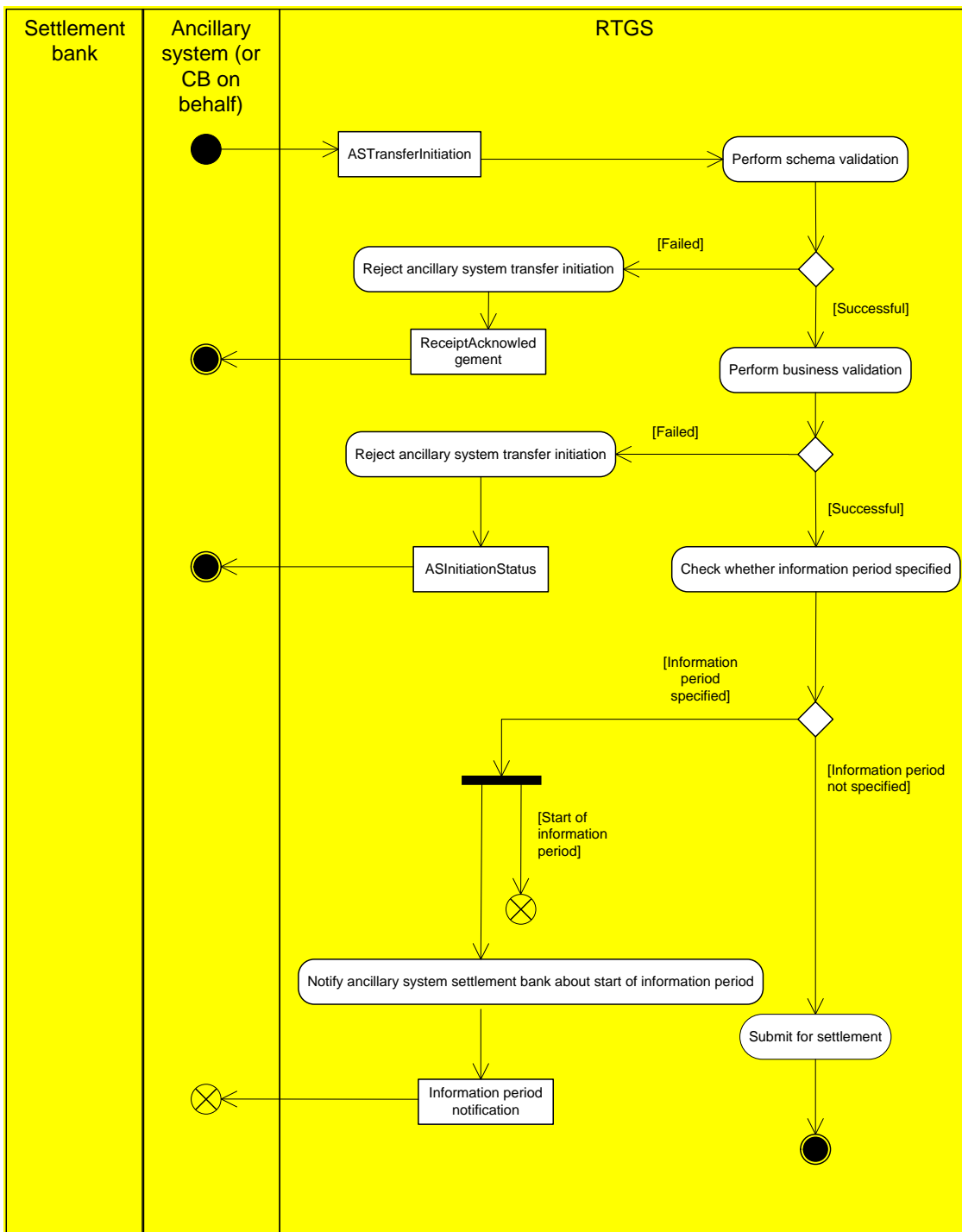


Figure 68 - Send ancillary system transfer initiation

Perform schema validation

In the first step, the RTGS component performs the schema validation of the [ASTransferInitiation \(pain.998\) \[633\]](#) message:

[Failed] In case the schema validation fails, the RTGS component rejects the *ASTransferInitiation* message and the submitting actor receives a “Negative receipt acknowledgement” [ReceiptAcknowledgement \(admi.007\)](#) [391].

[Successful] In case of a successful schema validation, the RTGS component continues with the business validation.

Perform business validation

In the second step, RTGS performs the business validation with possible outcomes being:

[Failed] In case the business validation fails (either on mandatory group level or on optional repetitive transaction level), the RTGS component rejects the *ASTransferInitiation* message and the submitting actor receives an [ASInitiationStatus \(pain.998\)](#) [620] message indicating the first error found.

[Successful] In case the business validation is successful, RTGS continues with the processing by checking whether the *ASTransferInitiation* is subject to information period.

Check whether information period specified

As last step of this process, in order to identify the next processing step RTGS determines whether the *ASTransferInitiation* and the involved ancillary system payment instructions are subject to information period mechanism. This can only be the case for:

1. *ASTransferInitiation* sent for ancillary system settlement procedures A or B and
2. The submitting actor has opted within the *ASTransferInitiation* message for an end time or duration of the information period.

In case the above prerequisites apply the check will be completed successful (i.e. the information period is specified):

[Information period specified] In case the information period option (relevant for ancillary system settlement procedures A and B) is used and applicable, the information period will immediately start. The process to notify the settlement banks about the start of information period is triggered. The settlement banks receive information period notifications as a broadcast.

[Information period not specified] in case the information period is not applicable, pertaining ancillary system settlement processes will be triggered.

10.3.3.2 Initiate ancillary system settlement for procedure A and B

This process is triggered after successfully validating *ASTransferInitiation* of settlement procedure A or B and, if opted, after the end of information period.

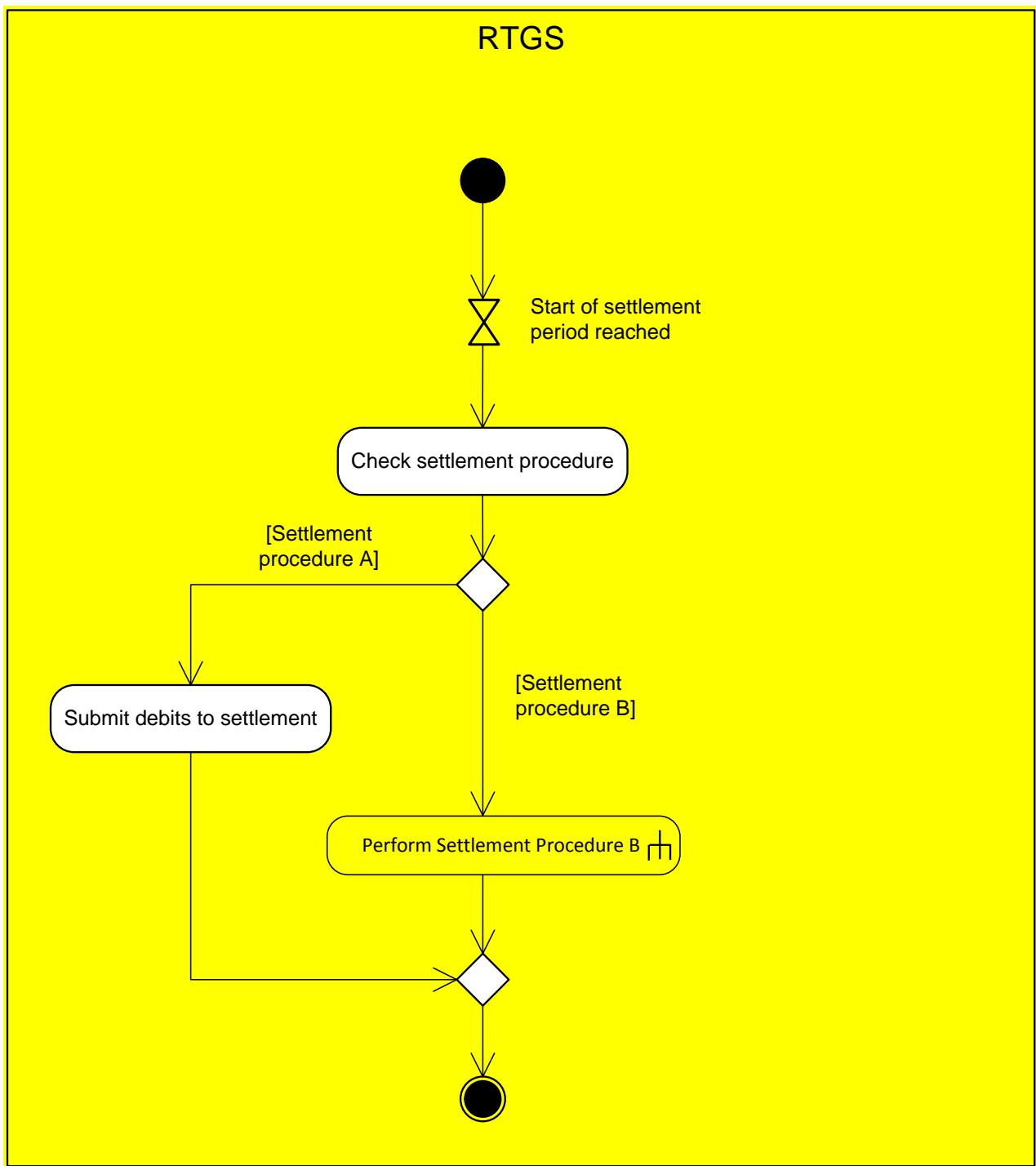


Figure 69 - Initiate ancillary system settlement procedure A and B

Start of settlement period reached

Once the start of settlement period is reached (i.e. after successful validation and, if opted for, after the end of the information period) a check on the used settlement procedure is made in order to distinguish the process to be triggered.

Check settlement procedure

[Settlement procedure A]: In case of settlement procedure A being used, related debit legs are submitted to standard settlement process. After successful settlement of the debit legs the credit legs will be sent to standard settlement process.

[Settlement procedure B]: For ancillary system transfers for settlement procedure B a dedicated process is started (please see [Perform settlement of settlement procedure B](#) [▶ 318])

10.3.3.3 Perform settlement of settlement procedure B

This process is triggered after reaching the settlement period for settlement procedure B by the process.

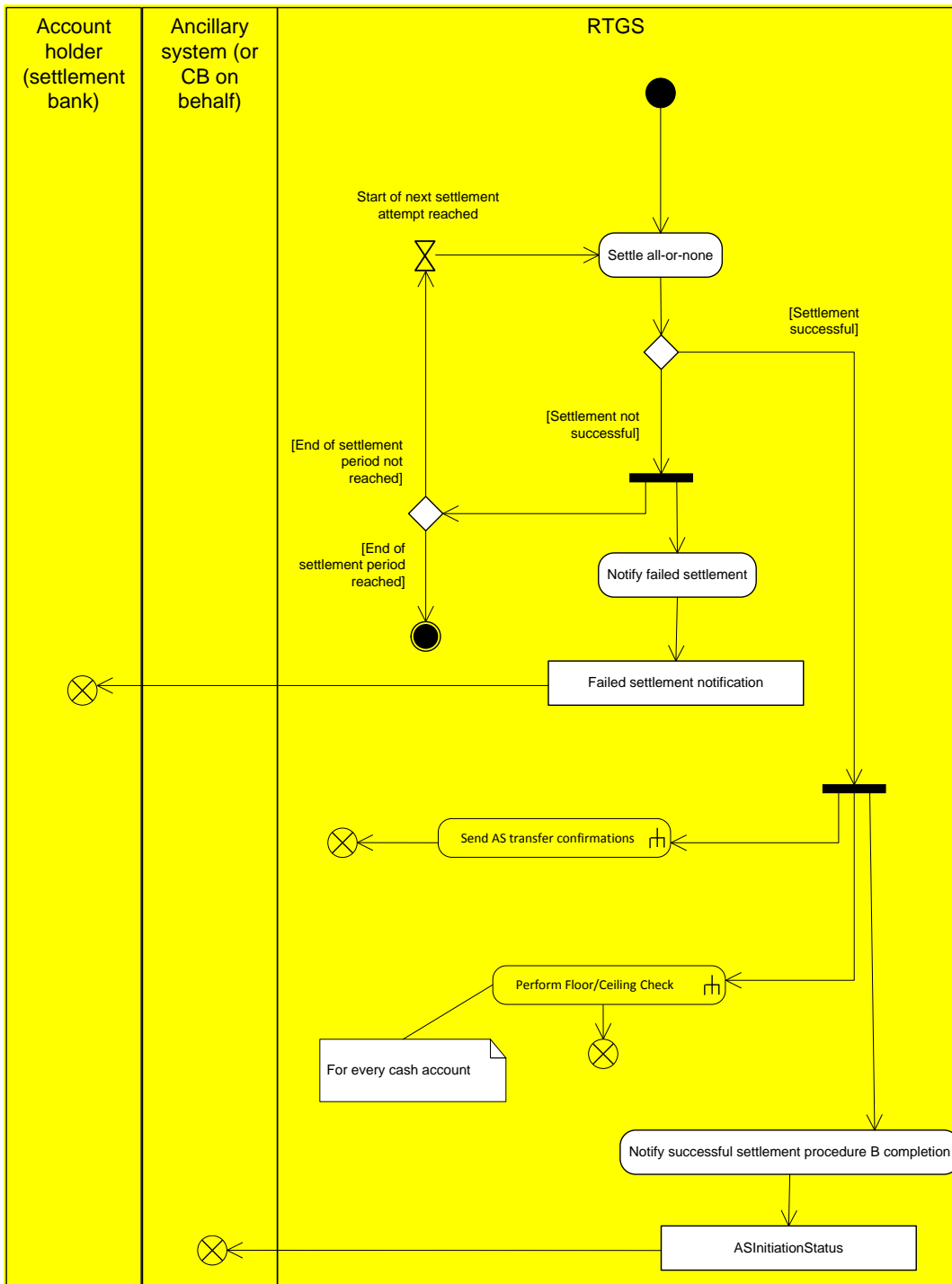


Figure 70 - Perform settlement procedure B

Settle all-or-none

The ancillary system transfers are sent to the all-or-none settlement, i.e. a dedicated settlement algorithm taking care to settle all debit and credit legs for settlement procedure B simultaneously.

[Settlement successful] When all debits are covered by the needed liquidity, all debits and credits are simultaneously settled by the optimisation settlement algorithm in one process. The processes “[Send ancillary](#)

[system transfer confirmations \[322\]](#) and [“Process floor and ceiling \[309\]”](#) are triggered in order to inform the settlement banks and to allow the related floor/ceiling actions. “Notify success settlement procedure B completion” process will take care for the information to ancillary systems.

[Settlement not successful]: Upon the first unsuccessful settlement attempt all settlement banks are notified by “Notify failed settlement” process.

[End of settlement period not reached] Unless the end of settlement period is reached the ancillary system transfers will have a settlement attempt with each launch of the optimisation algorithm.

[End of settlement period reached] At the end of settlement period either the process [„Notify guarantee fund mechanism initiation \[324\]”](#) is triggered (guarantee mechanism was opted for in reference data of the ancillary system) or, if no guarantee mechanism is envisaged, the process [“Terminate ancillary system processing \[328\]”](#) will be started.

Notify successful settlement procedure B completion

After successful settlement of all ancillary system transfers the submitter of the --- FEHLENDER LINK --- (ancillary system or CB on behalf) will receive an [ASTransferInitiation \(pain.998\) \[633\]](#) message.

Notify failed settlement process

The ancillary system and the settlement banks involved in the ancillary system batch message are informed on the queueing with a GUI broadcast.

10.3.3.4 Reverse previously settled debits

This process is triggered for settlement procedure A:

At end of settlement period of settlement procedure A with no guarantee account used.

At unsuccessful end of guarantee fund mechanism (usage not approved or usage was approved but liquidity on guarantee account was insufficient) for settlement procedures A or B.

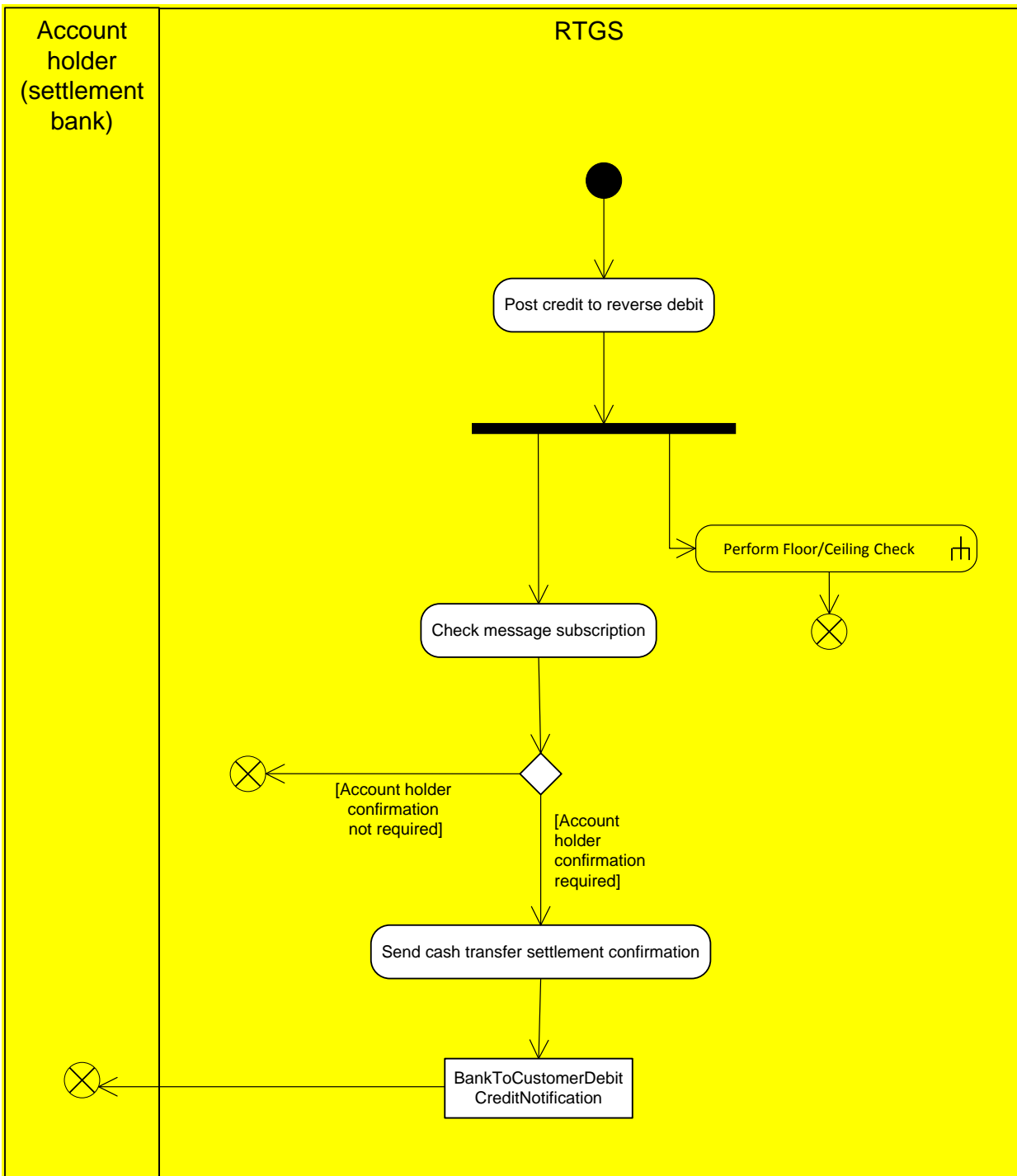


Figure 71 - Reverse debits

Post credit to reverse debit

In order to reverse previously settled debits RTGS creates credits with same amount for the impacted accounts. Consequently, following their settlement, the process for "Perform Floor/Ceiling Check" will be triggered as well as a check message subscription.

Check message subscription

Based on the message subscription the settlement banks as account holders are informed on the settlement of the reversal ancillary system transfers or not.

[Account holder confirmation not required] No information is sent to the settlement banks if not subscribed.

[Account holder confirmation required] If subscribed "Send cash transfer settlement confirmation" process will be started.

Send cash transfer settlement confirmation

The settlement banks receive a credit notification [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522] informing them on the settled credit.

10.3.3.5 Send ancillary system transfer confirmations

This process is triggered after each settlement of an ancillary system transfer except for the settlement of reversals for previously settled debits.

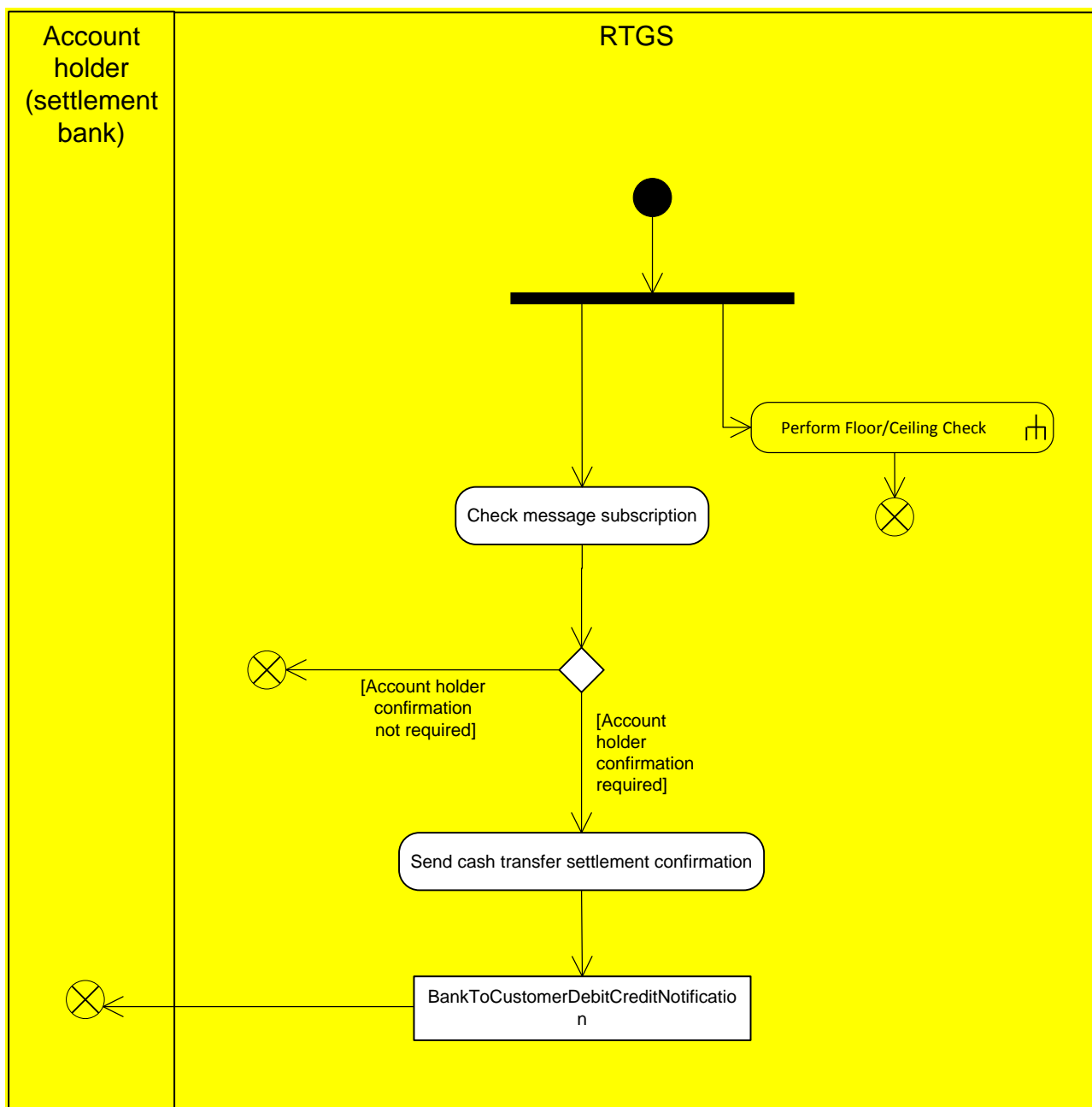


Figure 72 - Send ancillary system transfer confirmations

Check message subscription

Based on the message subscription the settlement banks as account holders are informed on the settlement of the ancillary system transfers or not.

[Account holder confirmation not required] No information is sent to the settlement banks if not subscribed.

[Account holder confirmation required] If subscribed “Send cash transfer settlement confirmation” process will started.

Send cash transfer settlement confirmation

The settlement banks receive a debit or credit notification [BankToCustomerDebitCreditNotification \(camt.054\)](#) [522] informing them on the settled credit.

10.3.3.6 Notify guarantee fund mechanism initiation

This process is started by RTGS at the end of settlement period (optional connected mechanism settlement period (“till”) must have been used) if the guarantee mechanism is opted for by the ancillary system in reference data for ancillary system transfers from ancillary system batch messages for settlement procedures A or B.

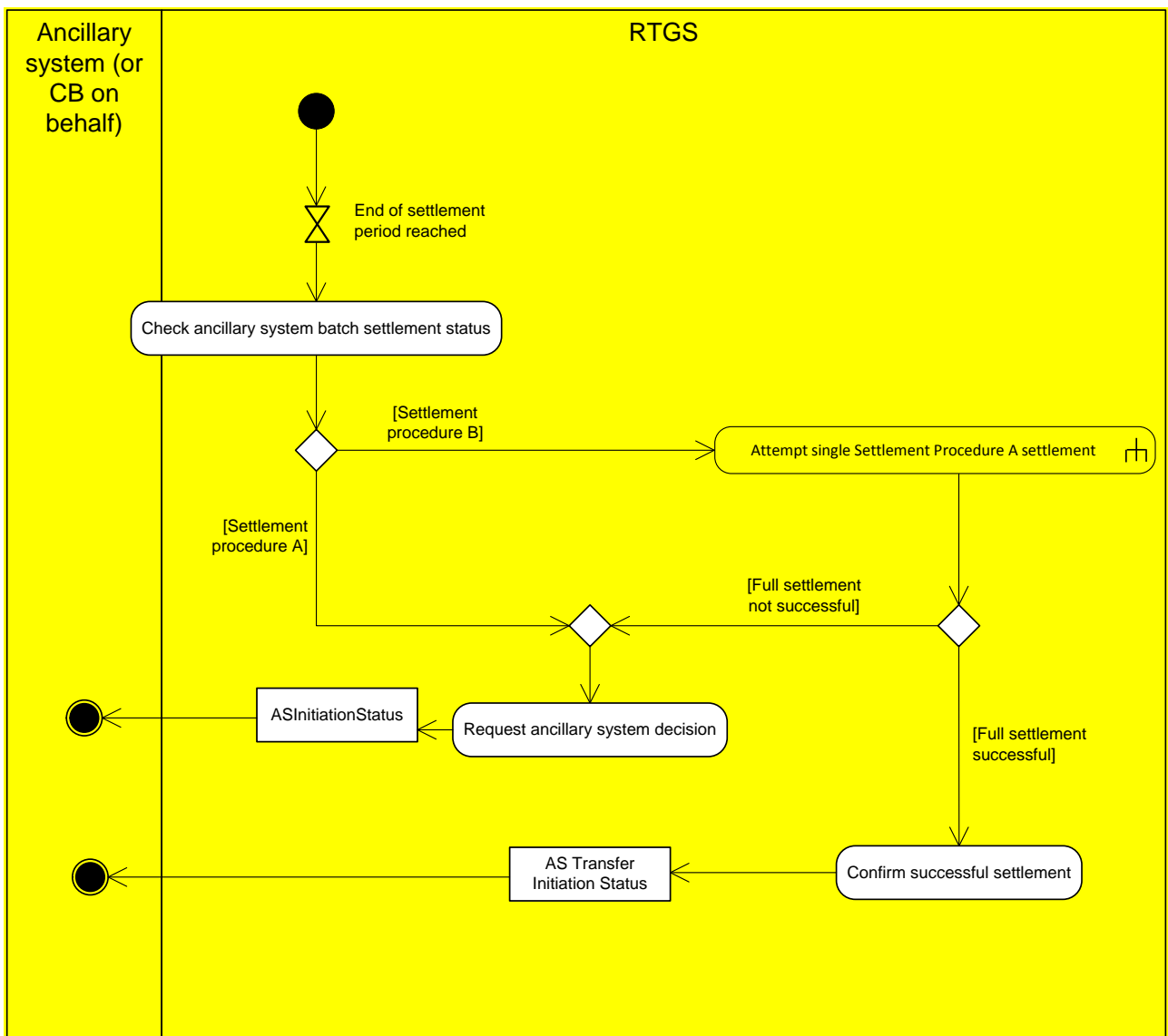


Figure 73 - Notify guarantee fund mechanism initiation

Check ancillary system batch settlement status

For identifying the next process step, it is needed to identify the underlying settlement procedure.

[Settlement procedure B] In order to single out the RTGS DCAs with a lack of liquidity the all-or-none scenario cannot be applied anymore. The ancillary system transfers are transferred into settlement procedure A and a single settlement attempt will be executed.

[Settlement procedure A] The process „Request ancillary system decision” is triggered by RTGS.

Attempt single settlement procedure A settlement

When being transferred from settlement procedure B to A, the ancillary system transfers on debit side will be sent to the regular settlement process.

[Full settlement successful] In case this single settlement attempt is successful for all debits also all credits are sent to regular settlement process and the process "Confirm successful settlement" is started.

[Full settlement not successful] After debiting the RTGS DCAs with sufficient liquidity the process „Request ancillary system decision" is started.

Confirm successful settlement

After full settlement of all ancillary system transfers of the initial ancillary system batch message the ancillary system receives an [ASInitiationStatus \(pain.998\)](#) [620] message indicating successful settlement.

Request ancillary system decision

After the successful settlement of debits being covered by liquidity, the ancillary system is notified with an [ASInitiationStatus \(pain.998\)](#) [620] message on those debits not being covered by liquidity and the pertaining amount. The [ASInitiationStatus \(pain.998\)](#) [620] has the decision indicator tag being set to true.

10.3.3.7 Trigger guarantee fund mechanism

This process is triggered when the ancillary system responds to the decision request to use or not the guarantee fund mechanism.

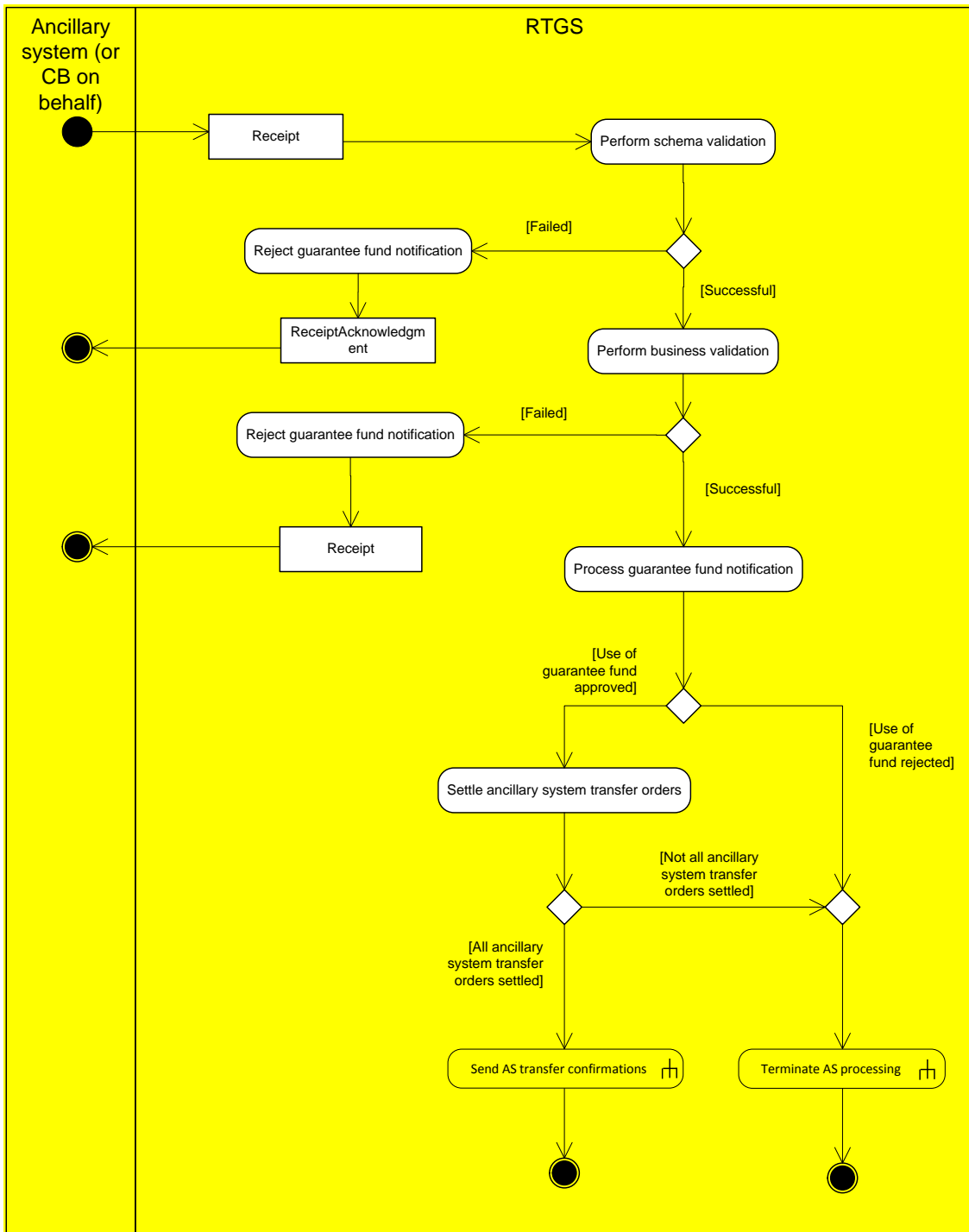


Figure 74 - Trigger guarantee fund mechanism

Perform schema validation

The *Receipt* message is validated against the pertaining schema file.

[Failed] If at least one validation error occurs, the ancillary system receives a [ReceiptAcknowledgment \(admi.007\)](#) [391] indicating the error.

[Successful] If no validation error occurs the receipt message will be passed on to business validation.

Perform business validation

The *Receipt* message is validated against the business validation rules.

[Failed] If the validations are carried with failure, the sender of the receipt message receives a [Receipt \(camt.025\)](#) [474] indicating the error.

[Successful] Once schema and business validations are successfully performed, “Process guarantee fund notification” starts.

Process guarantee fund notification

Upon reception of the valid *Receipt* message sent by the ancillary system, RTGS processes the response.

[Use of guarantee fund rejected] In case the ancillary system rejects the use of the guarantee fund mechanism the sub process “Terminate ancillary system processing” will be triggered.

[Use of guarantee fund approved] After approval of the use of guarantee fund mechanism the failing RTGS DCAs are substituted by the guarantee account and debits are sent to standard settlement. If all debits are settled also all credits are settled by standard settlement.

[Not all ancillary system transfers settled] If the liquidity on the guarantee account is insufficient to settle the remaining debits the process “Terminate ancillary system processing” is triggered.

[All ancillary system transfers settled] In case all ancillary system transfers were correctly settled, sub process “Send ancillary system transfer confirmations” is triggered.

10.3.3.8 Terminate ancillary system processing

This process is triggered for settlement procedures A and B by:

- | The rejection due to revocation by the CB responsible for the ancillary system or
- | The rejection at end of settlement period without guarantee mechanism or
- | The rejection after guarantee fund mechanism (not approved or unsettled) or
- | The rejection at EoD.

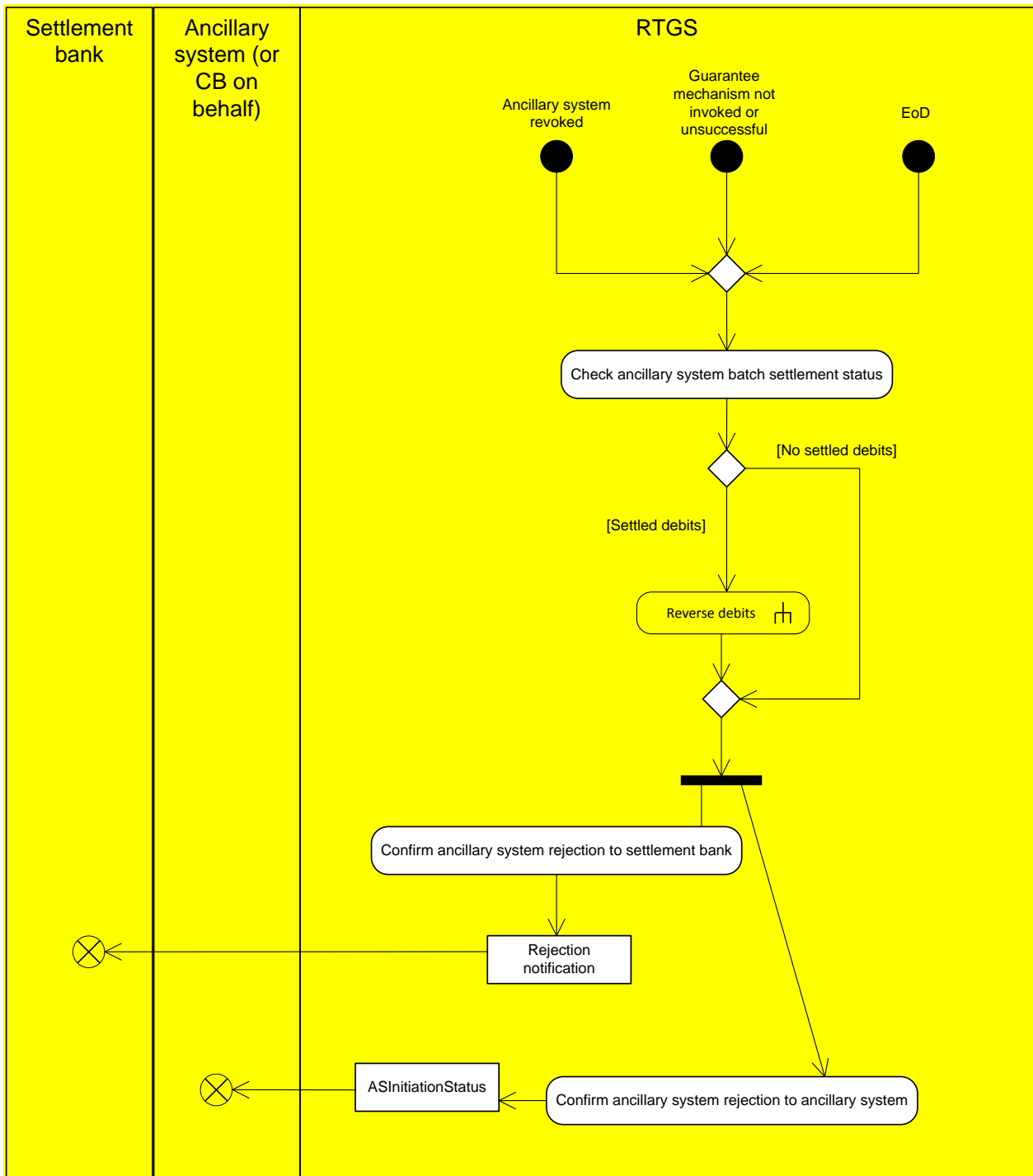


Figure 75 - Terminate ancillary system file processing

Check ancillary system batch settlement status

[Settled debits]: In case one or more debits are already settled they are subject to reversal by sub process "Reverse debits".

[No settled debits]: If no debit is settled yet or reversal is finalised, the processes to inform the settlement banks and ancillary system is started.

Confirm ancillary system rejection to settlement bank

The settlement banks receive a broadcast as rejection notification informing them on the rejection reason of the pertaining ancillary system transfers.

Confirm ancillary system rejection to ancillary system

Ancillary systems are informed about the rejection and related reason of their ancillary system transfers with an *ASInitiationStatus* message.

10.3.3.9 Execute start of procedure for ancillary system settlement procedures C and D

This process is triggered when the ancillary system opens a settlement procedure (only possible for settlement procedure C) or the procedure is started automatically at start of ancillary system business at 19:30.

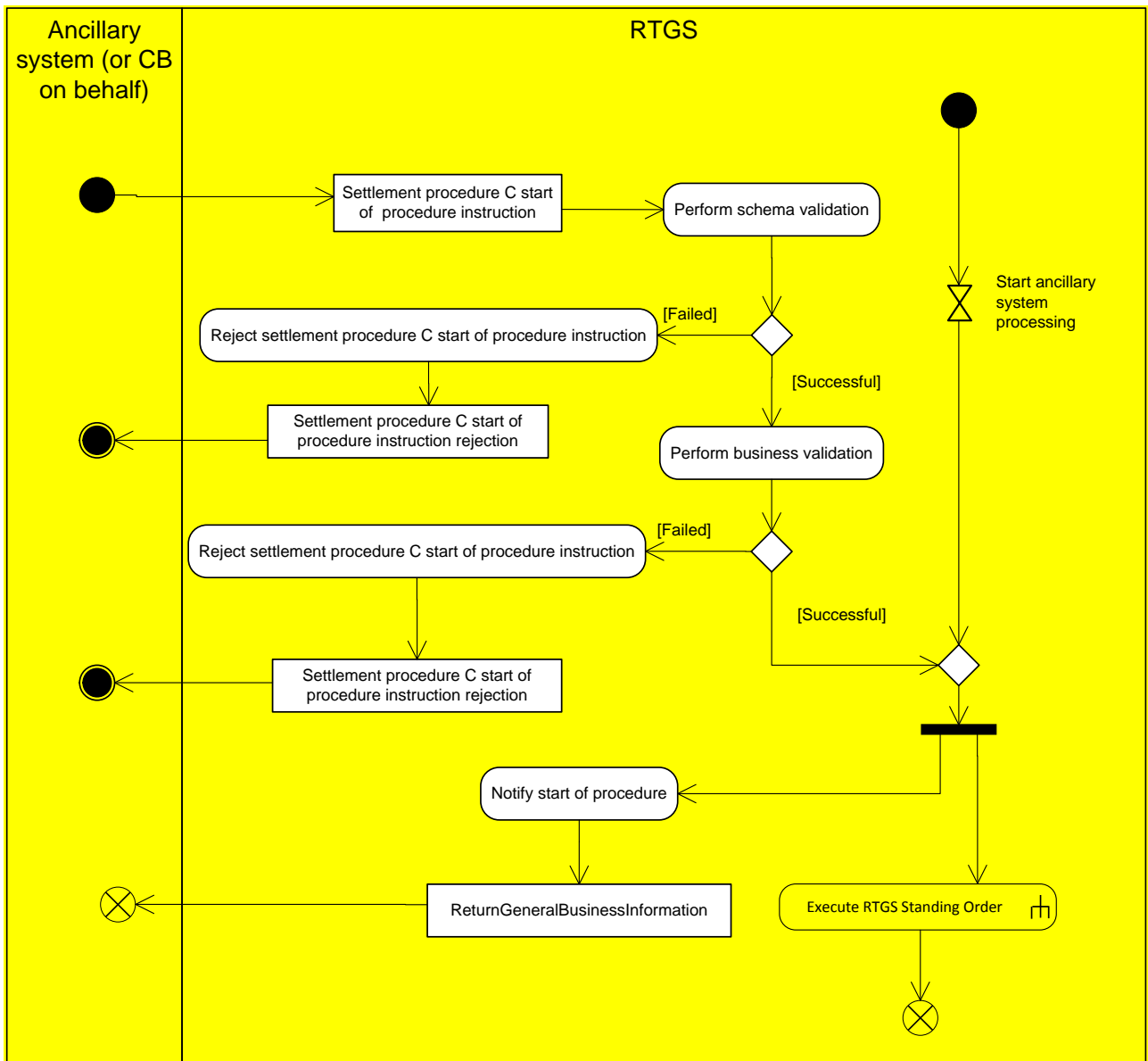


Figure 76 - Execute ancillary system settlement procedure C and D - start of procedure

Settlement procedure C start of procedure instruction

The ancillary system instructs the start of procedure with [ReturnGeneralBusinessInformation \(camt.021\) \[458\]](#) message.

Perform schema validation

The validity of incoming message according to the schema is validated by RTGS.

[Failed] If the schema validation fails a [ReceiptAcknowledgement \(admi.007\) \[391\]](#) is sent back to the ancillary system informing about the reasons.

[Successful] After successful schema validation the business validation is performed.

Perform business validation

The pertaining business rules for the message are being checked by RTGS.

[Failed] If one or more rules are disrespected, the “Reject settlement procedure C start of procedure instruction” process is triggered. A *Receipt* message indicating the error is returned to the ancillary system.

[Successful] After successful business validation or after automatic start of mandatory procedure at start time for ancillary system processing at 19:30 the procedure starts. The process to “Notify start of procedure” and sub process “Execute RTGS standing order” start.

Notify start of procedure

For the automatic start of mandatory procedure at 19:30 the ancillary system is informed via [ReturnGeneral-BusinessInformation \(camt.021\)](#) [▶ 458]. This is not true a manual start of procedure.

10.3.3.10 Execute start of cycle for settlement procedure C and D

This process is triggered when the ancillary system instructs the start of settlement cycle.

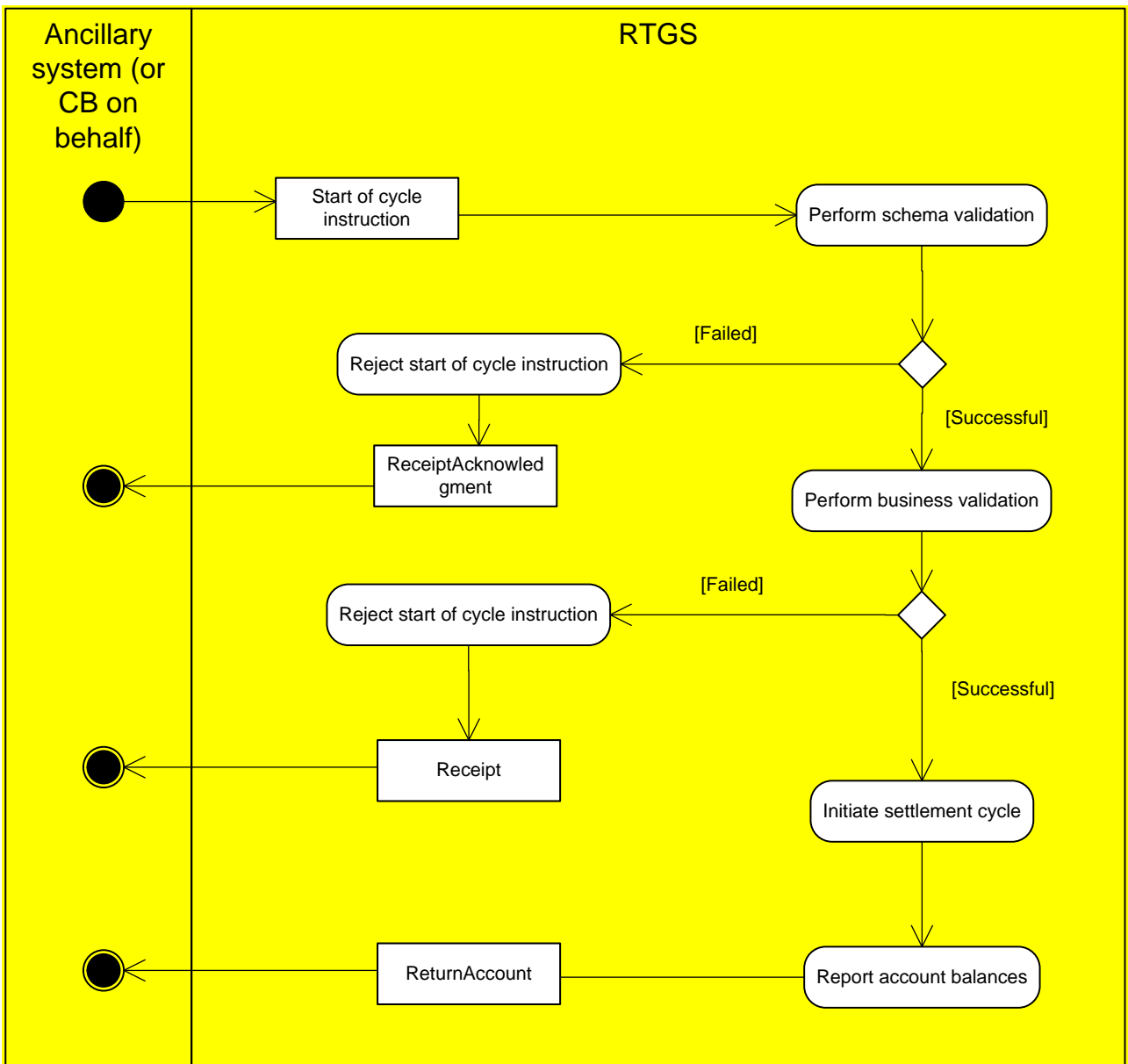


Figure 77 - Execute settlement procedure C and D - start of cycle

Start of cycle instruction

The ancillary system sends a [ReturnGeneralBusinessInformation \(camt.021\)](#) [458] indicating the start of settlement cycle within mandatory or optional procedure.

Perform schema validation

The validity of incoming message according to the schema is validated by RTGS.

[Failed] If the schema validation fails a [ReceiptAcknowledgement \(admi.007\)](#) [391] is sent back to the ancillary system informing about the reasons by "Reject start of cycle instruction" process.

[Successful]: After successful schema validation the business validation is performed.

Perform business validation

The pertaining business rules for the message are being checked by RTGS.

[Failed] If one or more rules are disrespected, the “Reject start of cycle instruction” process is triggered and the error is returned with a receipt to the ancillary system.

[Successful] After successful business validation “Initiate settlement cycle process” starts.

Initiate settlement cycle

With the start of cycle successfully received by RTGS, the liquidity on the ancillary system technical account is blocked and the ancillary system is informed by the “Report account balances” process.

Report account balances

The ancillary system receives a [ReturnAccount \(camt.004\)](#) [397] message informing on the liquidity blocked.

Procedure C: The blocked amount (different from zero) on the sub-accounts dedicated to the ancillary system.

Procedure D:

- First cycle: Liquidity from standing liquidity transfer orders and immediate liquidity transfer orders executed since the start of procedure.

- Following cycles: Liquidity from immediate liquidity transfer orders executed since the last end of cycle.

10.3.3.11 Execute end of cycle for settlement procedure C and D

This process is triggered when the ancillary system instructs the end of settlement cycle.

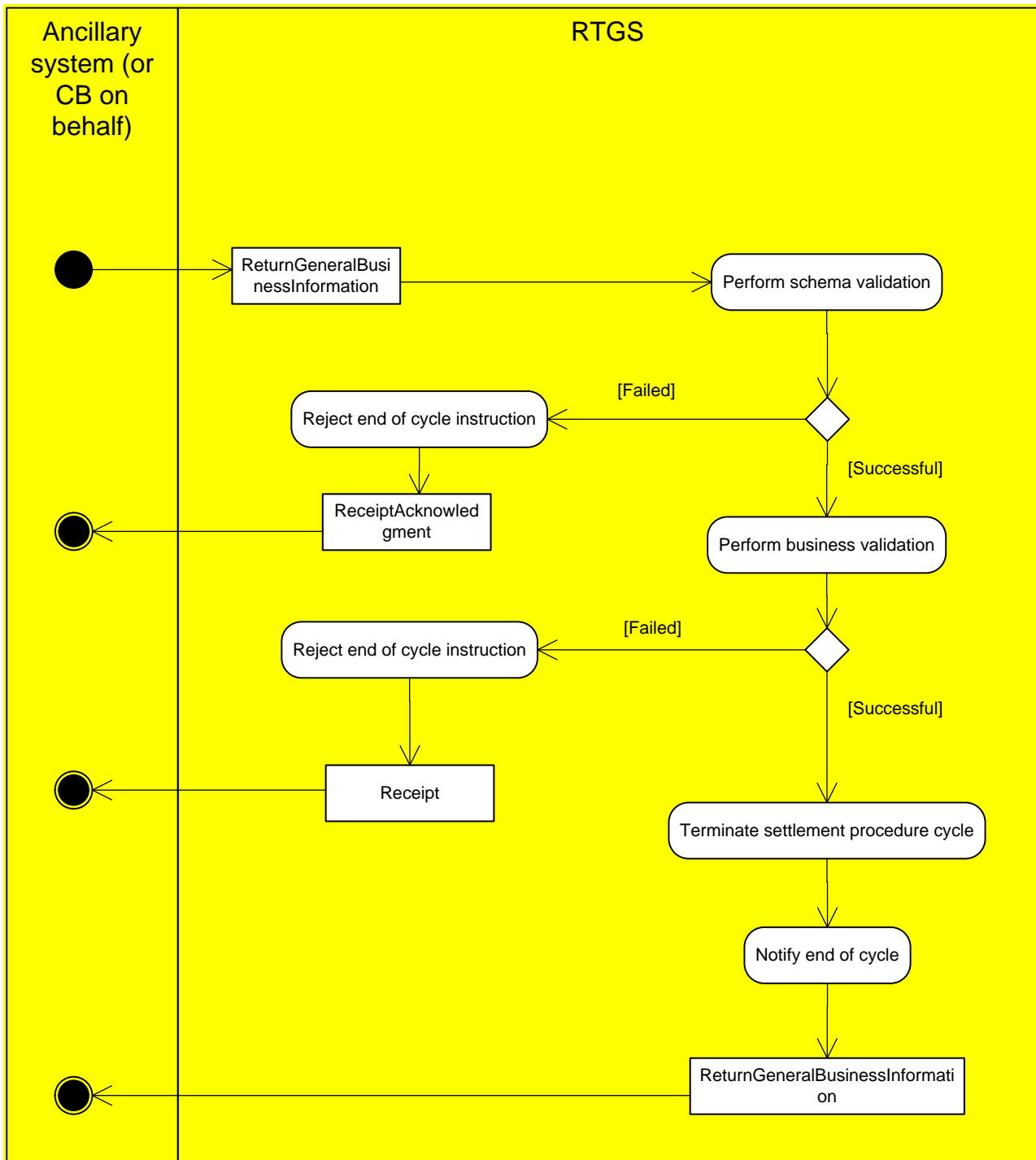


Figure 78 - Execute settlement procedure C and D - end of cycle

End of cycle instruction

The ancillary system sends a [ReturnGeneralBusinessInformation \(camt.021\)](#) [458] indicating the end of settlement cycle within mandatory or optional procedure.

Perform schema validation

The validity of incoming message according to the schema is validated by RTGS.

[Failed] If the schema validation fails a [ReceiptAcknowledgement \(admi.007\)](#) [391] is sent back to the ancillary system informing about the reasons by “Reject end of cycle instruction” process.

[Successful] After successful schema validation the business validation is performed.

Perform business validation

The pertaining business rules for the message are being checked by RTGS.

[Failed] If one or more rules are disrespected, the “Reject end of cycle instruction” process is triggered and the error is returned with a [Receipt \(camt.025\)](#) [474] to the ancillary system.

[Successful] After successful business validation “Terminate settlement procedure cycle” starts.

Terminate settlement procedure cycle

With the end of cycle successfully received by RTGS, queued ancillary system transfers will be rejected and stored immediate liquidity transfer orders received during the cycle are processed. If the cycle is closed at EoD automatically, after rejection neither liquidity transfers are processed nor will the ancillary system be informed about the end of cycle.

Notify end of cycle

A [ReturnGeneralBusinessInformation \(camt.021\)](#) [458] is sent to the ancillary system informing about the end of settlement cycle.

10.3.3.12 End of procedure for ancillary system settlement procedure C and D

This process is triggered whenever either the RTGS component initiates the end of procedure process at 18:00h of a business day or the ancillary system sends an end of procedure message [ReturnGeneralBusinessInformation \(camt.021\)](#) [458] (or optional in U2A via GUI) to the RTGS component.

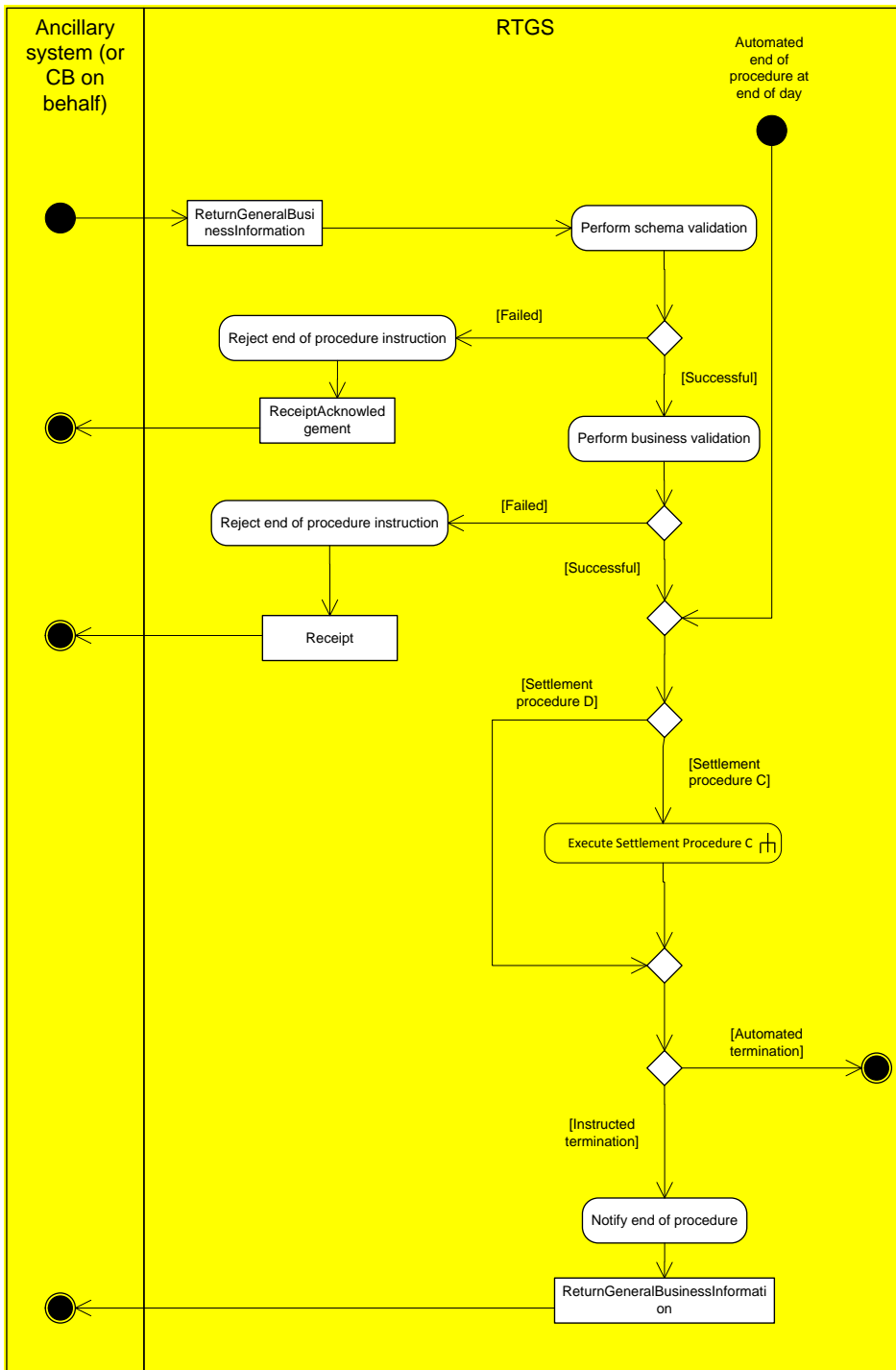


Figure 79 - Execute settlement procedure C and D - end of procedure

Perform schema validation

The *ReturnGeneralBusinessInformation* message is validated against the pertaining schema file.

[Failed] If at least one validation error occurs, the ancillary system receives a [ReceiptAcknowledgement \(admi.007\)](#) [391] indicating the error.

[Successful] If no validation error occurs the message will be passed on to business validation.

Perform business validation

The *ReturnGeneralBusinessInformation* message is validated against the business validation rules.

[Failed] If the validations are carried out with failure, the sender of the message receives a [Receipt \(camt.025\)](#) [474] indicating the error.

[Successful] Once schema and business validations are successfully performed, the process “Terminate settlement procedure” starts.

[Settlement procedure C] The remaining liquidity on sub-accounts is transferred back to the ancillary system settlement banks’ RTGS DCAs by “Execute settlement procedure C” sub-process.

[Settlement procedure D] After successful settlement of the retransfer from the sub-accounts of procedure C or in case of procedure D the procedure is marked as closed.

[Automated termination] If the end of procedure was initiated by RTGS at EoD, no information is sent to the ancillary system.

[Instructed termination] The sender of the A2A instruction to close the procedure is informed about closure of settlement procedure by “Notify end of procedure” process.

Notify end of procedure

The closure of the procedure is notified to the ancillary system or CB on behalf with [ReturnGeneralBusinessInformation \(camt.021\)](#) [458] message.

10.3.3.13 Execute settlement in ancillary system settlement procedure C

This process is triggered when a settlement for an ancillary system transfer for ancillary system settlement procedure C different from standing liquidity transfer orders takes place (settlement phase during a running cycle, cross ancillary system transfer impacting a sub-account or an immediate liquidity transfer order).

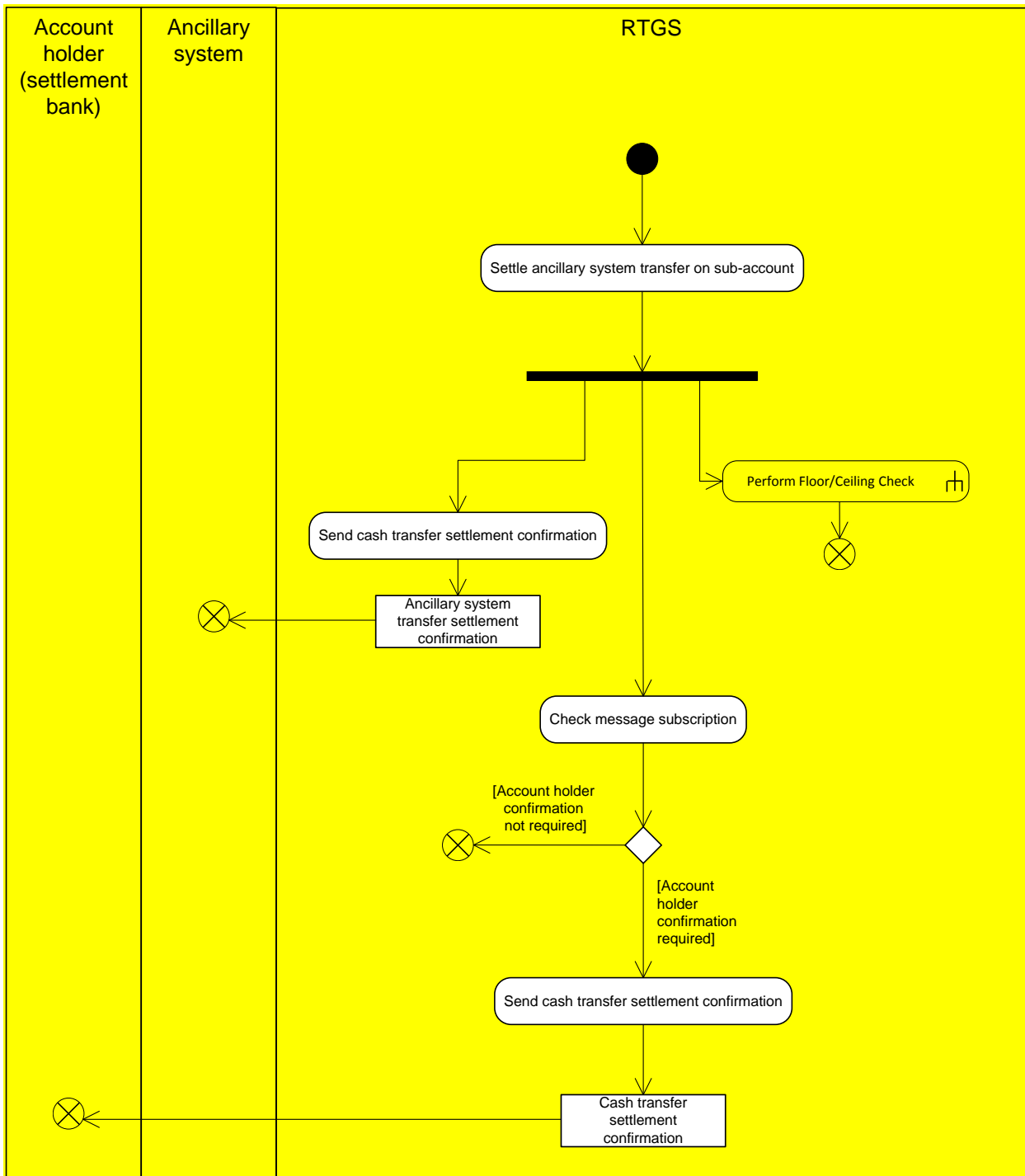


Figure 80 - Execute settlement procedure C

Settle ancillary system transfer on subaccount

The settlement of an ancillary system transfer for immediate liquidity transfer orders or settlement during the settlement phase takes place. Following the standard RTGS processing, sub process "Perform floor/ceiling check" is started.

Send cash transfer settlement confirmation

The ancillary system is notified on the settlement with an ancillary system transfer settlement confirmation ([ReturnAccount \(camt.004\)](#) [▶ 397] or [ASTransferNotice \(pain.998\)](#) [▶ 610]). Please see [Settlement on dedicated liquidity accounts \(ancillary system settlement procedure C and ancillary system settlement procedure D\)](#) [▶ 151] to understand which of the messages is used.

Check message subscription

Depending on the pertaining message subscription, the settlement banks receive confirmations about the settlement.

[Account holder confirmation required] If subscribed, the settlement bank receives a cash transfer settlement confirmation sent out by the process "Send cash transfer settlement confirmation".

[Account holder confirmation not required] In case there is no subscription, no settlement confirmation is sent to the settlement banks.

Send cash transfer settlement confirmation

The settlement banks receive a [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 522] as cash transfer settlement confirmation.

10.4 RTGS EoD processing

10.4.1 Reject payments (EoD)

If queued payments cannot be cleared during the ongoing optimisation procedures and are still queued by the EoD due to lack of liquidity (including urgent or high reservation of liquidity) or insufficient limits, these payments will be rejected during EoD processing.

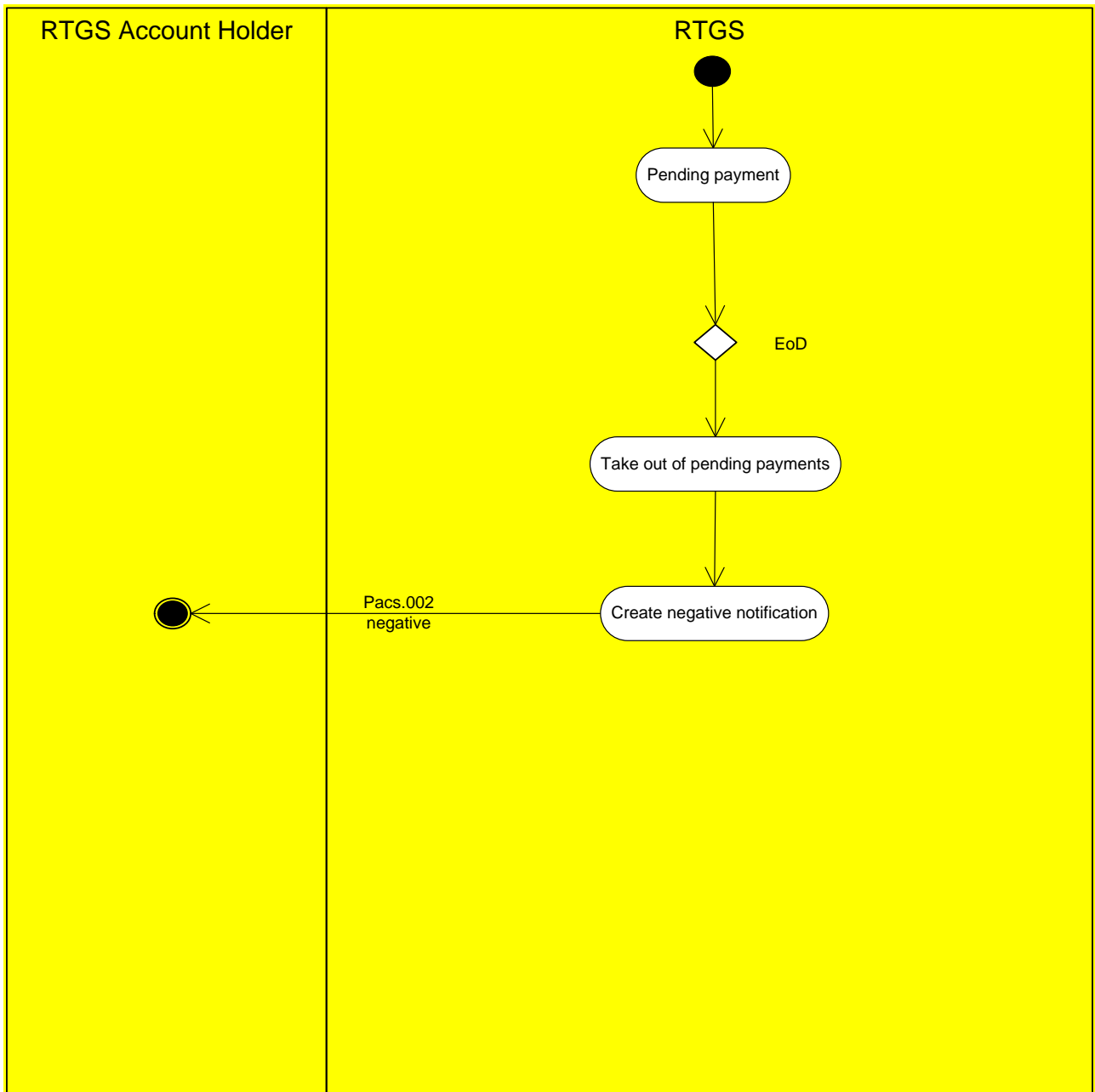


Figure 81 - Reject payments (EoD)

The sending RTGS Account Holder of the related inbound payment message will be notified by negative [PaymentStatusReport \(pacs.002\) \[568\]](#).

10.5 Revalidate warehoused payments at SoD

Basics

Warehoused payments are stored in RTGS with a certain payment status “warehoused”. They are validated every day between submission day and execution (value) day. The validation process starts when business day event SoD has been reached.

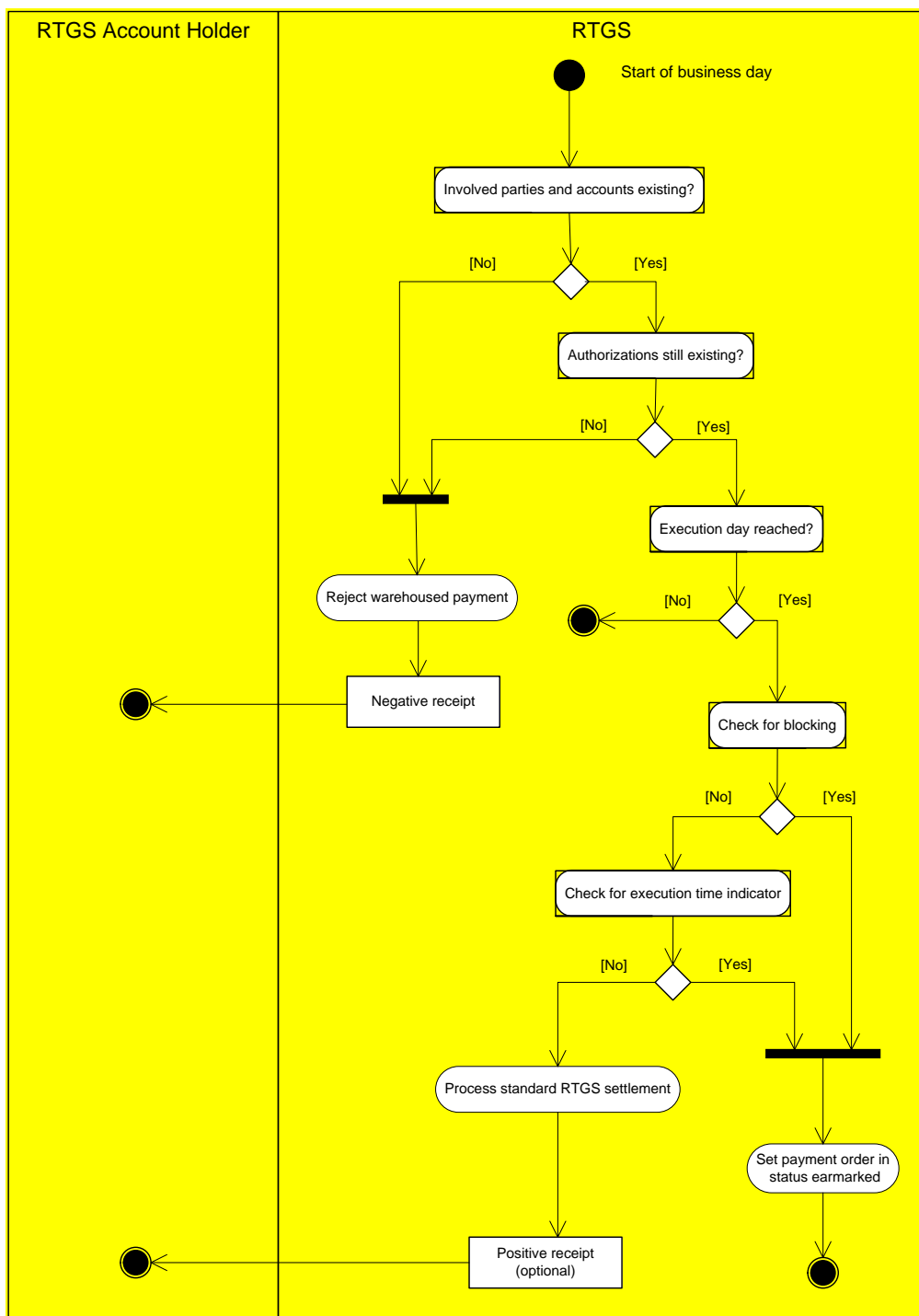


Figure 82 - Revalidate warehoused payments at SoD

Rules

The following validations are carried out at SOD:

- ! Check if the involved parties and accounts still exist and have not been closed meanwhile.
- ! Check if the authorisation on the involved accounts still exists.
- ! Check if the current business day is the intended settlement day.

If yes:

- | Check if any involved party or account is blocked.
- | Check for execution from time indicator (see chapter [Definition of execution time](#) [▶ 81])
- | Further checks as described in chapter [Entry disposition](#) [▶ 125]

Technical validations like schema validations are only carried out on message level on the submission day. The same is valid for the duplicate payment check. They are not repeated at start of day.

Processing on the intended settlement day

On the intended settlement date with the start of the processing time of the respective order type (e.g. liquidity transfer, credit transfers and direct debit) the warehoused payments are processed like described in standard RTGS settlement (see chapter [Standard RTGS settlement](#) [▶ 305]).

Exception: Warehoused payments with a set execution from time indicator which has not been reached are set to status "earmarked".

10.6 Reference data management

10.6.1 Maintain local reference data object

10.6.1.1 Maintain reservation

This is a general description of the RTGS process "Maintain reservation". For functional description please see chapter [Functionalities](#) [▶ 58]. The submitting actor sends a

- | modify reservation request (see chapter [ModifyReservation \(camt.048\)](#) [▶ 492]) to RTGS in which he instructs RTGS to create a new or modify an existing reservation or a
- | delete reservation request (see chapter [DeleteReservation \(camt.049\)](#) [▶ 494]) in which he instructs RTGS to delete an existing reservation.

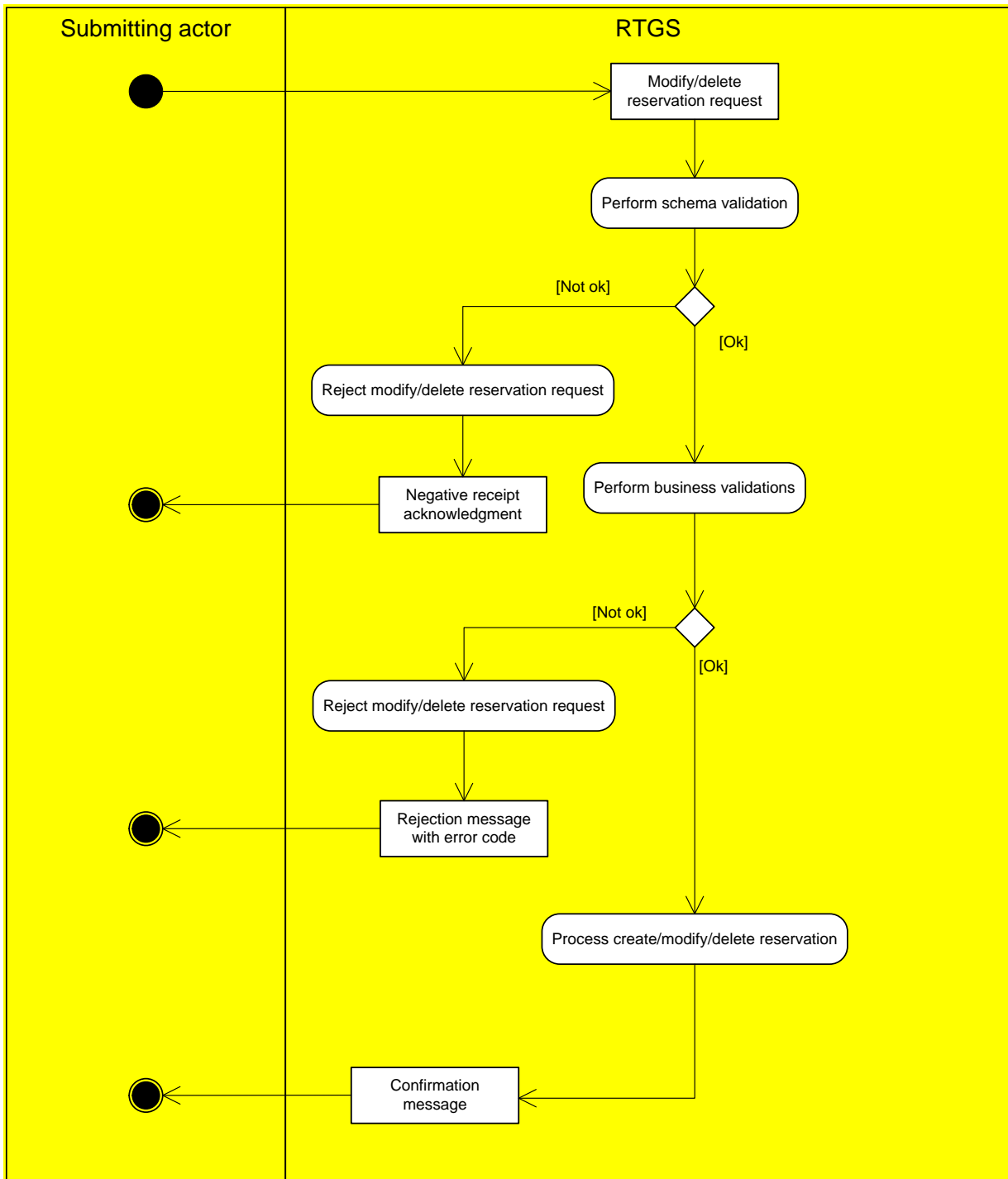


Figure 83 - Maintain reservation RTGS

Schema validation

As a first step within the respective component, the process “Perform schema validation” performs the schema validation of the respective [Modify/delete reservation request] schema.

[Not ok] In case of error, a negative receipt acknowledgment (see chapter [ReceiptAcknowledgement \(admi.007\)](#) [391]) is sent to the submitting actor on mandatory basis.

[Ok] If the schema validation was successful, the request is sent to the business validation.

Business validation

In a second step (i.e. after successful schema validation), the component performs the business validations (all business rules which are relevant for the [Modify/delete reservation request] including access rights). The validation procedure continues with business validations to the extent possible even after the business validation identifies one or more errors. It reports all identified validation errors.

[Not ok] In case of error, a rejection message with error code ([Receipt \(camt.025\)](#) [474]) is sent to the submitting actor on mandatory basis.

[Ok] If validation was successful, the request is being processed by RTGS.

Process create/modify/delete reservation

The request is processed by RTGS.

[Ok] In case of successful processing a respective confirmation message ([Receipt \(camt.025\)](#) [474]) is sent to the submitting actor.

Note: Please see chapters [Functionalities](#) [58] and [Liquidity reservation and management process](#) [193] for details on the reservation feature.

10.6.1.2 Maintain limit

This is a general description of the RTGS process "Maintain limit". For functional description please see chapter [Functionalities](#) [58]. The submitting actor sends a

modify limit request [ModifyLimit \(camt.011\)](#) [445] to RTGS in which it instructs RTGS to modify an existing limit or a

delete limit request [DeleteLimit \(camt.012\)](#) [448] in which he instructs RTGS to delete an existing limit.

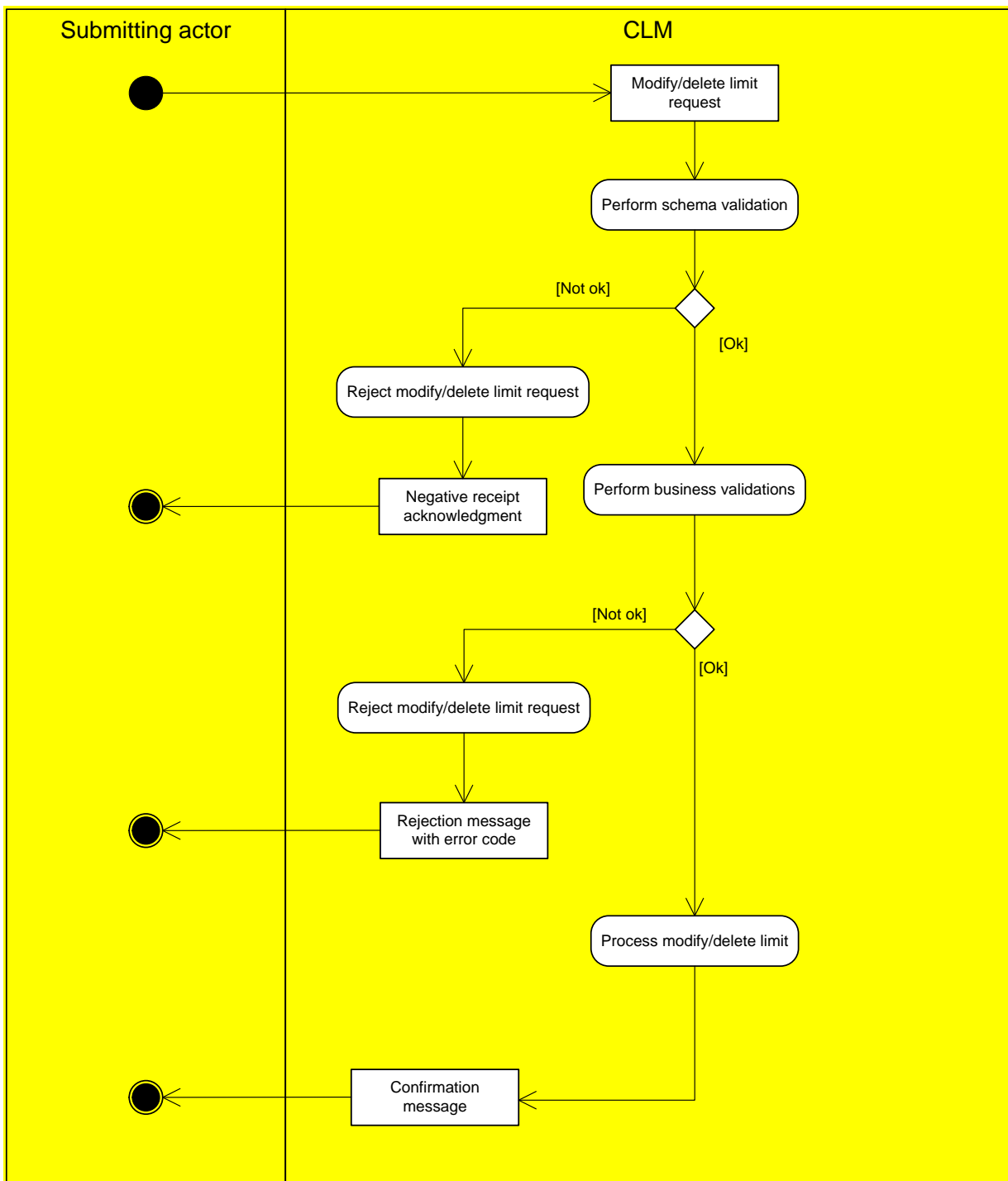


Figure 84 - Maintain limit

Schema validation

As a first step within the respective component, the process “Perform schema validation” performs the schema validation of the respective [Modify/delete limit request] schema.

- I **[Not ok]** In case of error, a negative receipt acknowledgment ([ReceiptAcknowledgement \(admi.007\) \[391\]](#)) is sent to the submitting actor on mandatory basis.

[Ok] If the schema validation was successful, the request is sent to the business validation.

Business validation

In a second step (i.e. after successful schema validation), the component performs the business validations (all business rules which are relevant for the [Modify/delete limit request] including access rights). The validation procedure continues with business validations to the extent possible even after the business validation identifies one or more errors. It reports all identified validation errors.

[Not ok] In case of error, a rejection message [Receipt \(camt.025\)](#) [▶ 474] with error code is sent to the submitting actor on mandatory basis.

[Ok] If validation was successful, the request is being processed by RTGS.

Process modify/delete limit

The request is processed by RTGS.

[Ok] In case of successful processing a respective confirmation message [Receipt \(camt.025\)](#) [▶ 474] is sent to the submitting actor.

Note: Please see chapters [Functionalities](#) [▶ 58] and [Limits](#) [▶ 200] for details on the limit feature.

10.7 Information services

10.7.1 Execute query

This is a general process description for query requests to RTGS in A2A mode. In order to retrieve information from a component, the submitting actor sends a query request message via ESMIG to the relevant component. Chapter [Query management for RTGS](#) [▶ 231] describes the respective business scope. Concerning information on CB specific queries please see chapter [Query management - specific functions for CBs](#) [▶ 290].

The following activity diagram provides respective processes in the context of the RTGS component:

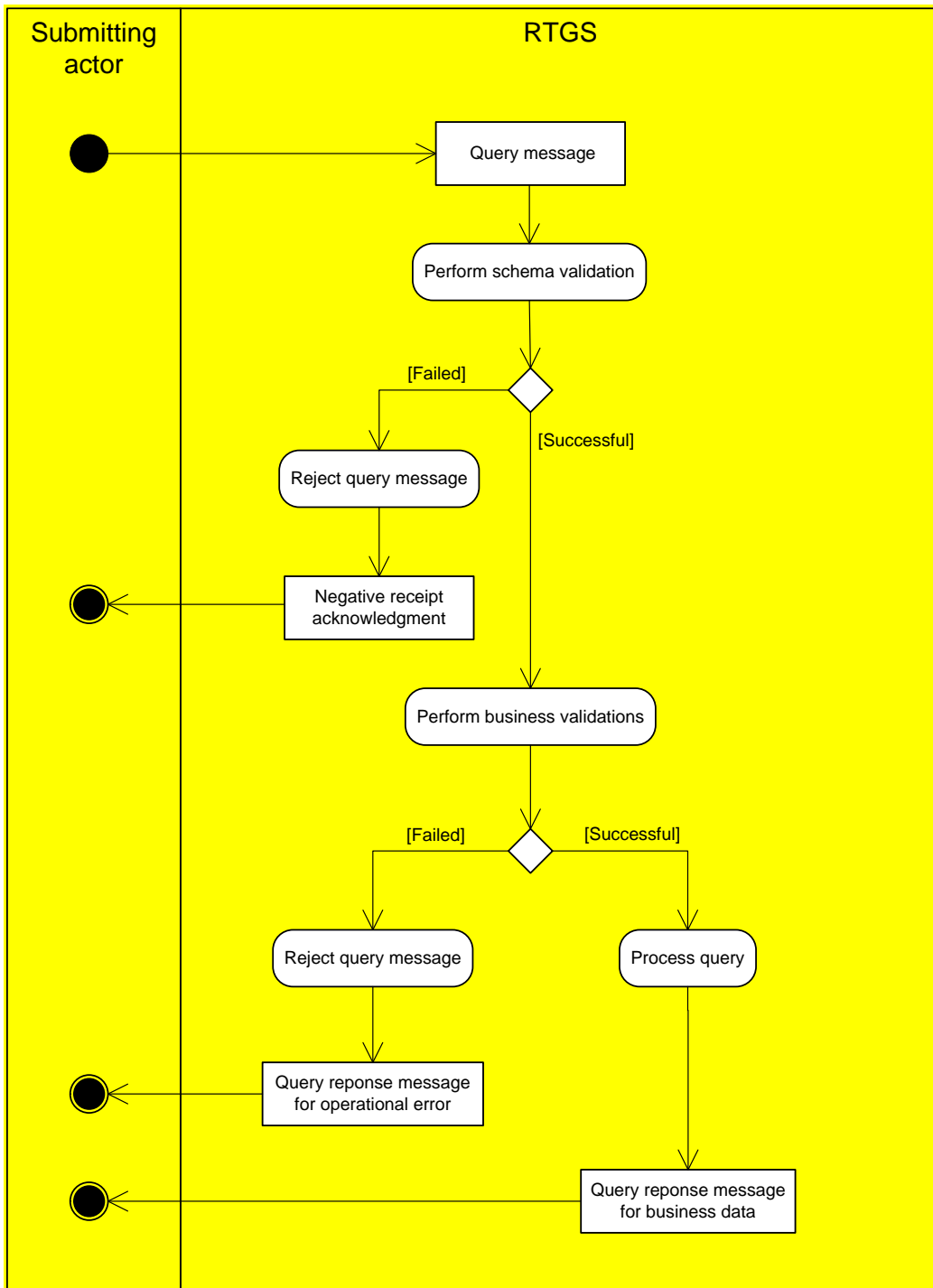


Figure 85 - RTGS send query

Schema validation

As a first step within the respective component, the process “Perform schema validation” performs the schema validation of the respective [Query request message] schema.

[Failed] The process “Reject query message” sends a [ReceiptAcknowledgement \(admi.007\)](#) [391] to the submitting actor including all information regarding the reasons for failed validation.

[Successful] The process triggers the business validation.

Business validation

In a second step (i.e. after successful schema validation), the component performs the business validations (all business rules which are relevant for the respective query including access rights). The validation procedure continues with business validations to the extent possible even after the business validation identifies one or more errors. It reports all identified validation errors.

[Failed] The process “Reject query message” sends a rejection message that includes the reasons for failing [Query response message for operational error] (as indicated in table A2A messages for query processing to the submitting actor).

[Successful] The process “Execute query” extracts the required business data, creates the [Query response message for business data] and sends the response via ESMIG to the submitting actor.

The following table provides a detailed list of A2A messages for query processing.

Query type	Query request message	Query response message for operational error	Query response message for business data
Account balance query	GetAccount (camt.003) [394]	ReturnAccount (camt.004) [397]	ReturnAccount (camt.004) [397]
Account statement query	ReportQueryRequest (admi.005) [389]	ReceiptAcknowledgement (admi.007) [391]	BankToCustomerStatement (camt.053) [505]
Audit trail for RTGS query	AuditTrailQuery (camt.097) [552]	AuditTrailReport (camt.098) [554]	AuditTrailReport (camt.098) [554]
Cash transfer query	GetTransaction (camt.005) [421]	ReturnTransaction (camt.006) [431]	ReturnTransaction (camt.006) [431]
Current limits query	GetLimit (camt.009) [440]	ReturnLimit (camt.010) [442]	ReturnLimit (camt.010) [442]
Current reservations query	GetReservation (camt.046) [487]	ReturnReservation (camt.047) [489]	ReturnReservation (camt.047) [489]
Event query	GetBusinessDayInformation (camt.018) [451]	ReturnBusinessDayInformation (camt.019) [453]	ReturnBusinessDayInformation (camt.019) [453]
System time query	GetBusinessDayInformation (camt.018) [451]	ReturnBusinessDayInformation (camt.019) [453]	ReturnBusinessDayInformation (camt.019) [453]

Table 134 - A2A messages for query processing

10.7.2 Receive report

This is a general description of the RTGS process “Receive report” in push mode. RTGS uses reports to periodically provide RTGS Actors with a defined set of data according to their data scope and access rights.

The chapter [RTGS report generation](#) [228] describes the respective business scope.

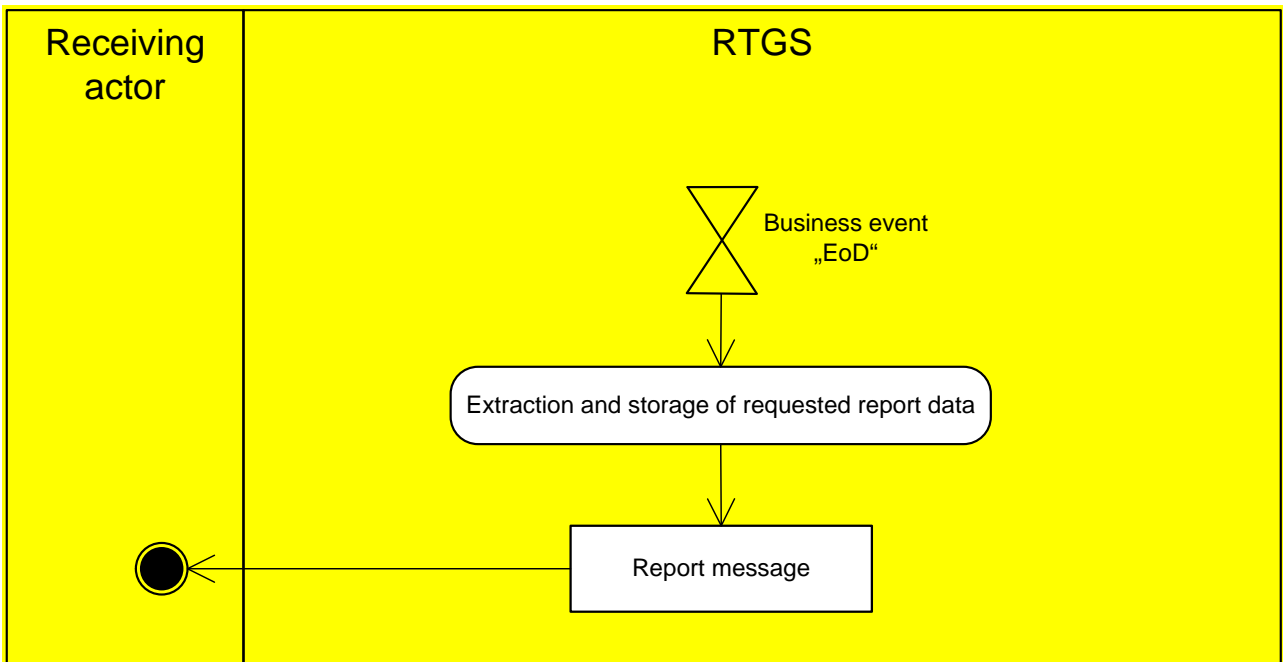


Figure 86 - RTGS receive report

The defined business EoD triggers the process “Extraction and storage of requested report data”. For statement of accounts it uses the report configuration in order to provide all necessary reports on the basis of the configured RTGS DCA. For general ledger files it provides the report on mandatory basis to the CB (see chapter [RTGS General Ledger](#) [289]). The RTGS component creates the report, including the execution of necessary calculations and storing the report for further processing. RTGS sends the [Report message] via ESMIG to the receiving actor when a report configuration for the report is set up.

Report name	ISO message	ISO code
Statement of accounts	BankToCustomerStatement	BankToCustomerStatement (camt.053) [505]
General ledger (CB only)	BankToCustomerStatement	BankToCustomerStatement (camt.053) [505]

Table 135 - A2A messages for receiving reports

10.7.3 Receive system notification

This is a general description of the RTGS process “Receive system notification”. RTGS uses system notifications to regularly provide RTGS Actors with a defined set of business events.

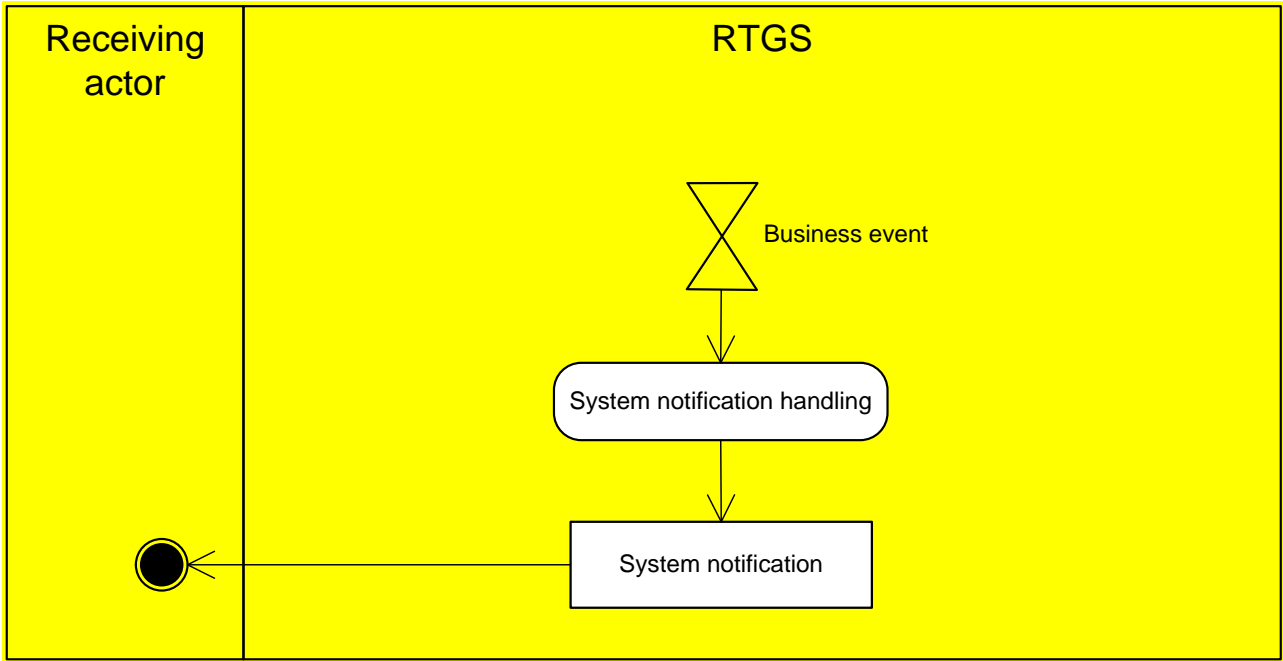


Figure 87 - RTGS receive system notification

The defined business events trigger the process “System notification handling”. RTGS sends the [System notification] via ESMIG to the receiving actor based on the respective message subscription in CRDM.

Notification name	ISO message	ISO code
BusinessDayInformation	ReturnBusinessDayInformation	ReturnBusinessDayInformation (camt.019 [453])

Table 136 - Receive system notification

11 Dialogues and processes

11.1 Dialogues and processes between CRDM and CRDM Actor

This chapter contains two main subsections describing interactions between a generic CRDM Actor and CRDM for universal use cases. Chapter [A2A Common reference data maintenance and query process](#) [▶ 353] describes the interactions for the maintenance and query of common reference data using the A2A channel. Chapter [DMT file upload](#) [▶ 358] describes the interaction for the configuration of common reference data using the Data Migration Tool.

11.1.1 A2A Common reference data maintenance and query process

This chapter covers the standard situation of a CB or payment bank as CRDM Actor interacting with CRDM through the A2A channel. The two sub- chapters present a standard use case for A2A reference data maintenance and A2A data query respectively.

11.1.1.1 Reference data maintenance process

The CRDM process can be described as a common message flow that applies to every business scenario.

Upon the sending of a request instructed with an input message, a related response message or a technical validation error message is returned.

11.1.1.1.1 Reference data objects

The shared generic message flow is as follows.

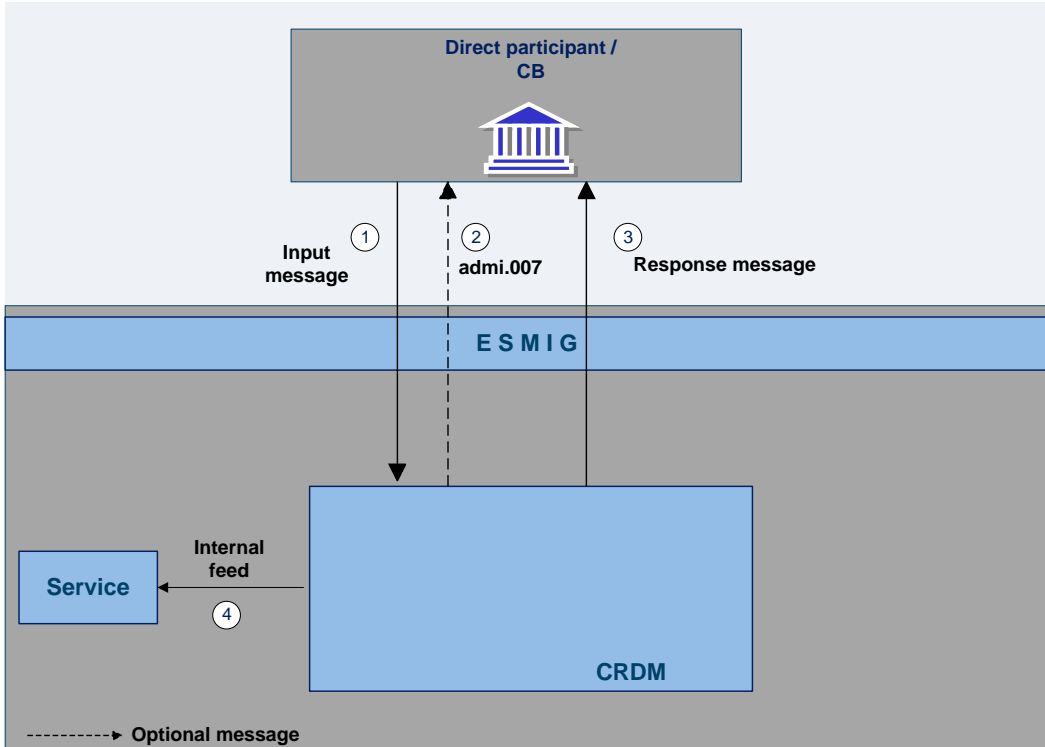


Figure 88 - Common reference data maintenance process

Step	Activity
1	The authorised actor (participant, responsible CB or another actor operating on behalf of the account owner under a contractual agreement) sends the input message to CRDM to create, modify or delete a common reference data entity.
2	In case of rejection upon technical validation, an admi.007, receipt acknowledgement is sent by CRDM to the sender of the originating request.
3	CRDM performs the business validation and sends to the authorised actor a response message to report processing result.
4	CRDM propagates the updated information to the subscribing services for their internal processing.

Table 137 - Common reference data maintenance process

The messages used in the interaction change depending on the business scenario to be covered.

In the following table, for every concerned common reference data entity and related business scenario, the input and response messages are defined.

Business scenario	Input message	Response message	Response message in case of error
Create/Modify standing Order	ModifyStandingOrder (camt.024)	Receipt (camt.025)	Receipt (camt.025)
Delete standing order	DeleteStandingOrder (camt.017)	Receipt (camt.025)	Receipt (camt.025)
Modify Standing Order for Limit	ModifyLimit (camt.011)	Receipt (camt.025)	Receipt (camt.025)
Delete Standing Order for Limit	DeleteReservation (camt.049)	Receipt (camt.025)	Receipt (camt.025)
Modify Standing Order for Reservation	ModifyReservation (camt.048)	Receipt (camt.025)	Receipt (camt.025)
Delete standing order for reservation	DeleteReservation (camt.049)	Receipt (camt.025)	Receipt (camt.025)
Create cash account	AccountOpeningRequest (acmt.007)	AccountRequestAcknowledgement (acmt.010)	AccountRequestRejection (acmt.011)
Delete cash account	AccountClosingRequest (acmt.019)	AccountRequestAcknowledgement (acmt.010)	AccountRequestRejection (acmt.011)
Modify cash account	AccountExcludedMandateMaintenanceRequest (acmt.015)	AccountRequestAcknowledgement (acmt.010)	AccountRequestRejection (acmt.011)
Create party	PartyCreationRequest (reda.014)	PartyStatusAdvice (reda.016)	PartyStatusAdvice (reda.016)
Modify party	PartyModificationRequest (reda.022)	PartyStatusAdvice (reda.016)	PartyStatusAdvice (reda.016)
Delete party	PartyDeletionRequest (reda.031)	PartyStatusAdvice (reda.016)	PartyStatusAdvice (reda.016)

Table 138 - CRDM messages

11.1.1.2 Common reference data query

The common reference data query can be described as a common message flow that applies to every business scenario.

Upon the sending of a query instructed with an input message, a related query response message or a technical validation error message is returned.

11.1.1.2.1 Reference data query message coverage

The shared generic message flow is as follows.

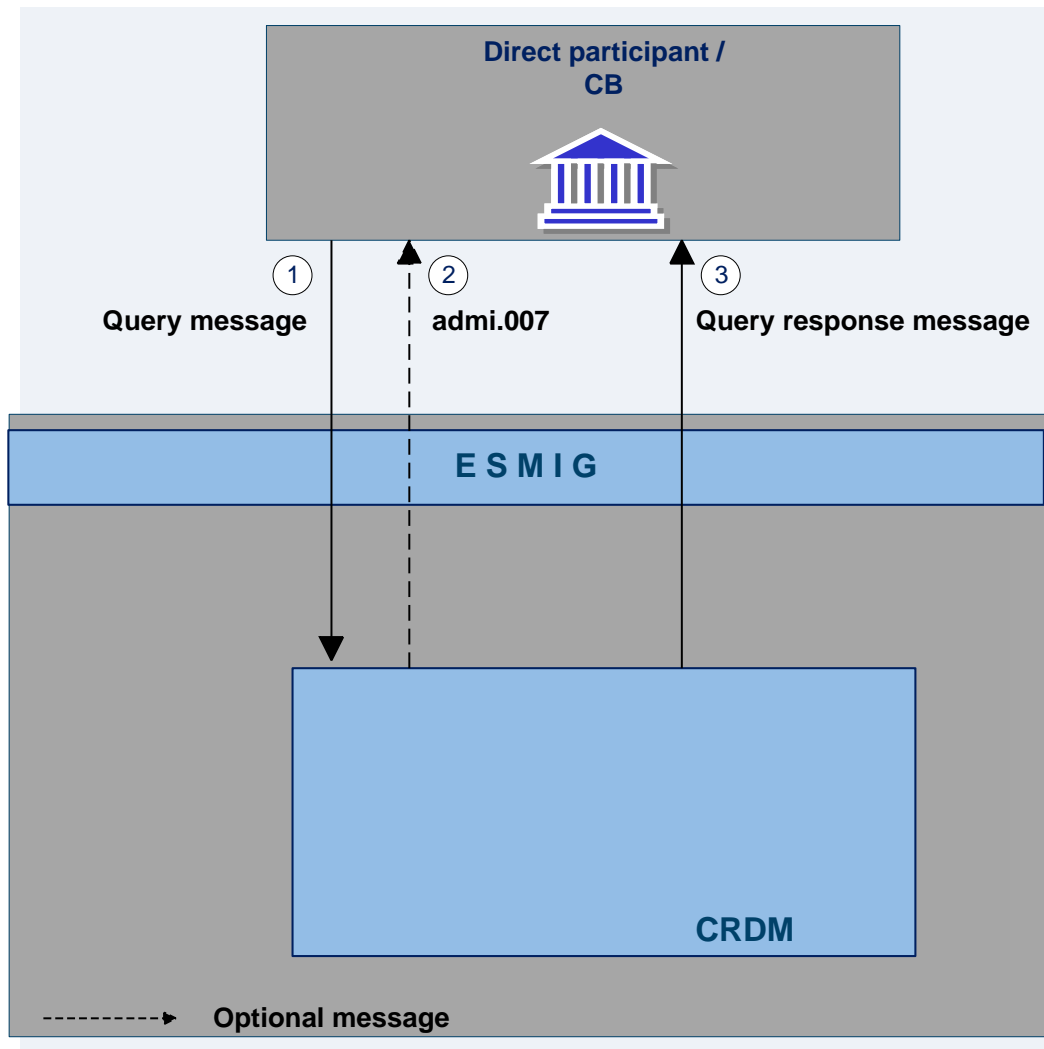


Figure 89 - Common reference data query process

Step	Activity
1	The authorised actor (participant or another actor operating on behalf of the owner under a contractual agreement) sends the query message to CRDM to retrieve a set of common reference data entity.
2	In case of rejection upon technical validation, an admi.007, receipt acknowledgement is sent by CRDM to the sender of the originating query.
3	CRDM performs the business validation and sends to the authorised actor a query response message to report processing result, which consists of the records found or business error found during the validation.

Table 139 - Common reference data query process

The messages used in the interaction change depending on the query to be performed.

In the following table, for every concerned common reference data entity, the query and query response messages are defined.

CRDM entity	Query request messages	Query response message for operational error	Query response message for business data
Standing order	GetStandingOrder (camt.069)	ReturnStandingOrder (camt.070)	ReturnStandingOrder (camt.070)
Account	AccountQueryList (acmt.025)	AccountListReport (acmt.026)	AccountListReport (acmt.026)
Account audit trail	CashAccountAuditTrailQuery(reda.039)	CashAccountAuditTrailReport(reda.040)	CashAccountAuditTrailReport(reda.040)
Party	PartyQuery (reda.015)	PartyReport (reda.017)	
Party audit trail	PartyAuditTrailQuery(reda.042)	PartyAuditTrailReport (reda.043)	PartyAuditTrailReport (reda.043)
Calendar	CalendarQuery(reda.064)	CalendarReport(reda.065)	CalendarReport(reda.065)
Direct debit mandate	DirectDebitMandateQuery(camt.099)	DirectDebitMandateReport(camt.100)	DirectDebitMandateReport(camt.100)

Table 140 - Common reference data query messages

11.1.2 DMT file upload

This use case covers the standard situation of a CB or payment bank as CRDM Actor loading reference data into CRDM. The upload use case is available via U2A through a dedicated section.

The user uploading the file is propagated to the related back-end functions and must have the appropriate access right configuration.

11.1.2.1 Activity diagram

The following diagram details all the processing steps of the DMT file upload use case.

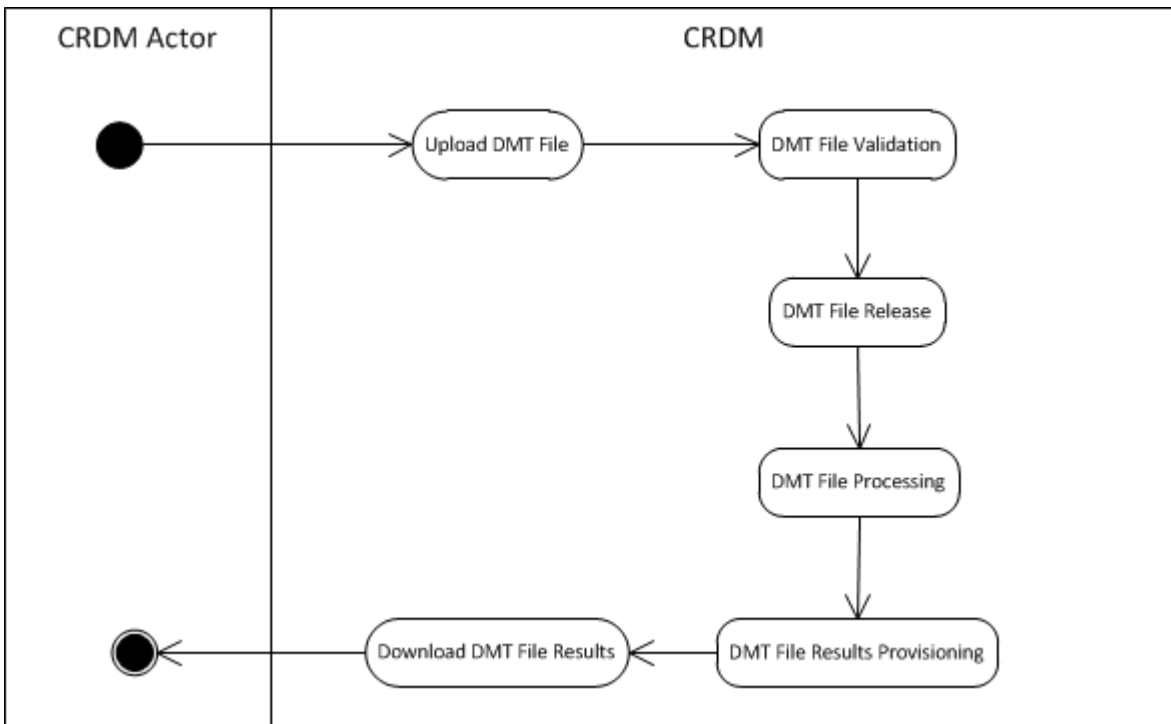


Figure 90 - DMT file upload process

11.1.2.1.1 Upload DMT file

The CRDM Actor uploads the required DMT file containing the reference data to be created in CRDM.

The file can be generated in Excel or Comma Separated Value format and follows the specifications described in Catalogue of messages.

11.1.2.1.2 DMT file validation

CRDM performs a technical validation on the uploaded file to ensure that the technical constraints are respected.

11.1.2.1.3 DMT file release

The operator releases the file for the back end module processing as agreed with the actor.

This step triggers the back end module function required by the file as described in the record type label.

11.1.2.1.4 DMT file processing

The DMT triggers the related back end module function passing information record by record.

Every call to the back end module function generates a result processing.

11.1.2.1.5 DMT file results provisioning

Once all of the records in the uploaded file are sent and processed by the back end module which provides the related result, the DMT file result is consolidated.

For every record, the successful processing or the business errors receives from the back end module is included in the DMT file results.

The file is published for the CRDM Actor to download.

11.1.2.1.6 Download DMT file results

The CRDM Actor downloads the result file reporting the number of migrated records and the detailed list of errors for rejected records.

The following table maps the reference data maintenance operations available in the DMT with the related reference data objects and the file specifications.

Reference data object	Operation	File specifications section
Authorised account user	Create	4.5.3.14
Cash account	Create	4.5.3.12
Certificate DN	Create	4.5.3.10
DN-BIC routing	Create	4.5.3.16
Limit	Create	4.5.3.13
Message subscription rule	Create	4.5.3.8
Message subscription rule set	Create	4.5.3.7
Party	Create	4.5.3.1
Party-service link	Create	4.5.3.15
Privilege	Grant	4.5.3.6
Report configuration	Create	4.5.3.9
Role	Create	4.5.3.4
Role	Grant	4.5.3.5

Reference data object	Operation	File specifications section
Technical address network service link	Create	4.5.3.2
User	Create	4.5.3.3
User certificate DN link	Create	4.5.3.11

Table 141 - DMT files specifications

11.2 Dialogues and processes between ESMIG and participant

Will be completed in v2.0.

11.2.1 Communication processing

11.2.1.1 Introduction

11.2.1.2 Schema validation

11.2.1.3 Technical message validation

11.2.1.4 Inbound and Outbound messages

11.2.1.4.1 Inbound messages

11.2.1.4.2 Outbound Messages

11.2.1.4.3 ReceiptAcknowledgement (admi.007.001.01)

The ReceiptAcknowledgement message is sent by ESMIG to the sender of the message to reject the reception of an A2A-message. Within the ESMIG for TARGET Services this message is generated after an inbound processing rejection, i.e. for missing authentication due to invalid signature.

The table below describes the message elements filled by ESMIG.

The SystemAcknowledgement message is used in this usage to report that ESMIG is not able to process incoming message because of failed authentication of the sending party due to invalid signature.

Specific message requirements

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax35Text	MsgId of the incoming message this ReceiptAcknowledgement is sent for
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status code indicating the error which occurred during the technical validation.
Description Document/RctAck/Rpt/ReqHdlg/Desc	RestrictedFINXMax140Text	Textual description of the technical validation error specified in the status code field.

In the example below a ReceiptAcknowledgement referring to an incoming message with the ID INCOMINGMSG02 with "Invalid Digital Signature" is sent to the corresponding party.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Digital signature check of an incoming message was not successful-->
<!-- Date: 12/06/2012-->
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:DRAFT2admi.007.001.01">
  <RctAck>
    <MsgId>
      <MsgId>NONREF</MsgId>
    </MsgId>
    <Rpt>
      <RltdRef>
        <Ref>INCOMINGMSG02</Ref>
      </RltdRef>
      <ReqHdlg>
        <StsCd>I071</StsCd>
        <Desc>ICSA010-Digital signature is not valid.</Desc>
      </ReqHdlg>
    </Rpt>
  </RctAck>
</Document>
```

11.3 Dialogues and processes with data warehouse

Will be completed in v2.0.

11.4 Dialogues and processes with billing

Will be completed in v2.0.

Part III - Catalogue of messages

12 Messages – introduction

Following on from the formalised illustration of the application processes, the “Part III - Catalogue of messages” section provides a detailed description of the entire set of ISO messages - customised to the specific needs of the RTGS component - available to the actors. The objective is to allow the reader to find the necessary information related to messaging which is needed to establish a functioning system of application-to-application communication.

The List of messages contains all the ISO messages required to support the actors’ business processes. This content is framed by an introductory section “General information”.

The introductory section “General information” provides general information on the concept of messaging or/and information applicable to all messages in RTGS. The appendix contains comprehensive lists of relevant technical details for each message.

The messages described in section “List of Messages” are grouped according to the “business areas” used in ISO 20022 to facilitate orientation for the reader. Each message description consists of three sections.

- I One section to explain the scope of the concerned message and to provide high-level information to the reader about its purpose
- I One section to provide detailed information on the schema file corresponding to the relevant message. Besides providing an overview of the message’s outline, this section contains a link to the online resources where the schema file in xsd- and Excel-format and the respective schema documentation in HTML- and pdf-format and the message examples can be accessed
- I One section to illustrate in detail the different usages or query and instruction types in accordance with the use cases

Overview and scope of the message

This section provides general information about the scope of the message within the context of RTGS. Besides illustrating the purpose of the message within the system, it informs about the sender and receiver of this particular message.

For an inbound message it mentions the possible different instructions or queries for the concerned message (if applicable) and informs the reader about the corresponding response message foreseen. For an outbound message it mentions the possible different usages covered by the message (if applicable).

Schema

This section starts with an outline of the message building blocks applicable to the schema. The reader can find guidance on whether this building block is optional or mandatory and what sort of information it contains.

The section also contains the respective hyperlinks for the online resources related to the message, including the in-depth schema file descriptions. The reader can access the schema file both in XSD and Excel format. These schema files were customised to the needs of the specific utilisation of the messages for the RTGS component and hence contain explanatory annotations and definitions clarifying these possible specificities. Besides the schema file representation, the reader can access documentation available in HTML and PDF providing further explanations on the specific utilisation of the concerned message.

The current messages for the RTGS component are based on ISO 20022 maintenance release 2017/18, whereas CSLD will start with ISO maintenance release 2018/19. The changes resulting from change requests raised for ISO maintenance release 2018/19 will be included at a later stage.

The customised schemas reflect the latest available status of the respective ISO message, i.e. they include all changes occurring during the regular ISO maintenance cycles for these messages. Under certain conditions, the schema documentation anticipates upcoming changes to the ISO messages which are caused by those ISO Change Requests launched specifically to cover RTGS requirements. These changes are not yet incorporated into the schema files as their availability follows the yearly maintenance cycle. Within the schema documentation the reader is nonetheless informed about such changes in advance and can identify future changes to the messages already at this point in time.

The message in business context

This section provides a concrete example on the utilisation of the message in the RTGS context.

For an inbound message with several purposes (instructions or queries) and for an outbound message with several usages, the section provides the specific setup of the message in order to perform the foreseen task.

- I It provides the scope and details of the specific types of instructions/queries or usages, e.g. the query parameters applicable to the specific case.
- I In a sub-section entitled “Specific message requirements”, a message extract is provided in a table format showing the necessary elements of the message to fulfil the purpose described. The extract only depicts the part/s of the message required for the particular necessary configuration for the usage case and may thus deviate from the overall XML structure of the message.
- I A complete message sample in XML format provides the reader with a concrete example on how the message is to be used in a specific business situation which refers to the particular instruction/query or usage. All data used are fictional.

The specific schema is the sole source of information. To avoid doubt, the information contained in the “Specific message content”- tables is not designed to be stand-alone and must be understood only as clarifying the respective specific schema and the related schema documentation.

Within the “Utilisation” column of the tables the reader is familiarised with the relevant content of the concerned message element in the context of the concerned message usage or instruction/query type. This column does not include any sample data but provides generic information applicable to the message element. In cases where codes or values are listed in this column, they should be understood to be the com-

prehensive set of all possible values for the element in the context of the concerned message usage or instruction/query type.

13 Messages - general information

13.1 Message validation

13.1.1 Structure of ISO 20022 messages

Basic information on the XML schema file

XML schema files conform to the compulsory overall structure foreseen for ISO 20022 messages.

Each schema file requires an XML declaration. This declaration provides information on the used XML version and the applicable character set within the message. XML declarations do not have an end tag as they are not part of the XML document itself and hence do not constitute an XML element.

Below the XML declaration, all schema files have a root element. This root element provides the name of the schema file, including information on the variant and the version²⁶ of the schema file. The actual content of the schema file is hence a sub-element of the root element. Similar to all other elements within the schema file, the root element also has an end tag at the end of the schema file.

Example

The below example provides an indication of the overall structure of ISO 20022 messages.

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:DRAFT3admi.007.001.01">
  <RctAck>
    <MsgId>
      <MsgId>NONREF</MsgId>
    </MsgId>
    <Rpt>
      <RltdRef>
        <Ref>LQMGREF1</Ref>
      </RltdRef>
      <ReqHdlg>
        <StsCd>I003</StsCd>
        <Desc>Duplicate checks (on BAH level) NOK</Desc>
      </ReqHdlg>
    </Rpt>
  </RctAck>
</Document>
```

²⁶ A "variant" is a restricted version of a global message which fits the needs of a particular community while remaining in strict compliance with the original ISO 20022 message. For example, optional items can be removed or made mandatory, choices can be removed to keep no or fewer options, internal code lists can be reduced to the subset of codes that is actually used, size of text fields can be reduced, etc. A "version" helps to cater for the evolution of message requirements and for the correction of possible problems and errors of a message. Upon the publication of a new message version a message switches from one way of being used to a new way of being used. Each message (variant) usually has one current version which is the most recent one. The former and the current version coexist for a certain while in order to ease the migration. Example: Within the ReturnAccount message camt.004.001.01 the number 001 reflects the variant of the message in use whereas the number 01 reflects the current version of the message variant in use.

ISO 20022 message

When being sent as an ISO 20022 message, an XML document is referred to as message instance. The underlying schema file “explains” what makes up a valid message (i.e. it contains the necessary rules and definitions). The message instances themselves consist of message components, choice components and message elements.

Message components are items which are used for setting up a message. These message components contain a set of message elements. In ISO 20022 these message components are usually linked to a particular business component. A comprehensive overview of all standardised ISO 20022 message components is available in the Data Dictionary of ISO 20022.

Message elements are the constituents of the message components and are uniquely identified in each component. In ISO 20022 these message elements are usually linked to a particular business element. Filled-in message elements occur as simple and complex data types. All message elements have such a particular type. These data types specify the format of the possible values of a message element.

Example

Simple types serve as a prescription on how to fill the respective message element in the message instance. The simple type shown below prescribes the way in which the currency code must be entered:

```
<xs:simpleType name="ActiveCurrencyCode">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Z]{3,3}" />
  </xs:restriction>
</xs:simpleType>
```

Complex types allow for choice and sequencing options within the message and do not (only) prescribe ways of filling message elements. They hence determine the structure of a message element. The complex type shown below allows for a choice on how to assure party identification in a message. :

```
<xs:complexType name="FinancialInstrumentQuantity15Choice">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Unit" type="RestrictedFINDecimalNumber" />
      <xs:element name="FaceAmt" type="RestrictedFINImpliedCurrencyAndAmount" />
      <xs:element name="AmtsdVal" type="RestrictedFINImpliedCurrencyAndAmount" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

ISO 2022 groups

ISO 20022 groups data types into standardised representation classes. These representation classes provide a set of possible data which can be inserted into the concerned message element.

For example, the message element “Bank Identifier” can be assigned to the representation class “BICIdentifier” or message element “Text” can be assigned to the representation class “Max35Text”.

Choice components allow the user of the message to choose between several possibilities. The message user may only choose one possible option in the instance.

Another term which specifies the partitioning within a message instance is the message item. Such a message item can be either a message building block or a message element. Message items which occur as XML tags within the message instance can appear at any level of nesting in the message.

A message building block is a message item which is specific to the concerned message (i.e. the user cannot find it in the ISO 20022 Data Dictionary). Within the corresponding schema file of the message the building block must be defined as an immediate child of the message. This is not to be confused with reusable groupings of one or more message elements, known as message components (i.e. that the user can find in the ISO 20022 Data Dictionary).

13.1.2 RTGS-specific schema customisation

Based upon the enriched ISO schema files for its messages, once available, (i.e. after the enrichment of newly-developed messages or after the publication of maintained messages in the context of a new standards release) these schema files are customised to adapt them to the specificities applicable in the context of RTGS.

The customisation of the schema files used in RTGS followed a particular approach which combines the needs of the RTGS Actors to have a coherent logic across the messages and the need within RTGS to have a usable and efficient schema definition. RTGS derived this approach from the following customisation principles:

- | customised RTGS schema files are compliant with the initial ISO 20022 schema files;
- | when possible, RTGS customisation drops all the message elements with no direct connection to the user requirements of RTGS;
- | when possible, RTGS customisation restricts element types to the RTGS-specific usage;
- | RTGS customisation defines the necessary content of mandatory fields which cannot be pruned (i.e. “removed”) from the ISO schema files;
- | RTGS customisation restricts the list of possible code values to the sole codes allowed in RTGS;
- | RTGS customisation sets the length of the values to the length applicable in RTGS;
- | RTGS customisation sets the occurrence of message elements to the occurrence applicable in RTGS;
- | RTGS customisation makes optional message elements mandatory if their usage in RTGS is always compulsory;
- | RTGS customisation restricts the allowed characters to those used in RTGS with a pattern;
- | RTGS customisation restricts numeric fields applicable to RTGS (e.g. for amounts).

Based on the chosen approach four scenarios apply to the customisation for RTGS purposes:

1. a (part of a) message only contains elements which are supported by RTGS and there is hence no need for any pruning;

2. RTGS does not need a certain element but it cannot be pruned in the message because of a particular customer need;
3. neither RTGS nor RTGS actors need a certain element and therefore it is pruned;
4. neither RTGS nor its users need a certain element but as mandatory element in the ISO schema file it cannot be pruned and may be filled with a dummy value in RTGS.

For the scenarios 1, 3 and 4, RTGS only allows message elements according to the customised schema file. RTGS rejects any inbound message containing message elements which are not part of the RTGS customised schema file. Message elements under the scope of scenario 4 are not subject to further processing in RTGS. RTGS actors can hence fill these fields either with dummy values or real data (inserting real data does not lead to any processing, either).

For scenario 2 an alternative procedure applies. If message elements are present in the message and in the RTGS customised schema file although the message element is per se dispensable, RTGS nevertheless processes the message. For these message elements only schema validations are applicable. RTGS does not validate these elements against its business rules.

However, for all messages, RTGS prunes elements which are not within the general scope of its functionalities.

RTGS rejects messages during schema validation in cases where actors:

- | Use elements in the message which are not present in the RTGS customised schema file;
- | Use values in allowed elements but do not respect the restrictions of these values foreseen in the RTGS customised schema.

For RTGS outbound messages the logic for filling message elements customised to be optional is derived from the concrete circumstances and purposes of the concerned messages:

- | For query response messages the filled message elements for outbound messages are those necessary to convey the information requested by the corresponding query message;
- | For report messages the same applies, in accordance to the concrete configuration for the subscribed reports;

For any other RTGS outbound message the filling of optional fields also depends on either:

- | The corresponding inbound message with its specific intention,
- | Or the purpose of the RTGS-generated outbound message in case no inbound message precedes.

The sections “The message in business context” may contain message usages and/or message samples in which the content of given fields for a specific purpose or as a reply to a specific inbound message are depicted.

13.1.3 XML character set

UTF-8 is a Unicode character encoding of variable length. It has the capacity to represent every character of the Unicode character set and is backwards compatible to ASCII (in contrast to UTF-16 or UTF-32). In the vast majority of character representations in UTF-8 it only takes one byte to code one character ²⁷.

UTF-8 is part of the ISO 10646 scheme which was published as a first draft in 1990. The idea is to assign a unique code point to every character (i.e. letters, numbers, symbols, ideograms, etc.) covered by this standard. Whereas the standard foresees a maximum amount of 1.1 million of such code points some 100.000 are attributed to abstract characters for the time being. The inclusiveness, however, is steadily augmenting as characters from previously unrepresented writing systems are added.

The ISO website offers a free-of-charge download of the complete definition of the ISO 10646 standard including all the later amendments (e.g. of additional languages).

In principle ISO 20022 caters for UTF8. CLM and RTGS follows the approach of HVPS+ supported character set, limited to basic Latin characters and additional special characters (see table below):

Message elements	Solution												
All Proprietary and/or Text elements, with exception of: Initiating Party, Debtor, Ultimate Debtor, Creditor, Ultimate Creditor, Related Remittance Information and Remittance	Use of FIN X-Character Set: <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre> a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 / - ? : () . , ' + CrLf Space </pre> </div>												
For Initiating Party, Debtor, Ultimate Debtor, Creditor, Ultimate Creditor, Related Remittance Information and Remittance	Use of FIN X-Character set (see above), plus !#\$%&'*+/-=?^_`{ }~"(),:;<>@[]. Note: 5 characters will need to be escaped: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Character</th> <th>XML Escape Sequence</th> </tr> </thead> <tbody> <tr> <td>"</td> <td>&quot;</td> </tr> <tr> <td>'</td> <td>&apos;</td> </tr> <tr> <td><</td> <td>&lt;</td> </tr> <tr> <td>></td> <td>&gt;</td> </tr> <tr> <td>&</td> <td>&amp;</td> </tr> </tbody> </table>	Character	XML Escape Sequence	"	"	'	'	<	<	>	>	&	&
Character	XML Escape Sequence												
"	"												
'	'												
<	<												
>	>												
&	&												

²⁷ UTF-8 uses a single byte to represent 7-bit ASCII characters. Representation of extended characters takes between two and six bytes and hence, between 14 and 42 bits".

13.1.3.1 Schema validation

All ISO 20022 messages which arrive at the RTGS Interface for further processing are subject to validation rules related to the syntax and structure of the message itself. In this context one can distinguish between well-formedness and validity of the message sent to RTGS.

An ISO 20022 message is well-formed if it satisfies the general syntactical rules foreseen for XML documents as outlined in the above chapter. The major aspects to be respected are the following:

- | the message only contains properly encoded Unicode characters;
- | the specific syntax characters (e.g. "<" and "&") are not used in the message except in their function as mark-up delineation;
- | the element-delimiting tags (i.e. start, end and empty-element tags) are correctly nested and paired and none of them is missing or overlapping;
- | the start and end tags match exactly and are case-sensitive;
- | the message has one root element which contains all other elements.

In contrast to other forms of representation the definition of XML documents is rather strict. XML processors cannot produce reasonable results if they encounter even slight violations against the principle of well-formedness. Any violation of this well-formedness automatically entails an interruption of the message processing and an error notification to the sender.

Every well-formed ISO 20022 message arriving at the RTGS interface undergoes a validity check according to the rules contained in the enriched RTGS schema files. These RTGS enriched schemas make the structure of the message visible to the user and provide all necessary explanations on the validations the message undergoes.

The RTGS enriched schema files serve different purposes:

- | they provide a definition of all the elements and attributes in the message;
- | they provide a definition on what elements are child elements and on their specific order and number;
- | they provide a definition of the data types applicable to a specific element or attribute;
- | they provide a definition of the possible values applicable to a specific element or attribute.

RTGS provides the RTGS enriched schema file description in several formats: in xsd., Excel and pdf. This shall allow the user to accommodate himself with the format of his choice while having recourse to computer processable information to the largest extent.

A short extract from an xsd schema file for exemplary purposes (ISO 20022 standard message):

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pacs.009.001.07">
  <FICdtTrf>
    <GrpHdr>
      <MsgId>MSGIDpacs.009</MsgId>
      <CreDtTm>2018-08-24T09:30:44Z</CreDtTm>
      <NbOfTxs>1</NbOfTxs>
      <SttlmInf>
        <SttlmMtd>CLRG</SttlmMtd>
      </SttlmInf>
    </GrpHdr>
    <CdtTrfTxInf>
      <PmtId>
        <InstrId>INSTRIDpacs.009</InstrId>
        <EndToEndId>NOTPROVIDED</EndToEndId>
        <TxId>TXIDpacs.009</TxId>
      </PmtId>
      <IntrBkSttlmAmt Ccy="EUR">1000000</IntrBkSttlmAmt>
      <IntrBkSttlmDt>2018-08-24</IntrBkSttlmDt>
      <SttlmPrty>NORM</SttlmPrty>
      <InstgAgt>
        <FinInstnId>
          <BICFI>BNKBXXYYXXX</BICFI>
        </FinInstnId>
      </InstgAgt>
      <InstdAgt>
        <FinInstnId>
          <BICFI>BNKCXXYYXXX</BICFI>
        </FinInstnId>
      </InstdAgt>
      <Dbtr>
        <FinInstnId>
          <BICFI>BNKBXXYYXXX</BICFI>
        </FinInstnId>
      </Dbtr>
      <Cdtr>
        <FinInstnId>
          <BICFI>BNKCXXYYXXX</BICFI>
        </FinInstnId>
      </Cdtr>
    </CdtTrfTxInf>
  </FICdtTrf>
</Document>
```

A short extract from an xsd. schema file for exemplary purposes (proprietary ISO 20022 based message):

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:swift:xsd:$pain.998.001.01">
  <pain.998.001.01>
    <PrtryDt>
      <Tp>a</Tp>
      <SspPrtryDt>
        <GrpHdr>
          <GrpId>a</GrpId>
          <CreDtTm>2001-12-17T09:30:47Z</CreDtTm>
          <SttlmMdlTp>5000</SttlmMdlTp>
        </GrpHdr>
        <PmtInf>
          <ReqdExctnDt>1957-08-13</ReqdExctnDt>
          <FrstAgt>
            <BIC>AAAAAA20</BIC>
          </FrstAgt>
          <PmtTx>
            <PmtId>
              <InstrId> </InstrId>
              <EndToEndId> </EndToEndId>
            </PmtId>
            <Amt>
              <InstAmt Ccy="AAA">0</InstAmt>
            </Amt>
            <FnlAgt>
              <BIC>AAAAAA20</BIC>
            </FnlAgt>
          </PmtTx>
        </PmtInf>
      </SspPrtryDt>
    </PrtryDt>
  </pain.998.001.01>
</Document>
```

Based on the relevant RTGS enriched schema, the RTGS interface performs the following validations for each incoming message instance:

- | validation of the XML structure (starting from the root element);
- | validation of the element sequencing (i.e. their prescribed order);
- | validation of the correctness of parent-child and sibling relations between the various elements;
- | validation of the cardinality of message elements (e.g. if all mandatory elements are present or if the overall number of occurrences is allowed);
- | validation of the choice options between the message elements;
- | validation of the correctness of the used character set;
- | validation of the correctness of the code list values and their format.

13.1.3.1.1 Business validation

Besides validations which verify the correctness of the ISO 20022 message as XML document itself RTGS also conducts validations which are based on the business context RTGS operates in.

This business validation in RTGS takes place on the basis of a set of pre-defined business rules which are available in the appendix to this document.

On a general level RTGS verifies the validity of the transmitted message content against its static data repository.

In case of violations against existing business rules, RTGS transmits them to the relevant RTGS Actors directly via an outbound message. This message contains all the information the RTGS actor needs to fully understand why e.g. an intended step of processing could not be completed by the system.

This example shows an extract of a camt.025 sent to the case of a business rule violation (RTGS_Receipt_Response to camt.012 Delete Limit RTGS to RTGS Account Holder Rejection):

```
<RctDtls>
  <OrgnlMsgId>
    <MsgId>MSGIDcamt.012</MsgId>
  </OrgnlMsgId>
  <ReqHdlg>
    <StsCd>RREJ</StsCd>
    <Desc>RTGS Rejection</Desc>
  </ReqHdlg>
</RctDtls>
```

13.2 Communication infrastructure

13.2.1 Envelope messages

13.2.1.1 Business Application Header

Regardless of any (ongoing) standardisation discussions at ISO level a business application header (BAH) is defined in general for all messages which are used in RTGS.

The BAH is not applicable when:

- Referring to the acknowledgement of the [ReceiptAcknowledgement \(admi.007\)](#) [391] of a message within RTGS;
- Technical validation errors identified during the "A2A Business File Validation and Splitting process" are answered from RTGS by a [ReceiptAcknowledgement \(admi.007\)](#) [391]

Technically speaking, the BAH is a separate XML document standing apart from the XML documents which represent the message instance itself. ISO structure/BAH structure see below:

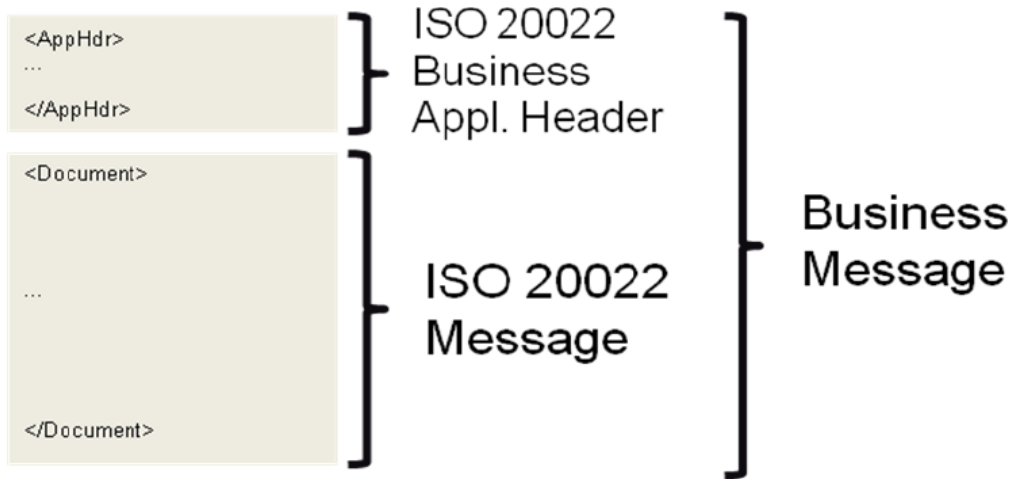


Figure 91 - Business application header

The BAH facilitates the message processing as it stores the information necessary for the processing at one central place. Without BAH this information would be either inside the message instance or in the “RequestHeader” of the ISO 20022 message. A uniform appearance (structure) of relevant information in the BAH improves the routing of the message once it arrives at the addressee’s interface.

The “Request Payload” stands for the whole communication data which is exchanged between and with RTGS.

BAH and business message (XML message instance) are part of this payload. BAH extract see below.

```
<?xml version="1.0" encoding="UTF-8"?>
<AppHdr xmlns="urn:iso:std:iso:20022:tech:xsd:head.001.001.01">
  <Fr>
    <FIId>
      <FinInstnId>
        <BICFI>NCBPARTYBIC</BICFI>
        <Othr>
          <Id>NCBPARNTBIC</Id>
        </Othr>
      </FinInstnId>
    </FIId>
  </Fr>
  <To>
    <FIId>
      <FinInstnId>
        <BICFI>RECEIVERBIC</BICFI>
        <Othr>
          <Id>RCVRPRNTBIC</Id>
          <SchmeNm>
            <Cd>CODE</Cd>
          </SchmeNm>
        </Othr>
      </FinInstnId>
    </FIId>
  </To>
  <BizMsgIdr>SENDERREFERENCE</BizMsgIdr>
  <MsgDefIdr>pacs.002.001.09</MsgDefIdr>
  <CreDt>2018-08-31T09:30:47Z</CreDt>
  <Sgntr>
  </Sgntr>
</AppHdr>
```

Figure 92 - BAH extract

For example, the message element contained in the BAH allows identifying immediately whether a sent message is a copy of a previously sent message.

13.2.1.2 Business File Header

Besides the sending of single messages RTGS supports the exchange of message batches (multi messages). Therefore, it is possible for the T2 actors in RTGS send a file composed of several messages. RTGS uses a business file header to assure the appropriate processing of such message batch. The file structure within is compliant to the requirement of the "Giovannini Protocol: File Transfer Rulebook (May 2007)".

The business file header contains information about the sender, the creation date of the file and the included number of messages. It therefore differs from the BAH which is only used to contain additional information regarding one message (i.e. the following message).

Equivalent to all incoming single messages, A2A files arriving at the RTGS interface entail a receipt confirmation from RTGS. After the successful authentication check RTGS divides the file into single messages. Every message undergoes a separate validation (schema validation). RTGS reports errors on message level either by the corresponding response message or by a status message.

To communicate an RTGS Account Holder can send single messages at a different time or a file containing several messages. Both the message and the file are sent within an envelope which can be compared to a cover page as it contains information about the content. General structure of business file and business message see below:

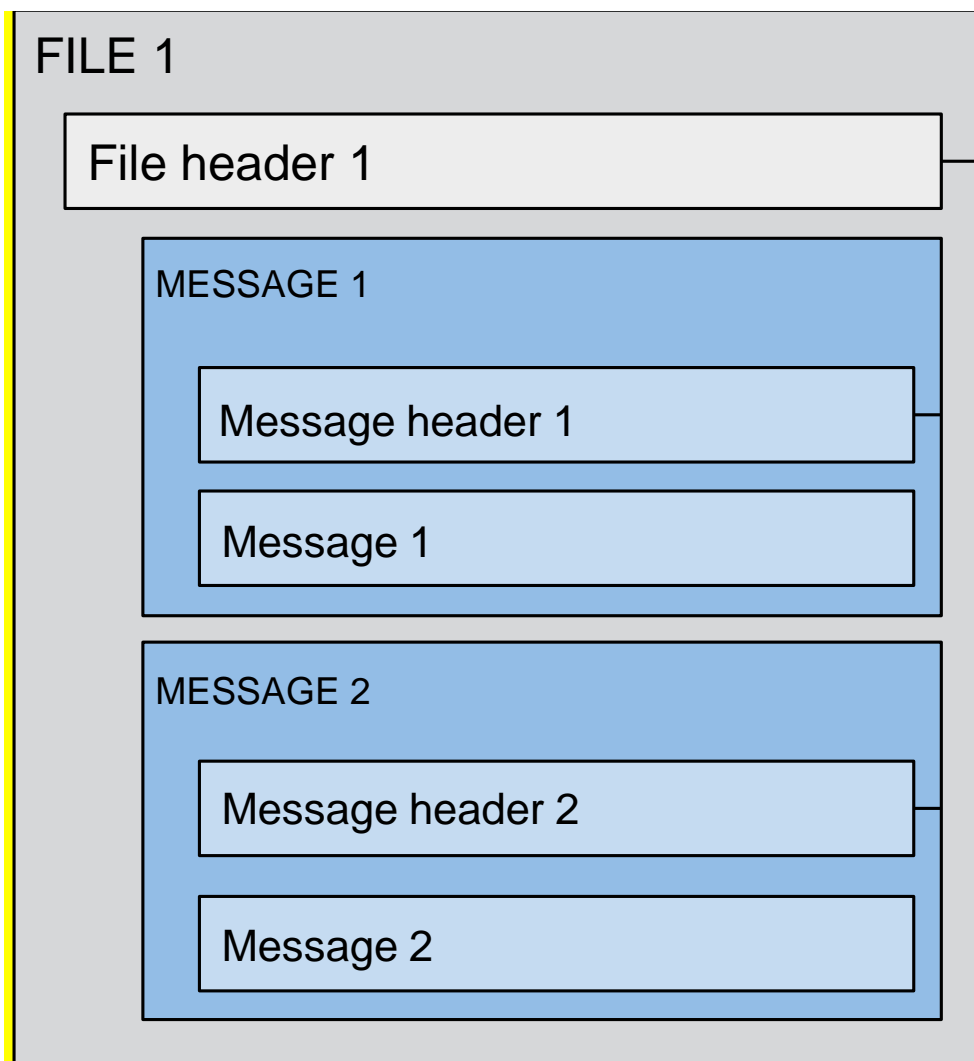


Figure 93 - Business file header

“Message header” in this respect is a synonym for BAH.

13.2.1.3 Digital Signature managed within the business layer

The purpose of this signature is to authenticate the business sender and guarantee the integrity of the business payload. This business signature should be compliant with the W3C XAdES²⁸ standard.

The (NRO)²⁹ signature is stored in the BAH in case of individual messages or in the file header in case of messages grouped into a file. In case messages grouped into a file, the BAH of the included individual messages does not include a signature.

File (meaning multi-message):

The signature is part of the file header. It is over the list of BAH's and ISO 20022 messages and covers the whole <XChg> element of the business file (head.002), except for the signature itself.

Single message:

The signature is over the ISO 20022 message and takes into account the business processing relevant information specified within the BAH (e.g. pair of BICs for definition of the instructing party), except for the signature itself. The digital signature grouped in the BAH itself is not part of this signature calculation.

Further details referring the digital signature construction on business layer can be retrieved from chapter [Digital signature on business layer](#) [> 730] .

13.2.1.4 Time zones

Messages exchanged between RTGS and its users consist of the BAH and the message payload. Both parts of the message contain time indications.

The relevant reference for all inbound and outbound communication in RTGS is Central European Time (CET) or Central European Summer Time (CEST). All indications contained in the payload of RTGS messages (based on given timestamps e.g.) refer to CET/CEST. The attribution of timestamps in the RTGS interface solely occurs on CET/CEST basis. All possible information related to time within the payload of messages sent to RTGS must refer to CET/CEST. The RTGS calendar as the relevant framework for all operational issues of RTGS contains CET/CEST only.

Due to the ISO definition of the application header the time indications within the application header refer to Zulu time. RTGS users must take into account the difference between the two time formats when exchanging messages with RTGS.

28 The XML Advanced Electronic Signatures is a W3C note which extends the [XMLDSIG] specification into the domain of non-repudiation by defining XML formats for advanced electronic signatures that remain valid over long periods and are compliant with the European "Directive 1999/93/EC of the European Parliament.

29 Non-repudiation of origin is intended to protect against the originator's false denial of having sent the message.

Example

A message sent to RTGS on 17 December 2021 at 09:30:47 CET/CEST would need to contain the following field in the BAH ("ZULU time" ³⁰):

```
<CreDt>2021-12-17T09:30:47Z</CreDt>
```

In case the same message contains within the payload an additional reference to the creation date of the message, it would need to contain the following information within the payload ("CET/CEST time"):

```
<CreDt>2021-12-17T09:30:47Z</CreDt>
```

13.2.1.5 Outbound traffic exceeding given size limitations

Traffic sent to or from RTGS is subject to a size limitation deriving from transport layer restrictions. The current message limit is foreseen at a size of 32 KB both for inbound and outbound traffic. In case of messages exceeding the maximum foreseen size technical solutions within RTGS allow for adequate processing of the messages and the contained information. The solution envisaged differs according to RTGS inbound and out-bound traffic.

For RTGS inbound traffic there is no need for the RTGS Actor to send information in one shot by making use of repetitive fields of a single message. Exceeding the maximum size of 32 KB will thus not happen. Instead of conveying the information in one (big) message the RTGS Actor can send two single (small) messages. In contrast to outgoing messages there is no need to see them as "one unit".

For RTGS outbound traffic the size limitation of 32 KB could lead to messages not being transmitted as their content unavoidably exceeds the maximum size. This is particularly the case for query responses and reports where a considerable amount of information referring to the same business case needs to be transported.

When the size of an outbound message exceeds the aforementioned size of 32 KB, RTGS automatically switches from a message-based network service to a file-based network service allowing for a maximum file size transmission of 32 MB. By doing so, splitting of the message into different messages below the 32 KB maximum limit can be avoided.

For query requests received via a message-based network service, the network service has to be switched if the query response exceeds the 32 KB (size restriction for message-based network service). RTGS then sends an error response via the channel in which the request was received and additionally "pushes" the query response details via the default routing for file-based communication.

In case the maximum size of 32 MB is exceeded by a RTGS outbound file, a technical solution is implemented to split this file technically in several parts.

30 Zulu time is the used format for the time indication.

In case the size of a RTGS outbound file is below 32 KB, the message-based network service is used for delivering it to its receiver.

In case a report exceeds the maximum size of 32 MB, the RTGS outbound message may split in several parts. This is the case for: camt.053 (Statement of accounts)

In order to indicate that a report was split, the message elements foreseen to indicate "pagination" is used (<Pgntn> ... </Pgntn>) or for camt.053 <MsgPgntn>...</MsgPgntn>).

For camt.053 a specific procedure for splitting is implemented. In order to avoid message parts exceeding 32 MB, the camt.053 is split at element BkToCstmrStmnt/Stmnt/Ntry.

In case splitting is applied, the following page starts with the same information within the <Stmnt> block as the last entry of the previous page (listing the same Account number and the relating balances) and continues in the <Ntry> block by listing all instructions that do not fit into the previous page.

The application takes care that the fixed elements plus the repetitive elements do not exceed 32 MB. Data compression is not taken into account when deciding on the need to split a message. The uncompressed data volume is the basis for the calculation.

13.2.1.6 Re-sending of messages

In case of need the customers can contact the NSP asking to re-send message/file as foreseen in the relevant NSP documentation.

The participants can also contact the service desk asking the re-send from the central platform. The procedure for engaging the service operators is described in the manual of operational procedures.

14 List of messages

Chapter	Message Code	Message Name
Account Management (acmt)		
AccountQueryList (acmt.025) [384]	acmt.025	AccountQuery
AccountListReport (acmt.026) [386]	acmt.026	AccountReport
Administration (admi)		
ReportQueryRequest (admi.005) [389]	admi.005	ReportQueryRequest
ResendRequest (admi.006) [390]	admi.006	ResendRequest
ReceiptAcknowledgement (admi.007) [391]	admi.007	ReceiptAcknowledgement
Cash Management (camt)		
GetAccount (camt.003) [394]	camt.003	GetAccount
ReturnAccount (camt.004) [397]	camt.004	ReturnAccount
GetTransaction (camt.005) [421]	camt.005	GetTransaction
ReturnTransaction (camt.006) [431]	camt.006	ReturnTransaction
ModifyTransaction (camt.007) [437]	camt.007	ModifyTransaction
GetLimit (camt.009) [440]	camt.009	GetLimit
ReturnLimit (camt.010) [442]	camt.010	ReturnLimit
ModifyLimit (camt.011) [445]	camt.011	ModifyLimit
DeleteLimit (camt.012) [448]	camt.012	DeleteLimit
GetBusinessDayInformation (camt.018) [451]	camt.018	GetBusinessDayInformation
ReturnBusinessDayInformation (camt.019) [453]	camt.019	ReturnBusinessDayInformation
ReturnGeneralBusinessInformation (camt.021) [458]	camt.021	ReturnGeneralBusinessInformation
ModifyStandingOrder (camt.024) [469]	camt.024	ModifyStandingOrder
Receipt (camt.025) [474]	camt.025	Receipt

Chapter	Message Code	Message Name
ResolutionOfInvestigation (camt.029) [▶ 484]	camt.029	ResolutionOfInvestigation
GetReservation (camt.046) [▶ 487]	camt.046	GetReservation
ReturnReservation (camt.047) [▶ 489]	camt.047	ReturnReservation
ModifyReservation (camt.048) [▶ 492]	camt.048	ModifyReservation
DeleteReservation (camt.049) [▶ 494]	camt.049	DeleteReservation
LiquidityCreditTransfer (camt.050) [▶ 497]	camt.050	LiquidityCreditTransfer
BankToCustomerStatement (camt.053) [▶ 505]	camt.053	BankToCustomerStatement
BankToCustomerDebitCreditNotifica- tion (camt.054) [▶ 522]	camt.054	BankToCustomerDebitCreditNotifica- tion
FIToFIPaymentCancellationRequest (camt.056) [▶ 536]	camt.056	FIToFIPaymentCancellationRequest
GetStandingOrder (camt.069) [▶ 542]	camt.069	GetStandingOrder
ReturnStandingOrder (camt.070) [▶ 543]	camt.070	ReturnStandingOrder
DeleteStandingOrder (camt.071) [▶ 550]	camt.071	DeleteStandingOrder
BillingReportRequest (camt.076) [▶ 552]	camt.076	BillingReportRequest
BillingReport (camt.077) [▶ 552]	camt.077	BillingReport
AuditTrailQuery (camt.097) [▶ 552]	camt.097	AuditTrailQuery
AuditTrailReport (camt.098) [▶ 554]	camt.098	AuditTrailReport
DirectDebitMandateQuery (camt.099) [▶ 558]	camt.099	DirectDebitMandateQuery
DirectDebitMandateReport(camt.100) [▶ 560]	camt.100	DirectDebitMandateReport
Headers (head)		
BusinessApplicationHeader (head.001) [▶ 564]	head.001	BusinessApplicationHeader

Chapter	Message Code	Message Name
BusinessFileHeader (head.002) [▶ 566]	head.002	BusinessFileHeader
Payments Clearing and Settlement (pacs)		
PaymentStatusReport (pacs.002) [▶ 568]	pacs.002	PaymentStatusReport
PaymentReturn (pacs.004) [▶ 571]	pacs.004	PaymentReturn
CustomerCreditTransfer (pacs.008) [▶ 577]	pacs.008	CustomerCreditTransfer
FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▶ 585]	pacs.009	FinancialInstitutionCreditTransfer
FinancialInstitutionDirectDebit (pacs.010) [▶ 603]	pacs.010	FinancialInstitutionDirectDebit
Payments Initiation (pain)		
ASInitiationStatus (pain.998) [▶ 620]	pain.998	ASInitiationStatus
ASTransferNotice (pain.998) [▶ 610]	pain.998	ASTransferNotice
ASTransferInitiation (pain.998) [▶ 633]	pain.998	ASTransferInitiation
Reference Data (reda)		
PartyQuery (reda.015) [▶ 651]	reda.015	PartyQuery
PartyReport (reda.017) [▶ 652]	reda.017	PartyReport
CashAccountAuditTrailQuery (reda.039) [▶ 655]	reda.039	CashAccountAuditTrailQuery
CashAccountAuditTrailReport (reda.040) [▶ 657]	reda.040	CashAccountAuditTrailReport
PartyAuditTrailQuery (reda.042) [▶ 661]	reda.042	PartyAuditTrailQuery
PartyAuditTrailReport (reda.043) [▶ 662]	reda.043	PartyAuditTrailReport
CalendarQuery (reda.064) [▶ 666]	reda.064	CalendarQuery
CalendarReport (reda.065) [▶ 667]	reda.065	CalendarReport

Table 142 - List of messages

14.1 Account management (acmt)

14.1.1 AccountQueryList (acmt.025)

14.1.1.1 Overview and scope of the message

This chapter illustrates the *AccountQueryList* message.

The *AccountQueryList* is sent by an actor authorised to query cash account reference data.

In response to the *AccountQueryList*, an [AccountListReport \(acmt.026\)](#) [▶ 386] containing the requested information is returned.

14.1.1.2 Schema

Outline of the schema

The *AccountQueryList* message is composed of the following message building blocks:

References

This block is mandatory and contains an identification used to uniquely and unambiguously identify the message.

AccountServicerIdentification

This block is mandatory. It contains the identification of the party receiving the request.

Organisation

This block is mandatory. It contains the identification of the party sending the request.

Account Search Criteria

This block is mandatory and it contains detailed information related to the business account query message. It includes the following elements:

- | identification
- | account type
- | currency

- | closing and opening date
- | account owner

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/acmt.025.001.002>

14.1.1.3 The message in business context

Usage case: Cash account reference data query

In this usage case reference data about a cash account are requested.

Specific message requirements

At least one of the search criteria must be provided.

Message item	Data type/code	Utilisation
Identification Document/AcctQryList/AcctSchCrit/Id	AccountIdentification4Choice	Account identifier
Type Document/AcctQryList/AcctSchCrit/Tp	CashAccountType2Choice	Account type
Currency Document/AcctQryList/AcctSchCrit/Ccy	ActiveCurrencyCode	Currency code
ClosingDate Docu- ment/AcctQryList/AcctSchCrit/ClsgDt	DateSearchChoice	Closing date
OpeningDate Docu- ment/AcctQryList/AcctSchCrit/OpngDt	DateSearchChoice	Opening date
BIC Docu- ment/AcctQryList/AcctSchCrit/AcctOwn r/BIC	AnyBICIdentifier	Account owner

Table 143 - AccountQueryList (acmt.025) – usage case Cash account reference data query**Usage case example: CashAccountReferenceDataQuery_example.xml**

In this example, a CB participating in T2S with BIC “NCBAXXYAAA” queries reference data for cash account “ACC0001” under its responsibility.

14.1.2 AccountListReport (acmt.026)

14.1.2.1 Overview and scope of the message

This chapter illustrates the *AccountListReport* message.

The *AccountListReport* is sent by CRDM to an authorised actor to provide with requested cash account information.

The *AccountListReport* is sent in response to the [AccountQueryList \(acmt.025\)](#) [▶ 384] message.

14.1.2.2 Schema

Outline of the schema

The *AccountListReport* message is composed of the following message building blocks:

References

This block is mandatory and contains the identification assigned by the sending party to uniquely and unambiguously identify the message and the identification of the original message.

AccountServicerIdentification

This building block is mandatory. It contains the identification of the CB responsible for the receiving party.

Organisation

This building block is mandatory. It contains the identification of the receiving party.

ReportOrError

This building block is mandatory. It provides either the information matching the search criteria or an error indication.

It includes the following elements identification, currency, opening and closing dates, restriction information, floor and ceiling notification amounts, external RTGS account reference, account owner.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/acmt.026.001.002>

14.1.2.3 The message in business context

Usage case: Cash account reference data query response

This message usage provides the sender with requested information about cash account reference data.

Specific message content

A cash account reference data query response contains the following set of information on queried cash account.

Message item	Data type/code	Utilisation
Identification Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/Id	AccountIdentification4Choice	Account identifier
Type Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/Tp	CashAccountType2Choice	Account type
Currency Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/Ccy	ActiveCurrencyCode	Currency code
FloorNotificationAmount Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/FlrNtfctnAmt	ImpliedCurrencyAndAmount	Floor notification amount
CeilingNotificationAmount Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/ClngNtfctnAmt	ImpliedCurrencyAndAmount	Ceiling notification amount

Message item	Data type/code	Utilisation
ClosingDate Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/ClsgDt	ISODate	Closing Date
Restriction Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/Rstrctn	Restriction1	Account restriction
OpeningDate Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/OpngDt	ISODate	Opening date
ReferenceAccount Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ref Acct	CashAccount24	External RTGS account reference
AccountOwner Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ctOwnr	AnyBICIdentifier	Account owner

Table 144 - AccountListReport (acmt.026) – usage case Cash account reference data query response

The returned data in case of an error response is listed below:

Message item	Data type/code	Utilisation
Proprietary Docu- ment/AcctListRpt/RptOrErr/Err/Err/Prtry	Max4AlphaNumericText	Specific error code
Description Docu- ment/AcctListRpt/RptOrErr/Err/Desc	Max140Text	Textual description in addition to the reported error code

Table 145 - AccountListReport (acmt.026) – usage case Error

Usage case example: CashAccountReferenceDataQueryResponse_example.xml

In this example, a CB with BIC “NCBAXXYAAA” queried Cash Account with Id “ACC0001”.

Reference data of the queried account is returned in the response.

14.2 Administration (admi)

14.2.1 ReportQueryRequest (admi.005)

14.2.1.1 Overview and scope of the message

This chapter illustrates the *ReportQueryRequest* message.

The *ReportQueryRequest* message is sent by a RTGS Account Holder (or a party authorised by them) to the RTGS component. It is used to query the latest available report data of a specific report type.

Within RTGS, the *ReportQueryRequest* message has the following usages:

- | Account Statement Query

In response to the *ReportQueryRequest* message, the requested report message is returned. In the case of an error resulting from the processing of the *ReportQueryRequest*, an error information is returned using a [ReceiptAcknowledgement \(admi.007\)](#) [391] message.

14.2.1.2 Schema

Outline of the schema

The *ReportQueryRequest* message is composed of the following building blocks.

MessageIdentification

This building block is mandatory and provides a set of elements to identify the report query request message.

ReportQueryCriteria

This building block is mandatory and repetitive. It defines the report query criteria. It contains the elements:

- | report name
- | report owing party BIC and name

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/admi.005.001.01_RTGS

Business rules applicable to the schema

For business rules applicable to *ReportQueryRequest* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.2.1.3 The message in business context

Usage case: Account Statement Query

In this usage case, the sender is requesting that the most recent [BankToCustomerStatement \(camt.053\)](#) [▶ 505] report fulfilling the given criteria, is sent back to them.

Specific message requirements

Message item	Data type/code	Utilisation
AccountIdentification RptQryCrit/SchCrit/AcctId/EQ/Othr/Id	Max34Text	DCA account number will be provided here.
Report Name RptQryCrit/SchCrit/RptNm/	ReportCode_CS LD	SACC code for statement of accounts report type is allowed.
PartyIdentification RptQryCrit/SchCrit/PtyId/AnyBIC	RTGS_BIC11Text	Either party BIC
Name of the Party RptQryCrit/SchCrit/PtyId/NmAndAdr/Nm	Max350Text	Or party name can be used

Table 146 - ReportQueryRequest (admi.005) – usage case Account Statement Query

Usage case example: admi.005_RTGS_ReportQueryRequest_AccountStatementQuery_Example.xml

Within message sample this party BIC “AAAAAA20000” is used as the only selection parameter to retrieve the latest available account statement report from last EoD generation.

14.2.2 ResendRequest (admi.006)

14.2.2.1 Overview and scope of the message

This chapter illustrates the *ResendRequest* message.

The *ResendRequest* message is sent by a RTGS Account Holder to ESMIG. It is used to request the resending of a message or a file (a duplicate of the original message/file) supported by the RTGS component.

The *ResendRequest* message supports resend requests for messages from the RTGS and other components. The resend process is under the control of ESMIG.

Please refer to the ESMIG component documentation for further information.

14.2.2.2 Schema

Outline of the schema

The *ResendRequest* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the request message.

ResendSearchCriteria

Defines the criteria required to unambiguously identify the information to be resent.

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/admi.006.001.01_RTGS

14.2.2.3 The message in business context

This section is intentionally left blank.

14.2.3 ReceiptAcknowledgement (admi.007)

14.2.3.1 Overview and scope of the message

This chapter illustrates the *ReceiptAcknowledgement* message.

The *ReceiptAcknowledgement* message is sent by the RTGS component to a RTGS Account Holder. It is used to reject the reception of a previously sent message.

The RTGS component generates this message after a negative authentication process. It can be also sent as an error reporting response to a report query. Within RTGS, the *ReceiptAcknowledgement* message has the following usages:

- | Negative Receipt Acknowledgement (e.g. Schema Validation Rejection, technical validation)

In general, the *ReceiptAcknowledgement* message is sent without a BAH.

14.2.3.2 Schema

Outline of the schema

The *ReceiptAcknowledgement* message is composed of the following message building blocks:

MessageIdentification

This building block is mandatory and provides a set of elements to uniquely identify the *ReceiptAcknowledgement* message.

RelatedReference

This building block is mandatory and non-repetitive. It provides a reference of the request message to which this *ReceiptAcknowledgement* message is responding.

RequestHandling

This building block is mandatory and non-repetitive. It gives the status of the request. It may contain:

- | status code
- | description

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/admi.007.001.01_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReceiptAcknowledgement* message.

14.2.3.3 The message in business context

Negative *Receipt Acknowledgement* is sent for the following functions:

- | NegativeReceiptAcknowledgement_SchemaValidation for all message functions
- | Reject RTGS Query Message

Usage case: Negative Receipt Acknowledgement/Reject Query Message

In the above mentioned usage case, the recipient is being informed that a message previously received from RTGS does not comply with RTGS technical rules and is not processable for RTGS.

Specific message requirements

Message item	Data type/code	Utilisation
Message Identification RctAck/Msgld	RTGS_RestrictedFINXMax35Text	Value "NONREF" as the message Id is already part of the BAH
Related Reference RltdRef/Ref	RTGS_RestrictedFINXMax35Text	Reference given by the original message: Msgld of the incoming message this receipt acknowledgement is sent for. In case the Msgld of the incoming message can't be identified: NONREF.
Status Code ReqHdlg/StsCd	Max4AlphaNumericText	Specifies the status of the request, based on the validation rule which occurred
Description ReqHdlg/StsCd	RestrictedFINXMax140Text	Description of the status and error defined (belonging to the validation rule)

Table 147 - ReceiptAcknowledgement (admi.007) – usage case Negative Receipt Acknowledgement

Usage case example 1: admi.007_RTGS_NegativeReceiptAcknowledgement_SchemaValidation_Example_1.xml

In this sample a [ModifyReservation \(camt.048\)](#) [▶ 492] is sent to RTGS for further processing, but rejected as the format of one field does not comply with the XSD scheme.

Usage case example 2: admi.007_RTGS_NegativeReceiptAcknowledgement_SchemaValidation_Example_2.xml

In this sample a [ModifyLimit \(camt.011\)](#) [▶ 445] is sent by a RTGS Account Holder (or on their behalf by an authorised party) to RTGS, but rejected as the format of one field does not comply with the XSD scheme.

Usage case example 3: admi.007_RejectRTGS Query Message _Example.xml

In this sample a [GetAccount \(camt.003\)](#) [▶ 394] is sent by a RTGS Account Holder to RTGS, but rejected as the digital signature is not valid.

14.3 Cash management (camt)

14.3.1 GetAccount (camt.003)

14.3.1.1 Overview and scope of the message

This chapter illustrates the *GetAccount* message.

The *GetAccount* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to request balances, of one RTGS DCA held at the RTGS component.

The *GetAccount* message contains the criteria which are used to select the response information.

Within RTGS, the *GetAccount* message has the following usages:

- Account Balance Query

In response to the *GetAccount* message, a [ReturnAccount \(camt.004\)](#) [▶ 397] message containing the requested information is returned.

14.3.1.2 Schema

Outline of the schema

The *GetAccount* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

AccountQueryDefinition

This building block is mandatory. It contains detailed information related to the business query criteria about the account.

SearchCriteria

This block is mandatory and non-repetitive. It defines the criteria to be used to extract the account information. It includes the following elements:

- | account identification
- | account owner

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.003.001.06_RTGS

Business rules applicable to the schema

For business rules applicable to *GetAccount* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.1.3 The message in business context

Usage case: Account Balance Query

In this usage case, the sender requests information regarding all balance information available on RTGS DCAs within their query criteria.

Specific message requirements

All content must comply with the business rules for the message.

If sending party and account owner are identical then the *GetAccount* message will only include message identification.

Message item	Data type/code	Utilisation
Message Identification	RestrictedFINXMax35Text	Identification of the message
Document/GetAcct/MsgHdr/MsgId		Copied from the BAH but not validated as a unique message ID is provided within the BAH

Table 148 - GetAccount (camt.003) – usage case AccountBalanceQuery

Request for specific sub-account, the *GetAccount* message will only include message identification and account identification.

Message item	Data type/code	Utilisation
Message Identification Document/GetAcct/MsgHdr/MsgId	RestrictedFINXMax35Text	Identification of the message Copied from the BAH but not validated as a unique message ID is provided within the BAH
Account Identification Docu- ment/GetAcct/AcctQryDef/AcctCrit/New Crit/SchCrit/AcctId/EQ/Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer Account ID or account owner must be present, but not both

Request on behalf of third party (account owner) e.g. BIC of CB or group of account manager is the sender of the *GetAccount* message. Sending party and account owner are different then, in the *GetAccount* message identification and account owner will be included.

Message item	Data type/code	Utilisation
Message Identification Document/GetAcct/MsgHdr/MsgId	RestrictedFINXMax35Text	Identification of the message Copied from the BAH but not validated as a unique message ID is provided within the BAH
Organisation Identification Docu- ment/GetAcct/AcctQryDef/AcctCrit/New Crit/SchCrit/AcctOwnr/Id/OrgId/AnyBIC	AnyBICIdentifier	Unique and unambiguous way to identify an organisation CLM-Use: <ul style="list-style-type: none"> BIC of the CLM MCA. In case the requestor is a normal CI, his DN must match to the stated BIC. A CB may only select accounts it is responsible for. AnyBICIdentifier/BIC used on behalf of third parties Account ID or account owner must be present, but not both

Usage case example 1: camt.003_RTGS_GetAccount_AccountBalanceQuery_MsgId_Example.xml

In this example, a *GetAccount* is instructed by the account owner. It illustrates the mandatory elements in the message.

Usage case example 2: camt.003_RTGS_GetAccount_AccountBalanceQuery_AcctId_Example.xml

In this example, a *GetAccount* is instructed by the account owner for a specific sub-account. It illustrates the mandatory elements in the message.

Usage case example 3: camt.003_RTGS_GetAccount_AccountBalanceQuery_AcctOwnr_Example.xml

In this example, a *GetAccount* is instructed by a third party. It illustrates the mandatory elements in the message.

14.3.2 ReturnAccount (camt.004)

14.3.2.1 Overview and scope of the message

This chapter illustrates the *ReturnAccount* (camt.004) message.

The *ReturnAccount* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to provide information on the balances of one RTGS DCA held at the RTGS component.

Within RTGS, the *ReturnAccount* message has the following usages:

- | Account Balance Query (Data or Error Response)
- | Floor Notification
- | Ceiling Notification
- | AS Procedure C (start)
 - Notification of Credit to all Sub-Accounts
 - Notification of Liquidity Blocked on all Sub-Accounts
- | AS Procedure C (end)
 - Notification of Execution of Stored Immediate Liquidity Transfers
 - Notification of Liquidity Re-transfer
- | AS Procedure D (start)
 - Notification of Global Amount on Liquidity Account

The *ReturnAccount* query account balance response message is sent in response to a [GetAccount \(camt.003\)](#) [▶ 394] message, which requested the information. The floor and ceiling notifications are sent based upon activity within the RTGS component.

14.3.2.2 Schema

Outline of the schema.

The *ReturnAccount* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about account, or an error indication.

AccountReport

This building block reports either on the account information or on a business error. When it reports the account information, it may contain:

- | account identification
- | account type
- | currency
- | account owner
- | multilateral balances (multiple)

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.004.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnAccount* message.

14.3.2.3 The message in business context

Usage case: Account Balance Query (Data Response)

In this usage case, the recipient of the message is being informed regarding all balance information available on RTGS DCAs within their query criteria.

Specific message requirements

Message item	Data type/code	Utilisation
Message Identification Document/RtrAcct/MsgHdr/MsgId	RestrictedFINXMax35Text	Identification of the message Value "NONREF" as the Message ID is already part of the BAH
Original Business Query Docu- ment/RtrAcct/MsgHdr/OrgnlBizQry/Msg Id	RestrictedFINXMax35Text	Point to point reference, as assigned by the original initiating party, to unambiguously identify the original query message. Message ID of the GetAccount (camt.003) [394] copied from the BAH.
Account Identification Docu- ment/RtrAcct/RptOrErr/AcctRpt/AcctId/ Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
Account Owner Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ownr/Id/OrgId/AnyBIC	AnyBIC	BIC of the RTGS Account Holder owning the account/sub-account which is being queried.
Multilateral Balance Amount Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Amt	ImpliedCurrencyAndAmount fractionDigits: 5 totalDigits: 18 minInclusive: 0	Amount of money of the cash balance.

Message item	Data type/code	Utilisation
Multilateral Balance Credit Debit Indicator Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/CdtDbtInd	CRDT DBIT	Indicates whether the balance is a credit or a debit balance. A zero balance is considered to be a credit balance. RTGS-Use: In the case of BalanceTypeCode NOTE it defines whether it is an earmarked payment for debits or credits.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp	SystemBalanceType1Code__1: OPNG CRRT AVLB LTSF CRDT BLCK XPCD DLOD ADJT PRAV DBIT NOTE Proprietary Code: FLOR CEIL	Specifies the nature of a balance which is being reported. Rule "RTGS_X123456_ Balance Types And Status _STLD": If balance type is AVLB (available liquidity), OPNG (start balance) or CRRT (current balance), then status must be STLD. Rule "RTGS_X123456_ Balance Types And Status _PDNG": If the balance type is NOTE (timed payments) or XPCD (projected liquidity), then status must be PDNG.
Multilateral Balance Status Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Sts	PDNG STLD	Current status of a cash balance which is being reported. RTGS-Use: Mandatory for: AVLB = available liquidity (only STLD is possible) OPNG = start balance (only STLD is possible) NOTE = timed payments (only PDNG is possible)

Message item	Data type/code	Utilisation
		CRDT = credits CRRT = current balance (only STLD is possible) DBIT = debits LTSF = liquidity transfer XPCD = projected liquidity (only PDNG is possible)

Table 149 - ReturnAccount camt.004 – usage case Account Balance Query (Data Response)

Usage case example: camt.004_RTGS_ReturnAccount_AccountBalanceQueryData_Example.xml

In this example a *ReturnAccount* containing a reference to an incoming message with the ID “MSGIDcamt.003” and the available balance information on the RTGS DCA is sent to the requesting party.

Usage case: Account Balance Query (Error Response)

In this usage case, the recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent [GetAccount \(camt.003\)](#) [▶ 394].

The identification of the previously sent query message is included in this error response for reconciliation purposes.

Specific message requirements

Message item	Data type/code	Utilisation
Message Identification Document/RtrAcct/MsgHdr/MsgId	RestrictedFINXMax35Text	Identification of the message Value "NONREF" as the Message ID is already part of the BAH
Original Business Query Docu- ment/RtrAcct/MsgHdr/OrgnlBizQry/Msg Id	RestrictedFINXMax35Text	Point to point reference, as assigned by the original initiating party, to unambiguously identify the original query message. Message ID of the GetAccount (camt.003) [394] copied from the BAH.
Error Code Docu- ment/RtrAcct/RptOrErr/OpriErr/Err/Prtr y	RestrictedFINXMax4Text	Specification of the error, in proprietary code.
Error Description Docu- ment/RtrAcct/RptOrErr/OpriErr/Desc	Max140Text	Specification of the error, in free format.

Table 150 - ReturnAccount (camt.004) – usage case Account Balance Query (Error Response)

Usage case example: camt.004_ RTGS_ReturnAccount_AccountBalanceQueryError_Example.xml

In this example a *ReturnAccount* containing a reference to an incoming message with the ID "MSGIDcamt.003", the error code "P055" and the error description "Account does not exist or is invalid" is sent to the requesting party.

Usage case: Floor Notification

In this usage case, the recipient of the message is being informed that the balance on one of their RTGS DCAs has fallen below the pre-defined floor threshold for the account.

Specific message requirements

Message item	Data type/code	Utilisation
Message Identification Document/RtrAcct/MsgHdr/MsgId	RestrictedFINXMax35Text	Identification of the message Value "NONREF" as the message ID is already part of the BAH
Original Business Query	RestrictedFINXMax35Text	RTGS-Use:

Message item	Data type/code	Utilisation
Docu- ment/RtrAcct/MsgHdr/OrgnIBizQry/Msg Id		Value "NONREF" for floor and ceiling notifications as <i>ReturnAccount</i> is sent in push mode to notify the recipient of the message that one of its RTGS DCAs has fallen below or risen above a pre-defined balance amount.
Account Identification Docu- ment/RtrAcct/RptOrErr/AcctRpt/AcctId/ Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	SACC	RTGS-Use: Only settlement account is used.
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
Account Owner Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ownr/Id/OrgId/AnyBIC	AnyBIC	BIC of the RTGS Account Holder owning the account/sub-account which is being queried.
Multilateral Balance Amount Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Amt	ImpliedCurrencyAndAmount fractionDigits: 5 totalDigits: 18 minInclusive: 0	Amount of money of the cash balance.

Message item	Data type/code	Utilisation
Multilateral Balance Credit Debit Indicator Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/CdtDbtInd	CRDT DBIT	Indicates whether the balance is a credit or a debit balance. A zero balance is considered to be a credit balance. RTGS-Use: In the case of BalanceTypeCode NOTE it defines whether it is an earmarked payment for debits or credits.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp	SystemBalanceType1Code__1: AVLB Proprietary Code: FLOR	Specifies the nature of a balance which is being reported. Rule "RTGS_X123456_ Balance Types And Status _STLD": If balance type is AVLB (available liquidity), OPNG (start balance) or CRRT (current balance), then status must be STLD.
Multilateral Balance Status Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Sts	PDNG STLD	Current status of a cash balance which is being reported. RTGS-Use: Mandatory for: AVLB = available liquidity (only STLD is possible)

Table 151 - ReturnAccount (camt.004) – usage case Floor Notification

Usage case example: camt.004_RTGS_ReturnAccount_FloorNotification_Example.xml

In this example a *ReturnAccount* is sent as a push notification to the account owner with information about the current balance and floor threshold of the account.

Usage case: Ceiling Notification

In this usage case, the recipient of the message is being informed that the balance on one of their RTGS DCAs has risen above the pre-defined ceiling threshold for the account.

Specific message requirements

Message item	Data type/code	Utilisation
Message Identification Document/RtrAcct/MsgHdr/MsgId	RestrictedFINXMax35Text	Identification of the message Value "NONREF" as the message ID is already part of the BAH
Original Business Query Docu- ment/RtrAcct/MsgHdr/OrgnlBizQry/Msg Id	RestrictedFINXMax35Text	RTGS-Use: Value "NONREF" for floor and ceiling notifications as <i>ReturnAccount</i> is sent in push mode to notify the recipient of the message that one of its RTGS DCAs has fallen below or risen above a pre-defined balance amount.
Account Identification Docu- ment/RtrAcct/RptOrErr/AcctRpt/AcctId/ Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	SACC	RTGS-Use: Only settlement account is used.
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
Account Owner Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ownr/Id/OrgId/AnyBIC	AnyBIC	BIC of the RTGS Account Holder owning the account/sub-account which is being queried.
Multilateral Balance Amount Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Amt	ImpliedCurrencyAndAmount fractionDigits: 5 totalDigits: 18 minInclusive: 0	Amount of money of the cash balance.

Message item	Data type/code	Utilisation
Multilateral Balance Credit Debit Indicator Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/CdtDbtInd	CRDT DBIT	Indicates whether the balance is a credit or a debit balance. A zero balance is considered to be a credit balance. RTGS-Use: In the case of BalanceTypeCode NOTE it defines whether it is an earmarked payment for debits or credits.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp	SystemBalanceType1Code__1: AVLB Proprietary Code: CEIL	Specifies the nature of a balance which is being reported. Rule "RTGS_X123456_ Balance Types And Status _STLD": If balance type is AVLB (available liquidity), OPNG (start balance) or CRRT (current balance), then status must be STLD.
Multilateral Balance Status Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Sts	PDNG STLD	Current status of a cash balance which is being reported. RTGS-Use: mandatory for: AVLB = available liquidity (only STLD is possible)

Table 152 - ReturnAccount (camt.004) – usage case Ceiling Notification

Usage case example: camt.004_RTGS_ReturnAccount_CeilingNotification_Example.xml

In this example a *ReturnAccount* is sent as a push notification to the account owner with information about the current balance and ceiling threshold of the account.

Usage case: Notification of Credit to all Sub-Accounts (Ancillary system Procedure C – start)

In this usage case, the RTGS component is informing the ancillary service of the credit amount applied to all sub-accounts of its RTGS DCAs.

Specific message requirements

Message item	Data type/code	Utilisation
<p>Message Identification</p> <p>Document/RtrAcct/MsgHdr/MsgId</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>When the reference is assigned by the ancillary system to identify the message, the format is composed of 6 characters followed by the RTGS business case identification. The triggering event of the <i>ReturnAccount</i> is precised by the first 6 characters.</p> <p>DAYSOP: start of procedure daylight</p> <p>OVNSOC: start of cycle in night-time</p> <p>DAYSOC: start of cycle in daylight</p> <p>OVNEOP: back transfer of liquidity at end of night-time procedure</p> <p>DAYEOP: back transfer of liquidity at end of daylight procedure</p> <p>OVNSOP: start of procedure night-time (execution of standing orders)</p> <p>SBKLCT: <i>LiquidityCreditTransfer</i> sent by settlement bank</p> <p>CBMAN: mandated payment</p> <p>CBKCSP: connected payment sent by CB</p> <p>CBKCHA: autocollateralisation "CHA" sent by CB</p> <p>CBKREP: autocollateralisation "REP" sent by CB</p> <p>CBKSTR: specific transactions "STR" sent by CB</p> <p>CBKCOL: autocollateralisation "COL" sent by CB</p> <p>CBKCDS: cross-ancillary system settlement by CB</p>
<p>Original Business Query</p> <p>Docu- ment/RtrAcct/MsgHdr/OrgnlBizQry/Msg</p>	RestrictedFINXMax35Text	<p>AS-Use:</p> <p>Value "NONREF" in model 6 as return account is sent in push mode to notify</p>

Message item	Data type/code	Utilisation
Id		the transfers of liquidity and the funds booked on the sub-accounts.
Account Identification Docu- ment/RtrAcct/RptOrErr/AcctRpt/AcctId/ Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	SACC	RTGS-Use: Only settlement account is used.
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
Account Owner Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ownr/Id/OrgId/AnyBIC	AnyBIC	BIC of the RTGS Account Holder owning the account/sub-account which is being queried.
Multilateral Balance Amount Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Amt	ImpliedCurrencyAndAmount fractionDigits: 5 totalDigits: 18 minInclusive: 0	Amount of money of the cash balance.

Message item	Data type/code	Utilisation
Multilateral Balance Credit Debit Indicator Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/CdtDbtInd	CRDT DBIT	Indicates whether the balance is a credit or a debit balance. A zero balance is considered to be a credit balance. RTGS-Use: In the case of BalanceTypeCode NOTE it defines whether it is an earmarked payment for debits or credits.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp/Cd	SystemBalanceType1Code__1: PYMT BOOK	Specifies the nature of a balance which is being reported.
Value Date	DateAndDateTimeChoice__1	Date and time at which the balance is or will be available.

Table 153 - ReturnAccount (camt.004) - usage case Notification of Credit to all Sub-Accounts

Usage case example:
camt.004_RTGS_ReturnAccount_NotificationOfCreditToSubAccounts_Example.xml

In this example a *ReturnAccount* is sent as a push notification to the ancillary system to notify the transfers of liquidity and the funds booked on the sub-accounts.

Usage case: Notification of Liquidity Blocked on all Sub-Accounts (Ancillary system Procedure C – start)

In this usage case, the RTGS component is informing the ancillary system of the amount of liquidity which has been blocked on all sub-accounts of its RTGS DCA.

Specific message content

Message item	Data type/code	Utilisation
<p>Message Identification</p> <p>Document/RtrAcct/MsgHdr/MsgId</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>When the reference is assigned by the ancillary system to identify the message, the format is composed of 6 characters followed by the RTGS business case identification. The triggering event of the <i>ReturnAccount</i> is precised by the first 6 characters.</p> <p>DAYSOP: start of procedure daylight</p> <p>OVNSOC: start of cycle in night-time</p> <p>DAYSOC: start of cycle in daylight</p> <p>OVNEOP: back transfer of liquidity at end of night-time procedure</p> <p>DAYEOP: back transfer of liquidity at end of daylight procedure</p> <p>OVNSOP: start of procedure night-time (execution of standing orders)</p> <p>SBKLCT: <i>LiquidityCreditTransfer</i> sent by settlement bank</p> <p>CBMAN: mandated payment</p> <p>CBKCSP: connected payment sent by CB</p> <p>CBKCHA: autocollateralisation "CHA" sent by CB</p> <p>CBKREP: autocollateralisation "REP" sent by CB</p> <p>CBKSTR: specific transactions "STR" sent by CB</p> <p>CBKCOL: autocollateralisation "COL" sent by CB</p> <p>CBKCDS: cross-ancillary system settlement by CB</p>
<p>Original Business Query</p> <p>Docu- ment/RtrAcct/MsgHdr/OrgnlBizQry/Msg</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>Value "NONREF" in model 6 as <i>ReturnAccount</i> is sent in push mode to</p>

Message item	Data type/code	Utilisation
Id		notify the transfers of liquidity and the funds booked on the sub-accounts.
Account Identification Docu- ment/RtrAcct/RptOrErr/AcctRpt/AcctId/ Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	SACC	RTGS-Use: Only settlement account is used.
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
Account Owner Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ownr/Id/OrgId/AnyBIC	AnyBIC	BIC of the RTGS Account Holder owning the account/sub-account which is being queried.
Multilateral Balance Amount Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Amt	ImpliedCurrencyAndAmount fractionDigits: 5 totalDigits: 18 minInclusive: 0	Amount of money of the cash balance.

Message item	Data type/code	Utilisation
Multilateral Balance Credit Debit Indicator Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/CdtDbtInd	CRDT DBIT	Indicates whether the balance is a credit or a debit balance. A zero balance is considered to be a credit balance. RTGS-Use: In the case of BalanceTypeCode NOTE it defines whether it is a earmarked payment for debits or credits.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp/Cd	SystemBalanceType1Code__1: BOOK	Specifies the nature of a balance which is being reported.
Value Date	DateAndDateTimeChoice__1	Date and time at which the balance is or will be available.

Table 154 - ReturnAccount (camt.004) – usage case Notification of Liquidity Blocked on Sub-Accounts

Usage case example: camt.004_RTGS_ReturnAccount_NotificationOfLiquidityBlockedOnSubAccounts_Example.xml

In this example a *ReturnAccount* is sent as a push notification to the ancillary system to notify about the liquidity blocked on all sub-accounts.

Usage case: Notification of Execution of Stored Immediate Liquidity Transfers (Ancillary system Procedure C – end)

In this usage case, the RTGS component is informing the ancillary system of the stored immediate liquidity orders which have been executed against its RTGS DCAs.

Specific message requirements

Message item	Data type/code	Utilisation
<p>Message Identification</p> <p>Document/RtrAcct/MsgHdr/MsgId</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>When the reference is assigned by the ancillary system to identify the message, the format is composed of 6 characters followed by the RTGS business case identification. The triggering event of the <i>ReturnAccount</i> is precised by the first 6 characters.</p> <p>DAYSOP: start of procedure daylight</p> <p>OVNSOC: start of cycle in night-time</p> <p>DAYSOC: start of cycle in daylight</p> <p>OVNEOP: back transfer of liquidity at end of night-time procedure</p> <p>DAYEOP: back transfer of liquidity at end of daylight procedure</p> <p>OVNSOP: start of procedure night-time (execution of standing orders)</p> <p>SBKLCT: <i>LiquidityCreditTransfer</i> sent by settlement bank</p> <p>CBMAN: mandated payment</p> <p>CBKCSP: connected payment sent by CB</p> <p>CBKCHA: autocollateralisation "CHA" sent by CB</p> <p>CBKREP: autocollateralisation "REP" sent by CB</p> <p>CBKSTR: specific transactions "STR" sent by CB</p> <p>CBKCOL: autocollateralisation "COL" sent by CB</p> <p>CBKCDS: cross-ancillary system settlement by CB</p>
<p>Original Business Query</p> <p>Docu- ment/RtrAcct/MsgHdr/OrgnlBizQry/Msg</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>Value "NONREF" in model 6 as <i>ReturnAccount</i> is sent in push mode to</p>

Message item	Data type/code	Utilisation
Id		notify the transfers of liquidity and the funds booked on the sub-accounts.
Account Identification Docu- ment/RtrAcct/RptOrErr/AcctRpt/AcctId/ Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	SACC	RTGS-Use: Only settlement account is used.
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
Account Owner Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ownr/Id/OrgId/AnyBIC	AnyBIC	BIC of the RTGS Account Holder owning the account/sub-account which is being queried.
Multilateral Balance Amount Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Amt	ImpliedCurrencyAndAmount fractionDigits: 5 totalDigits: 18 minInclusive: 0	Amount of money of the cash balance.

Message item	Data type/code	Utilisation
Multilateral Balance Credit Debit Indicator Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/CdtDbtInd	CRDT DBIT	Indicates whether the balance is a credit or a debit balance. A zero balance is considered to be a credit balance. RTGS-Use: In the case of BalanceTypeCode NOTE it defines whether it is a earmarked payment for debits or credits
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp/Cd	SystemBalanceType1Code__1: BOOK PYMT	Specifies the nature of a balance which is being reported.
Value Date	DateAndDateTimeChoice__1	Date and time at which the balance is or will be available.

Table 155 - ReturnAccount (camt.004) – usage case Notification of Execution of Stored Immediate Liquidity Transfers

Usage case example: camt.004_RTGS_ReturnAccount_NotificationOfExecutionOfStoredImmediateLTs_Example.xml

In this example a *ReturnAccount* is sent as a push notification to the ancillary system to notify the execution of the liquidity transfers booked on the sub-accounts.

Usage case: Notification of Liquidity Re-transfer (Ancillary system Procedure C – end)

In this usage case, the RTGS component is informing the ancillary system of the re-transfer of an amount of liquidity for its RTGS DCAs.

Specific message requirements

Message item	Data type/code	Utilisation
<p>Message Identification</p> <p>Document/RtrAcct/MsgHdr/MsgId</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>When the reference is assigned by the ancillary system to identify the message, the format is composed of 6 characters followed by the RTGS business case identification. The triggering event of the <i>ReturnAccount</i> is precised by the first 6 characters.</p> <p>DAYSOP: start of procedure daylight</p> <p>OVNSOC: start of cycle in night-time</p> <p>DAYSOC: start of cycle in daylight</p> <p>OVNEOP: back transfer of liquidity at end of night-time procedure</p> <p>DAYEOP: back transfer of liquidity at end of daylight procedure</p> <p>OVNSOP: start of procedure night-time (execution of standing orders)</p> <p>SBKLCT: <i>LiquidityCreditTransfer</i> sent by settlement bank</p> <p>CBMAN: mandated payment</p> <p>CBKCSP: connected payment sent by CB</p> <p>CBKCHA: autocolateralisation "CHA" sent by CB</p> <p>CBKREP: autocolateralisation "REP" sent by CB</p> <p>CBKSTR: specific transactions "STR" sent by CB</p> <p>CBKCOL: autocolateralisation "COL" sent by CB</p> <p>CBKCDS: cross-ancillary system settlement by CB</p>
<p>Original Business Query</p> <p>Docu- ment/RtrAcct/MsgHdr/OrgnlBizQry/Msg</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>Value "NONREF" in model 6 as <i>ReturnAccount</i> is sent in push mode to</p>

Message item	Data type/code	Utilisation
Id		notify the transfers of liquidity and the funds booked on the sub-accounts.
Account Identification Docu- ment/RtrAcct/RptOrErr/AcctRpt/AcctId/ Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	SACC	RTGS-Use: Only settlement account is used.
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
Account Owner Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ownr/Id/OrgId/AnyBIC	AnyBIC	BIC of the RTGS Account Holder owning the account/sub-account which is being queried.
Multilateral Balance Amount Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Amt	ImpliedCurrencyAndAmount fractionDigits: 5 totalDigits: 18 minInclusive: 0	Amount of money of the cash balance.

Message item	Data type/code	Utilisation
Multilateral Balance Credit Debit Indicator Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/CdtDbtInd	CRDT DBIT	Indicates whether the balance is a credit or a debit balance. A zero balance is considered to be a credit balance. RTGS-Use: In the case of BalanceTypeCode NOTE it defines whether it is an earmarked payment for debits or credits.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp/Cd	SystemBalanceType1Code__1: PYMT	Specifies the nature of a balance which is being reported.
Value Date	DateAndDateTimeChoice__1	Date and time at which the balance is or will be available.

Table 156 - ReturnAccount (camt.004) – usage case Notification of Liquidity Re-transfer

Usage case example: camt.004_RTGS_ReturnAccount_NotificationOfLiquidityRetransfer_Example.xml

In this example a *ReturnAccount* is sent as a push notification to the ancillary system to notify about the re-transfer of liquidity booked on the sub-accounts.

Usage case: Notification of Global Amount on Liquidity Account (Ancillary system Procedure D - start)

In this usage case, the RTGS component is informing the ancillary system of the overall amount of liquidity on the RTGS DCA owned by the ancillary system.

Specific message content

Message item	Data type/code	Utilisation
<p>Message Identification</p> <p>Document/RtrAcct/MsgHdr/MsgId</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>When the reference is assigned by the ancillary system to identify the message, the format is composed of 6 characters followed by the RTGS business case identification. The triggering event of the <i>ReturnAccount</i> is precised by the first 6 characters.</p> <p>DAYSOP: start of procedure daylight</p> <p>OVNSOC: start of cycle in night-time</p> <p>DAYSOC: start of cycle in daylight</p> <p>OVNEOP: back transfer of liquidity at end of night-time procedure</p> <p>DAYEOP: Back transfer of liquidity at end of daylight procedure</p> <p>OVNSOP: Start of procedure night-time (execution of standing orders)</p> <p>SBKLCT: <i>LiquidityCreditTransfer</i> sent by settlement bank</p> <p>CBMAN: mandated payment</p> <p>CBKCSP: connected payment sent by CB</p> <p>CBKCHA: autocollateralisation "CHA" sent by CB</p> <p>CBKREP: autocollateralisation "REP" sent by CB</p> <p>CBKSTR: specific transactions "STR" sent by CB</p> <p>CBKCOL: autocollateralisation "COL" sent by CB</p> <p>CBKCDS: cross-ancillary system settlement by CB</p>
<p>Original Business Query</p> <p>Docu- ment/RtrAcct/MsgHdr/OrgnlBizQry/Msg</p>	RestrictedFINXMax35Text	<p>Ancillary system - Use:</p> <p>Value "NONREF" in model 6 as <i>ReturnAccount</i> is sent in push mode to</p>

Message item	Data type/code	Utilisation
Id		notify the transfers of liquidity and the funds booked on the sub-accounts.
Account Identification Docu- ment/RtrAcct/RptOrErr/AcctRpt/AcctId/ Othr/Id	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	SACC	RTGS-Use: Only settlement account is used.
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
Account Owner Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ownr/Id/OrgId/AnyBIC	AnyBIC	BIC of the RTGS Account Holder owning the account/sub-account which is being queried.
Multilateral Balance Amount Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Amt	ImpliedCurrencyAndAmount fractionDigits: 5 totalDigits: 18 minInclusive: 0	Amount of money of the cash balance.

Message item	Data type/code	Utilisation
Multilateral Balance Credit Debit Indicator Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/CdtDbtInd	CRDT DBIT	Indicates whether the balance is a credit or a debit balance. A zero balance is considered to be a credit balance. RTGS-Use: In the case of BalanceTypeCode NOTE it defines whether it is an earmarked payment for debits or credits.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp/Cd	SystemBalanceType1Code__1: BOOK	Specifies the nature of a balance which is being reported.
Value Date	DateAndDateTimeChoice__1	Date and time at which the balance is or will be available.

Table 157 - ReturnAccount (camt.004) – usage case Notification of Global Amount on Liquidity Account

Usage case example: camt.004_RTGS_ReturnAccount_NotificationOfGlobalAmountOnLiquidityAccount_Example.xml

In this example a *ReturnAccount* is sent as a push notification to the ancillary system to notify about the global amount booked on the dedicated liquidity account.

14.3.3 GetTransaction (camt.005)

14.3.3.1 Overview and scope of the message

This chapter illustrates the *GetTransaction* message.

The *GetTransaction* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to request information about payments and payments and liquidity transfers held in the RTGS component.

The *GetTransaction* message can be used to request the above information based upon optional multiple criteria.

Within RTGS, the *GetTransaction* message has the following usages:

- Payment Query

In response to the *GetTransaction* message, a [ReturnTransaction \(camt.006\)](#) [▶ 431] message containing the requested information is returned.

14.3.3.2 Schema

Outline of the schema.

The *GetTransaction* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

TransactionQueryDefinition

This building block is mandatory. It contains detailed information related to the business query criteria about the transaction.

QueryType

Specifies the type of matching items to be returned in the response to the query.

QueryName

Recalls the criteria (search and return criteria) defined in a preceding query.

SearchCriteria

Non-repetitive when used. It defines the criteria on which the information is extracted. It includes the following elements:

- | payment to
- | payment from
- | entry information: ordering message identification, requested execution date, payment identification, status, instructed amount, instructed currency, debit/credit indicator, payment method, payment type, priority
- | account identification
- | account owner
- | entry date

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.005.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *GetTransaction* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.3.3 The message in business context

Usage case: Payment Query

In this usage case, the sender requests information regarding the details of all payments within their query criteria.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
MessageIdentification Document/GetTx/MsgHdr/MsgId	RTGS_RestrictedFINXMax35Text	Identification of the message
QueryType Document/GetTx/TxQryDef/QryTp	QueryType2Code	Specifies the type of matching items to be returned in the response to the query. Code (ALLL, CHNG, DELD, MODF)
QueryName Docu- ment/GetTx/TxQryDef/TxCrit/QryNm	Max35Text	Recalls the criteria (search and return criteria) defined in a preceding query.
SearchCriteria		
PaymentTo/MemberIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtTo/Mmbld/FinInstnId/BICFI	RTGS_BIC11Text	Search on member's BIC to which the payment is sent
PaymentTo/Country	CountryCode	Search on country to which the payment is sent

Message item	Data type/code	Utilisation
Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtTo/Ctry		
PaymentFrom/MemberIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtFr/Mmbld/FinInstnId/BICFI	RTGS_BIC11Text	Search on member's BIC from which the payment is originated
PaymentFrom/Country Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtFr/Ctry	CountryCode	Search on country from which the payment is originated
MessageIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/MsgId	RTGS_RestrictedFINXMax35Text	Search on MessageIdentification
RequestedExecutionDate Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/ReqdExctnDt/DtSch	DatePeriodSearch1Choice	Search on RequestedExecutionDate Range
LongBusinessIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/PmtId/LngBizId	LongPaymentIdentification1	Search on Payment LongBusinessI- dentification
PendingStatus Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Sts/PmtInstrSts/PdgSts	PendingStatus4Code	Search on pending status: ACPD, PSTL, STLE, STLM
FinalStatus Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Sts/PmtInstrSts/FnlSts	FinalStatusCode	Search on final status: CAND, RJTD, STLD
PendingOrFinal Docu-	CashPaymentStatus2Code	Search on final or pending status (FINL, PDNG)

Message item	Data type/code	Utilisation
ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Sts/PmtInstrSts/PdgAndFnlSts		
InstructedAmount /Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/InstdAmt/ImpldCcyAndAmtRg	ImpliedCurrencyAndAmountRange1	Search on instructed amount range and debit/credit indicator
InstructedAmountCurrency Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/InstdAmtCcy	ActiveOrHistoricCurrencyCode	Search on instructed amount currency
CreditDebitIndicator Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/CdtDbtInd	CreditDebitCode	Search on payment in debit or credit
InterbankSettlementAmount Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/IntrBkSttlmAmt/ImpldCcyAndAmtRg/	ImpliedCurrencyAndAmountRange1	Search on interbank settlement amount range and debit/credit indicator
InterbankSettlementAmountCurrency Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/IntrBkSttlmAmtCcy	ActiveCurrencyCode	Search on interbank settlement amount currency
PaymentMethod Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/PmtMtd	PaymentOrigin1Choice	Search on XML message name carrying the payment OR on proprietary codes (ASXML, INTERN, ORDER)
PaymentType Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/PmtTp/Prtry	RTGS_PaymentTypeCode	Search on BACP, LIQP, ASYP, MANP, BIDB, ASTI, REGP

Message item	Data type/code	Utilisation
PaymentPriority /Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Prty/Cd	Priority1Code	Search on HIGH, NORM, (URGT*) *CR to ISO
ProcessingValidityTime Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/PrcgVldtyTm	DateTimePeriod1Choice	Search on processing validity date time range
TransactionIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/TxId	Max35Text	Search on transaction identification
EndToEndIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/EndToEndId	Max35Text	Search on end to end identification
Debtor BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Pties/Dbtr/FinInstnId/BICFI	RTGS_BIC11Text	Search on debtor's BIC
DebtorAgent BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Pties/DbtrAgt/FinInstnId/BICFI	RTGS_BIC11Text	Search on debtor agent's BIC
IntermediaryAgent BIC /Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Pties/IntrmyAgt/FinInstnId/BICFI	RTGS_BIC11Text	Search on intermediary agent's BIC
CreditorAgent BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Pties/CdtrAgt/FinInstnId/BICFI	RTGS_BIC11Text	Search on creditor agent's BIC

Message item	Data type/code	Utilisation
Creditor' BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Pties/Cdtr/FinInstnId/BI CFI	RTGS_BIC11Text	Search on creditor's BIC
AccountIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/AcctNtrySch/AcctId/EQ/Othr/Id	RestrictedFINXMax34Text	Search on the cash entry account iden- tification
Entry DateTime Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/AcctNtrySch/NtryDt/DtTmSch	DateTimePeriod1Choice	Search on the entry date time range
Account Owner Name Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/AcctNtrySch/AcctOwnr/Nm	Max140Text	Search on the account owner's name
Account Owner BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/AcctNtrySch/AcctOwnr/Id/OrgId/ AnyBIC	AnyBICIdentifier	Search on the account owner's BIC
ReturnCriteria specify which information is requested in the camt.006		
Pay- mentTo/MemberIdentificationIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtToRtrCrit/MmbldInd	RequestedIndicator	True, false If absent, default value is true
Pay- mentFrom/MemberIdentificationIndicat or Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtFrRtrCrit/MmbldInd	RequestedIndicator	True, false If absent, default value is true
EntryDateIndicator	RequestedIndicator	True, false

Message item	Data type/code	Utilisation
Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/AcctCshNtryRtrCrit/NtryDtInd		If absent, default value is true
MessageIdentificationIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/MsgIdInd	RequestedIndicator	True, false If absent, default value is true
PaymentInstructionStatusIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/InstrStsRtrCrit/PmtIn strStsInd	RequestedIndicator	True, false If absent, default value is true
PaymentInstructionStatusDateTimelndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/InstrStsRtrCrit/PmtIn strStsDtTmInd	RequestedIndicator	True, false If absent, default value is true
PaymentInstructionStatusReasonIndi- cator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/InstrStsRtrCrit/PmtIn strStsRsnInd	RequestedIndicator	True, false If absent, default value is true
InstructedAmountIndicator /Document/GetTx/TxQryDef/TxCrit/Ne wCrit/RtrCrit/PmtRtrCrit/InstdAmtInd	RequestedIndicator	True, false If absent, default value is true
CreditDebitIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/CdtDbtInd	RequestedIndicator	True, false If absent, default value is false
InterbankSettlementAmountIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/IntrBkSttlmAmtInd	RequestedIndicator	True, false If absent, default value is true

Message item	Data type/code	Utilisation
PriorityIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/PrtyInd	RequestedIndicator	True, false If absent, default value is true
ProcessingValidityTimeIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/PrcgVldtyTmInd	RequestedIndicator	True, false If absent, default value is true
InstructionCopyIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/InstrCpyInd	RequestedIndicator	True, false If absent, default value is true
PaymentTypeIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/PmtTplInd	RequestedIndicator	True, false If absent, default value is true
TransactionIdentificationIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/TxIdInd	RequestedIndicator	True, false If absent, default value is true
InterbankSettlementDateIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/IntrBkSttlmDtInd	RequestedIndicator	True, false If absent, default value is true
EndToEndIdentificationIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/EndToEndIdInd	RequestedIndicator	True, false If absent, default value is true
PaymentMethodIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/PmtMtdInd	RequestedIndicator	True, false If absent, default value is true
DebtorIndicator Docu-	RequestedIndicator	True, false If absent, default value is true

Message item	Data type/code	Utilisation
ment/GetTx/TxQryDef/TxCrit/NewCrit/RtrCrit/PmtRtrCrit/DbtrInd		
DebtorAgentIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/RtrCrit/PmtRtrCrit/DbtrAgtInd	RequestedIndicator	True, false If absent, default value is true
IntermediaryIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/RtrCrit/PmtRtrCrit/IntrmyInd	RequestedIndicator	True, false If absent, default value is true
CreditorAgentIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/RtrCrit/PmtRtrCrit/CdtrAgtInd	RequestedIndicator	True, false If absent, default value is true
CreditorIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/RtrCrit/PmtRtrCrit/CdtrInd	RequestedIndicator	True, false If absent, default value is true

Table 158 - GetTransaction (camt.005) – usage case Payment Query

Usage case example 1: camt.005_RTGS_GetTransaction_PaymentOrderQuery_Example1.xml

Simple query based on the Payment TransactionIdentification. There is no return criteria specified meaning the respective return criteria default values are applied.

Usage case example 2: camt.005_RTGS_GetTransaction_PaymentOrderQuery_Example2.xml

Multiple criteria are defined in the query. The query purpose is to extract for the authorized user all the transactions (regular payments) with a “Pending” status, an amount greater of equal to 15.000 EUR, with a debit entry on the account number “Account123456”, and submitted to the system by using pacs.009.001.07 or pacs.008.001.07.

The return criteria are the respective default values except for InstructionCopy and EntryDate on the account which are not requested.

Usage case example 3: camt.005_RTGS_GetTransaction_PaymentOrderQuery_Example3.xml

The system has responded to a previous query (example 1) by returning a query name associated to the search and return criteria defined in that previous query. In the next queries using the same criteria, the user can just refer to the query name to activate the same search and return criteria as shown in example 3.

Usage case example 4: camt.005_RTGS_GetTransaction_PaymentOrderQuery_Example4.xml

In this example, the query requests only the new matching items since the last similar query based on camt.005.001.07.

14.3.4 ReturnTransaction (camt.006)

14.3.4.1 Overview and scope of the message

This chapter illustrates the *ReturnTransaction* message.

The *ReturnTransaction* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to provide information on the details of one or more payments and/or payments or liquidity transfers held in the RTGS component.

The *ReturnTransaction* message contains such information based upon RTGS DCAs held at the RTGS component and upon the criteria provided in the request.

Within RTGS, the *ReturnTransaction* message has the following usages:

- Payment Query (Data or Error Response)

The *ReturnTransaction* message is sent in response to a [GetTransaction \(camt.005\)](#) [▶ 421] message, which requested the information.

14.3.4.2 Schema

Outline of the schema.

The *ReturnTransaction* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about transaction, or an error indication.

TransactionReport

This building block is mandatory and repetitive. It reports either on the transaction information or on a business error. When it reports the transaction information, it may contain:

- | payment identification
- | payment to
- | payment from
- | debit/credit indicator
- | account
- | entry date
- | payment details: payment message identification, status, instructed amount, interbank settlement amount, payment method, priority, processing validity time, payment type, debtor, debtor agent, intermediary agent, creditor agent, creditor

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.006.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnTransaction* response message.

14.3.4.3 The message in business context

Usage case: Payment Query (Data Response)

In this usage case, the recipient of the message is being informed regarding the details of all payments within their query criteria.

If a problem is encountered while retrieving this information, the error information is reported instead.

Specific message requirements

Message item	Data type/code	Utilisation
MessageIdentification	RTGS_RestrictedFINXMax35Text	Identification of the message
Document/RtrTx/MsgHdr/MsgId		
OriginalBusinessQuery MessageIdenti-	RTGS_RestrictedFINXMax35Text	Identification of the original business

Message item	Data type/code	Utilisation
ification Docu- ment/RtrTx/MsgHdr/OrgnBizQry/Msgld		query message
QueryName Document/RtrTx/MsgHdr/QryNm	Max35Text	Name of the query allocated by the system in the return message
ReportOrError		
TransactionReport		
PaymentIdentification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Pmt Id	PaymentIdentification5Choice	Choice between QueueIdentification, LongBusinessIdentification and Short-BusinessIdentification
PaymentTo MemberIdentification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/PmtTo/Mmbld/FinInstnld/BIC FI	RTGS_BIC11Text	BIC of the member to which the pay- ment is sent
PaymentFrom MemberIdentification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/PmtFr/Mmbld/FinInstnld/BICF I	RTGS_BIC11Text	BIC of the member from which the payment is originated
CreditDebitIndicator Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/CdtDbtInd	CreditDebitCode	CRDT, DBIT
Payment MessageIdentification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Msgld	RTGS_RestrictedFINXMax35Text	Identification of the message contain- ing the payment
Status Code Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Sts/Cd	PaymentStatusCode6Choice	Pending (ACPD, PSTL, STLE, STLM), final (CAND, RJTD, STLD)
Status Date Time	ISODateTime	Date and time at which the status was

Message item	Data type/code	Utilisation
Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Sts/DtTm/DtTm		assigned
Rejection Reason Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Sts/Rsn/PrtryRjctn/PrtryS tsRsn	ProprietaryStatusJustification1	Rejection reason code and description
InstructedAmount Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/InstdAmt/AmtWthCcy	RTGS_Max14_Max5DecimalAmount	Amount and currency instructed in the payment
InterbankSettlementAmount /Document/RtrTx/RptOrErr/BizRpt/TxR pt/TxOrErr/Tx/Pmt/IntrBkSttlmAmt/Amt WthCcy	RTGS_Max14_Max2DecimalAmount	Interbank settlement amount with cur- rency
PaymentMethod Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/PmtMtd	PaymentOrigin1Choice	Payment method provided by XML message name or proprietary code (ASXML, INTERN, ORDER)
Payment Priority Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Prty/Cd	PriorityCode3Choice	HIGH, NORM, (URGT*) *CR submitted
ProcessingValidityTime /Document/RtrTx/RptOrErr/BizRpt/TxR pt/TxOrErr/Tx/Pmt/PrcgVldtyTm	DateTimePeriod1Choice	Date and time range within which the payment instruction must be pro- cessed.
InstructionCopy Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/InstrCpy	Max20000Text	Copy of the instruction in free format
Payment Type Docu-	RTGS_PaymentTypeCode	ASYP, ASTI, BACP, BIDB, LIQP, MANP, REGP

Message item	Data type/code	Utilisation
ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Tp/Prtry		
TransactionIdentification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/TxId	Max35Text	Unique identification, as assigned by the first instructing agent, to unambiguously identify the transaction that is passed on, unchanged, throughout the entire interbank chain.
InterbankSettlementDate Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/IntrBkSttlmDt	ISODate	Date on which the amount of money ceases to be available to the agent that owes it and when the amount of money becomes available to the agent to which it is due.
EndToEndIdentification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/EndToEndId	Max35Text	Unique identification, as assigned by the initiating party, to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.
Debtor's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Pties/Dbtr/FinInstnId/BICFI	RTGS_BIC11Text	BIC of the debtor
DebtorAgent's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Pties/DbtrAgt/FinInstnId/BICFI	RTGS_BIC11Text	BIC of the debtor agent
IntermediaryAgent's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Pties/IntrmyAgt/FinInstnId/BICFI	RTGS_BIC11Text	BIC of the intermediary agent
CreditorAgent's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx	RTGS_BIC11Text	BIC of creditor agent

Message item	Data type/code	Utilisation
OrErr/Tx/Pmt/Pties/CdtrAgt/FinInstnId/BICFI		
Creditor's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Pties/Cdtr/FinInstnId/BIC FI	RTGS_BIC11Text	BIC of creditor
Entry Account Identification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/AcctNtry/Acct/Id/Othr/Id	Max34Text	Account to or from which a cash entry is made.
Entry Date Time Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/AcctNtry/Ntry/Dt/DtTm	ISODateTime	Date and time of the posting cash entry on the account

Table 159 - ReturnTransaction (camt.006) – usage case Payment Query (Data Response)

Usage case example 1: camt.006_RTGS_ReturnTransaction_PaymentOrderQueryData_Example1.xml

The message returns the payment information identified with the transaction identification provided in the query message (camt.005_RTGS_GetTransaction_PaymentOrderQuery_Example1).

Usage case example 2: camt.006_RTGS_ReturnTransaction_PaymentOrderQueryData_Example2.xml

The message returns the 2 transactions meeting the search criteria defined in the query message (camt.005_RTGS_GetTransaction_PaymentOrderQuery_Example2).

Usage case: Payment Query (Error Response)

In this usage case, the recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent [GetTransaction \(camt.005\)](#) [▶ 421]

Specific message requirements

Message item	Data type/code	Utilisation
Error Docu- ment/RtrTx/RptOrErr/OprlErr/Err/Prtry	Max4Text	Proprietary code
Description Docu- ment/RtrTx/RptOrErr/OprlErr/Desc	Max140Text	Specification of the error, in free format.

Table 160 - ReturnTransaction (camt.006) – usage case Payment Query (Error Response)

Usage case example: camt.006_RTGS_ReturnTransaction_PaymentOrderQueryError_Example.xml

The query execution fails and the reason is reported in the operational error component with a code. The meaning of the code is defined in the UDFS.

14.3.5 ModifyTransaction (camt.007)

14.3.5.1 Overview and scope of the message

This chapter illustrates the *ModifyTransaction* message.

The *ModifyTransaction* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to modify one payment on the RTGS Participant's RTGS DCA.

The *ModifyTransaction* may only be used for a payment which is in a transient status (i.e. it has not reached a final status such as rejected, revoked or settled).

The *ModifyTransaction* message will contain the new value that the RTGS Account Holder wants to be applied to the relevant feature of the payment identified in the message. Only one feature, of one order, may be changed in a single *ModifyTransaction* message.

Within RTGS, the *ModifyTransaction* message has the following usages:

- I Amend Payment

In response to the *ModifyTransaction* message, a [Receipt \(camt.025\)](#) [▶ 474] is sent, indicating the success or rejection/failure of the modification.

14.3.5.2 Schema

Outline of the schema

The *ModifyTransaction* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Modification

This building block is mandatory and non-repetitive. It identifies the payment and the modification to be executed. The modifiable attributes are:

- | priority
- | processing validity time

References/Links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.007.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *ModifyTransaction* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.5.3 The message in business context

Usage case: Amend Payment

In this usage case, the sender is requesting that the priority or processing validity time for a previously sent payment should be changed to the values provided in this message.

The previously sent payment must not be already settled, for this amendment to take effect.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
MessageIdentification Document/ModifyTx/MsgHdr/Msgld	RTGS_RestrictedFINXMax35Text	Message identification will be copied from BAH
PaymentIdentification Docu- ment/ModifyTx/Mod/Pmtld/LngBizld	LongPaymentIdentification1	Payment is identified by the LongBusinessIdentification including TransactionIdentification, InterbankSettlementAmount, InterbankSettlementDate, PaymentMethod, InstructingAgent and InstructedAgent
New Payment Value Set		
Priority Docu- ment/ModifyTx/Mod/NewPmtValSet/Prt y	PriorityCode3Choice	Priority is a choice between a code (HIGH, NORM) or a queue reordering proprietary code (DECR, INCR)
ProcessingValidityTime Docu- ment/ModifyTx/Mod/NewPmtValSet/Prc gVldtyTm	DateTimePeriod1Choice	Choice between FromDateTime or ToDateTime

Table 161 - ModifyTransaction (camt.007) – usage case Amend Payment

Usage case example1: camt.007_RTGS_ModifyTransaction_AmendPaymentOrder_Example1.xml

ModifyTransaction sent to change the priority of a payment to HIGH.

Usage case example2: camt.007_RTGS_ModifyTransaction_AmendPaymentOrder_Example2.xml

ModifyTransaction sent for reordering a payment in the queue (top or bottom) to the top (INCR).

Usage case example3: camt.007_RTGS_ModifyTransaction_AmendPaymentOrder_Example3.xml

ModifyTransaction sent to change the earliest execution time of a payment.

Usage case example4: camt.007_RTGS_ModifyTransaction_AmendPaymentOrder_Example4.xml

ModifyTransaction sent to change the latest execution time of a payment.

14.3.6 GetLimit (camt.009)

14.3.6.1 Overview and scope of the message

This chapter illustrates the *GetLimit* message.

The *GetLimit* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to request details of one or more limits set by the RTGS Account Holder (or on their behalf by an authorised party) and managed by RTGS.

Within RTGS, the *GetLimit* message has the following usages:

- I Current Limits Query

In response to the *GetLimit* message, a [ReturnLimit \(camt.010\)](#) [▶ 442] message containing the requested information is returned.

14.3.6.2 Schema

Outline of the schema.

The *GetLimit* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

LimitQueryDefinition

This building block is mandatory. It contains detailed information related to the business query about limit.

SearchCriteria

This building block is mandatory and non-repetitive. It defines the criteria to extract the limit information. It includes the following elements:

- I account owner
- I account identification

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.009.001.06_RTGS

Business rules applicable to the schema

For business rules applicable to *GetLimit* please refer to the chapter [Index of business rules and error codes](#) [▶ 670]

14.3.6.3 The message in business context

Usage case: Current Limits Query

In this usage case, the sender requests information regarding the limits currently set against RTGS DCAs within their query criteria.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
MessageIdentification Document/GetLmt/MsgHdr/MsgId	RestrictedFINMax35Text	Message identification copied from BAH
LimitQueryDefinition – Search Criteria		
AccountIdentification Docu- ment/GetLmt/LmtQryDef/LmtCrit/NewC rit/SchCrit/AcctId/Othr/Id	Max34Text	Used in case of searching via RTGS DCA number
AccountOwner´s BIC Docu- ment/GetLmt/LmtQryDef/LmtCrit/NewC rit/SchCrit/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Used in case the request is on behalf of third party. Sending party and partic- ipant (account owner are different).
AccountOwner´s Name Docu- ment/GetLmt/LmtQryDef/LmtCrit/NewC rit/SchCrit/AcctOwnr/FinInstnId/Nm	Max140Text	Used in case the request is on behalf of third party. Sending party and partic- ipant (account owner are different).

Table 162 - GetLimit (camt.009) – usage case Current Limits Query

Usage case example 1: camt.009_RTGS_GetLimit_CurrentLimitsQuery_Example1.xml

Example with no query definition. By default the query is about the sending party´s RTGS DCA.

Usage case example 2: camt.009_RTGS_GetLimit_CurrentLimitsQuery_Example2.xml

This example illustrates a request on behalf of a third party meaning the sending party and participant (account's owner) are different.

Usage case example3: camt.009_RTGS_GetLimit_CurrentLimitsQuery_Example3.xml

This example illustrates a search via a RTGS DCA number.

14.3.7 ReturnLimit (camt.010)

14.3.7.1 Overview and scope of the message

This chapter illustrates the *ReturnLimit* message.

The *ReturnLimit* message is sent by RTGS to a RTGS Account Holder (or a party authorised by them). It is used to provide details of one or more limits set by the RTGS Account Holder (or on their behalf by an authorised party).

Within RTGS, the *ReturnLimit* message has the following usages:

- Current Limits Query (Data or Error Response)

The *ReturnLimit* message is sent in response to a [GetLimit \(camt.009\)](#) [▶ 440] message, which requested the information.

14.3.7.2 Schema

Outline of the schema

The *ReturnLimit* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the message and the original business query identification.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query message about limit, or an error indication.

Current limit

This building block is optional but repetitive. It reports on either a current limit or on a business error. When it reports the current limit information, it may contain:

- | limit identification
- | amount
- | debit/credit indicator

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.010.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnLimit* response message.

14.3.7.3 The message in business context

Usage case: Current Limits Query (Data Response)

In this usage case, the recipient of the message is being informed regarding the limits currently set against RTGS DCAs within their query criteria.

Specific message content

Message item	Data type/code	Utilisation
MessageIdentification Document/RtrLmt/MsgHdr/MsgId	RestrictedFINMax35Text	Message identification
OriginalBusinessQuery MessageIdentification Document/RtrLmt/MsgHdr/OrgnlBizQry/MsgId		Message identification of the <i>GetLimit</i> generating the current response It is copied from the BAH
BusinessReport		
Current BilateralLimitCounterpartyIdentification BIC Document/RtrLmt/RptOrErr/BizRpt/CurLmt/L	BICFIIdentifier	BIC of the bilateral limit counterparty used to identify the limit

Message item	Data type/code	Utilisation
mtId/BilLmtCtrPtyId/FinInstnId/BICF		
CurrentLimit Type Docu- ment/RtrLmt/RptOrErr/BizRpt/CurLmt/L mtId/Tp/Cd	LimitType3Code	Bilateral limit (BILI), multilateral limit (MULT)
AccountOwner's BIC Docu- ment/RtrLmt/RptOrErr/BizRpt/CurLmt/L mtId/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Account owner's BIC used to identify the limit
AccountIdentification Docu- ment/RtrLmt/RptOrErr/BizRpt/CurLmt/L mtId/AcctId/Othr/Id	Max34Text	Account identification used to identify the limit
Limit Amount Docu- ment/RtrLmt/RptOrErr/BizRpt/CurLmt/L mtOrErr/Lmt/Amt/AmtWthCcy	ActiveCurrencyAndAmount	Limit amount and currency
LimitCreditDebitIndicator Docu- ment/RtrLmt/RptOrErr/BizRpt/CurLmt/L mtOrErr/Lmt/CdtDbtInd	CreditDebitCode	Specifies if a limit is a debit limit or a credit limit. Only DBIT is used

Table 163 - ReturnLimit (camt.010) – usage case Current Limits Query (Data Response)

Usage case example 1: camt.010_RTGS_ReturnLimit_CurrentLimitsQueryData_Example1.xml

The example shows the *ReturnLimit* response for current multilateral limit.

Usage case example 2: camt.010_RTGS_ReturnLimit_CurrentLimitsQueryData_Example2.xml

The example shows the *ReturnLimit* response for current bilateral limit.

Usage case: Current Limits Query (Error response)

In this usage case, the recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent [GetLimit \(camt.009\)](#) [▶ 440].

Specific message requirements

Message item	Data type/code	Utilisation
Error Document/RtrLmt/RptOrErr/OprlErr/Err/ Prtry	Max4Text	Proprietary code reporting the error
Description Docu- ment/RtrLmt/RptOrErr/OprlErr/Desc	Max140Text	Additional information on the error

Table 164 - ReturnLimit (camt.010) – usage case Current Limits Query (Error response)

Usage case example: camt.010_RTGS_ReturnLimit_CurrentLimitsQueryError_Example.xml

The example shows the *ReturnLimit* response in case of error reported in the “CODE” which meaning is defined in the UDFS.

14.3.8 ModifyLimit (camt.011)

14.3.8.1 Overview and scope of the message

This chapter illustrates the *ModifyLimit* message.

The *ModifyLimit* is sent by an authorised party for instructing the update of a standing order for limit or RTGS limit, by providing details about the standing order for limit or RTGS limit to be updated.

The *ModifyLimit* message has the following usages:

- I CRDM update standing order for limit
- I RTGS update limit

In response to the modify standing order for limit message, CRDM sends a [Receipt \(camt.025\)](#) [▶ 474] message when the update of the standing order for limit has been successfully performed or rejected.

14.3.8.2 Schema

Outline of the schema

The *ModifyLimit* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

LimitDetails

This block contains detailed information related to the standing order for limit or RTGS limit to be updated.

It contains detailed information related to the limit to be updated. It includes the following elements:

- | The identification of the CB responsible for the account owner
- | the identification of the credit consumer
- | the type of limit to be updated
- | the identification of the credit provider
- | the identification of the account
- | the limit amount to set
- | the date from which the standing order for limit or RTGS limit is valid

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/camt.011.001.006>

14.3.8.3 The message in business context

Usage case: CRDM Update Standing Order for Limit

This usage case describes the update of a standing order for limit in CRDM.

Specific message requirements

Message item	Data type/code	Utilisation
Identification Docu- ment/ModfyLmt/LmtDtIs/LmtId/Dflt/Sysl d/MktInfrstrctrId/Prtry	BICFIIdentifier	NCB BIC
Identification Docu- ment/ModfyLmt/LmtDtIs/LmtId/Dflt/BiL mtCtrPtyId/FinInstnId/BICFI	BICFIIdentifier	Bilateral limit counterparty identification

Message item	Data type/code	Utilisation
Type Docu- ment/ModifyLmt/LmtDtls/LmtId/Dflt/Tp	LimitType1Choice	Limit Type
Account owner Docu- ment/ModifyLmt/LmtDtls/LmtId/Dflt/Acct Ownr/FinInstnId/BICFI	BICFIIdentifier	Account owner
Account identification Docu- ment/ModifyLmt/LmtDtls/LmtId/Dflt/Acct Id/Othr/Id	Max34Text	Account identifier
Valid from Docu- ment/ModifyLmt/LmtDtls/NewLmtValSet /StartDtTm/Dt	ISODate	Valid from
Amount Docu- ment/ModifyLmt/LmtDtls/NewLmtValSet /Amt/AmtWthCcy	ActiveCurrencyAndAmount	Amount

Table 165 - ModifyLimit (camt.011) – usage case CRDM Update Standing Order for Limit

Usage case example: CRDMUpdateLimit_example.xml

In this example a CB with BIC “NCBAXXYAAA” requests the update of multilateral standing order for limit set for payment bank with BIC “PAYBXYAAA” owning the cash account identified with “ACC001”.

Usage Case: RTGS Update Limit

This usage case describes the update of a current limit for RTGS.

Specific message requirements

Message item	Data type/code	Utilisation
Identification Docu- ment/ModifyLmt/LmtDtls/LmtId/Cur/BilL mtCtrPtyId/FinInstnId/BICFI	BICFIIdentifier	Bilateral limit counterparty identification
Type Docu- ment/ModifyLmt/LmtDtls/LmtId/Cur/Tp	LimitType1Choice	Limit type
Account owner Docu- ment/ModifyLmt/LmtDtls/LmtId/Cur/Acct Ownr/FinInstnId/BICFI	BICFIIdentifier	Account owner
Account identification Docu- ment/ModifyLmt/LmtDtls/LmtId/Cur/Acct Id/Othr/Id	Max34Text	Account identifier
Amount Docu- ment/ModifyLmt/LmtDtls/NewLmtValSet /Amt/AmtWthCcy	ActiveCurrencyAndAmount	Amount

Table 166 - ModifyLimit (camt.011) – usage case RTGS Update Limit

Usage case example: RTGSUpdateLimit_example.xml

In this example it is requested the update of multilateral limit set for payment bank with BIC “PAYBXXY-YAAA” owning the cash account identified with “ACC001”.

14.3.9 DeleteLimit (camt.012)

14.3.9.1 Overview and scope of the message

This chapter illustrates the *DeleteLimit* message.

The *DeleteLimit* is sent by an authorised actor for instructing the deletion of a standing order for limit or RTGS limit, by providing details about the standing order for limit or RTGS limit to be deleted.

The *DeleteLimit* message has the following usages:

- I CRDM delete standing order for limit
- I RTGS delete limit

In response to the delete standing order for limit message, CRDM sends a [Receipt \(camt.025\)](#) [▶ 474] message when the deletion of the standing order for limit has been successfully performed or rejected.

14.3.9.2 Schema

Outline of the schema

The *DeleteLimit* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and it contains an identification assigned by the sending party to uniquely and unambiguously identify the message.

LimitDetails

This building block is mandatory. It contains detailed information related to the standing order for limit or RTGS limit to be deleted. It includes elements uniquely identifying a standing order for limit or RTGS limit as responsible CB, credit consumer, type, credit provider and cash account identification.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/camt.011.001.006>

14.3.9.3 The message in business context

Usage case: CRDM Delete Standing Order for Limit

This usage case describes the deletion of a standing order for limit in CRDM.

Specific message requirements

Message item	Data type/code	Utilisation
Identification Document/DeLmt/LmtDtIs/CurLmtId/SysId/MktInfrstrctrId/Prtry	BICFIIdentifier	NCB BIC
Identification Document/DeLmt/LmtDtIs/CurLmtId/BilLmtCtrPtyId/FinInstnId/BICFI	BICFIIdentifier	Bilateral limit counterparty identification
Type Document/DeLmt/LmtDtIs/CurLmtId/Tp/Cd	LimitType3Code	Limit type
Account owner Document/DeLmt/LmtDtIs/CurLmtId/AcctOwner/FinInstnId/BICFI	BICFIIdentifier	Account owner
Account identification Document/DeLmt/LmtDtIs/CurLmtId/AcctId/Othr/Id	Max34Text	Account identifier

Table 167 - DeleteLimit (camt.012) – usage case CRDM Delete Standing Order for Limit

Usage case example: CRDMDeleteLimit_example.xml

In this example a CB with BIC “NCBAXXYAAA” requests the deletion of the multilateral standing order for limit set for payment bank identified with BIC “PAYBXYAAA” owning the cash account identified with “ACC001”.

Usage Case: RTGS Delete Limit

This usage case describes the deletion of a limit for RTGS.

Specific message requirements

Message item	Data type/code	Utilisation
Identification Docu- ment/DelLmt/LmtDtIs/CurLmtId/BilLmtC trPtyId/FinInstnId/BICFI	BICFIIdentifier	Bilateral limit counterparty identification
Type Docu- ment/DelLmt/LmtDtIs/CurLmtId/Tp/Cd	LimitType3Code	Limit type
Account owner Docu- ment/DelLmt/LmtDtIs/CurLmtId/AcctOw nr/FinInstnId/BICFI	BICFIIdentifier	Account owner
Account identification Docu- ment/DelLmt/LmtDtIs/CurLmtId/AcctId/ Othr/Id	Max34Text	Account identifier

Table 168 - DeleteLimit (camt.012) – usage case RTGS Delete Limit

Usage case example: RTGSDeleteLimit_example.xml

In this example it is requested the deletion of the multilateral limit set for payment bank identified with BIC “PAYBXXYAAA” owning the cash account identified with “ACC001”.

14.3.10 GetBusinessDayInformation (camt.018)

14.3.10.1 Overview and scope of the message

This chapter illustrates the *GetBusinessDayInformation* message.

The *GetBusinessDayInformation* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to request information on different types of administrative data linked to the RTGS component system.

Within RTGS, the *GetAccount* message has the following usages:

- | Event Query
- | System Time Query

In response to the *GetBusinessDayInformation* message, a [ReturnBusinessDayInformation \(camt.019\)](#) [▶ 453] message containing the requested information is returned.

14.3.10.2 Schema

Outline of the schema

The *GetBusinessDayInformation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

RequestType

This building block is mandatory and consists of pre-determined content.

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.018.001.04_RTGS

Business rules applicable to the schema

For business rules applicable to *GetBusinessDayInformation* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.10.3 The message in business context

Usage case: Event Query

In this usage case, the sender is requesting information regarding the execution of processing events in the RTGS component.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Document/GetBizDayInf/MsgHdr/MsgId	Max35Text	Unique ID for the message

Table 169 - GetBusinessDayInformation (camt.018) – usage case Event Query

Usage case example: camt.018_RTGS_GetBusinessDayInformation_EventQuery_Example.xml

Usage case: System Time Query

In this usage case, the sender is requesting to be informed of the RTGS system time.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Document/GetBizDayInf/MsgHdr/MsgId	Max35Text	Unique ID for the message
Docu- ment/GetBizDayInf/MsgHdr/ReqTp/Enq ry	ExternalEnquiryRequestTypeCode	Proprietary value for RTGS system time query - TBC

Table 170 - GetBusinessDayInformation (camt.018) – usage case System Time Query

Usage case example: camt.018_RTGS_GetBusinessDayInformation_SystemTime Query_Example.xml

14.3.11 ReturnBusinessDayInformation (camt.019)

14.3.11.1 Overview and scope of the message

This chapter illustrates the *ReturnBusinessDayInformation* message.

The *ReturnBusinessDayInformation* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to provide information on the details of on different types of administrative data linked to the RTGS component system.

The *ReturnBusinessDayInformation* message contains such administrative data information based upon the criteria provided in the request.

Within RTGS, the *ReturnBusinessDayInformation* message has the following usages:

- | System Time Query (Data or Error Response)
- | Event Query (Data or Error Response)
- | System Notification

The *ReturnBusinessDayInformation* message is sent in response to a [GetBusinessDayInformation \(camt.018\)](#) [▶ 451] message, which requested the information. The System Notification usage is sent in push mode, based upon operational settings with the CLM component.

14.3.11.2 Schema

Outline of the schema

The *ReturnBusinessDayInformation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about business day information, or an error indication.

BusinessDayOrError

This building block reports either the system availability for a specific business day or business error when information has not been found. When it reports the business day information, it may contain:

- | system date
- | events per currency
- | event scheduled and effective date/times

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

<http://www.swift.com/mystandards/RTGS/camt.019.001.06> RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnBusinessDayInformation* response message.

14.3.11.3 The message in business context

Usage case: System Time Query (Data Response)

In this usage case, the recipient of the message is being informed regarding the details of the status and time of the RGTS component.

Specific message requirements

Message item	Data type/code	Utilisation
System ID Docu- ment/RtrBizDayInf/RptOrErr/BizRpt/Sy sId/MktInfrstrctrId/Cd	ExternalMarketInfrastructureCode	RTG
Business date Docu- ment/RtrBizDayInf/RptOrErr/BizRpt/Biz DayOrErr/BizDayInf/SysDt	ISODate	Current business date of RTGS

Table 171 - ReturnBusinessDayInformation (camt.019) – usage case System Time Query (Data Response)

Usage case example:
camt.019_RTGS_ReturnBusinessDayInformation_SystemTimeQueryData_Example.xml

Usage case: System Time Query (Error Response)

In this usage case, the recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent System Time Query ([GetBusinessDayInformation \(camt.018\) \[▶ 451\]](#)).

The identification of the previously sent query message is included in this error response for reconciliation purposes.

Specific message requirements

Message item	Data type/code	Utilisation
Proprietary Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Prtry/	Max4Text	RTGS code for the problem being informed
Description Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Desc	Max140Text	Description of the problem being informed

Table 172 - ReturnBusinessDayInformation (camt.019) – usage case System Time Query (Error Response)

Usage case example:
camt.019_RTGS_ReturnBusinessDayInformation_SystemTimeQueryError_Example.xml

Usage case: Event Query (Data Response)

In this usage case, the recipient of the message is being informed regarding the details of a RTGS processing event.

Specific message requirements

Message item	Data type/code	Utilisation
Proprietary Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Prtry/	Max4Text	RTGS code for the problem being informed
Description Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Desc	Max140Text	Description of the event being informed

Table 173 - ReturnBusinessDayInformation (camt.019) – Event Query (Data Response)

Usage case example:
camt.019_RTGS_ReturnBusinessDayInformation_EventQueryData_Example.xml

Usage case: Event Query (Error Response)

In this usage case, the recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent Event Query ([GetBusinessDayInformation \(camt.018\)](#) [▶ 451]).

The identification of the previously sent query message is included in this error response for reconciliation purposes.

Specific message requirements

Message item	Data type/code	Utilisation
Proprietary Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Prtry/	Max4Text	RTGS code for the problem being informed
Description Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Desc	Max140Text	Description of the problem being informed

Table 174 - ReturnBusinessDayInformation (camt.019) – usage case Event Query (Error Response)

Usage case example:
camt.019_RTGS_ReturnBusinessDayInformation_EventQueryError_Example.xml

Usage case: System Notification

In this usage case, the recipient of the message is being informed of operational information situations as they arise within the RTGS component.

Specific message requirements

Message item	Data type/code	Utilisation
Proprietary Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Prtry/	Max4Text	RTGS code for the problem being informed
Description Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Desc	Max140Text	Description of the status

Table 175 - ReturnBusinessDayInformation (camt.019) – usage case System Notification

Usage case example:
camt.019_RTGS_ReturnBusinessDayInformation_SystemNotification_Example.xml

14.3.12 ReturnGeneralBusinessInformation (camt.021)

14.3.12.1 Overview and scope of the message

This chapter illustrates the *ReturnGeneralBusinessInformation* message.

The *ReturnGeneralBusinessInformation* message is sent in both directions between the RTGS component and an ancillary system. It is used to provide information related to the processing of the systems.

The *ReturnGeneralBusinessInformation* message is sent by the RTGS component, to inform about the start or end of a procedure or cycle within the RTGS component system.

The *ReturnGeneralBusinessInformation* message is received by the RTGS component from an ancillary system, to be informed about whether a procedure at the ancillary system is open or closed.

In both directions, the *ReturnGeneralBusinessInformation* message can contain either static data announcing foreseen events affecting the system operations, or dynamic data warning or notifying about unexpected events.

Within RTGS, the *ReturnGeneralBusinessInformation* message has the following usages:

- | AS Procedure C – Start Procedure
- | AS Procedure C – Notify Ancillary System About Start of Procedure
- | AS Procedure C – End of Procedure
- | AS Procedure C Cycle – Start of Cycle
- | AS Procedure C Cycle – Notify Ancillary System
- | AS Procedure C Cycle – End of Cycle
- | AS Procedure D – Notify Ancillary System About Start of Procedure
- | AS Procedure D Cycle – Start of Cycle
- | AS Procedure D Cycle – Notify Ancillary System
- | AS Procedure D Cycle – End of Cycle

The *ReturnGeneralBusinessInformation* is triggered by events and processing inside the RTGS component or an ancillary system. It is not a response to any form of query.

14.3.12.2 Schema

Outline of the schema.

The *ReturnGeneralBusinessInformation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains information about business day information. In RTGS there is no error usage.

GeneralBusiness

This building block is mandatory and non-repetitive. It reports the business information. It may contain:

- | qualifier for use with ancillary systems
- | subject (procedure or cycle indication)
- | subject details (BIC identifying the ancillary system)

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.021.001.06_RTGS

Business rules applicable to the schema

When used in its outbound form from the RTGS component, no business rules are applicable to a *ReturnGeneralBusinessInformation* message,

When used in its inbound form from an ancillary system, for business rules applicable to *ReturnGeneralBusinessInformation* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.12.3 The message in business context

Usage case: Ancillary System Procedure C – Start Procedure

In this usage case, the ancillary system (or the RTGS component on behalf of the ancillary system) sends this message to the RTGS component to indicate that ancillary system procedure C should be started by the RTGS component.

Specific message requirements

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the ancillary system: OVN-CYCL-OPEN = open overnight cycle
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 176 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C – Start Procedure

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureCStart_Example.xml

In this example, a *ReturnGeneralBusinessInformation* is instructed by the ancillary system to initiate ancillary system procedure C. It illustrates the mandatory elements in the message.

Usage case: AS Procedure C – Notify Ancillary System About Start of Procedure

In this usage case, the RTGS component is informing the ancillary system that ancillary system procedure C has been started within the RTGS component processing.

Specific message requirements

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmttd	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the RTGS: OVN-PROC-OPN = overnight procedure is open
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 177 - ReturnGeneralBusinessInformation (camt.021) – usage case AS Procedure C – Notify Ancillary System About Start of Procedure

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureCNotifyASAboutStart_Example.xml

In this example a *ReturnGeneralBusinessInformation* is sent as a push notification to the ancillary system to inform about the start of ancillary system procedure C.

Usage case: Ancillary System Procedure C – End of Procedure

In this usage case, the RTGS component is informing the ancillary system that ancillary system procedure C has finished processing within the RTGS component.

Specific message requirements

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle.
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: codes sent by the RTGS: LIQ-CYCL-END = release of liquidity after end of cycle
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 178 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C – End of Procedure

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureCEndOfProcedure_Example.xml

In this example a *ReturnGeneralBusinessInformation* is sent as a push notification to the ancillary system to inform about the end of ancillary system procedure C.

Usage case: Ancillary System Procedure C Cycle – Start of Cycle

In this usage case, the ancillary system (or the RTGS component on behalf of the ancillary system) sends this message to the RTGS component to indicate that the ancillary system procedure C cycle should be started by the RTGS component.

Specific message content

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the ancillary system: OVN-CYCL-OPEN = open overnight cycle
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtIs	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 179 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C Cycle – Start of Cycle

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureCCycleStart_Example.xml

In this example, a *ReturnGeneralBusinessInformation* is instructed by the ancillary system to initiate ancillary system procedure C cycle. It illustrates the mandatory elements in the message.

Usage case: Ancillary System Procedure C Cycle – Notify Ancillary System About Start of Cycle

In this usage case, the RTGS component is informing the ancillary system that the ancillary system procedure C cycle has been started within the RTGS component processing.

Specific message content

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the RTGS: OVN-CYCL-OPEN = open overnight cycle
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 180 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C Cycle – Notify Ancillary System About Start of Cycle

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureCCycleNotifyASAboutStart_Example.xml

In this example a *ReturnGeneralBusinessInformation* is sent as a push notification to the ancillary system to inform about the start of ancillary system procedure C cycle.

Usage case: Ancillary System Procedure C Cycle – End of Cycle

In this usage case, the RTGS component is informing the ancillary system that the AS Procedure C cycle has finished processing within the RTGS component.

Specific message content

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the RTGS: LIQ-CYCL-END = release of liquidity after end of cycle
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 181 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure C Cycle – End of Cycle

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureCCycleEndOfCycle_Example.xml

In this example a *ReturnGeneralBusinessInformation* is sent as a push notification to the ancillary system to inform about the end of ancillary system procedure C cycle.

Usage case: Ancillary System Procedure D – Notify Ancillary System About Start of Procedure

In this usage case, the RTGS component is informing the ancillary system that ancillary system procedure D has been started within the RTGS component processing.

Specific message content

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the RTGS: DAY-PROC-OPEN = open daylight procedure
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 182 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure D – Notify Ancillary System About Start of Procedure

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureDNotifyASAboutStart_Example.xml

In this example a *ReturnGeneralBusinessInformation* is sent as a push notification to the ancillary system to inform about the start of ancillary system procedure D.

Usage case: Ancillary System Procedure D Cycle – Start of Cycle

In this usage case, the ancillary system (or the RTGS component on behalf of the ancillary system) sends this message to the RTGS component to indicate that the ancillary system procedure D cycle should be started by the RTGS component.

Specific message content

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the ancillary system: DAY-CYCL-OPEN = open daylight cycle (interfaced ancillary system only)
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 183 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure D Cycle – Start of Cycle

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureDCycleStart_Example.xml

In this example, a *ReturnGeneralBusinessInformation* is instructed by the ancillary system to initiate ancillary system procedure D cycle. It illustrates the mandatory elements in the message.

Usage case: Ancillary System Procedure D Cycle – Notify Ancillary System About Start of Cycle

In this usage case, the RTGS component is informing the ancillary system that the ancillary system procedure D cycle has been started within the RTGS component processing.

Specific message content

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the AS: DAY-CYCL-OPEN = open daylight cycle (interfaced ancillary system only)
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 184 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure D Cycle – Notify Ancillary System About Start of Cycle

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureDCycleNotifyASAboutStart_Example.xml

In this example a *ReturnGeneralBusinessInformation* is sent as a push notification to the ancillary system to inform about the start of ancillary system procedure D cycle.

Usage case: Ancillary System Procedure D Cycle – End of Cycle

In this usage case, the RTGS component is informing the ancillary system that the ancillary system procedure D cycle has finished processing within the RTGS component.

Specific message content

Message item	Data type/code	Utilisation
Business Information Reference RtrGnlBizInf/RptOrErr/BizRpt/BizInfRef	Max16Text	BusinessInformationReference contained in the ReturnGeneralBusinessInformation sent by the ancillary system or the CB to request the end of cycle
Qualifier RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Qlfr/ IsFrmtD	InformationQualifierType1__1	True False
Subject RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ Sbjt	Max35Text	RTGS-Use: ancillary system usage: code sent by the RTGS: DAY-CYCL-CLOS = close daylight cycle (interfaced ancillary system only)
Subject Details RtrGnlBizInf/RptOrErr/ GnlBizOrErr/ GnlBiz/ SbjtDtls	Max350Text	Details of the code RTGS-Use: ancillary system usage: If the sender is a CB on behalf of the ancillary system, this field is filled with the BIC of the ancillary system.

Table 185 - ReturnGeneralBusinessInformation (camt.021) – usage case Ancillary System Procedure D Cycle – End of Cycle

Usage case example:
camt.021_RTGS_ReturnGeneralBusinessInformation_ASProcedureDCycleEndOfCycle_Example.xml

In this example a *ReturnGeneralBusinessInformation* is sent as a push notification to the ancillary system to inform about the end of ancillary system procedure D cycle.

14.3.13 ModifyStandingOrder (camt.024)

14.3.13.1 Overview and scope of the message

This chapter illustrates the *ModifyStandingOrder* message.

The *ModifyStandingOrder* message is sent by an actor authorised to create or modify standing orders for liquidity transfers.

The *ModifyStandingOrder* message has the following usages:

- I RTGS modify standing order
- I ASI6 RTGS modify standing order
- I CLM modify standing order

The *ModifyStandingOrder* message is replied by a [Receipt \(camt.025\)](#) [▶ 474] to return a positive technical response to the sender of the message or to provide detailed information in case of an error.

14.3.13.2 Schema

Outline of the schema

The *ModifyStandingOrder* message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides the message identification provided by the requesting actor.

StandingOrderIdentification

This block is mandatory and provides with all the key information to identify an existing standing order to be amended or a new standing order to be created.

NewStandingOrderValueSet

This block is mandatory and provide with the pieces of information related to the standing order to be modified or created.

It includes the amount to be transferred, the required account references to perform the transfer, the intended validity period and the execution type in terms of event identification.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/camt.024.001.05>

14.3.13.3 The message in business context

Usage case: RTGS modify standing order

This usage case describes the update of a standing order in CRDM for RTGS component.

Specific message requirements

Message item	Data type/code	Utilisation
Standing order identification Document/ModifyStgOrdr/StgOrdrId/Id	RestrictedFINMax16Text	Identification
Account identification Docu- ment/ModifyStgOrdr/StgOrdrId/Acct/Id/ Othr/Id	RestrictedFINMax34Text	Account identification
Type Docu- ment/ModifyStgOrdr/StgOrdrId/Acct/Tp/ Prtry	Exact4AlphaNumericText	Possible values are: CREA - To Create a Standing Order UPDA - To Modify a Standing Order
Amount Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /Amt/AmtWthCcy	RestrictedFINActiveCurrencyAndA- mount	Amount
Creditor account Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /CdtrAcct/Id/Othr/Id	RestrictedFINMax34Text	Creditor account
Execution type Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /ExctnTp/Evt/Cd	ExternalSystemEventType1Code	Execution type
Validity period Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /VldtyPrd	DatePeriod2Choice	Validity period

Table 186 - ModifyStandingOrder (camt.024) – usage case RTGS modify standing order

Usage case example: RTGSModifyStandingOrder_example.xml

In this example it is requested to update the standing order with id “STOID00001” for the account identified with “ACC001”.

Usage case: ASI6 RTGS Modify standing order

This usage case describes the update of a standing order in CRDM for RTGS component for ASI procedure 6.

Specific message requirements

Message item	Data type/code	Utilisation
Standing order identification Document/ModifyStgOrdr/StgOrdrId/Id	RestrictedFINMax16Text	Identification
Account identification Docu- ment/ModifyStgOrdr/StgOrdrId/Acct/Id/ Othr/Id	RestrictedFINMax34Text	Account identification
Type Docu- ment/ModifyStgOrdr/StgOrdrId/Acct/Tp/ Prtry	Exact4AlphaNumericText	Possible values are: CREA - To Create a Standing Order UPDA - To Modify a Standing Order
Account owner Docu- ment/ModifyStgOrdr/StgOrdrId/AcctOw nr/FinInstId/BICFI	BICFIIdentifier	Technical account BIC
Amount Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /Amt/AmtWthCcy	RestrictedFINActiveCurrencyAndA- mount	Amount
Creditor TBD	TBD	Creditor BIC
Creditor account Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /CdtrAcct/Id/Othr/Id	RestrictedFINMax34Text	Creditor account
Debtor TBD	TBD	Debtor BIC

Message item	Data type/code	Utilisation
Debtor account Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /DbtrAcct/Id/Othr/Id	RestrictedFINMax34Text	Debtor account
Execution type Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /ExctnTp/Evt/Cd	ExternalSystemEventType1Code	Execution type
Validity period Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /VldtyPrd	DatePeriod2Choice	Validity period

Table 187 - ModifyStandingOrder (camt.024) – usage case ASI6 RTGS Modify standing order

Usage case example: ASI6RTGSModifyStandingOrder_example.xml

Usage case: CLM modify standing order

This usage case describes the update of a standing order in CRDM for CLM component.

Specific message requirements

Message item	Data type/code	Utilisation
Standing order identification Document/ModifyStgOrdr/StgOrdrId/Id	RestrictedFINMax16Text	Identification
Account identification Docu- ment/ModifyStgOrdr/StgOrdrId/Acct/Id/ Othr/Id	RestrictedFINMax34Text	Account identification
Type Docu- ment/ModifyStgOrdr/StgOrdrId/Acct/Tp/ Prtry	Exact4AlphaNumericText	Possible values are: CREA - To Create a Standing Order UPDA - To Modifiy a Standing Order
Amount Docu- ment/ModifyStgOrdr/NewStgOrdrValSet	RestrictedFINActiveCurrencyAndA- mount	Amount

Message item	Data type/code	Utilisation
/Amt/AmtWthCcy		
Creditor account Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /CdtrAcct/Id/Othr/Id	RestrictedFINMax34Text	Creditor account
Execution type Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /ExctnTp/Evt/Cd	ExternalSystemEventType1Code	Execution type
Validity period Docu- ment/ModifyStgOrdr/NewStgOrdrValSet /VldtyPrd	DatePeriod2Choice	Validity period

Table 188 - ModifyStandingOrder (camt.024) – usage case CLM modify standing order

Usage case example: CLMModifyStandingOrder_example.xml

In this example it is requested to update the standing order with id “STOID00002” for the account identified with “ACC001”.

14.3.14 Receipt (camt.025)

14.3.14.1 Overview and scope of the message

This chapter illustrates the *Receipt* message.

The *Receipt* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to reply to a previously sent liquidity transfer, payment order or order-related activity.

The *Receipt* message will return a positive response to the sender of the previous message or will provide detailed information in case of an error.

The *Receipt* message can also be used by an ancillary system to advise the RTGS component of a decision regarding the use of ancillary system guarantee processing. This is the only case where the *Receipt* message is an inbound message into the RTGS component.

Within RTGS, the *Receipt* message has the following usages:

Note: due to the large number of usage cases and the similarities of the *Receipt* message, this section is dealt with by organising the usage cases into usage categories.

- | Usage category – Settlement
 - Payment Rejection Notification
 - Payment Settlement Notification (Liquidity Transfer)
 - Payment Settlement Notification (Inter-Service Liquidity Transfer)
- | Usage category – RTGS Status
 - Reject Limit Maintenance Request
 - Confirm Successful Limit Maintenance Request
- | Usage Category – Liquidity Management
 - Reject Reservation Maintenance Request
 - Confirm Successful Reservation Maintenance Request
- | Usage Category – CRDM
 - Create/Modify Standing Order
 - Delete Standing Order
 - Modify Limit
 - Delete Limit
 - Modify Standing Order for Reservation
 - Delete Standing Order for Reservation
- | Usage category – Ancillary System Processing
 - Invoke Guarantee Processing

The *Receipt* message is sent in response to several situations, both as a response to an action, and as an unsolicited update related to a previous action.

14.3.14.2 Schema

Outline of the schema

The *Receipt* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReceiptDetails

This building block is mandatory and non-repetitive. It provides information relating to the status of a previous instruction. It may contain:

- | original message identification
- | status code
- | description

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.025.001.04_RTGS

Business rules applicable to the schema

No business rules are applicable to a *Receipt* message.

14.3.14.3 The message in business context

Usage category – Settlement

All usage cases in this category will see a similar set of possible *Receipt* responses. For this reason, the usage category is described below and is relevant to all usage cases in this category.

Usage category case – Settlement – Rejected

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has been rejected and will not be processed further. A rejection code will be given and, in most cases, a reason code and reason text will be provided also.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnIMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	RRJT
Description Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ desc	RestrictedFINXMax140Text	RTGS rejection

Table 189 - Receipt (camt.025) – usage category case Settlement - Rejected

Usage category case example: camt.025_RTGS_Receipt_SettlementRejected_Example.xml

Usage category case – Settlement – Settled

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has been settled.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnIMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	SSET
Description Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	settled

Table 190 - Receipt (camt.025) – usage category case Settlement - Settled

Usage category case example: camt.025_RTGS_Receipt_SettlementSettled_Example.xml

Usage category case – Settlement – Unsettled

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has failed to reach full settlement.

A reason code will be given and, in most cases a reason text will be provided also.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnIMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	SUNS
Description Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	unsettled

Table 191 - Receipt (camt.025) – usage category case Settlement - Unsettled

Usage case example: camt.025_RTGS_Receipt_SettlementUnsettled_Example.xml

Usage category – RTGS Status

All usage cases in this category will see a similar set of possible *Receipt* responses. For this reason, the usage category is described below and is relevant to all usage cases in this category.

Usage category case – RTGS Status – Rejected

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has been rejected and will not be processed further. A rejection code will be given and, in most cases, a reason code and reason text will be provided also.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnIMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	RRJT
Description Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	RTGS rejection

Table 192 - Receipt (camt.025) – usage category case RTGS Status - Rejected

Usage category case example: camt.025_RTGS_Receipt_RTGSStatusRejected_Example.xml

Usage category case – RTGS Status – Confirmed

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has been confirmed.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnIMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	RCON
Description Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	RTGS confirmation

Table 193 - Receipt (camt.025) usage category case RTGS Status - Confirmed

Usage category case example: camt.025_RTGS_Receipt_RTGSStatusConfirmed_Example.xml

Usage category case – RTGS Status – Validation Error

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has failed CLM validation checks.

A reason code will be given and, in most cases a reason text will be provided also.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnIMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	VSTS
Description Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	Status codes/reason are defined according to chapter Index of business rules and error codes [▶ 670]

Table 194 - Receipt (camt.025) usage category case RTGS Status – Validation Error

Usage case example: camt.025_RTGS_Receipt_RTGSStatusValidationError_Example.xml

Usage category – Liquidity Management

All usage cases in this category will see a similar set of possible *Receipt* responses. For this reason, the usage category is described below and is relevant to all usage cases in this category.

Usage category case – Liquidity Management – Approved

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has been approved and successfully processed.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	REJT
Description Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	rejected

Table 196 - Receipt (camt.025) usage category case CRDM - Rejected

Usage category case example: camt.025_RTGS_Receipt_CRDMRejected_Example.xml

Usage category case – CRDM – Completed

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has been successfully completed.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	COMP
Description Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	completed

Table 197 - Receipt (camt.025) usage category case CRDM - Completed

Usage category case example: camt.025_RTGS_Receipt_CRDMCompleted_Example.xml

Usage category case – CRDM – Queued

In this usage category case, the recipient of the message is being informed that a previously sent message in this category sent by them (or on their behalf) has not yet been processed but is waiting a queue to be executed at a later time.

A reason code will be given and, in most cases a reason text will be provided also.

Specific message content

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	QUED
Description Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	queued

Table 198 - Receipt (camt.025) usage category case CRDM - Queued

Usage case example: camt.025_RTGS_Receipt_CRDMQueued_Example.xml

Usage category – Ancillary System Processing

All usage cases in this category are related specifically to ancillary system processing with the RTGS component. Each usage fulfils a single specific need.

Usage category case – Ancillary System Processing – Invoke Guarantee Processing

In this usage category case, the ancillary system is providing a “yes” or “no” response to a [ASInitiationStatus \(pain.998\)](#) [▶ 620] message previously sent by the RTGS component to the ancillary system.

The previous *ASInitiationStatus* message was sent to indicate that further settlement of payments under normal conditions was not possible and that the ancillary system has the option to invoke the use of further funds provided under the ancillary system guarantee process.

Specific message requirements

Message item	Data type/code	Utilisation
Original message Docu- ment/Rct/RctDtIs/OrgnIMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the <i>ASInitiationStatus</i> message from the RTGS component.
Status Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	APPR/REJT
Description Docu- ment/Rct/RctDtIs/OrgnIMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	Approved/rejected

Table 199 - Receipt (camt.025) usage category case Ancillary System Processing – Invoke Guarantee Processing

Usage category case example 1:
camt.025_RTGS_Receipt_ASProcessingInvokeGuaranteeProcessingYES_Example.xml

In this example, the ancillary system is responding to an *ASInitiationStatus* message from the RTGS component with reference ABCD. The ancillary system is indicating a YES response, meaning that guarantee processing should be invoked.

Usage category case example 2:
camt.025_RTGS_Receipt_ASProcessingInvokeGuaranteeProcessingNO_Example2.xml

In this example, the ancillary system is responding to an *ASInitiationStatus* message from the RTGS component with reference ABCD. The ancillary system is indicating a NO response, meaning that guarantee processing should not be invoked.

14.3.15 ResolutionOfInvestigation (camt.029)

14.3.15.1 Overview and scope of the message

This chapter illustrates the *ResolutionOfInvestigation* message.

The *ResolutionOfInvestigation* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to inform of the status of a previously requested payment cancellation.

The *ResolutionOfInvestigation* message only concerns the cancellation of one payment.

Within RTGS, the *ResolutionOfInvestigation* message has the following usages:

- I Rejection of Payment Cancellation Request (Negative Resolution to Investigation)

The *ResolutionOfInvestigation* message is sent in response to a [FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 536] message.

14.3.15.2 Schema

Outline of the schema

The *ResolutionOfInvestigation* message is composed of the following message building blocks:

Assignment

Identifies the assignment of an investigation case from an assigner to an assignee. The assigner must be the sender of this message and the assignee must be the receiver.

Status

Indicates the status of the investigation/cancellation.

Cancellation details

Specifies some of the details of the underlying transactions being cancelled.

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.029.001.08_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ResolutionOfInvestigation* response message.

14.3.15.3 The message in business context

Usage case: Rejection of Payment Cancellation Request

In this usage case, the recipient is being informed that a previously sent request to cancel a payment ([FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 536]), has been rejected. The payment will still go ahead settle, or has already settled.

Specific message content

Message item	Data type/code	Utilisation
Id RsltOfInvstgtn/ Assgnmt/Id	RTGS_RestrictedFINXMax35Text	PDCR
BICFI RsltOfInvstgtn/ Assgnr/Agt/FinInstnId/ BICFI	RTGS_BIC11Text	BIC
BICFI RsltOfInvstgtn/ Assgne /Agt/FinInstnId/ BICFI	RTGS_BIC11Text	BIC
Confirmation RsltOfInvstgtn/ Conf	ExternalInvestigationExecutionConfir- mation1Code	RJCR
Cancellation Status Identification RsltOfInvstgtn/ CxlDtls/TxInfAndSts/ CxlStsId	RTGS_RestrictedFINXMax35Text	The cancellation status identification can be used for reconciliation or to link tasks relating to the cancellation request.
Original Message Identification RsltOfInvstgtn/ CxlDtls/TxInfAndSts/ OrgnlMsgId	RTGS_RestrictedFINXMax35Text	If present in underlying FIToFIPaymentCancellationRequest (camt.056) [536], the original instruction identification is recommended to be transported in the <i>ResolutionOfInvestigation</i> under original instruction identification.
Original Message Name Identification RsltOfInvstgtn/ CxlDtls/TxInfAndSts/ OrgnlGrpInf/ OrgnlMsgNmId	RTGS_XMLMessageNamePattern	Specifies the original message name identifier to which the message refers, e.g. pacs.008.001.07

Message item	Data type/code	Utilisation
Original Instruction Identification RsltnOfInvstgtn/ CxlDtls/TxInfAndSts/ OrgnlInstrld	Max35Text	If present in underlying FIToFIPaymentCancellationRequest (camt.056) [▶ 536], the original instruction identification is recommended to be transported in the <i>ResolutionOfInvestigation</i> under original instruction identification.
Code RsltnOfInvstgtn/ CxlDtls/TxInfAndSts/ CxlStsRsnInf/ Rsn/ Cd	ExternalPaymentCancellationRejection1Code	RJCR
Additional Information RsltnOfInvstgtn/ CxlDtls/TxInfAndSts/ CxlStsRsnInf/ Rsn/ AddtlInf	Max105Text	“No return of funds”

Table 200 - ResolutionOfInvestigation (camt.029) – usage case Rejection of Payment Cancellation Request

Usage case example:
camt.029_RTGS_ResolutionOfInvestigation_RejectionOfPaymentOrderCancellationRequest_Example.xml

14.3.16 GetReservation (camt.046)

14.3.16.1 Overview and scope of the message

This chapter illustrates the *GetReservation* message.

The *GetReservation* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to request details of one or more reservation facilities set by the RTGS Account Holder (or on their behalf by an authorised party).

The *GetReservation* message can be used to request reservation information based on several criteria.

Within RTGS, the *GetReservation* message has the following usages:

- Current Reservations Query

In response to the *GetReservation* message, a [ReturnReservation \(camt.047\)](#) [▶ 489] message containing the requested information is returned.

14.3.16.2 Schema

Outline of the schema

The *GetReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

ReservationQueryDefinition

Definition of the reservation query.

SearchCriteria

Mandatory and non-repetitive. It defines the criteria to extract the reservation information. It includes the following elements:

- | account owner
- | account identification

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.046.001.04_RTGS

Business rules applicable to the schema

For business rules applicable to *GetReservation* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.16.3 The message in business context

Usage case: Current Reservations Query

In this usage case, the sender requests information regarding the all reservations currently set against RTGS DCAs within their query criteria.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
BICFI Document/GetRsvatn/ RsvatnQryDef/ RsvatnCrit/ NewCrit/ SchCrit/ Ac- ctOwnr/ FinInstnId/ BICFI	Max35Text	RTGS Account Holder BIC
Identification Document/GetRsvatn/ RsvatnQryDef/ RsvatnCrit/ NewCrit/ SchCrit/ AcctId / Othr / ID	Max35Text	RTGS Account Holder account ID

Table 201 - GetReservation (camt.046) – usage case Current Reservations Query

Usage case example: camt.046_RTGS_GetReservation_CurrentReservationsQuery_Example.xml

14.3.17 ReturnReservation (camt.047)

14.3.17.1 Overview and scope of the message

This chapter illustrates the *ReturnReservation* message.

The *ReturnReservation* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to provide information on the details of one or more reservation facilities set by the RTGS Account Holder (or on their behalf by an authorised party).

Within RTGS, the *ReturnReservation* message has the following usages:

- Current Reservations Query (Data or Error Response)

The *ReturnReservation* message is sent in response to a [GetReservation \(camt.046\)](#) [▶ 487] message which requested the information.

14.3.17.2 Schema

Outline of the schema

The *ReturnReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the message and the original business query identification.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query message, or an error indication.

CurrentReservation

This building block is optional but repetitive. It reports on either a current reservation or on a business error. When it reports the current reservation information, it may contain:

- | reservation identification
- | reservation type
- | account owner
- | account identification

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.047.001.05_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnReservation* response message.

14.3.17.3 The message in business context

Usage case: Current Reservations Query (Data Response)

In this usage case, the recipient of the message is being informed regarding the details of all reservations currently set against RTGS DCAs within their query criteria.

Specific message content

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Code Docu- ment/RtrRsvatn/RptOrErr/BizRpt/CurR svatn/RsvatnId/Tp/Cd	ReservationType2Code	HPAR UPAR
BICFI Docu- ment/RtrRsvatn/RptOrErr/BizRpt/CurR svatn/RsvatnId/AcctOwnr/FinInstnId/BI CFI	BICFIIdentifier	Current reservation – RTGS DCA own- er
Amount with Currency Docu- ment/RtrRsvatn/RptOrErr/BizRpt/CurR svatn/RsvatnOrErr/Rsvatn/Amt/AmtWth Ccy	ActiveCurrencyAndAmount	Current reservation – amount and currency
Current reservation – status Docu- ment/RtrRsvatn/RptOrErr/BizRpt/CurR svatn/RsvatnOrErr/Rsvatn/Sts/Cd	ReservationStatusCode	ENAB REQD

Table 202 - ReturnReservation (camt.047) – usage case Current Reservations Query (Data Response)

Usage case example:
camt.047_RTGS_ReturnReservation_CurrentReservationsQueryData_Example.xml

Usage case: Current Reservations Query (Error Response)

In this usage case, the recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent Current Reservations Query ([GetReservation \(camt.046\)](#) [▶ 487]).

The identification of the previously sent query message is included in this error response for reconciliation purposes.

Specific message content

Message item	Data type/code	Utilisation
Code Docu- ment/RtrRsvatn/RptOrErr/OprlErr/Err/P rtry/Cd	ErrorHandlingCode	RTGS code for the problem being informed.
Description Docu- ment/RtrRsvatn/RptOrErr/OprlErr/Desc	Max140Text	Description of the problem being informed.

Table 203 - ReturnReservation (camt.047) – usage case Current Reservations Query (Error Response)

Usage **case** **example:**
camt.047_RTGS_ReturnReservation_CurrentReservationsQueryError_Example.xml

14.3.18 ModifyReservation (camt.048)

14.3.18.1 Overview and scope of the message

This chapter illustrates the *ModifyReservation* message.

The *ModifyReservation* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) directly to CRDM. It is used to modify existing reservations set by the RTGS Account Holder (or on their behalf by an authorised party).

The *ModifyReservation* message will contain the new value that the RTGS Account Holder wants to be applied to the reservations identified in the message.

Within RTGS, the *ModifyReservation* message has the following usages:

- I Modify Reservation Request
- I Modify Standing Order for Reservation

In response to the *ModifyReservation* message, a [Receipt \(camt.025\)](#) [▶ 474] is sent, indicating the success or rejection/failure of the modification.

14.3.18.2 Schema

Outline of the schema

The *ModifyReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReservationIdentification

Identification of the reservation (current or default).

NewReservationValueSet

This building block is mandatory and non-repetitive. It identifies the modification to be executed. The modifiable attributes are:

- | amount with currency
- | start date

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.048.001.04_RTGS

Business rules applicable to the schema

For business rules applicable to *ModifyReservation* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.18.3 The message in business context

Usage case: Modify Reservation Request

In this usage case, the sender is requesting that a previously set reservation on a RTGS DCA is modified to the new attributes provided.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Code Docu- ment/ModifyRsvatn/RsvatnId/Cur/Tp/Cd	ReservationTypeCode	HPAR UPAR
BICFI Docu- ment/ModifyRsvatn/RsvatnId/Cur/AcctO wnr/FinInstnId/BICFI	BICFIIdentifier	Current reservation identification – RTGS DCA owner
New reservation amount Docu- ment/ModifyRsvatn/NewRsvatnValSet/ Amt/AmtWthCcy	ActiveCurrencyAndAmount	New reservation amount required

Table 204 - ModifyReservation (camt.048) – usage case Modify Reservation Request

Usage case example: camt.048_RTGS_ModifyReservation_ModifyReservationRequest_Example.xml

Usage case: Modify Standing Order for Reservation

In this usage case, the sender is changing elements of the reference data entry for a standing order for reservation currently stored on CRDM, to the new attributes provided.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
	TBC	

Table 205 - ModifyReservation (camt.048) – usage case Modify Standing Order for Reservation

Usage case example:
camt.048_RTGS_ModifyReservation_ModifyStandingOrderForReservationRequest_Example.xml

14.3.19 DeleteReservation (camt.049)

14.3.19.1 Overview and scope of the message

This chapter illustrates the *DeleteReservation* message.

The *DeleteReservation* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to request the deletion of one particular reservation set by the RTGS Account Holder (or on their behalf by an authorised party).

The *DeleteReservation* message allows for the deletion of only one reservation facility.

Within RTGS, the *DeleteReservation* message has the following usages:

- | Delete Reservation Request
- | Delete Standing Order for Reservation

In response to the *DeleteReservation* message, a [Receipt \(camt.025\)](#) [▶ 474] message is sent, indicating the success or rejection/failure of the deletion.

14.3.19.2 Schema

Outline of the schema

The *DeleteReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

CurrentReservation

This building block identifies the current reservation to delete. The available attributes to do this identification are:

- | reservation type
- | account owner
- | account identification

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.049.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *DeleteReservation* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.19.3 The message in business context

Usage case: Delete Reservation Request

In this usage case, the sender is requesting that a previously set reservation against a RTGS DCA should be deleted, thereby releasing the reserved amount.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Code CurRsvatn/Tp/Cd	ReservationType2Code	HPAR/UPAR
BICFI AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Current reservation - RTGS DCA owner
Id AcctId/Othr/Id	Max35Text	RTGS Account Holder account ID

Table 206 - DeleteReservation (camt.049) - usage case Delete Reservation Request

Usage case example: camt.049_RTGS_DeleteReservation_DeleteReservationRequest_Example.xml

Usage case: Delete Standing Order for Reservation

In this usage case, the sender is requesting that a current active standing order for reservation, defined in CRDM, should be deleted.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
	TBC	

Table 207 - DeleteReservation (camt.049) - usage case Delete Standing Order for Reservation

Usage case example:
camt.049_RTGS_DeleteReservation_DeleteStandingOrderForReservation_Example.xml

14.3.20 LiquidityCreditTransfer (camt.050)

14.3.20.1 Overview and scope of the message

This chapter illustrates the *LiquidityCreditTransfer* message.

The *LiquidityCreditTransfer* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component.

The *LiquidityCreditTransfer* message may also be sent by the RTGS component itself to the CLM component (on behalf of a RTGS Account Holder), in order to manipulate liquidity at need and to maintain the floor and ceiling balances of RTGS DCAs.

The *LiquidityCreditTransfer* message is used to request a transfer of funds

- | between two RTGS DCAs belonging to the RTGS Account Holder, or
- | between two RTGS DCAs within the same liquidity group of RTGS DCAs, defined within the RTGS component and identified via account IDs, or
- | from a RTGS DCA to a CLM MCA.

Within RTGS, the *LiquidityCreditTransfer* message has the following usages:

- | Payment Message
- | Inter-Service Liquidity Transfer Order (Floor Processing)
- | Inter-Service Liquidity Transfer Order (Ceiling Processing)
- | Automated Inter-Service Liquidity Transfer Order
- | Liquidity Adjustment (Ancillary System Settlement Procedure C)
- | Liquidity Adjustment (Ancillary System Settlement Procedure D)

In response to the *LiquidityCreditTransfer* message, a [Receipt \(camt.025\)](#) [▶ 474] message containing the status of the liquidity transfer is returned to the sending party.

14.3.20.2 Schema

Outline of the schema

The *LiquidityCreditTransfer* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

LiquidityCreditTransfer

This building block is mandatory. It contains detailed information related to the liquidity credit transfer being instructed. It contains the following elements:

- | liquidity transfer identification
- | creditor party and account
- | amount
- | debtor party and account
- | settlement date

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.050.001.04_RTGS

Business rules applicable to the schema

For business rules applicable to *LiquidityCreditTransfer* please refer to chapter [Index of business rules and error codes](#) [▶ 670].

14.3.20.3 The message in business context

Usage case: Payment Message

In this usage case, the message provides the details required for the RTGS component to execute a liquidity transfer between two RTGS DCAs belonging to the same RTGS Account Holder (or within a liquidity management group).

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
InstructingIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl d/InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification	RestrictedFINXMax16Text	Unique ID set by the initiating party

Message item	Data type/code	Utilisation
Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl d/EndToEndId		
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/CdtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	RTGS DCA/ sub-account to be credited
AmountWithCurrency Docu- ment/LqdyCdtTrf/LqdyCdtTrf/TrfdAmt/ AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/DbtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	RTGS DCA to be debited
SettlementDate Docu- ment/LqdyCdtTrf/LqdyCdtTrf/SettlmDt	ISODate	Current RTGS business date

Table 208 - LiquidityCreditTransfer (camt.050) – usage case Payment Order Message

Usage case example: camt.050_RTGS_LiquidityCreditTransfer_PaymentOrderMessage_Example.xml

Usage case: Inter-Service Liquidity Transfer Order (Floor Processing)

In this usage case, a RTGS DCA balance has fallen below its pre-defined floor amount.

This message is sent by the RTGS component to the CLM component to execute a movement of funds from a CLM MCA into the RTGS DCA.

This will increase the balance of the RTGS DCA and bring it back to its pre-defined floor amount.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
InstructingIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl d/InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl d/EndToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/CdtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	RTGS DCA to be credited
AmountWithCurrency Docu- ment/LqdyCdtTrf/LqdyCdtTrf/TrfdAmt/ AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/DbtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	CLM MCA to be debited
SettlementDate Docu- ment/LqdyCdtTrf/LqdyCdtTrf/SettlmDt	ISODate	Current RTGS business date

Table 209 - LiquidityCreditTransfer (camt.050) – usage case Inter-Service Liquidity Transfer Order (Floor Processing)

Usage case example:
camt.050_RTGS_LiquidityCreditTransfer_InterServiceLiquidityTransferOrderFloor_Example.xml

Usage case: Inter-Service Liquidity Transfer Order (Ceiling Processing)

In this usage case, a RTGS DCA balance has risen above its pre-defined ceiling amount.

This message is sent by the RTGS component to the CLM component to execute a movement of funds into a CLM MCA from the RTGS DCA.

This will decrease the balance of the RTGS DCA and bring it back to its pre-defined ceiling amount.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
InstructingIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrf/ InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrf/ EndToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/CdtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	CLM MCA to be credited
AmountWithCurrency Docu- ment/LqdyCdtTrf/LqdyCdtTrf/TrfdAmt/ AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/DbtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	RTGS DCA to be debited
SettlementDate Docu- ment/LqdyCdtTrf/LqdyCdtTrf/SettlmDt	ISODate	Current RTGS business date

Table 210 - LiquidityCreditTransfer (camt.050) – usage case Inter-Service Liquidity Transfer Order (Ceiling Processing)

Usage case example:
camt.050_RTGS_LiquidityCreditTransfer_InterServiceLiquidityTransferOrderCeiling_Example.xml

Usage case: Automated Inter-Service Liquidity Transfer Order

In this usage case, the message is created by the RTGS component on behalf of the debiting party of a payment, when the debiting party has insufficient balance in its RTGS DCA to fully settle the payment. The message is then sent to the CLM component.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
InstructingIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrf/ InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrf/ EndToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/CdtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	RTGS DCA to be credited
AmountWithCurrency Docu- ment/LqdyCdtTrf/LqdyCdtTrf/TrfdAmt/ AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/DbtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	CLM MCA to be debited
SettlementDate Docu- ment/LqdyCdtTrf/LqdyCdtTrf/SettlmDt	ISODate	Current RTGS business date

Table 211 - LiquidityCreditTransfer (camt.050) – usage case Automated Inter-Service Liquidity Transfer Order

Usage **case** **example:**
camt.050_RTGS_LiquidityCreditTransfer_AutomatedInterServiceLiquidityTransferOrder_Example.xml
I

Usage case: Liquidity Adjustment (Ancillary System Settlement Procedure C)

In this usage case, the message is sent by an ancillary system settlement bank (or the CB on its behalf) to execute a liquidity adjustment between sub-accounts. In this particular case, the adjustment must be executed during the start of ancillary system settlement procedure C.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
InstructingIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl d/InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl d/EndToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/CdtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	RTGS DCA to be credited
AmountWithCurrency Docu- ment/LqdyCdtTrf/LqdyCdtTrf/TrfdAmt/ AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/DbtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	CLM MCA to be debited
SettlementDate Docu- ment/LqdyCdtTrf/LqdyCdtTrf/SettlmDt	ISODate	Current RTGS business date

Table 212 - LiquidityCreditTransfer (camt.050) – usage case Liquidity Adjustment (Ancillary System Settlement Procedure C)

Usage case **example:**
camt.050_RTGS_LiquidityCreditTransfer_LiquidityAdjustmentASSettlementProcedureC_Example.xml

Usage case: Liquidity Adjustment (Ancillary System Settlement Procedure D)

In this usage case, the message is sent by an ancillary system settlement bank (or the CB on its behalf) to execute a liquidity adjustment between sub-accounts. In this particular case, the adjustment must be executed during the start of ancillary system settlement procedure D.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
InstructingIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl d/InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl d/EndToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/CdtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	RTGS DCA to be credited

Message item	Data type/code	Utilisation
AmountWithCurrency Docu- ment/LqdtYCdTrf/LqdtYCdTrf/TrfdAmt/ AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred
Identification Docu- ment/LqdtYCdTrf/LqdtYCdTrf/DbtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	CLM MCA to be debited
SettlementDate Docu- ment/LqdtYCdTrf/LqdtYCdTrf/SettlmDt	ISODate	Current RTGS business date

Table 213 - LiquidityCreditTransfer (camt.050) – usage case Liquidity Adjustment (Ancillary System Settlement Procedure D)

Usage case example:
camt.050_RTGS_LiquidityCreditTransfer_LiquidityAdjustmentASSettlementProcedureD_Example.xml

14.3.21 BankToCustomerStatement (camt.053)

14.3.21.1 Overview and scope of the message

This chapter illustrates the *BankToCustomerStatement* message.

The *BankToCustomerStatement* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to inform of the entries booked to a RTGS DCA and to provide account balance information at a given point in time as an account statement.

The *BankToCustomerStatement* message is also sent by the RTGS component to a CB. It is used to inform of the entries booked in general ledger accounts.

The *BankToCustomerStatement* message provides information for cash management and/or reconciliation of information on booked/settled entries only. Optionally it can include details of underlying payments and liquidity transfers that have been included in the entry.

Within RTGS, the *BankToCustomerStatement* message has the following usages:

- I Query Response Message for Business Data
- I Statement of Accounts

I CB General Ledger

The *BankToCustomerStatement* message is produced depending upon a party's reporting configurations.

14.3.21.2 Schema

Outline of the schema.

The *BankToCustomerStatement* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Statement

This building block is mandatory and repetitive. It shows information on booked entries and balances for a RTGS DCA. It may contain:

- I statement identification
- I report sequence information
- I creation timestamp
- I account identification
- I account balance/s
- I summary of transactions
- I details of each entry: entry reference, amount and currency, debit/credit indicator, status, booking date, value date, bank transaction code

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.053.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *BankToCustomerStatement* message.

14.3.21.3 The message in business context

Usage case: Query Response Message for Business Data

In this usage case, the RTGS Account Holder has specifically requested to be informed of movements for RTGS DCAs in its data scope.

Specific message content

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmt/GrpHdr/MsgId	Max35Text	Point to point reference, as assigned by the account servicing institution, and sent to the account owner or the party authorised to receive the message, to unambiguously identify the message
Docu- ment/BkToCstmrStmt/GrpHdr/CreDtTm	dateTime	Date and time at which the message was created
Docu- ment/BkToCstmrStmt/GrpHdr/MsgPgnt n /PgNb	Max5NumericText	Page number
Docu- ment/BkToCstmrStmt/GrpHdr/MsgPgnt n/LastPgInd	YesNoIndicator	Indicates the last page
Statement		
Statement ID Document/BkToCstmrStmt/Stmt/ID	Max35Text	Statement number
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/Fr Seq	Max35Text	Reporting sequence –from
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/To Seq	Max35Text	Reporting sequence –to
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/Fr	Max35Text	Reporting sequence –range from

Message item	Data type/code	Utilisation
ToSeq/FrSeq		
Reporting sequence Docu- ment/BkToCstmrStmnt/Stmnt/RptgSeq/Fr ToSeq/ToSeq	Max35Text	Reporting sequence –range to
Reporting sequence Docu- ment/BkToCstmrStmnt/Stmnt/RptgSeq/E qSeq	Max35Text	Reporting sequence –single sequence
Reporting sequence Docu- ment/BkToCstmrStmnt/Stmnt/RptgSeq/N EQSeq	Max35Text	Reporting sequence –excluding a se- quence
Creation date/time Docu- ment/BkToCstmrStmnt/Stmnt/CreDtTm	ISODatetime	Timestamp when the statement was created
Account Docu- ment/BkToCstmrStmnt/Stmnt/Acct/ID/Oth r/ID	Max34Text	RTGS DCA number.
Currency Docu- ment/BkToCstmrStmnt/Stmnt/Acct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS DCA
Account owner Docu- ment/BkToCstmrStmnt/Stmnt/Acct/Ownr/I D/OrgId/AnyBIC	AnyBICIdentifier	Owner of the RTGS DCA
Account owner country Docu- ment/BkToCstmrStmnt/Stmnt/Acct/Ownr/I D/CtryOfRes	CountryCode	Country of residence, of the owner of the RTGS DCA
Multiple repetitions of balance information		
Balance type	CSLD_BalanceTypeCode	Type of balance

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmnt/Stmnt/Bal/Trp/CdO rPrty/Cd		
Balance amount Docu- ment/BkToCstmrStmnt/Stmnt/Bal/Amt/	ActiveOrHistoricCurrencyAndAmount	Amount of balance
Balance credit/debit Docu- ment/BkToCstmrStmnt/Stmnt/Bal/Amt	CreditDebitCode	Credit or debit indicator for the balance amount
Balance date Docu- ment/BkToCstmrStmnt/Stmnt/Bal/Dt/Dt	ISODate	Date of the balance
Balance availability Docu- ment/BkToCstmrStmnt/Stmnt/Bal/Avlbtly	CashAvailability	Availability of balance. Might be needed for non-EURO – TBD.
Transactions summary		
Number of all entries Docu- ment/BkToCstmrStmnt/Stmnt/TxsSummry /TtlNtries/NbOfNtries	Max15NumericText	Total number of entries on statement
Sum of all entries Docu- ment/BkToCstmrStmnt/Stmnt/TxsSummry /TtlNtries/Sum	DecimalNumber	Total sum of all of entries on statement
Net sum of all entries Docu- ment/BkToCstmrStmnt/Stmnt/TxsSummry /TtlNtries/TtlNetNtry/Amt	NonNegativeDecimalNumber	Net total sum of all of entries on statement
Credit debit ind Docu- ment/BkToCstmrStmnt/Stmnt/TxsSummry /TtlNtries/TtlNetNtry/CdtDbtInd	CreditDebitCode	Credit debit indicator for the net total sum of all of entries on statement
Sum of all credit entries	DecimalNumber	Total sum of all of credit entries on

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlCdtNtries/Sum		statement
Sum of all debit entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlDbtNtries/Sum	DecimalNumber	Total sum of all of debit entries on statement
Multiple repetitions of entry information		
Entry ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryRe f	Max35Text	Unique reference for the entry
Amount Docu- ment/BkToCstmrStmt/Stmt/Ntry/Amt	CSLD_Max14_Max2DecimalAmount	Entry amount
Credit debit ind Docu- ment/BkToCstmrStmt/Stmt/Ntry/CrdDbt Ind	CreditDebitCode	Credit debit indicator for entry amount
Status Docu- ment/BkToCstmrStmt/Stmt/Ntry/Sts/Cd	CSLD_EntryStatusCode	Entry status
Booking datetime Docu- ment/BkToCstmrStmt/Stmt/Ntry/Bookg Dt/DtTm	ISODateTime	Date and time the entry was booked
Value date Docu- ment/BkToCstmrStmt/Stmt/Ntry/ValDt/ Dt	ISODate	Value date
Value datetime Docu- ment/BkToCstmrStmt/Stmt/Ntry/ValDt/ DtTm	ISODateTime	Value date and time

Message item	Data type/code	Utilisation
Bank transaction code Docu- ment/BkToCstmrStmt/Stmt/Ntry/BkTxCd/Prtry/Cd	CSLD_BankTransactionCode	Transaction code
Message ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/Refs/MsgId	Max35Text	Unique message reference of the instructing message
Instruction ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/Refs/InstrId	Max35Text	Unique reference set by the instructing party
End to end ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/Refs/EndToEndId	Max35Text	Additional unique reference set by the initiating party
Transaction ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/Refs/TxID	Max35Text	Transaction ID set by the instructing agent
Transaction ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/Amt	CSLD_Max14_Max2DecimalAmount	Entry detail amount

Message item	Data type/code	Utilisation
Transaction ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/CrdDbtInd	CreditDebitCode	Credit debit indicator of entry detail amount
Local instrument code Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/LclInstrm/Cd	ExternalLocalInstrumentCode	Local instrument code
Local instrument proprietary code Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/LclInstrm/Prtry	Max35Text	Local instrument – proprietary code

Table 214 - BankToCustomerStatement (camt.053) – usage case Query Response Message for Business Data

Usage case example:
camt.053_RTGS_BankToCustomerStatement_QueryresponseMessageForBusinessData_Example.xml

Usage case: Statement of Accounts

In this usage case, the recipient is being informed of all movements, including opening and closing balances, for RTGS DCAs in its data scope. This report message is automatically generated by the RTGS component in accordance with the reporting configuration settings applied by the RTGS Account Holder.

Specific message content

Message item	Data type/code	Utilisation
Statement ID Document/BkToCstmrStmt/Stmt/ID	Max35Text	Statement number
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/Fr Seq	Max35Text	Reporting sequence –from
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/To Seq	Max35Text	Reporting sequence –to

Message item	Data type/code	Utilisation
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/Fr ToSeq/FrSeq	Max35Text	Reporting sequence –range from
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/Fr ToSeq/ToSeq	Max35Text	Reporting sequence –range to
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/E qSeq	Max35Text	Reporting sequence –single sequence
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/N EQSeq	Max35Text	Reporting sequence –excluding a se- quence
Creation date/time Docu- ment/BkToCstmrStmt/Stmt/CreDtTm	ISODatetime	Timestamp when the statement was created
Account Docu- ment/BkToCstmrStmt/Stmt/Acct/ID/Oth r/ID	Max34Text	RTGS DCA
Currency Docu- ment/BkToCstmrStmt/Stmt/Acct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS DCA
Account owner Docu- ment/BkToCstmrStmt/Stmt/Acct/Ownr/I D/OrgId/AnyBIC	AnyBICIdentifier	Owner of the RTGS DCA
Account owner country Docu- ment/BkToCstmrStmt/Stmt/Acct/Ownr/I D/CtryOfRes	CountryCode	Country of residence, of the owner of the RTGS DCA

Message item	Data type/code	Utilisation
Multiple repetitions of balance information		
Balance type Docu- ment/BkToCstmrStmt/Stmt/Bal/Tp/CdO rPrty/Cd	CSLD_BalanceTypeCode	Type of balance
Balance amount Docu- ment/BkToCstmrStmt/Stmt/Bal/Amt/	ActiveOrHistoricCurrencyAndAmount	Amount of balance
Balance credit/debit Docu- ment/BkToCstmrStmt/Stmt/Bal/Amt	CreditDebitCode	Credit or debit indicator for the balance amount
Balance date Docu- ment/BkToCstmrStmt/Stmt/Bal/Dt/Dt	ISODate	Date of the balance
Balance availability Docu- ment/BkToCstmrStmt/Stmt/Bal/Avlbtly	CashAvailability	Availability of balance. Might be needed for non-EURO – TBD.
Transactions summary		
Number of all entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/NbOfNtries	Max15NumericText	Total number of entries on statement
Sum of all entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/Sum	DecimalNumber	Total sum of all of entries on statement
Net sum of all entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/TtlNetNtry/Amt	NonNegativeDecimalNumber	Net total sum of all of entries on statement
Credit debit ind Docu- ment/BkToCstmrStmt/Stmt/TxsSummry	CreditDebitCode	Credit debit indicator for the net total sum of all of entries on statement

Message item	Data type/code	Utilisation
/TtlNtries/TtlNetNtry/CdtDbtInd		
Sum of all credit entries Docu- ment/BkToCstmrStmnt/Stmnt/TxsSummry /TtlCdtNtries/Sum	DecimalNumber	Total sum of all of credit entries on statement
Sum of all debit entries Docu- ment/BkToCstmrStmnt/Stmnt/TxsSummry /TtlDbtNtries/Sum	DecimalNumber	Total sum of all of debit entries on statement
Multiple repetitions of entry information		
Entry ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryRe f	Max35Text	Unique reference for the entry
Amount Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/Amt	CSLD_Max14_Max2DecimalAmount	Entry amount
Credit debit ind Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/CrdDbt Ind	CreditDebitCode	Credit debit indicator for entry amount
Status Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/Sts/Cd	CSLD_EntryStatusCode	Entry status
Booking datetime Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/Bookg Dt/DtTm	ISODateTime	Date and time the entry was booked
Value date Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/ValDt/ Dt	ISODate	Value date
Value datetime	ISODateTime	Value date and time

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/ValDt/ DtTm		
Bank transaction code Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/BkTxC d/Prtry/Cd	CSLD_BankTransactionCode	Transaction code
Message ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/MsgId	Max35Text	Unique message reference of the in- structing message
Instruction ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/InstrId	Max35Text	Unique reference set by the instructing party
End to end ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/EndToEndId	Max35Text	Additional unique reference set by the initiating party
Transaction ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/TxID	Max35Text	Transaction ID set by the instructing agent
Transaction ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Amt	CSLD_Max14_Max2DecimalAmount	Entry detail amount

Message item	Data type/code	Utilisation
Transaction ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/CrdDbtInd	CreditDebitCode	Credit debit indicator of entry detail amount
Local instrument code Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/LclInstrm/Cd	ExternalLocalInstrumentCode	Local instrument code
Local instrument proprietary code Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/LclInstrm/Prtry	Max35Text	Local instrument – proprietary code

Table 215 - BankToCustomerStatement (camt.053) – usage case Statement of Accounts

Usage case example:
camt.053_RTGS_BankToCustomerStatement_StatementOfAccounts_Example.xml

Usage case: CB General Ledger

In this usage case, the CB is being informed of all movements occurring on general ledger accounts in its data scope. This report message is automatically generated by the RTGS component in accordance with the reporting configuration settings applied by the CB.

Specific message content

Message item	Data type/code	Utilisation
Statement ID Document/BkToCstmrStmt/Stmt/ID	Max35Text	Statement number
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/Fr Seq	Max35Text	Reporting sequence – from
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/To Seq	Max35Text	Reporting sequence – to

Message item	Data type/code	Utilisation
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/Fr ToSeq/FrSeq	Max35Text	Reporting sequence – range from
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/Fr ToSeq/ToSeq	Max35Text	Reporting sequence – range to
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/E qSeq	Max35Text	Reporting sequence – single sequence
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/N EQSeq	Max35Text	Reporting sequence – excluding a sequence
Creation date/time Docu- ment/BkToCstmrStmt/Stmt/CreDtTm	ISODateTime	Timestamp when the statement was created
Account Docu- ment/BkToCstmrStmt/Stmt/Acct/ID/Oth r/ID	Max34Text	RTGS DCA
Currency Docu- ment/BkToCstmrStmt/Stmt/Acct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS DCA
Account owner Docu- ment/BkToCstmrStmt/Stmt/Acct/Ownr/I D/OrgId/AnyBIC	AnyBICIdentifier	Owner of the RTGS DCA
Account owner country Docu- ment/BkToCstmrStmt/Stmt/Acct/Ownr/I D/CtryOfRes	CountryCode	Country of residence, of the owner of the RTGS DCA

Message item	Data type/code	Utilisation
Multiple repetitions of balance information		
Balance type Docu- ment/BkToCstmrStmt/Stmt/Bal/Tp/CdO rPrty/Cd	CSLD_BalanceTypeCode	Type of balance
Balance amount Docu- ment/BkToCstmrStmt/Stmt/Bal/Amt/	ActiveOrHistoricCurrencyAndAmount	Amount of balance
Balance credit/debit Docu- ment/BkToCstmrStmt/Stmt/Bal/Amt	CreditDebitCode	Credit or debit indicator for the balance amount
Balance date Docu- ment/BkToCstmrStmt/Stmt/Bal/Dt/Dt	ISODate	Date of the balance
Balance availability Docu- ment/BkToCstmrStmt/Stmt/Bal/Avlbtty	CashAvailability	Availability of balance. Might be needed for non-EURO – TBD.
Transactions summary		
Number of all entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/NbOfNtries	Max15NumericText	Total number of entries on statement
Sum of all entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/Sum	DecimalNumber	Total sum of all of entries on statement
Net sum of all entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/TtlNetNtry/Amt	NonNegativeDecimalNumber	Net total sum of all of entries on statement
Credit debit ind Docu- ment/BkToCstmrStmt/Stmt/TxsSummry	CreditDebitCode	Credit debit indicator for the net total sum of all of entries on statement

Message item	Data type/code	Utilisation
/TtlNtries/TtlNetNtry/CdtDbtInd		
Sum of all credit entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlCdtNtries/Sum	DecimalNumber	Total sum of all of credit entries on statement
Sum of all debit entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlDbtNtries/Sum	DecimalNumber	Total sum of all of debit entries on statement
Multiple repetitions of entry information		
Entry ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryRe f	Max35Text	Unique reference for the entry
Amount Docu- ment/BkToCstmrStmt/Stmt/Ntry/Amt	CSLD_Max14_Max2DecimalAmount	Entry amount
Credit debit ind Docu- ment/BkToCstmrStmt/Stmt/Ntry/CrdDbt Ind	CreditDebitCode	Credit debit indicator for entry amount
Status Docu- ment/BkToCstmrStmt/Stmt/Ntry/Sts/Cd	CSLD_EntryStatusCode	Entry status
Booking datetime Docu- ment/BkToCstmrStmt/Stmt/Ntry/Bookg Dt/DtTm	ISODateTime	Date and time the entry was booked
Value date Docu- ment/BkToCstmrStmt/Stmt/Ntry/ValDt/ Dt	ISODate	Value date
Value datetime	ISODateTime	Value date and time

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/ValDt/ DtTm		
Bank transaction code Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/BkTxC d/Prtry/Cd	CSLD_BankTransactionCode	Transaction code
Message ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/MsgId	Max35Text	Unique message reference of the in- structing message
Instruction ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/InstrId	Max35Text	Unique reference set by the instructing party
End to end ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/EndToEndId	Max35Text	Additional unique reference set by the initiating party
Transaction ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/TxID	Max35Text	Transaction ID set by the instructing agent
Transaction ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Amt	CSLD_Max14_Max2DecimalAmount	Entry detail amount

Message item	Data type/code	Utilisation
Transaction ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/CrdDbtInd	CreditDebitCode	Credit debit indicator of entry detail amount
Local instrument code Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/LclInstrm/Cd	ExternalLocalInstrumentCode	Local instrument code
Local instrument proprietary code Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/LclInstrm/Prtry	Max35Text	Local instrument – proprietary code

Table 216 - BankToCustomerStatement (camt.053) – usage case CB General Ledger

Usage **case** **example:**
camt.053_RTGS_BankToCustomerStatement_CentralBankGeneralLedger_Example.xml

14.3.22 BankToCustomerDebitCreditNotification (camt.054)

14.3.22.1 Overview and scope of the message

This chapter illustrates the *BankToCustomerDebitCreditNotification* message.

The *BankToCustomerDebitCreditNotification* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to confirm the credit or the debit of a certain amount on one of their RTGS DCAs.

The *BankToCustomerDebitCreditNotification* message is sent by the RTGS component when the RTGS DCA owner was not the instructor of the movement.

The *BankToCustomerDebitCreditNotification* message is only concerned with one single debit or credit movement on one single RTGS DCA.

Within RTGS, the *BankToCustomerDebitCreditNotification* message has the following usages:

- | Payment Settlement Notification (Intra-Service Liquidity Transfer)
- | Payment Settlement Notification (Inter-Service Liquidity Transfer)
- | Payment Settlement Notification (Payments)

- | Payment Settlement Notification (RTGS Standing Order SoD)
- | Liquidity Transfer Settlement Notification
- | Ancillary System Processing
 - Procedure A
 - Send Debit Notification (due to recurrent optimisation)
 - Send Debit Notification (any debit against the ancillary systems RTGS DCA)
 - Send Credit Notification (any credit into the ancillary systems RTGS DCA)
 - Procedure B
 - Send Debit/Credit Notification (due to recurrent optimisation)
 - Send Debit/Credit Notification (due to first settlement attempt)
 - Procedure A & B
 - Send Credit Notification (due to reversal, resulting from revoke of batch)
 - Send Credit Notification (due to reversal, resulting from rejection)
 - Send Debit Notification (due to guarantee processing)
 - Send Credit Notification (due to guarantee processing)
 - Procedure C - start
 - Notify Ancillary System Settlement Bank (Debit notification)
 - Notify Ancillary System Settlement Bank (Credit notification)
 - Procedure C – settlement
 - Notify Ancillary System Settlement Bank (Debit/Credit on sub-account movements)
 - Procedure C - end
 - Notify Ancillary System Settlement Bank (Debit/Credit from executed stored liquidity transfer orders)
 - Notify Ancillary System Settlement Bank (Debit/Credit from re-transfer of liquidity)
 - Procedure D
 - Notify Ancillary System Settlement Bank (Debit/Credit from executed stored liquidity transfer orders)
 - Notify Ancillary System Settlement Bank (Debit/Credit from re-transfer of liquidity)

The *BankToCustomerDebitCreditNotification* message is sent in response to a debit/credit movement activity within the RTGS component.

14.3.22.2 Schema

Outline of the schema

The *BankToCustomerDebitCreditNotification* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the message.

Notification

This building block is mandatory and non-repetitive. It notifies of a debit or credit entry for the RTGS DCA. It may contain:

- | identification
- | creation timestamp
- | account identification
- | amount
- | debit/credit indicator
- | status
- | booking date
- | value date
- | bank transaction code
- | amount details
- | related parties and agents
- | local instrument

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.054.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *BankToCustomerDebitCreditNotification* message.

14.3.22.3 The message in business context

Usage case: Payment Settlement Notification (Intra-Service Liquidity Transfer)

In this usage case, the RTGS component sends a confirmation of credit movement to a RTGS Account Holder if one of its RTGS DCAs was credited as the result of an intra-service liquidity transfer order. It can also be used for a credit notification on the RTGS dedicated transit account.

Specific message content

In the rules and further descriptions, the confirmation contains the exact amount and the reason for the credit. In the case of a credit on the RTGS dedicated transit account, the notification is sent to the CB.

Message item	Data type/code	Utilisation
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/I D	Max35Text	Unique ID for this notification, set by RTGS
CreationDateTime Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/C reDtTm	ISODateTime	Date time when the notification was created
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/ID/Othr/ID	Max34Text	RTGS DCA ID
Currency Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS DCA
AnyBIC Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/Ownr/ID/OrgId/AnyBIC	CSLD_BIC11Text	Party owning the RTGS DCA
Amount Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/Amt	CSLD_Max14_Max2DecimalAmount	Originally instructed amount of the transaction (no partially settled amount)

Message item	Data type/code	Utilisation
CreditDebitIndicator Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/CrdDbtInd	CreditDebitCode	CRDT
Code Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/Sts/Cd	ExternalEntryStatus1Code	BOOK
DateTime Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BookgDt/DtTm	ISODateTime	Time when the credit entry was booked (business day and settlement time)
DateTime Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/ValDt/DtTm	ISODateTime	Time when the credit entry amount became available (business day and settlement time)
Code Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BnkTxCd/Prtry/Cd	CSLD_BankTransactionCode	PMNT
MessageIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/MsgId	CSLD_RestrictedFINXMax35Text	Message ID of the underlying instruc- tion which caused the credit entry
InstructionIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/InstrId	Max35Text	Identification of the underlying instruc- tion which caused the credit entry
EndToEndIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/EndToEndId	Max35Text	End-to-end ID of the underlying instruc- tion which caused the credit entry
TransactionIdentification Docu-	Max35Text	Transaction ID of the underlying in- struction which caused the credit entry

Message item	Data type/code	Utilisation
ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtIs/TxDtIs/Refs/TxId		
Amount Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtIs/TxDtIs/Amt	CSLD_Max14_Max5DecimalAmount	Transaction amount
CreditDebitIndicator Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtIs/TxDtIs/CrdDbtInd	CreditDebitCode	CRDT
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtIs/TxDtIs/RltdPties/DbtrAcct/ID/Othr/ID	Max34Text	Debtor account in the underlying transaction
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtIs/TxDtIs/RltdPties/CdtrAcct/ID/Othr/ID	Max34Text	Creditor account in the underlying transaction

Table 217 - BankToCustomerDebitCreditNotification (camt.054) – usage case Payment Settlement Notification (Intra-Service Liquidity Transfer)

Usage case example:
camt.054_RTGS_BankToCustomerDebitCreditNotification_PaymentOrderSettlementNotification_Example.xml

In this example a confirmation of credit movement on the “CASHACCT01” resulting from a payment is sent to the corresponding party.

Usage case: Payment Settlement Notification (Inter-Service Liquidity Transfer)

In this usage case, the RTGS component sends a confirmation of credit movement to a RTGS Account Holder if one of its RTGS DCAs was credited as the result of the settlement of an inter-service liquidity transfer order. It can also be used for a credit notification on the RTGS dedicated transit account.

Specific message content

For content and examples, please see usage case:

Payment Settlement Notification (Intra-Service Liquidity Transfer)

Usage case: Payment Settlement Notification (Payment)

In this usage case, the RTGS component sends a confirmation of credit movement to a RTGS Account Holder if one of its RTGS DCAs was credited as the result of the settlement of a payment order. It can also be used for a credit notification on the RTGS dedicated transit account.

Specific message content

For content and examples, please see usage case:

Payment Settlement Notification (Intra-Service Liquidity Transfer)

Usage case: Payment Order Settlement Notification (RTGS Standing Order SoD)

In this usage case, the RTGS component sends a confirmation of debit or credit movement to a RTGS Account Holder or a CB if one of its RTGS DCAs was credited or debited as the result of the settlement of a RTGS standing order at SoD.

Specific message content

For content and examples, please see usage case:

Payment Settlement Notification (Intra-Service Liquidity Transfer)

Usage case: Liquidity Transfer Settlement Notification

In this usage case, the RTGS component sends a confirmation of credit movement to a RTGS Account Holder if one of its RTGS DCAs was credited as the result of the settlement of a liquidity transfer. It can also be used for a credit notification on the RTGS dedicated transit account.

Specific message content

In the rules and further descriptions, the confirmation contains the exact amount and the reason for the credit. In the case of a credit on the RTGS dedicated transit account, the notification is sent to the CB.

Message item	Data type/code	Utilisation
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/I D	Max35Text	Unique ID for this notification, set by RTGS
CreationDateTime Docu-	ISODateTime	Date time when the notification was created

Message item	Data type/code	Utilisation
ment/BkToCstmrDbtCdtNtfctn/NtFctn/C reDtTm		
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/ID/Othr/ID	Max34Text	RTGS DCA ID
Currency Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS DCA
AnyBIC Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/Ownr/ID/OrgId/AnyBIC	CSLD_BIC11Text	Party owning the RTGS DCA
Amount Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/Amt	CSLD_Max14_Max2DecimalAmount	Originally instructed amount of the transaction (no partially settled amount)
CreditDebitIndicator Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/CrdDbtInd	CreditDebitCode	CRDT
Code Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/Sts/Cd	ExternalEntryStatus1Code	BOOK
DateTime Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BookgDt/DtTm	ISODateTime	Time when the credit entry was booked (business day and settlement time)
DateTime Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/ValDt/DtTm	ISODateTime	Time when the credit entry amount became available (business day and settlement time)

Message item	Data type/code	Utilisation
Code Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BnkTxCd/prtry/Cd	CSLD_BankTransactionCode	CAMT
MessageIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/MsgId	CSLD_RestrictedFINXMax35Text	Message ID of the underlying instruc- tion which caused the credit entry
InstructionIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/InstrId	Max35Text	Identification of the underlying instruc- tion which caused the credit entry
EndToEndIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/EndToEndId	Max35Text	End-to-end ID of the underlying instruc- tion which caused the credit entry
TransactionIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/TxId	Max35Text	Transaction ID of the underlying in- struction which caused the credit entry
Amount Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Amt	CSLD_Max14_Max5DecimalAmount	Transaction amount

Message item	Data type/code	Utilisation
CreditDebitIndicator Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/CrdDbtInd	CreditDebitCode	CRDT
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/RltdPties/DbtrAcct/I D/Othr/ID	Max34Text	Debtor account in the underlying trans- action
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/RltdPties/CdtrAcct/I D/Othr/ID	Max34Text	Creditor account in the underlying transaction

Table 218 - BankToCustomerDebitCreditNotification (camt.054) – usage case Liquidity Transfer Settlement Notification

Usage case example:
camt.054_RTGS_BankToCustomerDebitCreditNotification_LiquiditytransferSettlementNotification_Example.xml

In this example a confirmation of credit movement on the “CASHACCT01” resulting from a liquidity transfer is sent to the corresponding party.

Usage case: Ancillary System Processing

In this usage case, the RTGS component sends a confirmation of credit or a debit movement to an ancillary system if one of its RTGS DCAs was credited or debited result of the specific ancillary system processing procedures within the RTGS component.

Specific message content

Message item	Data type/code	Utilisation
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/I D	Max35Text	Unique ID for this notification, set by RTGS
CreationDateTime	ISODatetime	Date time when the notification was

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/C reDtTm		created
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/ID/Othr/ID	Max34Text	RTGS DCA ID
Currency Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS DCA
AnyBIC Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/Ownr/ID/OrgId/AnyBIC	CSLD_BIC11Text	Party owning the RTGS DCA
Amount Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/Amt	CSLD_Max14_Max2DecimalAmount	Originally instructed amount of the transaction (no partially settled amount)
CreditDebitIndicator Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/CrdDbtInd	CreditDebitCode	CRDT or DBIT
Code Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/Sts/Cd	ExternalEntryStatus1Code	BOOK
DateTime Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BookgDt/DtTm	ISODateTime	Time when the credit entry was booked (business day and settlement time)
DateTime Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N	ISODateTime	Time when the credit entry amount became available (business day and settlement time)

Message item	Data type/code	Utilisation
try/ValDt/DtTm		
Code Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BnkTxCd/prtry/Cd	CSLD_BankTransactionCode	CAMT, PMNT, other
MessageIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/MsgId	CSLD_RestrictedFINXMax35Text	Message ID of the underlying instruc- tion which caused the credit entry
InstructionIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/InstrId	Max35Text	Identification of the underlying instruc- tion which caused the credit entry
EndToEndIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/EndToEndId	Max35Text	
TransactionIdentification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Refs/TxId	Max35Text	Transaction ID of the underlying in- struction which caused the credit entry
Amount Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/Amt	CSLD_Max14_Max5DecimalAmount	Transaction amount

Message item	Data type/code	Utilisation
CreditDebitIndicator Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/CrdDbtInd	CreditDebitCode	CRDT or DBIT
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/RltdPties/DbtrAcct/I D/Othr/ID	Max34Text	Debtor account in the underlying trans- action
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/NtryDtIs/TxDtIs/RltdPties/CdtrAcct/I D/Othr/ID	Max34Text	Creditor account in the underlying transaction

Table 219 - BankToCustomerDebitCreditNotification (camt.054) – usage case Ancillary System Processing

Usage case example 1:
camt.054_RTGS_BankToCustomerDebitCreditNotification_AncillarySystemProcessing_Example1.xml

Revoke procedure A/B AS batch. Credit notification is sent to ancillary system settlement banks (holders of RTGS DCA) once a previously debited ancillary system payment instruction is reversed. Ancillary system BIC is used in CreditorAgent element.

Usage case example 2:
camt.054_RTGS_BankToCustomerDebitCreditNotification_AncillarySystemProcessing_Example2.xml

Settlement of ancillary system payment instructions for ancillary system procedure A. Debit notification is sent to ancillary system settlement banks for the debit on their RTGS DCA. Ancillary system BIC is used in DebtorAgent element.

Usage case example 3:
camt.054_RTGS_BankToCustomerDebitCreditNotification_AncillarySystemProcessing_Example3.xml

Settlement of ancillary system payment instructions for ancillary system procedure A. Credit notification is sent to ancillary system settlement banks for the credit on their RTGS DCA. Ancillary system BIC is used in CreditorAgent element.

Usage	case	example	10:
camt.054_RTGS_BankToCustomerDebitCreditNotification_AncillarySystemProcessing_Example10.xml			

End procedure C cycle. Ancillary system Settlement banks are informed about the execution of stored immediate liquidity transfer via camt.054. Sample illustrates the RTGS DCA debited (Example10_1).

An equivalent message is sent for the credited sub-account with the same reference to the underlying transaction (Transaction Id). In RelatedParties, the debtor account contains the ID of the RTGS DCA of the settlement bank (Example10_2).

Usage	case	example	11:
camt.054_RTGS_BankToCustomerDebitCreditNotification_AncillarySystemProcessing_Example11.xml			

End of procedure ancillary system settlement procedure C. The remaining liquidity on sub-accounts is transferred back to the ancillary system settlement banks' RTGS DCA. Ancillary system settlement banks are informed about the re-transfer of liquidity via camt.054. Sample illustrates the RTGS DCA credited (Example11_1).

An equivalent message is sent for the debited sub-accounts with the same reference to the underlying transaction (Transaction Id). In RelatedParties, the creditor account contains the ID of the RTGS DCA of the settlement bank (Example11_2).

Usage	case	example	12:
camt.054_RTGS_BankToCustomerDebitCreditNotification_AncillarySystemProcessing_Example12.xml			

End procedure D cycle. Ancillary system settlement banks are informed about the execution of stored immediate liquidity transfer via camt.054. The reverse liquidity transfer issued by the ancillary system aims at debiting dedicated liquidity account and crediting the RTGS DCA. In RelatedParties, the creditor account contains the ID of the RTGS DCA of the settlement bank.

14.3.23 FIToFiPaymentCancellationRequest (camt.056)

14.3.23.1 Overview and scope of the message

This chapter illustrates the *FIToFiPaymentCancellationRequest* message.

The *FIToFiPaymentCancellationRequest* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to request the cancellation of an original payment.

The *FIToFiPaymentCancellationRequest* message concerns only one original payment.

Within RTGS, the *FIToFIPaymentCancellationRequest* message has the following usages:

- I Cancel Payment

In response to the *FIToFIPaymentCancellationRequest* message, a [ResolutionOfInvestigation \(camt.029\)](#) [▶ 484] is sent, indicating the success or rejection/failure of the cancellation.

14.3.23.2 Schema

Outline of the schema.

The *FIToFIPaymentCancellationRequest* message is composed of the following message building blocks:

Assignment

Identifies the assignment of an investigation case from an assigner to an assignee. The assigner must be the sender of this message and the assignee must be the receiver.

Underlying

This block is mandatory and non-repetitive. It identifies the original liquidity transfer order to be cancelled. It contains the following elements:

- I cancellation identification
- I original group information
- I original: instruction identification, end-to-end identification, transaction identification, clearing system reference
- I original interbank settlement amount
- I original interbank settlement date
- I cancellation reason information

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.056.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *FIToFIPaymentCancellationRequest* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.3.23.3 The message in business context

Usage case: Cancel Payment

In this usage case, a RTGS Account Holder (or a party authorized by them) is requesting that a previously sent payment should be cancelled.

If the previously sent payment has not yet been settled, then this message intends that it will never reach settlement. If the previously sent payment has achieved settlement, then this message intends that a reversal of such settlement is implied.

Note: cancellation and/or reversal will be subject to the appropriate rules and privileges.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Docu- ment/FIToFIPmtCxlReq/Assgnmt/Id	RTGS_RestrictedFINXMax35Text	Assignment identification
Docu- ment/FIToFIPmtCxlReq/Assgnmt/Assg nr/Agt/FinInstnId/BICFI	RTGS_BIC11Text	Assigner BIC
Docu- ment/FIToFIPmtCxlReq/Assgnmt/Assg ne/Agt/FinInstnId/BICFI	RTGS_BIC11Text	Assignee BIC
Docu- ment/FIToFIPmtCxlReq/Assgnmt/CreD tTm	ISODateTime	Creation timestamp
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlId	RTGS_RestrictedFINXMax35Text	Cancellation ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlGrpInf/OrgnlMsgId	RTGS_RestrictedFINXMax35Text	Original instruction message ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlGrpInf/OrgnlMsgNmId	RTGS_XMLMessageNamePattern	Original instruction message name id
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/	ISODateTime	Original instruction message time

Message item	Data type/code	Utilisation
OrgnlGrpInf/OrgnlCreDtTm		
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlInstrId	Max35Text	Original instruction ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlEndToEndId	Max35Text	Original end-to-end ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlTxId	Max35Text	Original transaction ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlClrSysRef	Max35Text	Original clearingsystem reference
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlIntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Original amount & currency
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlIntrBkSttlmDt	ISODate	Original interbank settlement date
Originator (requestor) of cancellation		
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Nm	Max140Text	Originator's name
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/PstlAdr/StrtNm	Max70Text	Postal address street
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/PstlAdr/BldgNb	Max16Text	Postal address building number
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/PstlAdr/PstCd	Max16Text	Postal address postcode
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/PstlAdr/TwnNm	Max35Text	Postal address town name

Message item	Data type/code	Utilisation
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/PstlAdr/CtrySubDvsn	Max35Text	Postal address country subdivision
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/PstlAdr/Ctry	CountryCode	Postal address country
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/Orgld/AnyBIC	AnyBICIdentifier	Identification by BIC
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/Orgld/Othr/Id	Max35Text	Identification by other ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/Orgld/Othr/SchmeN m/Cd	ExternalOrganisationIdentifica- tion1Code	Identification by other scheme
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/Orgld/Othr/SchmeN m/Prtry	Max35Text	Identification by other scheme-code
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/Orgld/Othr/Issr	Max35Text	Identification by other scheme issr
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/DtAndPlcOfBi rth/BirthDt	ISODate	Identification by private – birth date
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/DtAndPlcOfBi rth/PrvcOfBirth	Max35Text	Identification by private – birth place
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/DtAndPlcOfBi rth/CityOfBirth	Max35Text	Identification by private – birth city

Message item	Data type/code	Utilisation
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/DtAndPlcOfBi rth/CtryOfBirth	CountryCode	Identification by private – birth country
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/Othr/Id	Max35Text	Identification by private other – ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/Othr/SchmeN m/Cd	ExternalPersonIdentification1Code	Identification by private other - scheme
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/Othr/SchmeN m/Prtry	Max35Text	Identification by private other - scheme- code
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/Othr/Issr	Max35Text	Identification by private other - scheme issr
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/CtryOfRes	CountryCode	Identification by private other – country of residence
Cancellation reason information		
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Rsn/Cd	ExternalCancellationReason1Code	Reason for cancellation request

Table 220 - FIToFIPaymentCancellationRequest (camt.056) – usage case Cancel Payment

Usage **case** **example:**
camt.056_RTGS_FIToFIPaymentCancellationRequest_CancelPaymentOrder_Example.xml

The example illustrates a request from a participant to the RTGS system to cancel a [FinancialInstitution-CreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585] identified by several identifications from the original transaction. The reason is that the transaction is duplicated.

14.3.24 GetStandingOrder (camt.069)

14.3.24.1 Overview and scope of the message

This chapter illustrates the *GetStandingOrder* message.

The *GetStandingOrder* message is sent by an authorised actor to retrieve standing order information.

The *GetStandingOrder* message is replied by a [ReturnStandingOrder \(camt.070\)](#) [▶ 543] to return the retrieved standing order information or to provide detailed information in case of an error (e.g. no rows retrieved).

14.3.24.2 Schema

Outline of the schema

The *GetStandingOrder* message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message Identification provided by the requesting actor.

It is also used to specify which kind of query must be performed.

Only standing order details query is allowed.

StandingOrderQueryDefinition

This block is mandatory and provides with all the search criteria that must be used to filter standing order records in CRDM. Possible criteria are account and BIC.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/camt.069.001.02>

14.3.24.3 The message in business context

Usage case: Get Standing order details

This usage case describes a query used to retrieve the standing order details in CRDM.

Specific message requirements and search criterias.

Message item	Data type/code	Utilisation
Request type Docu- ment/GetStgOrdr/MsgHdr/ReqTp/Cd	StandingOrderQueryType1Code	Request type STDL
Account identification Docu- ment/GetStgOrdr/StgOrdrQryDef/StgOr drCrit/NewCrit/SchCrit/Acct/Id/Othr/Id	RestrictedFINMax34Text	Account identification
Party BIC Docu- ment/GetStgOrdr/StgOrdrQryDef/StgOr drCrit/NewCrit/SchCrit/RspnsblPty/Finl nstnId/BICFI	BICFIIdentifier	Party BIC

Table 221 - GetStandingOrder (camt.069) – usage case Get Standing order details

Usage case example: GetStandingOrderDetails_example.xml

In this example details of a standing order for the account identified with “ACC001” and Owner “PAYBXXY-YAAA” are requested.

14.3.25 ReturnStandingOrder (camt.070)

14.3.25.1 Overview and scope of the message

This chapter illustrates the *ReturnStandingOrder* message.

The *ReturnStandingOrder* message is sent by CRDM to an authorised actor to provide with requested standing order information.

The *ReturnStandingOrder* message has the following usages:

- | RTGS return standing order details
- | ASI6 RTGS return standing order details
- | CLM return standing order details

The *ReturnStandingOrder* message is sent as a response to a previously sent [GetStandingOrder \(camt.069\)](#) [► 542].

14.3.25.2 Schema

Outline of the schema

The *ReturnStandingOrder* message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message identification provided by the requesting actor as well as the original business query message identification and the request type (only standing order details query response is allowed).

ReportOrError

This block is mandatory and includes either the retrieved records or the error occurred during the query processing (e.g. no records retrieved).

Report

This block is mandatory and provides with all the pieces of information related to the retrieved standing order.

- | Standing order identification
- | account identification
- | account owner
- | amount
- | credit/debit indicator
- | validity period
- | execution type
- | creditor
- | credited account
- | debtor
- | debited account

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/camt.070.001.03>

14.3.25.3 The message in business context

Usage case: RTGS return standing order details

In this usage case, data about a standing order for RTGS is queried. Standing order details are returned.

Specific message content

ReturnStandingOrder contains the following set of information.

Message item	Data type/code	Utilisation
Request type Docu- ment/RtrStgOrdr/MsgHdr/ReqTp/Cd	StandingOrderQueryType1Code	Request type SDTL
Standing order identification Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrId/Id	RestrictedFINMax16Text	Identification
Account identification Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrId/Acct/Id/Othr/Id	RestrictedFINMax34Text	Account identification
Account owner Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrId/AcctOwner/FinInstnId/BICFI	BICFIIdentifier	Account owner
Amount Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/Amt/AmtWthtCcy	RestrictedFINImpliedCurrencyAndAmount	Amount
CreditDebit indicator Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/CdtDbtInd	CreditDebitCode	CreditDebit indicator
Validity period Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/VldtyPrd	DatePeriodDetails1	Validity period

Message item	Data type/code	Utilisation
Responsible party Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/RspnsblPty/FinInstnId/BI CFI	BICFIIdentifier	Responsible NCB
Execution type Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/ExctnTp/Evt/Cd	ExternalSystemEventType1Code	Execution type
Creditor account Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/CdtrAcct/Id/Othr/Id	RestrictedFINMax34Text	Creditor account

Table 222 - ReturnStandingOrder (camt.070) – usage case RTGS return standing order details

Usage case example: RTGSReturnStandingOrderDetails_example.xml

In this example reference data of the standing order with ID “STOID00001” is returned.

Usage case: ASI6 RTGS return standing order details

In this usage case data about a standing order for RTGS ASI procedure 6 is requested.

Specific message content

Return standing order contains the following set of information:

Message item	Data type/code	Utilisation
Request type Docu- ment/RtrStgOrdr/MsgHdr/ReqTp/Cd	StandingOrderQueryType1Code	Request Type SDTL
Standing order identification Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr/Id	RestrictedFINMax16Text	Identification
Account identification Docu-	RestrictedFINMax34Text	Account identification

Message item	Data type/code	Utilisation
Document/Report/Order/Status/OrderReference/Account/OtherReference		
Account owner Document/Report/Order/Status/OrderReference/AccountOwner/FinancialInstitution/BICFI	BICFIIdentifier	Technical account BIC
Amount Document/Report/Order/Status/OrderReference/Order/Status/Amount/AmountWithCurrency	RestrictedFINImpliedCurrencyAndAmount	Amount
CreditDebit indicator Document/Report/Order/Status/OrderReference/Order/Status/CreditDebitIndicator	CreditDebitCode	CreditDebit indicator
Validity period Document/Report/Order/Status/OrderReference/Order/Status/ValidityPeriod	DatePeriodDetails1	Validity period
Responsible party Document/Report/Order/Status/OrderReference/Order/Status/ResponsibleParty/FinancialInstitution/BICFI	BICFIIdentifier	Responsible NCB
Execution type Document/Report/Order/Status/OrderReference/Order/Status/ExecutionType/Event/Code	ExternalSystemEventType1Code	Execution type
Creditor TBD	TBD	Creditor BIC

Message item	Data type/code	Utilisation
Creditor account Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/CdtrAcct/Id/Othr/Id	RestrictedFINMax34Text	Creditor account
Debtor TBD	TBD	Debtor BIC
Debtor account Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/DbtrAcct/Id/Othr/Id	RestrictedFINMax34Text	Debtor account

Table 223 - ReturnStandingOrder (camt.070) – usage case ASI6 RTGS return standing order details

Usage case example: ASI6 RTGSReturnStandingOrderDetails_example.xml

Usage case: CLM Return standing order details

In this usage case data about a standing order for RTGS is requested.

Specific message content

Return standing order contains the following set of information.

Message item	Data type/code	Utilisation
Request type Docu- ment/RtrStgOrdr/MsgHdr/ReqTp/Cd	StandingOrderQueryType1Code	Request type STDL
Standing order identification Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr/Id/Id	RestrictedFINMax16Text	Identification
Account identification Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr/Id/Id/Id/Othr/Id	RestrictedFINMax34Text	Account identification
Account owner Docu-	BICFIIdentifier	Account owner

Message item	Data type/code	Utilisation
ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrId/AcctOwnr/FinInstnId/BICFI		
Amount Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/Amt/AmtWthtCcy	RestrictedFINImpliedCurrencyAndAmount	Amount
CreditDebit indicator Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/CdtDbtInd	CreditDebitCode	CreditDebit indicator
Validity period Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/VldtyPrd	DatePeriodDetails1	Validity period
Responsible party Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/RspnsblPty/FinInstnId/BICFI	BICFIIdentifier	Responsible party
Execution type Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/ExctnTp/Evt/Cd	ExternalSystemEventType1Code	Execution type
Creditor account Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrOrErr/StgOrdr/CdtrAcct/Id/Othr/Id	RestrictedFINMax34Text	Creditor account

Table 224 - ReturnStandingOrder (camt.070) – usage case CLM Return standing order details

Usage case example: CLMReturnStandingOrderDetails_example.xml

In this example reference data of the standing order with ID “STOID00002” is returned.

For all the usage cases, the returned data in case of an error response is listed below:

Message item	Data type/code	Utilisation
Code Docu- ment/RtrStgOrdr/RptOrErr/OprlErr/Err/ Cd	ErrorHandling1Code	Specific error
Description Docu- ment/RtrStgOrdr/RptOrErr/OprlErr/Des c	Max140Text	Textual description in addition to the reported error

Table 225 - ReturnStandingOrder (camt.070) – usage case Error

14.3.26 DeleteStandingOrder (camt.071)

14.3.26.1 Overview and scope of the message

This chapter illustrates the *DeleteStandingOrder* message.

The *DeleteStandingOrder* message is sent by an actor authorised to delete standing orders for liquidity transfers.

The *DeleteStandingOrder* message is replied by a [Receipt \(camt.025\)](#) [▶ 474] to return a positive technical response to the sender of the message or to provide detailed information in case of an error.

14.3.26.2 Schema

Outline of the schema

The *DeleteStandingOrder* message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message identification provided by the requesting actor.

StandingOrderDetails

This block is mandatory and provides with all the key information to identify an existing standing order to be deleted. Both identification and account identification must be provided.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/camt.071.001.02>

14.3.26.3 The message in business context

Usage case: Delete standing order

This usage case describes the deletion of a standing order in CRDM.

Specific message requirements

Message item	Data type/code	Utilisation
Standing order identification Docu- ment/DelStgOrdr/StgOrdrDtls/StgOrdr/ d	RestrictedFINMax16Text	Standing order identification
Account identification Docu- ment/DelStgOrdr/StgOrdrDtls/StgOrdr/ Acct/Id/Othr/Id	RestrictedFINMax34Text	Account identification

Table 226 - DeleteStandingOrder (camt.071) – usage case Delete standing order

Usage case example: DeleteStandingOrder_example.xml

In this example it is requested the deletion of the standing order with Identification “STDID001” for the account identified with “ACC001”.

14.3.27 BillingReportRequest (camt.076)

14.3.27.1 Overview and scope of the message

14.3.27.2 Schema

14.3.27.3 The message in business context

14.3.28 BillingReport (camt.077)

14.3.28.1 The message in business context

14.3.28.2 Schema

14.3.28.3 The message in business context

14.3.29 AuditTrailQuery (camt.097)

14.3.29.1 Overview and scope of the message

This chapter illustrates the *AuditTrailQuery* message.

The *AuditTrailQuery* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to request information on audit trails which have been logged by the RTGS component.

The *AuditTrailQuery* message can be used to query audit trails relating to: transactions, reservations or limits.

Within RTGS, the *AuditTrailQuery* message has the following usages:

In response to the *AuditTrailQuery* message, an [AuditTrailReport \(camt.098\)](#) [▶ 554] message containing the requested information is returned.

14.3.29.2 Schema

Outline of the schema.

The *AuditTrailQuery* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

ReservationQueryDefinition

Definition of the reservation query.

SearchCriteria

Mandatory and non-repetitive. It defines the criteria to extract the reservation information. It includes the following elements:

- | account owner
- | account identification
- | date period (optional)
- | audit trail type (transaction, reservation, limit)

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.097.001.01_RTGS

14.3.29.3 The message in business context

Usage case: Audit Trail for RTGS Query

In this usage case, the sender requests information regarding the audit trail of the transaction, reservation or limit activity defined within their query criteria.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Message Identification Document/AudtTriQry/MsgHdr/MsgId	RestrictedFINMax35Text	Point to point reference, as assigned by the sender, to unambiguously identify the message
Account Id Docu- ment/AudtTriQry/SchCrit/AcctId/Othr/Id	RestrictedFINX2Max34Text	Identification assigned by an institution
Account Owner Docu- ment/AudtTriQry/SchCrit/AcctOwnr/Finl nstnId	RTGS_BIC11Text	Owner of the account which is being queried
Audit Trail Type Docu- ment/AudtTriQry/SchCrit/AudTriTp	LIMI RSVT TRXN	Specifies which type of audit trails data must be returned.

Table 227 - AuditTrailQuery (camt.097) – usage case Audit Trail for RTGS Query

Usage case example: camt.097_RTGS_AuditTrailQuery_AuditTrailForRTGSQuery_Example.xml

In this example, an *AuditTrailQuery* is instructed by the account owner with transactions as target data. It illustrates the mandatory elements in the message.

14.3.30 AuditTrailReport (camt.098)

14.3.30.1 Overview and scope of the message

This chapter illustrates the *AuditTrailReport* message.

The *AuditTrailReport* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to provide audit trail information which has been logged by the RTGS component.

The *AuditTrailReport* message may contain audit trail information relating to: transactions, reservations or limits.

Within RTGS, the *AuditTrailReport* message has the following usages:

- I Audit Trail for RTGS Query (Data or Error response)

The *AuditTrailReport* message is sent in response to an [AuditTrailQuery \(camt.097\)](#) [▶ 552] message, which requested the information.

14.3.30.2 Schema

Outline of the schema.

The *AuditTrailReport* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and the original business query.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query message, or an error indication.

AuditTrailReport

This building block is mandatory and non-repetitive. It includes the following elements:

- | account owner
- | account identification
- | date period
- | audit trail blocks

AuditTrail

This building block is repetitive. It contains details of an audit trail entry fulfilling the query criteria.

- | audit trail type (transaction, reservation, limit)
- | timestamp
- | approval status
- | processing status

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.098.001.01_RTGS

14.3.30.3 The message in business context

Usage case: Audit Trail for RTGS Query (Data response)

In this usage case, the recipient of the message is being informed regarding the audit trail of the transaction, reservation or limit activity defined within their query criteria.

Specific message content

Message item	Data type/code	Utilisation
Message Identification Document/AudtTrlRpt/MsgHdr/MsgId	RestrictedFINMax35Text	Point to point reference, as assigned by the sender, to unambiguously identify the message.
Original Business Query Docu- ment/AudtTrlRpt/MsgHdr/OrgnlBizQry/ MsgId	RestrictedFINMax35Text	Point to point reference, as assigned by the original initiating party, to unambiguously identify the original query message.
Account Identification Docu- ment/AudtTrlRpt/RptOrErr/AudtTrlRpt/ AcctId/Othr/Id	RestrictedFINX2Max34Text	Identification of the account on which information is requested.
Account Owner Docu- ment/AudtTrlRpt/RptOrErr/AudtTrlRpt/ AcctOwnr/FinInstnId/BICFI	RTGS_BIC11Text	Owner of the account which is being queried.
Audit Trail Record Docu- ment/AudtTrlRpt/RptOrErr/AudtTrlRpt/ AudtTrlOrErr/AudtTrl/Rcrd	Transaction Limit Reservation	Requested information on the audit trail. Provides the business item record for which details of the audit trail data are provided.

Message item	Data type/code	Utilisation
Operation Time Stamp Docu- ment/AudtTrlRpt/RptOrErr/AudtTrlRpt/ AudtTrlOrErr/AudtTrl/OprTmStmp	ISODateTime	Timestamp of the change.
Approval Status Docu- ment/AudtTrlRpt/RptOrErr/AudtTrlRpt/ AudtTr- lOrErr/AudtTrl/ApprovISts/ApprovIReqd	TrueFalseIndicator	Provides the details related to the ap- proval of the change reported in the audit trail.
Docu- ment/AudtTrlRpt/RptOrErr/AudtTrlRpt/ AudtTr- lOrErr/AudtTrl/ApprovISts/InstgUsr	RestrictedFINMax16Text	
Processing Status	Exact4AlphaNumericText	Provides details about the processing status of the change reported in the audit trail

Table 228 - AuditTrailReport (camt.098) – usage case Audit Trail for RTGS Query (Data response)

Usage case example: camt.098_RTGS_AuditTrailReport_AuditTrailForRTGSQueryData_Example.xml

In this example an *AuditTrailReport* containing a reference to an incoming message with the ID “MSGIDcamt.097”, and the available audit trail related to transactions changes is sent to the requesting party.

Usage case: Audit Trail for RTGS Query (Error response)

In this usage case, the recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent audit trail for RTGS query ([AuditTrailQuery \(camt.097\)](#) [▶ 552]).

The identification of the previously sent query message is included in this error response for reconciliation purposes.

Specific message content

Message item	Data type/code	Utilisation
Message Identification Document/AudtTrlRpt/MsgHdr/MsgId	RestrictedFINMax35Text	Point to point reference, as assigned by the sender, to unambiguously identify the message.
Original Business Query Docu- ment/AudtTrlRpt/MsgHdr/OrgnlBizQry/ MsgId	RestrictedFINMax35Text	Point to point reference, as assigned by the original initiating party, to unambiguously identify the original query message.
Operational Error Docu- ment/AudtTrlRpt/RptOrErr/OprlErr/Err/ Prtry	Max4Text	Proprietary error code
Docu- ment/AudtTrlRpt/RptOrErr/OprlErr/Des c	Max140Text	Description of error

Table 229 - AuditTrailReport (camt.098) – usage case Audit Trail for RTGS Query (Error response)

Usage case example: camt.098_RTGS_AuditTrailReport_AuditTrailForRTGSQueryError_Example.xml

14.3.31 DirectDebitMandateQuery (camt.099)

14.3.31.1 Overview and scope of the message

This chapter illustrates the *DirectDebitMandateQuery* message.

The *DirectDebitMandateQuery* is sent by an actor authorised to query direct debit mandate data.

In response to the *DirectDebitMandateQuery*, a [DirectDebitMandateReport\(camt.100\)](#) [▶ 560] containing the requested information is returned.

14.3.31.2 Schema

Outline of the schema

The *DirectDebitMandateQuery* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and it contains an identification assigned by the sending party to uniquely and unambiguously identify the message.

Search Criteria

This block is optional and it contains detailed information related to the direct debit mandate query message.

Allowed search criteria are:

- | creditor
- | cash account
- | direct debit mandate reference
- | service, for the specification of the service for which the query must be executed, with the currency details

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/camt.099.001.001>

14.3.31.3 The message in business context

Usage case: Direct debit mandate query

In this usage case data about direct debit mandate is requested.

Specific message requirements

Message item	Data type/code	Utilisation
Creditor Docu- ment/DrctDbtMndtQry/SchCrit/Cdtr/Id/Id/AnyBIC	AnyBICIdentifier	Creditor
Responsible party Docu- ment/DrctDbtMndtQry/SchCrit/Cdtr/Rsp nsblPtyId/Id/AnyBIC	AnyBICIdentifier	NCB

Message item	Data type/code	Utilisation
Cash account Docu- ment/DrctDbtMndtQry/SchCrit/CshAcct /Othr/Id	Max34Text	Account
Direct debit mandate reference Docu- ment/DrctDbtMndtQry/SchCrit/DrctDbt MndtRef	Max35Text	Direct debit mandate reference
Service Docu- ment/DrctDbtMndtQry/SchCrit/Svc/Sysl d/MktInfrstrctrId/Prtry	Max35Text	Service

Table 230 - DirectDebitMandateQuery (camt.099) – usage case Direct debit mandate query

Usage case example: DirectDebitMandateQuery_example.xml

14.3.32 DirectDebitMandateReport(camt.100)

14.3.32.1 Overview and scope of the message

This chapter illustrates the *DirectDebitMandateReport* message.

The *DirectDebitMandateReport* is sent by CRDM to an authorised actor to provide with requested direct debit mandate information.

The *DirectDebitMandateReport* is sent in response to the [DirectDebitMandateQuery \(camt.099\)](#) [▶ 558] message.

14.3.32.2 Schema

Outline of the schema

The *DirectDebitMandateReport* message is composed of the following message building blocks:

MessageHeader

It contains an identification assigned to uniquely and unambiguously identify the message and the identification of the original business query generating the report.

ReportOrError

This building block is mandatory. It contains either the information matching the search criteria of the related query or an error indication.

Direct Debit Mandate Report

It provides requested information on direct debit mandate, with the service information.

The direct debit mandate data includes the following elements:

- | creditor
- | cash account
- | maximum amounts
- | direct debit mandate reference
- | valid from
- | valid to

OperationalError

In case of error, it provides the reason why the requested information cannot be given.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/camt.100.001.001>

14.3.32.3 The message in business context

Usage case: Direct debit mandate report

This message usage provides the sender with requested information about direct debit mandate data.

Specific message content

A direct debit mandate report contains the following set of information.

Message item	Data type/code	Utilisation
Service Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/Svc/SysId/MktInfrstrctrId/Prtry	Max35Text	Service
Creditor Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/MndtOrErr/DrctDbtMndt/Cdtr/Id/Id/ AnyBIC	AnyBICIdentifier	Creditor
Responsible party Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/MndtOrErr/DrctDbtMndt/Cdtr/Rspn sblPtyId/Id/AnyBIC	AnyBICIdentifier	NCB
Cash account Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/MndtOrErr/DrctDbtMndt/CshAcct/ Othr/Id	Max34Text	Account
Amount type Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/MndtOrErr/DrctDbtMndt/MaxAmt/T p/Cd	ExternalMaximumAmountType1Code	Amount type
Amount Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/MndtOrErr/DrctDbtMndt/MaxAmt/ Amt	ActiveCurrencyAndAmount	Amount

Message item	Data type/code	Utilisation
Direct debit mandate reference Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/MndtOrErr/DrctDbtMndt/DrctDbtM ndtRef	Max35Text	Direct debit mandate reference
Valid from Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/MndtOrErr/DrctDbtMndt/VldFr/DtT m	ISODateTime	Valid from
Valid to Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDb tMnd- tRpt/MndtOrErr/DrctDbtMndt/VldTo/DtT m	ISODateTime	Valid from

Table 231 - DirectDebitMandateReport (camt.100) – usage case Direct debit mandate report

The returned data in case of an error response is listed below:

Message item	Data type/code	Utilisation
Proprietary Docu- ment/DrctDbtMndtRpt/RptOrErr/OpriErr /Err/Prtry	Max35Text	Specific error
Description Docu- ment/DrctDbtMndtRpt/RptOrErr/OpriErr /Desc	Max140Text	Textual description in addition to the reported error

Table 232 - DirectDebitMandateReport (camt.100) – usage case Error

Usage case example: DirectDebitMandateReportrReport_example.xml

14.4 Headers (head)

14.4.1 BusinessApplicationHeader (head.001)

14.4.1.1 Overview and scope of the message

This chapter illustrates the *BusinessApplicationHeader* message.

For payment messages between bank A and bank B, FROM identifies bank A and TO identifies bank B. For service messages between bank A and the MI (e.g. [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585] connected payment, liquidity messages etc.), FROM identifies bank A and TO identifies the MI.

14.4.1.2 Schema

Outline of the schema

The BAH message is composed of the following message building blocks:

FROM

The sender that has created this message for the receiver that processes this message. FROM BIC must have exactly 11 characters.

TO

The receiver designated by the sender who ultimately processes this message. TO BIC must have exactly 11 characters.

BusinessMessageIdentifier

Identifies unambiguously the message. The BusinessMessageIdentifier has maximum 35 characters.

MessageDefinitionIdentifier

Contains the MessageIdentifier that defines the message. It must contain a MessageIdentifier published on the ISO 20022 website.

CreationDate

Date and time when this message (header) was created.

CopyDuplicate (optional)

Indicates whether the message is a copy, a duplicate or a copy of a duplicate of a previously sent ISO 20022 message.

PossibleDuplicate (optional)

Is a flag indicating if the message exchanged between sender and receiver is possibly a duplicate.

Signature (optional)

Contains the digital signature of the business entity authorised to sign this message.

Related (optional)

Specifies the BAH of the message to which this message relates. It can be used when replying to a query; it can also be used when canceling or amending.

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/head.001.001.01_RTGS

Business rules applicable to the schema

For business rules applicable to *BusinessApplicationHeader* please refer to the business rules table below.

14.4.1.3 The message in business context

The BAH contains information to correctly process the message payload. Every message exchanged between the RTGS component and a connected party, includes such information. The relationship between the BAH and the message payload is one to one.

The BAH includes the following main information:

- | document routing (e.g. sender, receiver, information about the message)
- | document identification (e.g. MessageDefinitionIdentifier, creation date and time)
- | document processing information (e.g. sender, service, COPY, possible duplicate)

Message example 1: head.001_RTGS_IncomingMessageWithinRTGS_Example.xml

In this example the BusinessApplicationHeader (BAH) is used for an incoming message within RTGS. It is sent from a NCB with parent BIC "NCBPARNTBIC" and party BIC "NCBPARTYBIC" to RTGS. The BAH is filled with the corresponding digital signature.

Message example 2: head.001_RTGS_OutgoingMessageBeingSentByRTGSAsCopy_Example.xml

In this example the BusinessApplicationHeader is used for an outgoing message being sent by RTGS as a copy to a party other than the account owner, e.g. NCB, for information purposes. Sending and receiving system entity is the NCB "NCBBICEUXXX". The BAH includes the digital signature.

Message example 3: head.001_RTGS_BankToCustomerStatementSentByRTGS_Example.xml

In this example the BusinessApplicationHeader is used for a bank to customer statement sent by RTGS to the account owner "NCBBICEUXXX". The BAH includes the digital signature.

Message example 4: head.001_RTGS_IncomingMessag_pacs.009COV_Example.xml

In this example the BusinessApplicationHeader (BAH) is used to sent a pacs.009 COV to RTGS. It is sent from a NCB with BIC "NCBPARNTBIC" to RTGS. The BAH is filled with the corresponding digital signature.

14.4.2 BusinessFileHeader (head.002)

14.4.2.1 Overview and scope of the message

This chapter illustrates the *BusinessFileHeader* message.

The *BusinessFileHeader* is used by the RTGS component to receive several business messages within one file to the RTGS component.

Under a single *BusinessFileHeader*, every message within a file has to be an ISO 20022 Message together with its business application header (business message). A file can contain one or several business messages.

Within RTGS, the *BusinessFileHeader* information is used for:

- | consistency and completeness checks

In response to an incoming file which fails validation, the RTGS component sends a [ReceiptAcknowledgement \(admi.007\)](#) [▶ 391] message containing information on negative validation.

Results from validation which is performed at file level, are sent without BAH information.

14.4.2.2 Schema

Outline of the schema.

The *BusinessFileHeader* is composed of the following building blocks:

PayloadDescription

The PayloadDescription is a mandatory block and contains the following information tags:

- | PayloadDetails: with PayloadIdentifier; CreationDateAndTime and PossibleDuplicateFlag
- | ApplicationSpecificInformation: which contains information about the total number of instances (messages) within the file
- | PayloadTypeDetails: which declares the payload content (describes the type of business document being exchanged)
- | ManifestDetails: with information to each DocumentType and the number of instances (messages) for each declared type.

Payload

The payload is a mandatory block and contains the set of business messages, each built of an ISO 20022 message together with its business application header.

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/head.002.001.01_RTGS

Business rules applicable to the schema

For business rules applicable to *BusinessFileHeader* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.4.2.3 The message in business context

Message example: head.002_RTGS_IncomingMessageFileWithinRTGS_Example.xml

In this example the *BusinessFileHeader* is used for an incoming file within RTGS. The file payload contains a [GetAccount \(camt.003\)](#) [▶ 394]. The file envelope is filled with the corresponding digital signature.

Message example: head.002_RTGS_OutgoingFileSentfromRTGS_Example.xml

In this example the *BusinessFileHeader* is used for an outgoing file. The file includes a [PaymentStatusReport \(pacs.002\)](#) [▶ 568] message. The file envelope includes the corresponding digital signature.

14.5 Payments clearing and settlement (pacs)

14.5.1 PaymentStatusReport (pacs.002)

14.5.1.1 Overview and scope of the message

This chapter illustrates the *PaymentStatusReport* message.

The *PaymentStatusReport* message is sent by the RTGS component to a RTGS Account Holder (or a party authorised by them). It is used to inform this party about the status of a previous payment.

The *PaymentStatusReport* message is treated as mandatory for all processing failure situations. To receive a *FIToFIPaymentStatusReport* message for normal successful processing situations, subscription is required.

Within RTGS, the *PaymentStatusReport* message has the following usages:

- | Payment Rejection Notification
- | Payment Settlement Notification

The *PaymentStatusReport* message is sent in response to a previously sent payment message ([PaymentReturn \(pacs.004\)](#) [▶ 571], [CustomerCreditTransfer \(pacs.008\)](#) [▶ 577], [FinancialInstitutionCreditTransfer \(GEN and COV\) \(pacs.009\)](#) [▶ 585] or [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 603]).

14.5.1.2 Schema

Outline of the schema.

The *PaymentStatusReport* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

TransactionInformationAndStatus

This building block is mandatory and non-repetitive. It provides information concerning the original transactions, to which the status report message refers. It may contain:

- | original group information
- | original instruction identification
- | original transaction identification

- | status
- | status reason information
- | CLM component reference

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.002.001.09_RTGS

Business rules applicable to the schema

No business rules are applicable to a *PaymentStatusReport* message.

14.5.1.3 The message in business context

Usage case: Payment Rejection Notification

In this usage case, the recipient of the message is being informed that a previous payment sent by them (or on their behalf) has been rejected and will not be processed further. A rejection code will be given and, in most cases, a reason code and reason text will be provided also.

Specific message content

Message item	Data type/code	Utilisation
OriginalMessageIdentification Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/OrgnlGrpInf/OrgnlMsgId	Max35Text	Message ID of original instruction
OriginalMessageNameIdentification Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/OrgnlGrpInf/OrgnlMsgNmId	RTGS_XMLMessageNamePattern	Message name of the original instruction
OriginalInstructionIdentification Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/OrgnlInstrId	Max35Text	Identification of the original instruction

Message item	Data type/code	Utilisation
TransactionStatus Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/Tx Sts	RTGS_TransactionStatusCode	RJCT
Proprietary Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/St sRsnInf/Rsn/Prtry	Max35Text	Status reason
AdditionalInformation Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/St sRsnInf/Addtlinf	Max105Text	Detailed error description

Table 233 - PaymentStatusReport (pacs.002) – usage case Payment Rejection Notification

Usage case example: pacs.002_RTGS_FIToFIPaymentStatusReportPORejection_Example.xml

In this example a *PaymentStatusReport* “Payment Rejection Notification” containing a reference to an incoming message with the ID “INSTRIDpacs.008”, the error code “2862” and the description “Request out of cut-off time” is sent to the corresponding party.

Usage case: Payment Settlement Notification

In this usage case, the recipient of the message is being informed that a previous payment sent by them (or on their behalf) has been actioned successfully (i.e. payment has been settled).

Specific message content

Message item	Data type/code	Utilisation
OriginalMessageIdentification Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/Or gnlGrpInf/OrgnlMsgId	Max35Text	Message ID of original instruction
OriginalMessageNameIdentification Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/Or gnlGrpInf/OrgnlMsgNmId	RTGS_XMLMessageNamePattern	Message name of the original instruction

Message item	Data type/code	Utilisation
OriginalTransactionIdentification Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/Or gnITxId	Max35Text	Transaction ID of the original instruc- tion
TransactionStatus Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/Tx Sts	RTGS_TransactionStatusCode	ACSC
ClearingSystemReference Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/Clr SysRef	Max105Text	RTGS booking reference

Table 234 - PaymentStatusReport (pacs.002) – usage case Payment Settlement Notification

Usage case example: pacs.002_RTGS_FIToFIPaymentStatusReportPOSettlement_Example.xml

In this example a *PaymentStatusReport* “Payment Settlement Notification” resulting from a payment containing the RTGS booking reference is sent to the corresponding party.

14.5.2 PaymentReturn (pacs.004)

14.5.2.1 Overview and scope of the message

This chapter illustrates the *PaymentReturn* message.

The *PaymentReturn* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to undo a previously settled payment.

The *PaymentReturn* message concerns only one payment.

Within RTGS, the *PaymentReturn* message has the following usages:

- I Payment Message
- I Payment Settlement Notification

In response to the *PaymentReturn* message, a [PaymentStatusReport \(pacs.002\)](#) [▶ 568] message containing the status of the payment return is returned to the sending party.

In addition, if the payment return is successfully processed, the *PaymentReturn* message is forwarded to the re-credited RTGS Account Holder (or a party authorised by them).

14.5.2.2 Schema

Outline of the schema.

The *PaymentReturn* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

TransactionInformation

Information concerning the original transactions, to which the return message refers.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.004.001.08_RTGS

Business rules applicable to the schema

For business rules applicable to *PaymentReturn* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.5.2.3 The message in business context

Usage case: Payment Message

In this usage case, the message provides the details required for the RTGS component to execute a reversal of a previously settled payment.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
OriginalMessageIdentification Docu- ment/PmtRtr/TxInf/OrgnlGrpInf/OrgnlM sgld	Max35Text	Message ID of original instruction
OriginalMessageNameIdentification Docu- ment/PmtRtr/TxInf/OrgnlGrpInf/OrgnlM sgNmld	RTGS_XMLMessageNamePattern	Message name of the original instruc- tion
OriginalInstructionIdentification Document/PmtRtr/TxInf/OrgnlInstrld	Max35Text	Identification of the original instruction
OriginalEndToEndIdentification Docu- ment/PmtRtr/TxInf/OrgnlEndToEndld	Max35Text	End-to-end identification of the original instruction
OriginalInterbankSettlementAmount Docu- ment/PmtRtr/TxInf/OrgnlIntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Interbank settlement amount of the original instruction
OriginalInterbankSettlementDate Docu- ment/PmtRtr/TxInf/OrgnlIntrBkSttlmDt	ISODate	Interbank settlement date of the origi- nal instruction
ReturnedInterbankSettlementAmount Docu- ment/PmtRtr/TxInf/RtrdIntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Settlement amount of the return pay- ment
InterbankSettlementDate Document/PmtRtr/TxInf/IntrBkSttlmDt	ISODate	Interbank settlement date of the return payment
SettlementPriority Document/PmtRtr/TxInf/SttlmPrty	Priority3Code	Priority for the return payment
ReturnedInstructedAmount Document/PmtRtr/TxInf/RtrdInstdAmt	RTGS_Max14_Max5DecimalAmount	Gross amount of return payment
ExchangeRate Document/PmtRtr/TxInf/XchgRate	BaseOneRate	Exchange rate
CompensationAmount Document/PmtRtr/TxInf/CompstnAmt	RTGS_Max14_Max5DecimalAmount	Compensation amount

Message item	Data type/code	Utilisation
ChargeBearer Document/PmtRtr/TxInf/ChrgBr	ChargeBearerType1Code	Charge bearer for the return payment
Amount Document/PmtRtr/TxInf/ChrgsInf/Amt	RTGS_Max14_Max5DecimalAmount	Charges amount
BICFI Docu- ment/PmtRtr/TxInf/ChrgsInf/Agt/FinInst nId/BICFI	BICFIIdentifier	Charges agent (BIC)
BICFI Docu- ment/PmtRtr/TxInf/InstgAgt/FinInstnId/ BICFI	BICFIIdentifier	Party instructing the return payment
BICFI Docu- ment/PmtRtr/TxInf/InstdAgt/FinInstnId/ BICFI	BICFIIdentifier	Party receiving the return payment instruction
Return chain	various	Full analysis to follow.
Originator Document/PmtRtr/TxInf/RtrRsnInf/Orgtr	PartyIdentification125	Party issuing the return payment in- struction
Code Docu- ment/PmtRtr/TxInf/RtrRsnInf/Rsn/Cd	ExternalReturnReason1Code	Reason code for return payment
AdditionalInformation Docu- ment/PmtRtr/TxInf/RtrRsnInf/AddtlInf	Max105Text	Additional reason information

Table 235 - PaymentReturn (pacs.004) – usage case Payment Message

Usage case example: pacs.004_RTGS_PaymentReturn_PaymentOrder_Example.xml

In this example a *PaymentReturn* “Payment Message” is sent by a direct participant to the RTGS component for the execution of a return settlement.

Usage case: Payment Settlement Notification

In this usage case, the message is outbound from the RTGS component. It identifies to the recipient a payment which has been reversed and the funds credited to one of its RTGS DCAs. The payment return would have been ordered by the originally credited payment party (or on their behalf) using a *PaymentReturn* message.

Specific message content

Message item	Data type/code	Utilisation
OriginalMessageIdentification Docu- ment/PmtRtr/TxInf/OrgnlGrpInf/OrgnlM sgId	Max35Text	Message ID of original instruction
OriginalMessageNameIdentification Docu- ment/PmtRtr/TxInf/OrgnlGrpInf/OrgnlM sgNmId	RTGS_XMLMessageNamePattern	Message name of the original instruc- tion
OriginalInstructionIdentification Document/PmtRtr/TxInf/OrgnlInstrId	Max35Text	Identification of the original instruction
OriginalEndToEndIdentification Docu- ment/PmtRtr/TxInf/OrgnlEndToEndId	Max35Text	End-to-end identification of the original instruction
OriginalInterbankSettlementAmount Docu- ment/PmtRtr/TxInf/OrgnlIntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Interbank settlement amount of the original instruction
OriginalInterbankSettlementDate Docu- ment/PmtRtr/TxInf/OrgnlIntrBkSttlmDt	ISODate	Interbank settlement date of the origi- nal instruction
ReturnedInterbankSettlementAmount Docu- ment/PmtRtr/TxInf/RtrdIntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Settlement amount of the return pay- ment
InterbankSettlementDate Document/PmtRtr/TxInf/IntrBkSttlmDt	ISODate	Interbank settlement date of the return payment
SettlementPriority Document/PmtRtr/TxInf/SttlmPrty	Priority3Code	Priority for the return payment

ReturnedInstructedAmount Document/PmtRtr/TxInf/RtrdInstdAmt	RTGS_Max14_Max5DecimalAmount	Gross amount of return payment
ExchangeRate Document/PmtRtr/TxInf/XchgRate	BaseOneRate	Exchange rate
CompensationAmount Document/PmtRtr/TxInf/CompstnAmt	RTGS_Max14_Max5DecimalAmount	Compensation amount
ChargeBearer Document/PmtRtr/TxInf/ChrgBr	ChargeBearerType1Code	Charge bearer for the return payment
Amount Document/PmtRtr/TxInf/ChrgsInf/Amt	RTGS_Max14_Max5DecimalAmount	Charges amount
BICFI Docu- ment/PmtRtr/TxInf/ChrgsInf/Agt/FinInst nId/BICFI	BICFIIdentifier	Charges agent (BIC)
ClearingSystemReference Document/PmtRtr/TxInf/ClrSysRef	Max35Text	Reference at the clearing system for the return payment.
BICFI Docu- ment/PmtRtr/TxInf/InstgAgt/FinInstnId/ BICFI	BICFIIdentifier	Party instructing the return payment
BICFI Docu- ment/PmtRtr/TxInf/InstdAgt/FinInstnId/ BICFI	BICFIIdentifier	Party receiving the return payment instruction
Return chain	various	Full analysis to follow.
Originator Document/PmtRtr/TxInf/RtrRsnInf/Orgtr	PartyIdentification125	Party issuing the return payment instruction
Code Docu- ment/PmtRtr/TxInf/RtrRsnInf/Rsn/Cd	ExternalReturnReason1Code	Reason code for return payment
AdditionalInformation Docu- ment/PmtRtr/TxInf/RtrRsnInf/AddtlInf	Max105Text	Additional reason information

Table 236 - PaymentReturn (pacs.004) – usage case Payment Settlement Notification

Usage **case** **example:**
pacs.004_RTGS_PaymentReturn_PaymentOrderSettlementNotification_Example.xml

In this example a *PaymentReturn* “Payment Settlement Notification” is sent to the corresponding party after the execution of a return settlement in the RTGS component.

14.5.3 CustomerCreditTransfer (pacs.008)

14.5.3.1 Overview and scope of the message

This chapter illustrates the *CustomerCreditTransfer* message.

This message type is used in the RTGS component to execute a payment if the debtor or the creditor or both are non-financial institutions.

The payment message can be sent by a

- | RTGS Account Holder
- | BIC of the multi-addressee access and
- | CB as a RTGS Account Holder

Credited and debited RTGS DCAs must be denominated in the same currency.

Within RTGS, the *CustomerCreditTransfer* message has the following usages:

- | Payment Message
- | Payment Settlement Notification

In response to the *CustomerCreditTransfer* message, a [PaymentStatusReport \(pacs.002\)](#) [▶ 568] message containing the status of the payment return is returned to the sending party.

In addition, if the payment is successfully processed, the *CustomerCreditTransfer* message is forwarded to the credited RTGS Account Holder (or a party authorised by them).

14.5.3.2 Schema

Outline of the schema

The *CustomerCreditTransfer* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

CreditTransferTransactionInformation

Set of elements providing information specific to the individual credit transfer. It contains the following elements:

- | payment identification
- | payment type
- | interbank settlement amount
- | interbank settlement date
- | settlement priority
- | settlement time indication and request
- | instructed amount
- | exchange rate, charges and charge bearer information
- | 3 previous instructing agents and their accounts
- | instructing and instructed agents
- | 3 intermediary agents and their accounts
- | ultimate debtor
- | initiating party
- | debtor, debtor agent, creditor agent and credit parties, and their accounts
- | ultimate creditor
- | instruction for creditor agent
- | purpose
- | regulatory reporting
- | related remittance information
- | remittance information

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.008.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *CustomerCreditTransfer* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.5.3.3 The message in business context

Usage case: Payment Message

In this usage case, the message provides the details required for the RTGS component to execute a payment in which either the debiting party or the crediting party (or both) is a non-financial institution.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Payment ID Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtId/InstrId	Max35Text	Payment ID – instruction ID
Payment ID Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtId/EndToEndId	Max35Text	Payment ID – end to end ID
Payment ID Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtId/TxId	Max35Text	Payment ID – transaction ID
Amount Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/In trBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Payment amount
Date Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/In trBkSttlmDt	ISODate	Payment date
Settlement priority	Priority3Code	NORM

Message item	Data type/code	Utilisation
Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/St tImPrty		HIGH
Charge bearer Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/C hrgBr	ChargeBearerType1Code	CRED DEBT SHAR
BICFI Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/In stgAgt/FinInstnId/BICFI	BICFIIdentifier	Instructing agent BIC
BICFI Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/In stdAgt/FinInstnId/BICFI	BICFIIdentifier	Instructed agent BIC
Debtor Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/D btr	PartyIdentification125	Debtor
Debtor agt Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/D btrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Debtor agent
Creditor agt Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/C dtrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Creditor agent

Table 237 - CustomerCreditTransfer (pacs.008) – usage case Payment Message

Usage case example:
pacs.008_RTGS_FIToFICustomerCreditTransfer_PaymentOrderMessage_Example.xml

Usage case: Payment Settlement Notification

In this usage case, the message is outbound from the RTGS component. It identifies to the recipient a payment which has been made involving one of its RTGS DCAs. The payment would have been ordered by the other payment party (or on their behalf) using this same *CustomerCreditTransfer* message.

Specific message content

Message item	Data type/code	Utilisation
Payment ID Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtId/InstrId	Max35Text	Customers instruction ID
EndToEnd ID Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtId/EndToEndId	Max35Text	Customers end to end ID
Transaction ID Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtId/TxId	Max35Text	Customers transaction ID
Clearing System Reference Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtId/ClrSysRef	Max35Text	RTGS system reference
Payment type priority Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtTpInf/InstrPrty	Priority2Code	NORM HIGH
Payment service level code Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtTpInf/SvcLvl/Cd	ExternalServiceLevel1Code	External Service level code
Payment local instrument code Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtTpInf/LclInstrm/Cd	ExternalLocalInstrument1Code	External Local instrument code

Message item	Data type/code	Utilisation
Payment category purpose code Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/P mtTpInf/CtgyPurp/Cd	ExternalCategoryPurpose1Code	External Category purpose code
Amount Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/In trBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Payment amount
Interbank Settlement Date Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/In trBkSttlmDt	ISODate	Payment date
Settlement priority Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/St tlmPrty	Priority3Code	Settlement priority
Settlement date time Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/St tlmTmIndctn/CrdDtTm	ISODateTime	Settlement date time
Settlement time request - till Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/St tlmTmReq/TillTm	ISOTime	Settlement time request – till time
Settlement time request - from Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/St tlmTmReq/FrTm	ISOTime	Settlement time request – from time
Settlement time request - reject Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/St tlmTmReq/RjctTm	ISOTime	Settlement time request – reject time
Instructed amount Docu-	ActiveOrHistoricCurrencyAndAmount	Instructed amount

Message item	Data type/code	Utilisation
ment/FIToFICstmrCdtTrf/CdtTrfTxInf/InstdAmt		
Exchange rate Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/XchgRate	BaseOneRate	Exchange rate
Charge bearer Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgBr	ChargeBearerType1Code	Charge bearer
Charge amount Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Amt	RTGS_Max14_Max5DecimalAmount	Charge amount
BIC Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/BICFI	BICFIIdentifier	Charge agent - BIC
Clearing System ID Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/ClrSysMmbld/ClrSysId/Cd	ExternalClearingSystemIdentification1Code	RTGS system reference
Member ID Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/ClrSysMmbld/Mmbld	RTGS_RestrictedFINXMax28Text	Charge agent – clearing system member ID
Name Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/Nm	Max140Text	Name of Financial Institution
Charge agent Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/C	ExternalFinancialInstitutionIdentification1Code	Charge agent - other ID, scheme code

Message item	Data type/code	Utilisation
hrgsInf/Agt/FinInstnId/Othr/SchmeNm/ Cd		
Issuer Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/C hrgsInf/Agt/FinInstnId/Othr/Issr	Max35Text	Assignee
Instructing Agent Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/In stgAgt/FinInstnId/BICFI	BICFIIdentifier	Instructing agent
Instructed Agent Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/In stdAgt/FinInstnId/BICFI	BICFIIdentifier	Instructed agent
Debtor Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/D btr	PartyIdentification125	Debtor
Debtor account Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/D btrAcct	CashAccount24	Debtor account
Debtor agt Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/D btrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Debtor agent
Debtor agt account Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/D btrAgtAcct	CashAccount24	RTGS DCA
Creditor agt Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/C dtrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Creditor agent

Message item	Data type/code	Utilisation
Creditor agt account Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/C dtrAgtAcct	CashAccount24	Creditor agent account
Creditor Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/C dtr	PartyIdentification125	Creditor
Creditor account Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/C dtrAcct	CashAccount24	Creditor account
Related remittance information Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/RI tdRmtInf	RemittanceLocation4	Related remittance information
Remittance information Docu- ment/FIToFICstmrCdtTrf/CdtTrfTxInf/R mtInf	RemittanceInformation15	Remittance information

Table 238 - CustomerCreditTransfer (pacs.008) – usage case Payment Settlement Notification

Usage case example: pacs.008_ RTGS_FIToFICustomerCreditTransfer_PaymentOrderSettlementNotification_Example.xml

14.5.4 FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009)

14.5.4.1 Overview and scope of the message

This chapter illustrates the *FinancialInstitutionCreditTransfer* message.

This message type can be used for different RTGS services:

- | high value payments
- | ancillary systems transactions

High value payments can be sent by a

- | RTGS Account Holder
- | BIC of the multi-addressee access and
- | CB as a direct participant or on behalf of a RTGS Account Holder (mandated payments)

Transactions to serve ancillary system settlement procedures can be sent by

- | ancillary systems for procedures real-time settlement and bilateral settlement sent in batch
- | settlement banks to provide liquidity for settlement on dedicated ancillary systems liquidity account (technical account for real-time settlement procedure)

Credited and debited RTGS DCAs must be denominated in the same currency.

Within RTGS, the *FinancialInstitutionCreditTransfer* message has the following usages:

- | settlement of an interbank payment
- | settlement of an interbank payment (customer cover)
- | settlement of ancillary system movement
- | liquidity transfer from RTGS DCA to sub-account

In response to the *FinancialInstitutionCreditTransfer* message, a [PaymentStatusReport \(pacs.002\)](#) [▶ 568] is returned.

14.5.4.2 Schema

Outline of the schema.

The *FinancialInstitutionCreditTransfer* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

CreditTransferTransactionInformation

Set of elements providing information specific to the individual credit transfer(s).

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.009.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *FinancialInstitutionCreditTransfer* please refer to the business rules table below.

14.5.4.3 The message in business context

Usage case: Settlement of an interbank payment

In this usage case, the message describes a payment between two financial institutions.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
FinancialInstitutionCreditTransferV07 Document/FICdtTrf	FinancialInstitutionCreditTransferV07	
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/InstrId	Max35Text	Payment ID – instruction ID
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/EndT oEndId	Max35Text	Payment ID – end to end ID
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/TxId	Max35Text	Payment ID – transaction ID
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/ClrSy sRef	Max35Text	Payment ID – RTGS system reference
Payment service level code Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Sv cLvl/Cd	ExternalServiceLevel1Code	Service level code
Payment local instrument code Docu-	ExternalLocalInstrument1Code	Local instrument code

Message item	Data type/code	Utilisation
ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Lcl Instrm/Cd		
Payment category purpose code Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Ct gyPurp/Cd	ExternalCategoryPurpose1Code	Category purpose code
Amount Docu- ment/FICdtTrf/CdtTrfTxInf/IntrBkSttlmA mt	RTGS_Max14_Max2DecimalAmount	Payment amount
Date Docu- ment/FICdtTrf/CdtTrfTxInf/IntrBkSttlmD t	ISODate	Payment date
Settlement priority Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmPrty	Priority3Code	Settlement priority
Settlement date time Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmIndc tn/CrdDtTm	ISODateTime	Settlement date time
Settlement time request - CLS Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /CLSTm	ISOTime	Settlement time request – CLS time
Settlement time request - till Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /TillTm	ISOTime	Settlement time request – till time
Settlement time request - from Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /FrTm	ISOTime	Settlement time request – from time
Settlement time request - reject	ISOTime	Settlement time request – reject time

Message item	Data type/code	Utilisation
Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /RjctTm		
Prev instructing agt1 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 1	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 1
Prev instructing agt1 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 1Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 2	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 2
Prev instructing agt2 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 2Acct	CashAccount24	Previous instructing agent 2 account
Prev instructing agt3 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 3	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 3
Prev instructing agt3 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 3Acct	CashAccount24	Previous instructing agent 3 account
Instructing agt Docu- ment/FICdtTrf/CdtTrfTxInf/InstgAgt/Finl nstnId/BICFI	BICFIIdentifier	Instructing agent
Instructed agt Docu- ment/FICdtTrf/CdtTrfTxInf/InstdAgt/Finl	BICFIIdentifier	Instructed agent

Message item	Data type/code	Utilisation
nstnId/BICFI		
Intermed. Agt1 Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt1	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 1
Intermed. Agt1 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt1A cct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt2	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 2
Intermed. Agt2 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt2A cct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 3
Intermed. Agt3 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3A cct	CashAccount24	Intermediary agent 3 account
Debtor Document/FICdtTrf/CdtTrfTxInf/Dbtr	PartyIdentification125	Debtor
Debtor account Docu- ment/FICdtTrf/CdtTrfTxInf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Document/FICdtTrf/CdtTrfTxInf/DbtrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Debtor agent
Debtor agt account Docu- ment/FICdtTrf/CdtTrfTxInf/DbtrAgtAcct	CashAccount24	Debtor agent account
Creditor agt	BranchAndFinancialInstitutionIdentifi-	Creditor agent

Message item	Data type/code	Utilisation
Document/FICdtTrf/CdtTrfTxInf/CdtrAgt	PartyIdentification5	
Creditor agt account Docu- ment/FICdtTrf/CdtTrfTxInf/CdtrAgtAcct	CashAccount24	Creditor agent account
Creditor Document/FICdtTrf/CdtTrfTxInf/Cdtr	PartyIdentification125	Creditor
Creditor account Docu- ment/FICdtTrf/CdtTrfTxInf/CdtrAcct	CashAccount24	Creditor account
Remittance information Document/FICdtTrf/CdtTrfTxInf/RmtInf	RemittanceInformation2	Remittance information

Table 239 - FinancialInstitutionCreditTransfer(GEN and COV) (pacs.009) – usage case Settlement of an interbank payment

Usage case example:
pacs.009.001.07_RTGS_FinancialInstitutionCreditTransferIBPayment_Example.xml

Usage case: Settlement of an interbank customer cover payment

In this usage case, the message describes a payment between two financial institutions, performed as a cover for an underlying customer payment.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Institutional information as above PLUS underlying customer payment information below		
Ultimate debtor Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/UltmtDbtr	PartyIdentification125	Ultimate debtor
Initiating party	PartyIdentification125	Initiating party

Message item	Data type/code	Utilisation
Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/InitgPty		
Debtor Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/Dbtr	PartyIdentification125	Debtor
Debtor account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/DbtrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Debtor agent
Debtor agt account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/DbtrAgtAcct	CashAccount24	Debtor agent account
Prev instructing agt1 Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/PrvsInstgAgt1	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 1
Prev instructing agt1 account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/PrvsInstgAgt1 Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/PrvsInstgAgt2	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 2
Prev instructing agt2 account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm	CashAccount24	Previous instructing agent 2 account

Message item	Data type/code	Utilisation
rCdtTrf/PrvsInstgAgt2Acct		
Prev instructing agt3 Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/PrvsInstgAgt3	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 3
Prev instructing agt3 account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/PrvsInstgAgt3Acct	CashAccount24	Previous instructing agent 3 account
Intermed. Agt1 Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/IntrmyAgt1	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 1
Intermed. Agt1 account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/IntrmyAgt1Acct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/IntrmyAgt2	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 2
Intermed. Agt2 account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/IntrmyAgt2Acct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/IntrmyAgt3	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 3
Intermed. Agt3 account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/IntrmyAgt3Acct	CashAccount24	Intermediary agent 3 account
Creditor agt	BranchAndFinancialInstitutionIdentifi-	Creditor agent

Message item	Data type/code	Utilisation
Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/CdtrAgt	cation5	
Creditor agt account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/CdtrAgtAcct	CashAccount24	Creditor agent account
Creditor Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/Cdtr	PartyIdentification125	Creditor
Creditor account Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/CdtrAcct	CashAccount24	Creditor account
Ultimate creditor Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/UltmtCdtr	PartyIdentification125	Ultimate creditor
Instruction for creditor agent Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/InstrForCdtrAgt	InstructionForCreditorAgent1	Instruction for creditor agent
Remittance information Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/RmtInf	RemittanceInformation15	Remittance information
Instructed amount Docu- ment/FICdtTrf/CdtTrfTxInf/UndrlygCstm rCdtTrf/InstdAmt	RTGS_Max14_Max5DecimalAmount	Instructed amount

Table 240 - FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) – usage case Settlement of an interbank customer cover payment

Usage case example:
pacs.009.001.07_RTGS_FinancialInstitutionCreditTransferIBCcustomerCover_Example.xml

Usage case: Settlement of ancillary system movement

In this usage case, the message describes a payment movement instructed by an ancillary system.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
FinancialInstitutionCreditTransferV07 Document/FICdtTrf	FinancialInstitutionCreditTransferV07	
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/InstrId	Max35Text	Payment ID – instruction ID
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/EndT oEndId	Max35Text	Payment ID – end to end ID
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/TxId	Max35Text	Payment ID – transaction ID
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/ClrSy sRef	Max35Text	Payment ID – RTGS system reference
Payment service level code Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Sv cLv/Cd	ExternalServiceLevel1Code	Service level code
Payment local instrument code Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Lcl Instrm/Cd	ExternalLocalInstrument1Code	Local instrument code

Message item	Data type/code	Utilisation
Payment category purpose code Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Ct gyPurp/Cd	ExternalCategoryPurpose1Code	Category purpose code
Amount Docu- ment/FICdtTrf/CdtTrfTxInf/IntrBkSttlmA mt	RTGS_Max14_Max2DecimalAmount	Payment amount
Date Docu- ment/FICdtTrf/CdtTrfTxInf/IntrBkSttlmD t	ISODate	Payment date
Settlement priority Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmPrty	Priority3Code	Settlement priority
Settlement date time Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmIndc tn/CrdDtTm	ISODateTime	Settlement date time
Settlement time request - CLS Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /CLSTm	ISOTime	Settlement time request – CLS time
Settlement time request - till Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /TillTm	ISOTime	Settlement time request – till time
Settlement time request - from Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /FrTm	ISOTime	Settlement time request – from time
Settlement time request - reject Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq	ISOTime	Settlement time request – reject time

Message item	Data type/code	Utilisation
/RjctTm		
Prev instructing agt1 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 1	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 1
Prev instructing agt1 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 1Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 2	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 2
Prev instructing agt2 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 2Acct	CashAccount24	Previous instructing agent 2 account
Prev instructing agt3 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 3	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 3
Prev instructing agt3 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 3Acct	CashAccount24	Previous instructing agent 3 account
Instructing agt Docu- ment/FICdtTrf/CdtTrfTxInf/InstgAgt/Finl nstnId/BICFI	BICFIIdentifier	Instructing agent
Instructed agt Docu- ment/FICdtTrf/CdtTrfTxInf/InstdAgt/Finl nstnId/BICFI	BICFIIdentifier	Instructed agent
Intermed. Agt1	BranchAndFinancialInstitutionIdentifi-	Intermediary agent 1

Message item	Data type/code	Utilisation
Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt1	cation5	
Intermed. Agt1 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt1A cct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt2	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 2
Intermed. Agt2 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt2A cct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 3
Intermed. Agt3 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3A cct	CashAccount24	Intermediary agent 3 account
Debtor Document/FICdtTrf/CdtTrfTxInf/Dbtr	PartyIdentification125	Debtor
Debtor account Docu- ment/FICdtTrf/CdtTrfTxInf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Document/FICdtTrf/CdtTrfTxInf/DbtrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Debtor agent
Debtor agt account Docu- ment/FICdtTrf/CdtTrfTxInf/DbtrAgtAcct	CashAccount24	Debtor agent account
Creditor agt Document/FICdtTrf/CdtTrfTxInf/CdtrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Creditor agent

Message item	Data type/code	Utilisation
Creditor agt account Docu- ment/FICdtTrf/CdtTrfTxInf/CdtrAgtAcct	CashAccount24	Creditor agent account
Creditor Document/FICdtTrf/CdtTrfTxInf/Cdtr	PartyIdentification125	Creditor
Creditor account Docu- ment/FICdtTrf/CdtTrfTxInf/CdtrAcct	CashAccount24	Creditor account
Remittance information Document/FICdtTrf/CdtTrfTxInf/RmtInf	RemittanceInformation2	Remittance information

Table 241 - FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) – usage case Settlement of ancillary system movement

Usage case example:
pacs.009.001.07_RTGS_FinancialInstitutionCreditTransferASMovement_Example.xml

Usage case: Liquidity transfer to sub-account

In this usage case, the message describes a payment movement instructed by an ancillary system.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
FinancialInstitutionCreditTransferV07 Document/FICdtTrf	FinancialInstitutionCreditTransferV07	
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/InstrId	Max35Text	Payment ID – instruction ID
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/EndT oEndId	Max35Text	Payment ID – end to end ID
Payment ID Docu-	Max35Text	Payment ID – transaction ID

Message item	Data type/code	Utilisation
ment/FICdtTrf/CdtTrfTxInf/PmtId/TxId		
Payment ID Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/ClrSy sRef	Max35Text	Payment ID – RTGS system reference
Payment service level code Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Sv cLvl/Cd	ExternalServiceLevel1Code	Service level code
Payment local instrument code Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Lcl Instrm/Cd	ExternalLocalInstrument1Code	Local instrument code
Payment category purpose code Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Ct gyPurp/Cd	ExternalCategoryPurpose1Code	Category purpose code
Amount Docu- ment/FICdtTrf/CdtTrfTxInf/IntrBkSttlmA mt	RTGS_Max14_Max2DecimalAmount	Payment amount
Date Docu- ment/FICdtTrf/CdtTrfTxInf/IntrBkSttlmD t	ISODate	Payment date
Settlement priority Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmPrty	Priority3Code	Settlement priority
Settlement date time Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmIndc tn/CrdDtTm	ISODateTime	Settlement date time
Settlement time request - CLS Docu-	ISOTime	Settlement time request – CLS time

Message item	Data type/code	Utilisation
ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /CLSTm		
Settlement time request - till Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /TillTm	ISOTime	Settlement time request – till time
Settlement time request - from Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /FrTm	ISOTime	Settlement time request – from time
Settlement time request - reject Docu- ment/FICdtTrf/CdtTrfTxInf/SttlmTmReq /RjctTm	ISOTime	Settlement time request – reject time
Prev instructing agt1 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 1	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 1
Prev instructing agt1 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 1Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 2	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 2
Prev instructing agt2 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 2Acct	CashAccount24	Previous instructing agent 2 account
Prev instructing agt3 Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 3	BranchAndFinancialInstitutionIdentifi- cation5	Previous instructing agent 3

Message item	Data type/code	Utilisation
Prev instructing agt3 account Docu- ment/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt 3Acct	CashAccount24	Previous instructing agent 3 account
Instructing agt Docu- ment/FICdtTrf/CdtTrfTxInf/InstgAgt/Finl nstnId/BICFI	BICFIIdentifier	Instructing agent
Instructed agt Docu- ment/FICdtTrf/CdtTrfTxInf/InstdAgt/Finl nstnId/BICFI	BICFIIdentifier	Instructed agent
Intermed. Agt1 Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt1	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 1
Intermed. Agt1 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt1A cct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt2	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 2
Intermed. Agt2 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt2A cct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3	BranchAndFinancialInstitutionIdentifi- cation5	Intermediary agent 3
Intermed. Agt3 account Docu- ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3A cct	CashAccount24	Intermediary agent 3 account
Debtor	PartyIdentification125	Debtor

Message item	Data type/code	Utilisation
Document/FICdtTrf/CdtTrfTxInf/Dbtr		
Debtor account Docu- ment/FICdtTrf/CdtTrfTxInf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Document/FICdtTrf/CdtTrfTxInf/DbtrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Debtor agent
Debtor agt account Docu- ment/FICdtTrf/CdtTrfTxInf/DbtrAgtAcct	CashAccount24	Debtor agent account
Creditor agt Document/FICdtTrf/CdtTrfTxInf/CdtrAgt	BranchAndFinancialInstitutionIdentifi- cation5	Creditor agent
Creditor agt account Docu- ment/FICdtTrf/CdtTrfTxInf/CdtrAgtAcct	CashAccount24	Creditor agent account
Creditor Document/FICdtTrf/CdtTrfTxInf/Cdtr	PartyIdentification125	Creditor
Creditor account Docu- ment/FICdtTrf/CdtTrfTxInf/CdtrAcct	CashAccount24	Creditor account
Remittance information Document/FICdtTrf/CdtTrfTxInf/RmtInf	RemittanceInformation2	Remittance information

Table 242 - FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) – usage case Liquidity transfer to sub-account

Usage **case** **example:**
pacs.009.001.07_RTGS_FinancialInstitutionCreditTransferLTtoSubaccount_Example.xml

14.5.5 FinancialInstitutionDirectDebit (pacs.010)

14.5.5.1 Overview and scope of the message

This chapter illustrates the *FinancialInstitutionDirectDebit* message.

The *FinancialInstitutionDirectDebit* message is sent by a RTGS Account Holder (or on their behalf by an authorised party) to the RTGS component. It is used to move an amount from the RTGS DCA of another RTGS Account Holder, to an RTGS DCA of the sending RTGS Account Holder.

The *FinancialInstitutionDirectDebit* message can also be sent by the RTGS component to a RTGS Account Holder (or to a party authorized by them). In this case, it is used to notify the recipient about a direct debit, initiated by their payment counterparty, that has been executed.

The *FinancialInstitutionDirectDebit* message concerns only one direct debit movement.

Within RTGS, the *FinancialInstitutionDirectDebit* message has the following usages:

- | Payment Message
- | Payment Settlement Notification

In response to the *FinancialInstitutionDirectDebit* message, a [PaymentStatusReport \(pacs.002\)](#) [▶ 568] message containing the status of the movement is returned to the sending party.

In addition, if the direct debit movement is successfully processed, the *FinancialInstitutionDirectDebit* message is forwarded to the debited RTGS Account Holder (or a party authorised by them).

14.5.5.2 Schema

Outline of the schema.

The *FinancialInstitutionDirectDebit* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

CreditInstruction

This building block is mandatory and non-repetitive. It holds the characteristics that apply to the credit side of the payment transaction included in the message. It contains the following elements:

- | credit identification
- | instructing and instructed agents
- | creditor party and account
- | creditor agent and account
- | debit information: payment identification, payment type, interbank settlement amount, interbank settlement date, settlement priority, settlement time request, debtor and account, debtor agent and account, remittance information

References/links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.010.001.02_RTGS

Business rules applicable to the schema

For business rules applicable to *FinancialInstitutionDirectDebit* please refer to the chapter [Index of business rules and error codes](#) [670]

14.5.5.3 The message in business context

Usage case: Payment Message

In this usage case, the message provides the details required for the RTGS component to execute a direct debit payment between two financial institutions. It is a pre-requisite that a valid direct debit agreement must exist between the two financial institutions.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Message Identification Document/FIDrctDbt/GrpHdr/MsgId	Max35Text	Message ID of the payment
Creation Date Time Document/FIDrctDbt/GrpHdr/CreDtTm	ISODateTime	Date and time at which the message was created.
Number Of Transactions Document/FIDrctDbt/GrpHdr/NbOfTxs	Max15NumericText_fixed	1
Credit Instruction Document/FIDrctDbt/CdtInstr/CdtId	Max35Text	Customers ID
Instructing Agent Document/FIDrctDbt/CdtInstr/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	BIC of the instructing agent
Instructed Agent Docu-	BICFIIdentifier	BIC of the instructed agent

Message item	Data type/code	Utilisation
ment/FIDrctDbt/CdtInstr/InstdAgt/FinInstnId/BICFI		
Creditor Docu- ment/FIDrctDbt/CdtInstr/Cdtr/FinInstnId/BICFI	BICFIIdentifier	Financial institution servicing an account for the creditor.
Instruction Identification Document/DrctDbtTxInf/PmtId/InstrId	Max35Text	ID set by the instructing agent
End To End Identification Docu- ment/DrctDbtTxInf/PmtId/EndToEndId	Max35Text	ID set by the instructing agent. This identification is passed on, unchanged, throughout the entire end-to-end chain.
Transaction Identification Document/DrctDbtTxInf/PmtId/TxId	Max35Text	ID assigned by the first instructing agent, to unambiguously identify the transaction.
Interbank Settlement Amount Document/DrctDbtTxInf IntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Amount of money moved between the instructing agent and the instructed agent.
Interbank Settlement Date Document/DrctDbtTxInfIntrBkSttlmDt	ISODate	Date on which the amount becomes available for the creditor.
Settlement Priority Document/DrctDbtTxInf SttlmPrty	Priority3Code	Urgent(URGT) High(HIGH) Normal(NORM)
Debitor Docu- ment/FIDrctDbt/CdtInstr/Dbtr/FinInstnId/BICFI	BICFIIdentifier	Financial institution that owes an amount of money to the (ultimate) financial institutional creditor.

Table 243 - FinancialInstitutionDirectDebit (pacs.010) – usage case Payment Message

Usage case example: pacs.010_RTGS_FinancialInstitutionDirectDebit_PaymentOrder_Example.xml

Usage case: Payment Settlement Notification

In this usage case, the message is outbound from the RTGS component. It identifies to the recipient a payment which has been debited against one of its RTGS DCAs, under a valid pre-existing direct debit agree-

ment. The payment would have been ordered by the credited payment party (or on their behalf) using this same *FinancialInstitutionDirectDebit* message.

Specific message content

Message item	Data type/code	Utilisation
Message Identification Document/FIDrctDbt/GrpHdr/Msgld	Max35Text	Message ID of the paymentorder set by the instructing agent of the payment
Creation Date Time Document/FIDrctDbt/GrpHdr/CreDtTm	ISODateTime	Date and time at which the message was created.
Number Of Transactions Document/FIDrctDbt/GrpHdr/NbOfTx	Max15NumericText_fixed	1
Credit Instruction Document/FIDrctDbt/CdtInstr/CdtId	Max35Text	Customers ID
Instructing Agent Docu- ment/FIDrctDbt/CdtInstr/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	BIC of the instructing agent
Instructed Agent Docu- ment/FIDrctDbt/CdtInstr/InstdAgt/FinInstnId/BICFI	BICFIIdentifier	BIC of the instructed agent
Creditor Agent Document/FIDrctDbt/ CdtrAgt/ Fin-InstnId/ BICFI	BICFIIdentifier	BIC of Creditor
ID Docu- ment/FIDrctDbt/CdtrAgtAcct/Id/Othr/Id	Max34Text	RTGS DCA
Currency Docu- ment/FIDrctDbt/CdtrAgtAcct/Id/Othr/Ccy	ActiveOrHistoricCurrencyCode	EUR
Name Docu- ment/FIDrctDbt/CdtrAgtAcct/Id/Othr/N	Max70Text	Name of the account

Message item	Data type/code	Utilisation
m		
Creditor Docu- ment/FIDrctDbt/CdtrInstr/Cdtr/FinInstnId /BICFI	BICFIIdentifier	Financial institution servicing an ac- count for the creditor.
ID Docu- ment/FIDrctDbt/CdtrAcct/Id/Othr/Id	Max34Text	RTGS DCA
Currency Docu- ment/FIDrctDbt/CdtrAcct/Id/Othr/Ccy	ActiveOrHistoricCurrencyCode	EUR
Name Docu- ment/FIDrctDbt/CdtrAcct/Id/Othr/Nm	Max70Text	Name of the account
Instruction ation Document/DrctDbtTxInf/PmtId/InstrId	Max35Text	ID set by the instructing agent
End To End Identification Docu- ment/DrctDbtTxInf/PmtId/EndToEndId	Max35Text	ID set by the instructing agent. This identification is passed on, unchanged, throughout the entire end-to-end chain.
Transaction Identification Document/DrctDbtTxInf/PmtId/TxId	Max35Text	ID assigned by the first instructing agent, to unambiguously identify the transaction.
Interbank Settlement Amount Document/DrctDbtTxInf IntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Amount of money moved between the instructing agent and the instructed agent.
Interbank Settlement Date Document/DrctDbtTxInfIntrBkSttlmDt	ISODate	Date on which the amount becomes available for the creditor.
Settlement Priority Document/DrctDbtTxInf SttlmPrty	Priority3Code	Urgent(URGT) High(HIGH) Normal(NORM)
Debitor	BICFIIdentifier	Financial institution that owes an amount of money to the (ultimate)

Message item	Data type/code	Utilisation
Docu- ment/FIDrctDbt/CdtInstr/Dbtr/FinInstnld /BICFI		financial institutional creditor.
ID Docu- ment/FIDrctDbt/DebtAcct/Id/Othr/Id	Max34Text	RTGS DCA
Currency Docu- ment/FIDrctDbt/DebtAcct/Id/Othr/Ccy	ActiveOrHistoricCurrencyCode	EUR
Name Docu- ment/FIDrctDbt/DebtAcct/Id/Othr/Nm	Max70Text	Name of the account
ID Docu- ment/FIDrctDbt/DebtAgtAcct/Id/Othr/Id	Max34Text	RTGS DCA
Currency Docu- ment/FIDrctDbt/DebtAgtAcct/Id/Othr/Cc y	ActiveOrHistoricCurrencyCode	EUR
Name Docu- ment/FIDrctDbt/DebtAgtAcct/Id/Othr/N m	Max70Text	Name of the account
Unstructured Document/FIDrctDbt/ RmtInf/Ustrd	Max140Text	For matching of an entry

Table 244 - FinancialInstitutionDirectDebit (pacs.010) – usage case Payment Settlement Notification

Usage **case** **example:**
pacs.010_RTGS_FinancialInstitutionDirectDebit_PaymentOrderSettlementNotification_Example.xml

14.6 Payment initiation (pain)

14.6.1 ATransferNotice (pain.998)

14.6.1.1 Overview and scope of the message

This chapter illustrates the *ProprietaryMessage ATransferNotice* message.

The *ProprietaryMessage* is an ISO 20022 wrapper that can be used to transport a proprietary format of message data. In this instance it is used to transport a proprietary T2 format *ATransferNotice* message.

The *ATransferNotice* message is sent from the RTGS component to an ancillary system. It is used to notify the ancillary system of the settlement of credit amount on a RTGS DCA owned by the ancillary system.

Within RTGS, the *ATransferNotice* message has the following usages:

- | Procedure D
- | Notify ancillary system of Credit Liquidity Bookings
- | Notify ancillary system of Immediate Liquidity Transfer Order Issued by Settlement Bank
- | Notify ancillary system of Execution of Stored Liquidity Transfer Orders

The *ATransferNotice* is sent as a result of processing within the RTGS component.

14.6.1.2 Schema

Outline of the schema.

The *ProprietaryMessage* message is composed of the following message building blocks:

ProprietaryData

Type of the proprietary message and the actual *ATransferNotice* message itself.

References/Links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pain.998.001.01_RTGS_ATransferNotice

Business rules applicable to the schema

No business rules are applicable to a *ASTransferNotice* message.

14.6.1.3 The message in business context

Usage case: Procedure D – Notify ancillary system of Credit Liquidity Bookings

In this usage case, the RTGS component is advising the ancillary system of the actual credit bookings resulting from the execution of standing liquidity transfers defined for procedure D, at start of day.

Specific message content

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/Tp	Max35Text	Always "ASTransferNotice"
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/GrpId	Max35text	Group identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CreDtTm	ISODateTime	Notification creation time
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CtrlSum	DecimalNumber	Control total of payments
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/NbOfTxs	Max15NumericText	Number of transactions reported
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/Prty	PriorityCode_EMIP1	Notification priority
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/SttlmMdlTp	SettlementModelType_EMIP1	Settlement model
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CtpAS/Fl/BIC	PartyIdentification2_EMIP1	Ancillary system counterpart identifica- tion
Payment information		

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/ReqdExctnDt	ISODate	Request execution date
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/CdtTrfTpId/StlmPrty/PmtSch me/Cd	CashClearingSystemCode_EMIP2	Payment scheme code SET
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/Dbtr/Nm	Max70Text	Debtor name
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/Dbtr/FI/BIC	BICIdentifier	Debtor financial institution BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/DbtrAcct/DmstAcct/Id	Max35Text	Debtor account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/FirstAgt/BIC	BICIdentifier	First agent BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/FirstAgtAcct/DmstAcct/Id	Max35Text	First agent account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/PmtId/InstrId	Max16Text	Payment identification – instruction identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/PmtId/EndToEndId	Max16Text	Payment identification – end-to-end identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Amt/InstAmt	ActiveCurrencyAndAmount_EMIP1	Instructed amount
Docu-	ActiveCurrencyCode	Instructed amount currency

Message item	Data type/code	Utilisation
ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/Amt/InstAmt/Ccy		
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Cdtr/Nm	Max70text	Creditor name
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Cdtr/Fl/BIC	BICIdentifier	Creditor BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/CdtrAcct/DmstAcct/Id	Max35Text	Creditor account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/FnlAgt/BIC	BICIdentifier	Final agent BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/FnlAgtAcct/DmstAcct/I d	Max35Text	Final agent account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/RmtInf/Ustrd	Max140Text	Remittance information
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/Amt	ActiveCurrencyAndAmount_EMIP1	Resulting balance
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/Amt/Ccy	ActiveCurrencyCode	Resulting balance currency
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/CdtDbtInd	CreditDebitCode_EMIP1	Resulting debit/credit indicator
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/ValDt/DtTm	ISODatetime	Resulting timestamp

Table 245 - ATransferNotice (pain.998) – usage case Procedure D – Notify ancillary system of Credit Liquidity Bookings

Usage case example:
pain.998_RTGS_ASInitiationNotice_ProcDNotifyASOfCreditLiquidityBookings_Example.xml

In the example, the final agent is the ancillary system. The payment scheme code used is “SET” (TBC).

Usage case: Procedure D – Notify ancillary system of Immediate Liquidity Transfer Order Issued by Settlement Bank

In this usage case, the RTGS component is advising the ancillary system that a liquidity adjustment (instructed by the settlement bank using a [LiquidityCreditTransfer \(camt.050\)](#) [▶ 497] message or the GUI) has successfully been processed during an execution of ancillary system Settlement Procedure D.

Specific message content

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/Tp	Max35Text	Always "ATransferNotice"
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/GrpId	Max35text	Group identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CreDtTm	ISODateTime	Notification creation time
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CtrlSum	DecimalNumber	Control total of payments
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/NbOfTxs	Max15NumericText	Number of transactions reported
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/Prty	PriorityCode_EMIP1	Notification priority
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/StlmMdlTp	SettlementModelType_EMIP1	Settlement model
Docu- ment/pain.998.001.01/PrtryData/EMIP	PartyIdentification2_EMIP1	Ancillary system counterpart identifica- tion

Message item	Data type/code	Utilisation
PrtryData/GrpHdr/CtpAS/FI/BIC		
Payment information		
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/ReqdExctnDt	ISODate	Request execution date
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/CdtTrfTpId/StlmPrty/PmtSch me/Cd	CashClearingSystemCode_EMIP2	Payment scheme code CUO
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/Dbtr/Nm	Max70Text	Debtor name
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/Dbtr/FI/BIC	BICIdentifier	Debtor financial institution BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/DbtrAcct/DmstAcct/Id	Max35Text	Debtor account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/FrstAgt/BIC	BICIdentifier	First agent BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/FrstAgtAcct/DmstAcct/Id	Max35Text	First agent account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/PmtId/InstrId	Max16Text	Payment identification – instruction identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/PmtId/EndToEndId	Max16Text	Payment identification – end-to-end identification
Docu- ment/pain.998.001.01/PrtryData/EMIP	ActiveCurrencyAndAmount_EMIP1	Instructed amount

Message item	Data type/code	Utilisation
PrtryData/PmtInf/PmtTx/Amt/InstAmt		
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/Amt/InstAmt/Ccy	ActiveCurrencyCode	Instructed amount currency
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Cdtr/Nm	Max70text	Creditor name
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Cdtr/Fl/BIC	BICIdentifier	Creditor BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/CdtrAcct/DmstAcct/Id	Max35Text	Creditor account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/FnlAgt/BIC	BICIdentifier	Final agent BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/FnlAgtAcct/DmstAcct/Id	Max35Text	Final agent account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/RmtInf/Ustrd	Max140Text	Remittance information
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/Amt	ActiveCurrencyAndAmount_EMIP1	Resulting balance

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/Amt/Ccy	ActiveCurrencyCode	Resulting balance currency
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/CdtDbtInd	CreditDebitCode_EMIP1	Resulting debit/credit indicator
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/ValDt/DtTm	ISODateTime	Resulting timestamp

Table 246 - ASTransferNotice (pain.998) – usage case Procedure D – Notify ancillary system of Immediate Liquidity Transfer Order Issued by Settlement Bank

Usage case example:
pain.998_RTGS_ASInitiationNotice_ProcDNotifyASOfILTOIssuedBySettlementBank_Example.xml

In the example, the first agent is the settlement bank. The final agent is the ancillary system. The payment scheme code used is "CUO".

Usage case: Procedure D – Notify ancillary system of Execution of Pending Liquidity Transfer Orders

In this usage case, the RTGS component is advising the ancillary system of the actual credit bookings resulting from the execution of pending liquidity transfers remaining at the end of ancillary system Settlement Procedure D.

Specific message content

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/Tp	Max35Text	Always "ASTransferNotice"
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/GrpId	Max35text	Group identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CreDtTm	ISODateTime	Notification creation time
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CtrlSum	DecimalNumber	Control total of payments

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/NbOfTxs	Max15NumericText	Number of transactions reported
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/Prty	PriorityCode_EMIP1	Notification priority
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/SttlmMdlTp	SettlementModelType_EMIP1	Settlement model
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CtpAS/Fl/BIC	PartyIdentification2_EMIP1	Ancillary system counterpart identifica- tion
Payment information		
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/ReqdExctnDt	ISODate	Request execution date
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/CdtTrfTpId/SttlmPrty/PmtSch me/Cd	CashClearingSystemCode_EMIP2	Payment scheme code SOR
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/Dbtr/Nm	Max70Text	Debtor name
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/Dbtr/Fl/BIC	BICIdentifier	Debtor financial institution BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/DbtrAcct/DmstAcct/Id	Max35Text	Debtor account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/FrstAgt/BIC	BICIdentifier	First agent BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP	Max35Text	First agent account

Message item	Data type/code	Utilisation
PrtryData/PmtInf/FrstAgtAcct/DmstAcct/Id		
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/PmtId/InstrId	Max16Text	Payment identification – instruction identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/PmtId/EndToEndId	Max16Text	Payment identification – end-to-end identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Amt/InstAmt	ActiveCurrencyAndAmount_EMIP1	Instructed amount
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/Amt/InstAmt/Ccy	ActiveCurrencyCode	Instructed amount currency
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Cdtr/Nm	Max70text	Creditor name
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Cdtr/Fl/BIC	BICIdentifier	Creditor BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/CdtrAcct/DmstAcct/Id	Max35Text	Creditor account
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/FnlAgt/BIC	BICIdentifier	Final agent BIC
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/FnlAgtAcct/DmstAcct/I d	Max35Text	Final agent account
Docu- ment/pain.998.001.01/PrtryData/EMIP	Max140Text	Remittance information

Message item	Data type/code	Utilisation
PrtryData/PmtInf/PmtTx/RmtInf/Ustrd		
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/Amt	ActiveCurrencyAndAmount_EMIP1	Resulting balance
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/Amt/Ccy	ActiveCurrencyCode	Resulting balance currency
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/CdtDbtInd	CreditDebitCode_EMIP1	Resulting debit/credit indicator
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/RsltGBal/ValDt/DtTm	ISODateTime	Resulting timestamp

Table 247 - ASTransferNotice (pain.998) – usage case Procedure D – Notify ancillary system of Execution of Pending Liquidity Transfer Orders

Usage case example:
pain.998_RTGS_ASInitiationNotice_ProcDNotifyASOfExecutionOfPendingLTOs_Example.xml

In the example, the first agent is the settlement bank which account is debited while the final agent’s account (ancillary system account) is credited. The creditor is the final beneficiary to be credited. The payment scheme code used is “SOR” (TBC).

14.6.2 ASInitiationStatus (pain.998)

14.6.2.1 Overview and scope of the message

This chapter illustrates the *ProprietaryMessage ASInitiationStatus* message.

The *ProprietaryMessage* is an ISO 20022 wrapper that can be used to transport a proprietary format of message data. In this instance it is used to transport a proprietary T2 format *ASInitiationStatus* message.

The *ASInitiationStatus* message is sent from the RTGS component to an ancillary system. It is used to notify the ancillary system of the status of a payment previously initiated by the ancillary system.

Within RTGS, the *ASInitiationStatus* message has the following usages:

Note: due to the large number of usage cases and the similarities of *ASInitiationStatus* for these usages, this section is dealt with by organising the usage cases into a single usage category.

I Usage category – settlement

- Rejection of Ancillary System Transfer Initiation
- Procedure A - Notify Ancillary System of Successful Settlement
- Procedure B - Notify Ancillary System of Successful Settlement at First Attempt
- Procedure B - Notify Ancillary System of Successful Settlement using Recurrent Optimisation
- Procedure A/B - Notify Ancillary System about Rejection due to Disagreement
- Procedure A/B - Notify Ancillary System Regarding Revocation of Complete Ancillary System Batch Message
- Procedure A/B - Notify Ancillary System of Rejection at End of Settlement Period
- Procedure A/B - Notify Ancillary System of Decision Required to Invoke Guarantee Process
- Procedure A/B - Notify Ancillary System of Successful or Failed Settlement after Guarantee Process
- Procedure C - Notify Ancillary System of Immediate Liquidity Order Issued by Ancillary System
- Procedure C - Notify Ancillary System of Successful Settlement (Procedure C)
- Procedure C - Notify Ancillary System with Individual Status of Each Transaction
- Procedure D - Notify Ancillary System of Immediate Liquidity Order Issued by Ancillary System

The *ASInitiationStatus* message is sent in response to a previously received [ASTransferInitiation \(pain.998\)](#) [▶ 633] message.

14.6.2.2 Schema

Outline of the schema.

The *ProprietaryMessage* message is composed of the following message building blocks:

ProprietaryData

Type of the proprietary message and the actual *ASInitiationStatus* message itself.

References/Links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pain.998.001.01_RTGS_ASInitiationStatus

Business rules applicable to the schema

No business rules are applicable to a *ASInitiationStatus* message.

14.6.2.3 The message in business context

Usage Category – Settlement

All usage cases in this category will see a similar set of possible *ASInitiationStatus* responses. For this reason, the usage category is described below and is relevant to all usage cases in this category.

Usage Category case – Settlement - Settled

In this usage case, the RTGS component is advising the ancillary system that a previously received [ASTransferInitiation \(pain.998\)](#) [▶ 633] message has successfully been settled.

Specific message content

Message item	Data type/code	Utilisation
Proprietary Data Type Docu- ment/ASInitiationStatus/PrtryData/Tp	Max35Text	Always : ASInitiationStatus
EMIP proprietary message information		
payment initiation status identification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/GnlInf/Pmtlni tnStsId	Max35Text	Payment initiation status identification
Creation Date Time Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/GnlInf/CreDt Tm	ISODateTime	Creation date/time
Group Identification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/GrpId	Max35Text	Reference assigned by the sender to identify the group of individual transfers being sent Filled with the field “GroupIdentifica- tion” of the original message.

Message item	Data type/code	Utilisation
Original Message Type Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/OrgnlMsgTp	OriginalMessageType_EMIP1	Always: ASTransferInitiation
Settlement Model Type Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/StlmMdlTp	SettlementModelType_EMIP1	Specifies the generic settlement procedure 1000 = Procedure 1 (Liquidity transfer) 2000 = Procedure 2 (Real time settlement) 3000 = Procedure 3 (Bilateral settlement) 4000 = Procedure 4 (Standard multilateral settlement) 5000 = Procedure 5 (Simultaneous multilateral settlement) 6000 = Procedure 6 (Settlement on dedicated liquidity) It is identical to the model of payment indicated in the original message
Group Status Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/GrpSts	PaymentGroupStatusCodeT_EMIP1	Status information concerning the group of payment transactions included in the original message. ACSC = AcceptedSettlementCompleted Settlement on the debtor's account has been completed for all the transactions in the file. There is no list of single positions because they are all settled. INVL = Invalid The input file cannot be processed because of error PART = PartiallyAccepted A number of transactions have been accepted, whereas another number of

Message item	Data type/code	Utilisation
		<p>transactions have not achieved "settled" status. The status of each transaction is indicated at payment level in the list of single positions. 'PART' is also used if no transactions have been executed.</p> <p>In case of model 3 with single notification the group status will be always PART when the file is not globally rejected.</p> <p>REVR = Reversed</p> <p>The file which was previously "PartiallyAccepted" is now totally rejected after a reversing procedure of the transactions which were settled</p> <p>RJCT = Rejected</p> <p>Payment initiation or individual transaction included in the payment initiation has been rejected or revoked.</p> <p>The whole file is rejected</p> <p>RJDA = RejectedDisagreement</p> <p>Disagreement of the CB in case of transactions relative to excluded settlement bank or excluded ancillary system. The whole file is rejected</p>

Table 248 - ASInitiationStatus (pain.998) – usage category case Settlement - Settled

Usage case example: pain.998_RTGS_ASInitiationStatus_SettlementSettled_Example.xml

In this usage case, the RTGS component is advising the ancillary system that a previously received [ASTransferInitiation \(pain.998\)](#) [▶ 633] message has successfully been settled.

Usage Category case – Settlement - Rejected

In this usage case, the RTGS component is advising the ancillary system that a previously received [ASTransferInitiation \(pain.998\)](#) [▶ 633] message has either failed business validation or encountered a processing problem, and will not be processed further.

Specific message content

Message item	Data type/code	Utilisation
Proprietary Data Type Docu- ment/ASInitiationStatus/PrtryData/Tp	Max35Text	Always : ASInitiationStatus
EMIP proprietary message information		
payment initiation status identification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/GnlInf/Pmtlni tnStsId	Max35Text	Payment initiation status identification
Creation Date Time Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/GnlInf/CreDt Tm	ISODateTime	Creation date/time
Group Identification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/GrpId	Max35Text	Original group id
Original Message Type Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/OrgnlMsgTp	OriginalMessageType_EMIP1	Always: ASTransferInitiation
Settlement Model Type Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/StlmMdlTp	SettlementModelType_EMIP1	Specifies the generic settlement Pro- cedure 1000 = Procedure 1 (Liquidity transfer) 2000 = Procedure 2 (Real time settle- ment 3000 = Procedure 3 (Bilateral settle- ment) 4000 = Procedure 4 (Standard multilat- eral settlement) 5000 = Procedure 5 (Simultaneous multilateral settlement) 6000 = Procedure 6 (Settlement on dedicated liquidity It is identical to the model of payment

Message item	Data type/code	Utilisation
		indicated in the original message
Group Status Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/GrpSts	PaymentGroupStatusCodeT_EMIP1	<p>Status information concerning the group of payment transactions included in the original message.</p> <p>ACSC = AcceptedSettlementCompleted</p> <p>Settlement on the debtor's account has been completed for all the transactions in the file. There is no list of single positions because they are all settled.</p> <p>INVL = Invalid</p> <p>The input file cannot be processed because of error</p> <p>PART = PartiallyAccepted</p> <p>A number of transactions have been accepted, whereas another number of transactions have not achieved "settled" status. The status of each transaction is indicated at payment level in the list of single positions.</p> <p>'PART' is also used if no transactions have been executed.</p> <p>In case of model 3 with single notification the group status will be always PART when the file is not globally rejected.</p> <p>REVR = Reversed</p> <p>The file which was previously "PartiallyAccepted" is now totally rejected after a reversing procedure of the transactions which were settled</p> <p>RJCT = Rejected</p> <p>Payment initiation or individual transaction included in the payment initiation has been rejected or revoked.</p>

Message item	Data type/code	Utilisation
		<p>The whole file is rejected</p> <p>RJDA = RejectedDisagreement</p> <p>Disagreement of the CB in case of transactions relative to excluded settlement bank or excluded ancillary system. The whole file is rejected</p>
<p>Bilaterally Agreed</p> <p>Document/ ASInitiationStatus</p> <p>/PrtryData/EMIPPrtryData/OrgnlGrpRef</p> <p>InfAndSts/StsRsn/BilyAgrd</p>	Max4AlphaNumericText_EMIP1	<p>-Error codes specified by ASI (A0xx) in case of invalid input message</p> <p>-In case of reject of a file after Reverse: "GANR" if the ancillary system decision to use the guarantee account was negative, "GALL" if there is a lack of liquidity on the guarantee account - Failure reason in case of complete rejection of a file:</p> <p>RVOK = Revoke. The file has been revoked</p> <p>RJSP = RejectedSettlementPeriod. The file is rejected because the Settlement period time is reached.</p> <p>RDIB = File rejected at cutoff time due to insufficiency balance in the account to be decreased</p> <p>EXAS = File rejected for AS exclusion</p> <p>EXSB = File rejected because it contains a transaction relative to an excluded settlement bank</p> <p>DPNS = Daylight settlement period has not started</p>
<p>Instruction Identification</p> <p>Document/ ASInitiationStatus</p> <p>/PrtryData/EMIPPrtryData/OrgnlPmtInf/</p> <p>OrgnlTxRefInfAndSts/PmtId/InstrId</p>	Max16Text	<p>Unique and unambiguous identifier for a payment instruction assigned by the initiating party</p> <p>=Identification indicated in the original message</p>

Message item	Data type/code	Utilisation
EndToEndIdentification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnPmtInf/ OrgnITxRefInfAnd- Sts/PmtId/EndToEndId	Max16Text	Unique and unambiguous identification of a payment transaction, as assigned by any of the parties on the initiating side, that will be passed on throughout the entire end-to-end chain
/ TransactionStatus Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnPmtInf/ OrgnITxRefInfAndSts/TxSts	PaymentTransactionStatus- CodeT_EMIP1	Status of a transaction included in the original message ACSC = Accepted-SettlementCompleted Settlement on the debtor's account has been completed. INVL = Invalid The input payment cannot be processed because of error COPS = CurrentOrderPartiallySettled Current order from ancillary system has been partially settled REVR = Reversed Reject of a transaction which was previously settled RJCT = Rejected Payment initiation or individual transaction included in the payment initiation has been rejected or revoked. RJDA = RejectedDisagreement Disagreement of the CB in case of transactions relative to excluded settlement bank or excluded ancillary system. The transaction is rejected.
BilaterallyAgreed Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnPmtInf/ OrgnITxRefInfAndSts/StsRsn/BilyAgrd	Max4AlphaNumericText_EMIP1	-Error code specified by ASI (A0xx) in case of invalid payment -Failure reason in case of payment rejected RVOK = Revoke The payment has been revoked RJSP = RejectedSettlementPeriod.

Message item	Data type/code	Utilisation
		<p>The payment is rejected because the Settlement period time is reached.</p> <p>DPNS = DaySettlementPeriodNot-Started. Daylight settlement period has not started</p> <p>RDIB = RejectedDecreaseInsufficient-Balance. Decrease rejected due to insufficient balance</p> <p>GENE = Generic Error</p> <p>EXSB = Exclusion Settlement Bank</p>

Table 249 - ASInitiationStatus (pain.998) – usage category case Settlement - Rejected

Usage case example: pain.998_RTGS_ASInitiationStatus_SettlementRejected_Example.xml

In this usage case, the RTGS component is advising the ancillary system that a previously received [ASTransferInitiation \(pain.998\)](#) [▶ 633] message has either failed business validation or encountered a processing problem, and will not be processed further.

Usage Category case – Settlement - Partial

In this usage case, the RTGS component is advising the ancillary system that a previously received [ASTransferInitiation \(pain.998\)](#) [▶ 633] message has only been partially settled, and the remainder is waiting for a future opportunity to settle.

Specific message content

Message item	Data type/code	Utilisation
Proprietary Data Type Docu- ment/ASInitiationStatus/PrtryData/Tp	Max35Text	Always: ASInitiationStatus
EMIP proprietary message information		
payment initiation status identification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/Gnllnf/Pmtlni tnStsld	Max35Text	payment initiation status identification
Creation Date Time Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/Gnllnf/CreDt	ISODateTime	creation date/time timestamp of status

Message item	Data type/code	Utilisation
Tm		
Group Identification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/GrpId	Max35Text	Reference assigned by the sender to identify the group of individual transfers being sent [1..1] Mandatory validation: Filled with the field "GroupIdentification" of the original message.
Original Message Type Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/OrgnlMsgTp	OriginalMessageType_EMIP1	Always: ASTransferInitiation
Settlement Model Type Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/StlmMdlTp	SettlementModelType_EMIP1	Specifies the generic settlement Procedure 1000 = Procedure 1 (Liquidity transfer) 2000 = Procedure 2 (Real time settlement) 3000 = Procedure 3 (Bilateral settlement) 4000 = Procedure 4 (Standard multilateral settlement) 5000 = Procedure 5 (Simultaneous multilateral settlement) 6000 = Procedure 6 (Settlement on dedicated liquidity) It is identical to the model of payment indicated in the original message
Group Status Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlGrpRef InfAndSts/GrpSts	PaymentGroupStatusCodeT_EMIP1	Status information concerning the group of payment transactions included in the original message. ACSC = AcceptedSettlementCompleted Settlement on the debtor's account has been completed for all the transactions

Message item	Data type/code	Utilisation
		<p>in the file. There is no list of single positions because they are all settled.</p> <p>INVL = Invalid</p> <p>The input file cannot be processed because of error</p> <p>PART = PartiallyAccepted</p> <p>A number of transactions have been accepted, whereas another number of transactions have not achieved "settled" status. The status of each transaction is indicated at payment level in the list of single positions.</p> <p>'PART' is also used if no transactions have been executed.</p> <p>In case of model 3 with single notification the group status will be always PART when the file is not globally rejected.</p> <p>REVR = Reversed</p> <p>The file which was previously "PartiallyAccepted" is now totally rejected after a reversing procedure of the transactions which were settled</p> <p>RJCT = Rejected</p> <p>Payment initiation or individual transaction included in the payment initiation has been rejected or revoked.</p> <p>The whole file is rejected</p> <p>RJDA = RejectedDisagreement</p> <p>Disagreement of the CB in case of transactions relative to excluded Settlement Bank or excluded ancillary system. The whole file is rejected</p>

Message item	Data type/code	Utilisation
InstructionIdentification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlPmtInf/ OrgnlTxRefInfAndSts/PmtId/InstrId	Max16Text	Unique and unambiguous identifier for a payment instruction assigned by the initiating party =Identification indicated in the original message
EndToEndIdentification Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlPmtInf/ OrgnlTxRefInfAnd- Sts/PmtId/EndToEndId	Max16Text	Unique and unambiguous identification of a payment transaction, as assigned by any of the parties on the initiating side, that will be passed on throughout the entire end-to-end chain
TransactionStatus Document/ ASInitiationStatus /PrtryData/EMIPPrtryData/OrgnlPmtInf/ OrgnlTxRefInfAndSts/TxSts	PaymentTransactionStatus- CodeT_EMIP1	<p>Status of a transaction included in the original message ACSC = Accepted-SettlementCompleted</p> <p>Settlement on the debtor's account has been completed.</p> <p>INVL = Invalid</p> <p>The input payment cannot be processed because of error</p> <p>COPS = CurrentOrderPartiallySettled</p> <p>Current order from ancillary system has been partially settled</p> <p>REVR = Reversed</p> <p>Reject of a transaction which was previously settled</p> <p>RJCT = Rejected</p> <p>Payment initiation or individual transaction included in the payment initiation has been rejected or revoked.</p> <p>RJDA = RejectedDisagreement</p> <p>Disagreement of the CB in case of transactions relative to excluded Settlement Bank or excluded ancillary system. The transaction is rejected..</p>

Table 250 - ASInitiationStatus (pain.998) – usage category case Settlement - Partial

Usage case example: pain.998_RTGS_ASInitiationStatus_SettlementPartial_Example.xml

In this usage case, the RTGS component is advising the ancillary system that a previously received [ASTransferInitiation \(pain.998\)](#) [633] message has only been partially settled, and the remainder is waiting for a future opportunity to settle.

14.6.3 ASTransferInitiation (pain.998)

14.6.3.1 Overview and scope of the message

This chapter illustrates the *ProprietaryMessage ASTransferInitiation* message.

The *ProprietaryMessage* is an ISO 20022 wrapper that can be used to transport a proprietary format of message data. In this instance it is used to transport a proprietary T2 format *ASTransferInitiation* message.

The *ASTransferInitiation* message is sent from an ancillary system to the RTGS component. It is used to instruct a payment order to be executed on the RTGS component system.

The *ASTransferInitiation* message may contain several *PaymentInformation* blocks. Each *PaymentInformation* block may only contain one payment.

Within RTGS, the *ASTransferInitiation* message has the following usages:

- | Send Ancillary System Transfer Initiation
- | Procedure C
- | Liquidity Adjustment
- | Instruct the Settlement Transactions
- | Procedure D
- | Liquidity Adjustment

The *ASTransferInitiation* message is sent as the result of a processing requirement from within the ancillary system.

14.6.3.2 Schema

Outline of the schema.

The *ProprietaryMessage* message is composed of the following message building blocks:

ProprietaryData

Type of the proprietary message and the actual *ASTransferInitiation* message itself.

References/Links

The RTGS-specific schema and documentation in XSD/EXCEL/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pain.998.001.01_RTGS_ASTransferInitiation

Business rules applicable to the schema

For business rules applicable to *ASTransferInitiation* please refer to the chapter [Index of business rules and error codes](#) [▶ 670].

14.6.3.3 The message in business context

Usage case: Send Ancillary System Transfer Initiation

In this usage case, the ancillary system is instructing the RTGS component to execute payments from RTGS DCAs owned or managed by the ancillary system. These payments are in response to operational requirements within the ancillary system's own processing. The various combinations of field usage dictate whether the information period is to be used and which Settlement Procedure.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Proprietary Data Type Docu- ment/pain.998.001.01/PrtryData/Tp	Max35Text	always ASTransferInitiation
Group Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/GrpId	Max35Text	Reference assigned by the sender to identify the group of individual transfers being sent
Creation Date Time Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CreDtTm	RestrictedISODateTime	Date and time at which the transfer initiation was created
Control Sum Docu- ment/pain.998.001.01/PrtryData/EMIP	ControlSum_ASI_EMIP1	Total of all the individual instructed amounts

Message item	Data type/code	Utilisation
PrtryData/GrpHdr/CtrlSum		
Number of Transactions Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/NbOfTxs	Max15NumericText	Number of individual transactions contained in the message, i.e. the number of occurrences of PaymentTransaction
Priority Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/Prty	PriorityCode_EMIP1	Priority of all the payments at group level
Settlement Model Type Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/StlmMdlTp	SettlementModelType_EMIP1	Specifies the generic settlement Procedure 1000 = Procedure 1 (Liquidity transfer) 2000 = Procedure 2 (Real time settlement) 3000 = Procedure 3 (Bilateral settlement) 4000 = Procedure 4 (Standard multilateral settlement) 5000 = Procedure 5 (Simultaneous multilateral settlement) 6000 = Procedure 6 (Settlement on dedicated liquidity)
Initiating Party BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/InitgPty/FI/BIC	BICIdentifier	The optional sequence will be filled in only in case a CB or SSP operator is sending the message on behalf of an ancillary system
Requested Execution Date Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/ReqdExctnDt	RestrictedISODate_EMIP1	Date at which the initiating party requests that the payment instruction be processed (=settlement date) Must be the current business day
Debtor Name Docu- ment/pain.998.001.01/PrtryData/EMIP	Max62FINCharacterText_EMIP1	Name of the debtor

Message item	Data type/code	Utilisation
PrtryData/PmtInf/Dbtr/Nm		
Debtor BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/Dbtr/FI/BIC	BICIdentifier	BIC of the ordering institution
Debtor Account Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/DbtrAcct/DmstAcct/Id	Max35FINCharacterText_EMIP1	Simple identification information
First Agent BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/FrstAgt/BIC	BICIdentifier	First Agent identification
Instruction Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/PmtId/InstrId	RestrictedFINXMax16Text	Unique and unambiguous identifier for a payment instruction assigned by the initiating party
End to End Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/PmtId/EndToEndId	RestrictedFINXMax16Text	<p>Unique and unambiguous identification of a payment transaction, as assigned by any of the parties on the initiating side, that will be passed on throughout the entire end-to-end chain</p> <p>This identification must comply with the FIN set of characters and must not contain slashes, it will be mapped to the settlement bank on MT900/910 field 21 For the Model1, shall be on MT202 field 21</p>
Amount Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Amt/InstAmt	ActiveCurrencyAndAmount_ASI_EMIP2	Currency and amount of money to be transferred between debtor and creditor
Creditor Name Docu- ment/pain.998.001.01/PrtryData/EMIP	Max62FINCharacterText_EMIP1	Creditor name

Message item	Data type/code	Utilisation
PrtryData/PmtInf/PmtTx/Cdtr/Nm		
Creditor BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Cdtr/FI/BIC	BICIdentifier	BIC of the final beneficiary (after the final agent)
Creditor Account Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/CdtrAcct/DmstAcct/Id	Max35FINCharacterText_EMIP1	Simple identification information. Account of the final beneficiary
Final Agent BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/FnlAgt/BIC	BICIdentifier	Provides details about a system and about a member of a system It is the BIC of the account to be credited in the SSP
Remittance Information Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/RmtInf/Ustrd	Max140FINCharacterText_EMIP1	The contents must comply with the FIN set of characters. Model 1: Shall be mapped to the outgoing MT202 field 72 Model 6 (code CUO) : Shall be mapped to the remittance information of the payment transaction branch of the ATransferNotice sent to the ancillary system Others models : Shall be mapped to the MT900/910 field 72

Table 251 - ATransferInitiation SendATransferInitiation (pain.998) – usage case Send Ancillary System Transfer Initiation

Usage case example: pain.998_RTGS_ATransferInitiation_SendATransferInitiation_Example.xml

In this example, an *ATransferInitiation* is sent by the ancillary system to instruct the RTGS to execute a payment. It illustrates the mandatory elements in the message.

Usage case: Procedure C – Liquidity Adjustment

In this usage case, the ancillary system is providing an instruction to the RTGS component to execute a Liquidity Adjustment in its RTGS DCAs, or subaccounts, following the settlement of standing liquidity transfer orders during Ancillary System Settlement Procedure C.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Proprietary Data Type Docu- ment/pain.998.001.01/PrtryData/Tp	Max35Text	always ASTransferInitiation
Group Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/GrpId	Max35Text	Reference assigned by the sender to identify the group of individual transfers being sent
Creation Date Time Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CreDtTm	RestrictedISODateTime	Date and time at which the transfer initiation was created
Control Sum Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CtrlSum	ControlSum_ASI_EMIP1	Total of all the individual instructed amounts
Number of Transactions Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/NbOfTxS	Max15NumericText	Number of individual transactions contained in the message, i.e. the number of occurrences of PaymentTransaction
Priority Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/Prty	PriorityCode_EMIP1	Priority of all the payments at group level
Settlement Model Type Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/SttlmMdlTp	SettlementModelType_EMIP1	Specifies the generic settlement Procedure 1000 = Procedure 1 (Liquidity transfer) 2000 = Procedure 2 (Real time settle-

Message item	Data type/code	Utilisation
		<p>ment)</p> <p>3000 = Procedure 3 (Bilateral settlement)</p> <p>4000 = Procedure 4 (Standard multilateral settlement)</p> <p>5000 = Procedure 5 (Simultaneous multilateral settlement)</p> <p>6000 = Procedure 6 (Settlement on dedicated liquidity)</p>
Initiating Party BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/InitgPty/FI/BIC	BICIdentifier	The optional sequence will be filled in only in case a CB or SSP operator is sending the message on behalf of an ancillary system
Requested Execution Date Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/ReqdExctnDt	RestrictedISODate_EMIP1	<p>Date at which the initiating party requests that the payment instruction be processed (=settlement date)</p> <p>Must be the current business day</p>
Payment Scheme Code Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/CdtTrfTpId/StlmPrty/PmtSch me/Cd	CashClearingSystemCode_EMIP1 CHA	<p>Specific rulebook governing the rules of settlement</p> <p>"CDS" = Model 6 Cross ancillary system settlement between two ancillary system</p> <p>"CUO" = Model 6 Current Order Current order sent either by an ancillary system on behalf of a Settlement Bank</p> <p>"CSP" = Model 6 Connected SSP Automatic increase/decrease of liquidity by connected payments on SSP account</p> <p>"CHA" = Model 6 Connected Home Account Automatic increase/decrease of liquidity with credit lines managed in proprie-</p>

Message item	Data type/code	Utilisation
		<p>tary home account</p> <p>"COL" = Model 6 Auto-Collateral</p> <p>File of mandated payments to debit (credit) ancillary system Auto collateral account and credit (debit) sub-accounts of</p> <p>Settlement Banks</p> <p>"REP" = Model 6 Auto-collateral for Repo countries</p> <p>Automatic increase/decrease of blocked liquidity by autocollateralisation for repo countries</p> <p>"REP" = Model 1 to 5</p> <p>REPO operations</p> <p>"STR" = Model 6 Specific Transactions</p> <p>Increase of dedicated liquidity triggered by specific transactions</p> <p>"SET" = Model 6 Settlement</p> <p>Settlement</p> <p>"SOR" = Model 6 Standing Order</p> <p>Code used only in <i>ASTransferNotice</i> to notify to the ancillary system the funds booked on the Technical account - procedure 6</p> <p>real-time after standing order execution</p>
First Agent BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/FrstAgt/BIC	BICIdentifier	First agent identification
Instruction Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/PmtId/InstrId	RestrictedFINXMax16Text	Unique and unambiguous identifier for a payment instruction assigned by the initiating party
End to End Identification	RestrictedFINXMax16Text	Unique and unambiguous identification of a payment transaction, as assigned

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/PmtId/EndToEndId		by any of the parties on the initiating side, that will be passed on throughout the entire end-to-end chain This identification must comply with the FIN set of characters and must not contain slashes, it will be mapped to the settlement bank on MT900/910 field 21 For the Model1, shall be on MT202 field 21
Amount Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Amt/InstAmt	ActiveCurrencyAndA- mount_ASI_EMIP2	Currency and amount of money to be transferred between debtor and creditor
Final Agent BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/FnlAgt/BIC	BICIdentifier	Provides details about a system and about a member of a system It is the BIC of the account to be credited in the SSP
Remittance Information Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/RmtInf/Ustrd	Max140FINCharacterText_EMIP1	The contents must comply with the FIN set of characters. Model 1: Shall be mapped to the outgoing MT202 field 72 Model 6 (code CUO) : Shall be mapped to the remittance information of the payment transaction branch of the <i>ASTransferNotice</i> sent to the ancillary system Others models : Shall be mapped to the MT900/910 field 72

Table 252 - ASTransferInitiation SendASTransferInitiation (pain.998) – usage case Procedure C – Liquidity Adjustment

Usage case example: pain.998_RTGS_ASTransferInitiation_ProcCLiquidityAdjustment_Example.xml

In this example, an *ASTransferInitiation* is sent by the Ancillary System to instruct the RTGS to execute a liquidity adjustment. It illustrates the mandatory elements in the message.

Usage case: Procedure C – Instruct the Settlement Transactions

In this usage case, the ancillary system is instructing the RTGS component to execute payments from RTGS DCAs owned or managed by the ancillary system. These payments are to be executed specifically using the Ancillary System Settlement procedure cycle C.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Proprietary Data Type Docu- ment/pain.998.001.01/PrtryData/Tp	Max35Text	always ASTransferInitiation
Group Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/GrpId	Max35Text	Reference assigned by the sender to identify the group of individual transfers being sent
Creation Date Time Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CreDtTm	RestrictedISODateTime	Date and time at which the transfer initiation was created
Control Sum Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CtrlSum	ControlSum_ASI_EMIP1	Total of all the individual instructed amounts
Number of Transactions Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/NbOfTxS	Max15NumericText	Number of individual transactions contained in the message, i.e. the number of occurrences of PaymentTransaction
Priority Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/Prty	PriorityCode_EMIP1	Priority of all the payments at group level
Settlement Model Type	SettlementModelType_EMIP1	Specifies the generic settlement Pro-

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/SttlmMdlTp		cedure 1000 = Procedure 1 (Liquidity transfer) 2000 = Procedure 2 (Real time settle- ment) 3000 = Procedure 3 (Bilateral settle- ment) 4000 = Procedure 4 (Standard multilat- eral settlement) 5000 = Procedure 5 (Simultaneous multilateral settlement) 6000 = Procedure 6 (Settlement on dedicated liquidity)
Initiating Party BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/InitgPty/Fl/BIC	BICIdentifier	The optional sequence will be filled in only in case a CB or SSP operator is sending the message on behalf of an ancillary system
Requested Execution Date Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/ReqdExctnDt	RestrictedISODate_EMIP1	Date at which the initiating party re- quests that the payment instruction be processed (=settlement date) Must be the current business day
Payment Scheme Code Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/CdtTrfTpId/SttlmPrty/PmtSch me/Cd	CashClearingSystemCode_EMIP1 CUO	Specific rulebook governing the rules of settlement "CDS" = Model 6 Cross ancillary sys- tem settlement between two ancillary system "CUO" = Model 6 Current Order Current order sent either by an ancil- lary system on behalf of a Settlement Bank "CSP" = Model 6 Connected SSP Automatic increase/decrease of liquidi- ty by connected payments on SSP account "CHA" = Model 6 Connected Home

Message item	Data type/code	Utilisation
		<p>Account</p> <p>Automatic increase/decrease of liquidity with credit lines</p> <p>managed in proprietary home account</p> <p>"COL" = Model 6 Auto-Collateral</p> <p>File of mandated payments to debit (credit) ancillary system Auto collateral account and credit (debit) sub-accounts of settlement banks</p> <p>"REP" = Model 6 Auto-collateral for Repo countries</p> <p>Automatic increase/decrease of blocked liquidity by autocolateralisation for repo countries</p> <p>"REP" = Model 1 to 5</p> <p>REPO operations</p> <p>"STR" = Model 6 Specific Transactions</p> <p>Increase of dedicated liquidity triggered by specific transactions</p> <p>"SET" = Model 6 Settlement</p> <p>Settlement</p> <p>"SOR" = Model 6 Standing Order</p> <p>Code used only in <i>ASTransferNotice</i> to notify to the ancillary system the funds booked on the Technical account - procedure 6</p> <p>real-time after standing order execution</p>
First Agent BIC	BICIdentifier	First Agent identification
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/FrstAgt/BIC		
Instruction Identification	RestrictedFINXMax16Text	Unique and unambiguous identifier for a payment instruction assigned by the initiating party
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/PmtId/InstrId		

Message item	Data type/code	Utilisation
End to End Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/PmtId/EndToEndId	RestrictedFINXMax16Text	Unique and unambiguous identification of a payment transaction, as assigned by any of the parties on the initiating side, that will be passed on throughout the entire end-to-end chain This identification must comply with the FIN set of characters and must not contain slashes, it will be mapped to the settlement bank on MT900/910 field 21 For the Model1, shall be on MT202 field 21
Amount Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Amt/InstAmt	ActiveCurrencyAndA- mount_ASI_EMIP2	Currency and amount of money to be transferred between debtor and creditor
Final Agent BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/FnlAgt/BIC	BICIdentifier	Provides details about a system and about a member of a system It is the BIC of the account to be credited in the SSP
Remittance Information Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/RmtInf/Ustrd	Max140FINCharacterText_EMIP1	The contents must comply with the FIN set of characters. Model 1: Shall be mapped to the outgoing MT202 field 72 Model 6 (code CUO) : Shall be mapped to the remittance information of the payment transaction branch of the <i>ASTransferNotice</i> sent to the ancillary system Others models : Shall be mapped to the MT900/910 field 72

Table 253 - ASTransferInitiation SendASTransferInitiation (pain.998) – usage case Procedure C – Instruct the Settlement Transactions

Usage case example: pain.998_RTGS_ASTransferInitiation_ProcCInstructThe SettlementInstructions_Example.xml

In this example, an *ASTransferInitiation* is sent by the ancillary system to instruct the RTGS to execute a payment. It illustrates the mandatory elements in the message.

Usage case: Procedure D – Liquidity Adjustment

In this usage case, the ancillary system is providing an instruction to the RTGS component to execute a Liquidity Adjustment in its RTGS DCAs, or subaccounts, following the settlement of standing liquidity transfer orders during Ancillary System Settlement Procedure D.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
Proprietary Data Type Docu- ment/pain.998.001.01/PrtryData/Tp	Max35Text	always ASTransferInitiation
Group Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/GrpId	Max35Text	Reference assigned by the sender to identify the group of individual transfers being sent
Creation Date Time Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CreDtTm	RestrictedISODateTime	Date and time at which the transfer initiation was created
Control Sum Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/CtrlSum	ControlSum_ASI_EMIP1	Total of all the individual instructed amounts
Number of Transactions Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/NbOfTxs	Max15NumericText	Number of individual transactions contained in the message, i.e. the number of occurrences of PaymentTransaction
Priority	PriorityCode_EMIP1	Priority of all the payments at Group

Message item	Data type/code	Utilisation
Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/Prty		level
Settlement Model Type Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/SttlmMdlTp	SettlementModelType_EMIP1	Specifies the generic settlement Pro- cedure 1000 = Procedure 1 (Liquidity transfer) 2000 = Procedure 2 (Real time settle- ment) 3000 = Procedure 3 (Bilateral settle- ment) 4000 = Procedure 4 (Standard multilat- eral settlement) 5000 = Procedure 5 (Simultaneous multilateral settlement) 6000 = Procedure 6 (Settlement on dedicated liquidity)
Initiating Party BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/GrpHdr/InitgPty/FI/BIC	BICIdentifier	The optional sequence will be filled in only in case a CB or SSP operator is sending the message on behalf of an ancillary system
Requested Execution Date Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/ReqdExctnDt	RestrictedISODate_EMIP1	Date at which the initiating party re- quests that the payment instruction be processed (=settlement date) Must be the current business day
Payment Scheme Code Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/CdtTrfTpId/SttlmPrty/PmtSch me/Cd	CashClearingSystemCode_EMIP1 CUO	Specific rulebook governing the rules of settlement "CDS" = Model 6 Cross Ancillary Sys- tem Settlement between two Ancillary System "CUO" = Model 6 Current Order Current order sent either by an ancil- lary system on behalf of a Settlement Bank "CSP" = Model 6 Connected SSP

Message item	Data type/code	Utilisation
		<p>Automatic increase/decrease of liquidity by connected payments on SSP account</p> <p>"CHA" = Model 6 Connected Home Account</p> <p>Automatic increase/decrease of liquidity with credit lines managed in proprietary home account</p> <p>"COL" = Model 6 Auto-Collateral</p> <p>File of mandated payments to debit (credit) Ancillary System Auto collateral account and credit (debit) sub-accounts of settlement banks</p> <p>"REP" = Model 6 Auto-collateral for Repo countries</p> <p>Automatic increase/decrease of blocked liquidity by autocolateralisation for repo countries</p> <p>"REP" = Model 1 to 5</p> <p>REPO operations</p> <p>"STR" = Model 6 Specific Transactions</p> <p>Increase of dedicated liquidity triggered by specific transactions</p> <p>"SET" = Model 6 Settlement</p> <p>Settlement</p> <p>"SOR" = Model 6 Standing Order</p> <p>Code used only in <i>ASTransferNotice</i> to notify to the ancillary system the funds booked on the Technical account - procedure 6</p> <p>real-time after standing order execution</p>
First Agent BIC Docu- ment/pain.998.001.01/PrtryData/EMIP	BICIdentifier	First Agent identification

Message item	Data type/code	Utilisation
PrtryData/PmtInf/FrstAgt/BIC		
Instruction Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/PmtId/InstrId	RestrictedFINXMax16Text	Unique and unambiguous identifier for a payment instruction assigned by the initiating party
End to End Identification Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryDa- ta/PmtInf/PmtTx/PmtId/EndToEndId	RestrictedFINXMax16Text	<p>Unique and unambiguous identification of a payment transaction, as assigned by any of the parties on the initiating side, that will be passed on throughout the entire end-to-end chain</p> <p>This identification must comply with the FIN set of characters and must not contain slashes, it will be mapped to the settlement bank on MT900/910 field 21 For the Model1, shall be on MT202 field 21</p>

Message item	Data type/code	Utilisation
Amount Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/Amt/InstAmt	ActiveCurrencyAndA- mount_ASI_EMIP2	Currency and amount of money to be transferred between debtor and creditor
Final Agent BIC Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/FnlAgt/BIC	BICIdentifier	Provides details about a system and about a member of a system It is the BIC of the account to be credited in the SSP
Remittance Information Docu- ment/pain.998.001.01/PrtryData/EMIP PrtryData/PmtInf/PmtTx/RmtInf/Ustrd	Max140FINCharacterText_EMIP1	The contents must comply with the FIN set of characters. Model 1: Shall be mapped to the outgoing MT202 field 72 Model 6 (code CUO) : Shall be mapped to the remittance information of the payment transaction branch of the <i>ASTransferNotice</i> sent to the ancillary system Others models : Shall be mapped to the MT900/910 field 72

Table 254 - ASTransferInitiation SendASTransferInitiation (pain.998) – usage case Procedure D – Liquidity Adjustment

Usage case example: pain.998_RTGS_ASTransferInitiation_ProcDLiquidityAdjustment_Example.xml

In this example, an *ASTransferInitiation* is sent by the ancillary system to instruct the RTGS to execute a liquidity transfer between a sub-account and a DCA. It illustrates the mandatory elements in the message.

14.7 Reference data (reda)

14.7.1 PartyQuery (reda.015)

14.7.1.1 Overview and scope of the message

This chapter illustrates the PartyQuery message.

The PartyQuery is sent by an actor authorised to query party reference data.

In response to the PartyQuery, a [PartyReport \(reda.017\)](#) [▶ 652] containing the requested information is returned.

14.7.1.2 Schema

Outline of the schema

The *PartyQuery* message is composed of the following message building blocks:

Message identification

This building block is mandatory. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Search Criteria

This block is mandatory and it contains detailed information related to the business party query message. It includes the following elements:

- | identification
- | opening and closing date
- | type of the party
- | CB identification

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/reda.015.001.001>

14.7.1.3 The message in business context

Usage case: Party reference data query

In this usage case reference data about a party is requested.

Specific message requirements

At least one of the search criteria must be provided.

Message item	Data type/code	Utilisation
Identification Docu- ment/PtyQry/MsgId/ReqTp/Prtry/Id	Exact4AlphaNumericText_T2S_5	Fixed value "PYRD"
OpeningDate Document/PtyQry/SchCrit/OpngDt	DateSearchChoice	Opening date
ClosingDate Document/PtyQry/SchCrit/ClsgDt	DateSearchChoice	Closing date
Type Document/PtyQry/SchCrit/Tp	SystemPartyType1Code	Party type
CSDOrNCB Document/PtyQry/SchCrit/CSDOrNCB	CSDOrNCB1Choice	NCB BIC
Identification Document/PtyQry/SchCrit/Id	BICFIIdentifier	Party BIC

Table 255 - PartyQuery (reda.015) – usage case Party reference data query

Usage case example: PartyReferenceDataQuery_example.xml

In this example a CB with BIC "NCBAXXYAAA" queries reference data of the payment bank with BIC "PMBKAXXYAA" under its responsibility.

14.7.2 PartyReport (reda.017)

14.7.2.1 Overview and scope of the message

This chapter illustrates the *PartyReport* message.

The *PartyReport* is sent by CRDM to an authorised actor to provide with requested party information.

The *PartyReport* is sent in response to the [PartyQuery \(reda.015\)](#) [▶ 651] message.

14.7.2.2 Schema

Outline of the schema

The *PartyReport* message is composed of the following message building blocks:

MessageHeader

It contains an identification assigned to uniquely and unambiguously identify the message and the identification of the original business query generating the report.

ReportOrError

This building block is mandatory. It contains either the information matching the search criteria of the related query or an error indication.

PartyReport

This building block is optional. It provides requested information on party.

It includes the following elements:

- | identification
- | opening and closing date
- | party type
- | technical address
- | long and short names
- | address
- | restriction information

OperationalError

This building block is optional. It provides the reason why the requested information cannot be given.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/reda.017.001.001>

14.7.2.3 The message in business context

Usage case: Party reference data response

This message usage provides the sender with requested information about party reference data.

Specific message content

A party reference data response contains the following set of information on queried party.

Message item	Data type/code	Utilisation
PartyIdentification Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyId	SystemPartyIdentification3	Identification of the party to be reported
OpeningDate Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/OpngDt	ISODate	Opening date for the party
ClosingDate Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/ClsgDt	ISODate	Closing date for the party
Type Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/Tp	SystemPartyType1Code	Party type
TechnicalAddress Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/TechAdr	TechnicalIdentification1Choice	Technical addresses for the party
Identification Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/Id	SystemPartyIdentification1	Party code for the party

Message item	Data type/code	Utilisation
Name Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/Nm	PartyName3	Long and short names for the party
Address Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/Adr	PostalAddress10	Address for the party
Restriction Docu- ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/Rstrctn	SystemRestriction1	Restrictions issued on the party

Table 256 - PartyReport (read.017) – usage case Party reference data response

The returned data in case of an error response is listed below:

Message item	Data type/code	Utilisation
Proprietary Docu- ment/PtyRpt/RptOrErr/OprlErr/Err/Prtry	Exact4AlphaNumericText	Specific error code
Description Docu- ment/PtyRpt/RptOrErr/OprlErr/Desc	Max140Text	Textual description in addition to the reported error code

Table 257 - PartyReport (read.017) – usage case Error

Usage case example: PartyReferenceDataResponse_example.xml

In this example, a CB with BIC “NCBAXXYYAAA” queried reference data of the payment bank with BIC “PMBKAXXYYAA” under its responsibility.

Reference data of the party “PMBKAXXYYAA” is returned.

14.7.3 CashAccountAuditTrailQuery (reda.039)

14.7.3.1 Overview and scope of the message

This chapter illustrates the *CashAccountAuditTrailQuery* message.

The *CashAccountAuditTrailQuery* is sent by an actor authorised to query on audit trail for cash account reference data.

In response to the *CashAccountAuditTrailQuery*, a [CashAccountAuditTrailReport \(reda.040\)](#) [▶ 657] containing the requested information is returned.

14.7.3.2 Schema

Outline of the schema

The *CashAccountAuditTrailQuery* message is composed of the following message building blocks:

Message identification

This building block is mandatory. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Search criteria

This block is mandatory and it contains detailed information related to the business cash account audit trail query message. It includes the following elements:

- | cash account identification
- | date period

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/reda.039.001.001>

14.7.3.3 The message in business context

Usage case: Cash account audit trail query

In this usage case audit trail reference data for cash account is requested.

Specific message requirements

Message item	Data type/code	Utilisation
Cash Account Identification Docu- ment/CshAcctAudtTrlQry/SchCrit/CshA cctId/Id/Othr/Id	RestrictedFINX2Max34Text	Cash account identification
Date period Docu- ment/CshAcctAudtTrlQry/SchCrit/DtPrd	DateSearchChoice	Date period

Table 258 - CashAccountAuditTrailQuery (camt.039) – usage case Cash account audit trail query

Usage case example: CashAccountAuditTrailQuery_example.xml

In this example a CB queries audit trail information for Cash Account identified with “ACC001” and date period from 2018-01-01 to 2018-01-05.

14.7.4 CashAccountAuditTrailReport (reda.040)

14.7.4.1 Overview and scope of the message

This chapter illustrates the *CashAccountAuditTrailReport* message.

The *CashAccountAuditTrailReport* is sent by CRDM to an authorised actor to provide with requested cash account audit trail information.

The *CashAccountAuditTrailReport* is sent in response to the [CashAccountAuditTrailQuery \(reda.039\)](#) [▶ 655] message.

14.7.4.2 Schema

Outline of the schema

The *CashAccountAuditTrailReport* message is composed of the following message building blocks:

MessageHeader

It contains an identification assigned to uniquely and unambiguously identify the message and the identification of the original business query generating the report.

ReportOrError

This building block is mandatory it contains either the information matching the search criteria of the related query or an error indication.

CashAccountAuditTrailReport

This building block is optional. It provides requested information on Cash Account audit trail. It includes the following elements:

- | identification of the cash account
- | name of the field changed
- | value of the field before the change
- | value of the field after the change
- | timestamp of the change
- | name of the user who instructed the change
- | name of the user who approved the change in a four eyes scenario

BusinessError

This building block is optional. It provides the reason why the requested information cannot be given.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/reda.040.001.001>

14.7.4.3 The message in business context

Usage case: Cash account audit trail report

This message usage provides the sender with requested information about cash account audit trail reference data.

Specific message requirements

A cash account audit trail report contains the following set of information on queried object.

Message item	Data type/code	Utilisation
Field name Docu- ment/CshAcctAudtTrIRpt/RptOrErr/Csh Ac- ctAudtTrIRpt/CshAcctAudtTrlOrErr/Aud tTrl/FldNm	RestrictedFINXMax35Text	Field name
Old field value Docu- ment/CshAcctAudtTrIRpt/RptOrErr/Csh Ac- ctAudtTrIRpt/CshAcctAudtTrlOrErr/Aud tTrl/OdFldVal	RestrictedFINXMax350Text	Old field value
New field value Docu- ment/CshAcctAudtTrIRpt/RptOrErr/Csh Ac- ctAudtTrIRpt/CshAcctAudtTrlOrErr/Aud tTrl/NewFldVal	RestrictedFINXMax350Text	New field value
Timestamp Docu- ment/CshAcctAudtTrIRpt/RptOrErr/Csh Ac- ctAudtTrIRpt/CshAcctAudtTrlOrErr/Aud tTrl/OprTmStmp	ISODateTime	Timestamp

Message item	Data type/code	Utilisation
Instructing user Docu- ment/CshAcctAudtTrlRpt/RptOrErr/Csh Ac- ctAudtTrlRpt/CshAcctAudtTrlOrErr/Aud tTrl/InstgUsr	RestrictedFINXMax256Text	Instructing user
Approving user Docu- ment/CshAcctAudtTrlRpt/RptOrErr/Csh Ac- ctAudtTrlRpt/CshAcctAudtTrlOrErr/Aud tTrl/ApprovgUsr	RestrictedFINXMax256Text	Approving user
Account identification Docu- ment/CshAcctAudtTrlRpt/RptOrErr/Csh AcctAudtTrlRpt/CshAcctId/Id/Othr/Id	RestrictedFINX2Max34Text	Account identification

Table 259 - CashAccountAuditTrailReport (reda.040) – usage case Cash account audit trail report

The returned data in case of an error response is listed below:

Message item	Data type/code	Utilisation
Proprietary Docu- ment/CshAcctAudtTrlRpt/RptOrErr/Opri Err/Err/Prtry	Exact4AlphaNumericText	Specific error code
Description Docu- ment/CshAcctAudtTrlRpt/RptOrErr/Opri Err	Max140Text	Textual description in addition to the reported error code

Table 260 - CashAccountAuditTrailReport (reda.040) – usage case Error

Usage case example: CashAccountAuditTrailReport_example.xml

In this example a CB participating with BIC “NCBAXXYAAAA” queried audit trail information for cash account identified with “ACC001” during the period from 2018-01-01 to 2018-01-05.

One occurrence is returned reporting a change for the cash account. Ceiling notification amount has been set to 1.000.000 instead of 500.000. Modification has been instructed by user “USERTWOWEYES” on 2018-01-03 at 17:59.

14.7.5 PartyAuditTrailQuery (reda.042)

14.7.5.1 Overview and scope of the message

This chapter illustrates the *PartyAuditTrailQuery* message.

The *PartyAuditTrailQuery* is sent by an actor authorised to query on audit trail for party reference data.

In response to the *PartyAuditTrailQuery*, a [PartyAuditTrailReport \(reda.043\)](#) [▶ 662] containing the requested information is returned.

14.7.5.2 Schema

Outline of the schema

The *PartyAuditTrailQuery* message is composed of the following message building blocks:

MessageIdentification

This building block is mandatory. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Search criteria

This block is mandatory and it contains detailed information related to the business party audit trail query message. It includes the following elements:

- | party identification
- | date period

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/reda.042.001.001>

14.7.5.3 The message in business context

Usage case: Party audit trail query

In this usage case audit trail reference data for party is requested.

Specific message requirements

Message item	Data type/code	Utilisation
Party identification Docu- ment/PtyAudtTrlQry/SchCrit/PtyId/Rltd PtyId	BICFIIdentifier	Party identification
Responsible party identification Docu- ment/PtyAudtTrlQry/SchCrit/PtyId/Rspn sblPtyId	BICFIIdentifier	NCB
Date period Docu- ment/CshAcctAudtTrlQry/SchCrit/DtPrd	DateSearchChoice	Date period

Table 261 - PartyAuditTrailQuery (reda.042) – usage case Party audit trail query

Usage case example: PartyAuditTrailQuery_example.xml

In this example a CB with BIC “NCBAXXYAAA” queries audit trail information for party with BIC “PAYBXXYYAAA” for which it is responsible.

14.7.6 PartyAuditTrailReport (reda.043)

14.7.6.1 Overview and scope of the message

This chapter illustrates the *PartyAuditTrailReport* message.

The *PartyAuditTrailReport* is sent by CRDM to an authorised actor to provide with requested Party audit trail information.

The *PartyAuditTrailReportV01* reports changes applied to the following entities:

- | party
- | party name
- | party address
- | party code

The *PartyAuditTrailReport* is sent in response to the [PartyAuditTrailQuery \(reda.042\)](#) [▶ 661] message.

14.7.6.2 Schema

Outline of the schema

The *PartyAuditTrailReport* message is composed of the following message building blocks:

MessageHeader

It contains an identification assigned to uniquely and unambiguously identify the message and the identification of the original business query generating the report.

ReportOrError

This building block is mandatory it contains either the information matching the search criteria of the related query or an error indication.

PartyAuditTrailReport

This building block is optional. It provides requested information on party account audit trail. It includes the following elements:

- | identification of the party
- | name of the field changed
- | value of the field before the change
- | value of the field after the change
- | timestamp of the change
- | name of the user who instructed the change
- | name of the user who approved the change in a four eyes scenario

BusinessError

This building block is optional. It provides the reason why the requested information cannot be given.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/reda.043.001.001>

14.7.6.3 The message in business context

Usage case: Party audit trail report

This message usage provides the sender with requested information about party audit trail reference data.

Specific message requirements

A party audit trail report contains the following set of information on queried object.

Message item	Data type/code	Utilisation
Field name Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtTrlRpt/PtyAudtTrlOrErr/AudtTrl/FldNm	RestrictedFINXMax35Text	Field name
Old field value Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtTrlRpt/PtyAudtTrlOrErr/AudtTrl/OdFldVal	RestrictedFINXMax350Text	Old field value
New field value Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtTrlRpt/PtyAudtTrlOrErr/AudtTrl/NewFldVal	RestrictedFINXMax350Text	New field value
Timestamp Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtTrlRpt/PtyAudtTrlOrErr/AudtTrl/OprTmStamp	ISODateTime	Timestamp
Instructing user Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtTrlRpt/PtyAudtTrlOrErr/AudtTrl/InstgUsr	RestrictedFINXMax256Text	Instructing user

Message item	Data type/code	Utilisation
Approving user Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtT rlRpt/PtyAudtTrlOrErr/AudtTrl/ApprovgU sr	RestrictedFINXMax256Text	Approving user
Party identification Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtT rlRpt/Ptyld/RltdPtyldhr/ld	BICFIIdentifier	Party identification
Responsible party identification Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtT rlRpt/Ptyld/RltdPtyldhr/ld	BICFIIdentifier	NCB

Table 262 - PartyAuditTrailReport (reda.043) – usage case Party audit trail report

The returned data in case of an error response is listed below:

Message item	Data type/code	Utilisation
Proprietary Docu- ment/PtyAudtTrlRpt/RptOrErr/OprlErr/E rr/Prtry	Exact4AlphaNumericText	Specific error code
Description Docu- ment/PtyAudtTrlRpt/RptOrErr/OprlErr/D esc	Max140Text	Textual description in addition to the reported error code

Table 263 - PartyAuditTrailReport (reda.043) – usage case Error

Usage case example: PartyAuditTrailReport_example.xml

In this example a CB participating with BIC “NCBAXXYAAAA” queried audit trail information for payment bank with BIC “PAYBXXYAAAA”.

One occurrence is returned reporting a change for the party. Postal code has been changed from “54321” to “12345”. Modification has been instructed by user “USER1” and confirmed on 2018-01-03 at 17:59 by user “USER2”.

14.7.7 CalendarQuery(reda.064)

14.7.7.1 Overview and scope of the message

This chapter illustrates the *CalendarQuery* message.

The *CalendarQuery* is sent by an actor authorised to query calendar data.

In response to the *CalendarQuery*, a [CalendarReport\(reda.065\)](#) [▶ 667] containing the requested information is returned.

14.7.7.2 Schema

Outline of the schema

The *CalendarQuery* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and It contains an identification assigned by the sending party to uniquely and unambiguously identify the message.

Search criteria

This block is optional and it contains detailed information related to the calendar query message.

Allowed search criteria are:

- | year
- | month
- | service, for the specification of the service for which the query must be executed, with the currency details.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/reda.064.001.001>

14.7.7.3 The message in business context

Usage case: Calendar query

In this usage case data about calendar is requested.

Specific message requirements

Message item	Data type/code	Utilisation
Year Document/CalQry/SchCrit/Yr	ISOYear	Year
Month Document/CalQry/SchCrit/Mnth	ISOMonth	Month
Service Docu- ment/CalQry/SchCrit/Svc/SysId/MktInfr strctrId/Prtry	Max35Text	Service
Currency Document/CalQry/SchCrit/Svc/SysCcy	ActiveCurrencyCode	Currency of the service for which the calendar is requested.

Table 264 - CalendarQuery (reda.064) – usage case Calendar query

Usage case example: CalendarQuery_example.xml

14.7.8 CalendarReport(reda.065)

14.7.8.1 Overview and scope of the message

This chapter illustrates the *CalendarReport* message.

The *CalendarReport* is sent by CRDM to an authorised actor to provide with requested calendar information.

The *CalendarReport* is sent in response to the [CalendarQuery\(reda.064\)](#) [666] message.

14.7.8.2 Schema

Outline of the schema

The *CalendarReport* message is composed of the following message building blocks:

MessageHeader

It contains an identification assigned to uniquely and unambiguously identify the message and the identification of the original business query generating the report.

ReportOrError

This building block is mandatory it contains either the information matching the search criteria of the related query or an error indication.

CalendarReport

This building block is mandatory. It provides requested information on calendar, with the service information.

The CalendarData includes the following elements:

- | date
- | status

OperationalError

This building block is optional. It provides the reason why the requested information cannot be given.

References/links

The schema and the related documentation in XSD/EXCEL/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

<https://www.swift.com/mystandards/CSLD/reda.065.001.001>

14.7.8.3 The message in business context

Usage case: Calendar report

This message usage provides the sender with requested information about calendar data.

Specific message requirements

A calendar report contains the following set of information on queried calendar.

Message item	Data type/code	Utilisation
Date Docu- ment/CalRpt/RptOrErr/CalRpt/CalOrErr /CalData/SysDt	ISODate	Date
Status Docu- ment/CalRpt/RptOrErr/CalRpt/CalOrErr /CalData/SysSts/Cd	SystemStatus3Code	Status
Service Docu- ment/CalRpt/RptOrErr/CalRpt/Svc/Sysl d/MktInfrstrctrld/Prtry	Max35Text	Service
Currency Docu- ment/CalRpt/RptOrErr/CalRpt/Svc/Sys Ccy	ActiveCurrencyCode	Currency of the service for which the calendar is returned.

Table 265 - CalendarReport (reda.065) – usage case Calendar report

The returned data in case of an error response is listed below:

Message item	Data type/code	Utilisation
Proprietary Docu- ment/CalRpt/RptOrErr/OprlErr/Err/Prtry	Max35Text	Specific error code
Description Docu- ment/CalRpt/RptOrErr/OprlErr/Desc	Max140Text	Textual description in addition to the reported error code

Table 266 - CalendarReport (reda.065) – usage case Error

Usage case example: CalendarReport_example.xml

Part IV - Appendixes

15 Index and digital signature

15.1 Index of business rules and error codes

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	The technical sender (=DN) must be allowed to send messages for the business sender (=head.001 "From")	head.001	admi.007			The technical sender is not allowed to send messages for the business sender
	The digital signature on communication level has to be valid for the technical sender (DN)	DEP	admi.007			The digital signature is not valid for the technical sender
	The system user sending the inbound A2A communication has to be known.	DEP	admi.007			The system user sending the inbound A2A communication is unknown.
	The business sending user has to be known.	head.001	admi.007			The business sending user is unknown.
	The message type must be supported in the component	head.001	admi.007			The message type is not supported in the component
	The syntax of the file header has to conform to the scheme	head.002	admi.007			The syntax of the file header does not conform to the scheme
	The technical sending user has to be allowed to send for the business sending party.	head.002	pacs.002			The technical sending user is not allowed to send for the business sending party.

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Non-exhaustive
	The file must not be sent twice or payload identifier must not be used twice.	head.002	admi.007			The file is sent twice or the payload identifier is used twice.
	The syntax of the BAH has to conform to the scheme	head.001	admi.007			The syntax of the BAH does not conform to the scheme
	A message with the same BizMsgIdr and the same Business sender "From" at the same day will be detected as duplicate.	head.001	admi.007			A message with the same BizMsgIdr and the same Business sender "From" at the same day was detected as duplicate.
	The message definition identifier must be allowed ("RTGS_ Message Definition Identifier": Example of format, pacs.008.001.07 For pacs.009, it will also be indicated if the payment is a CORE or COV payment. Example, pacs.009.001.07COVE and pacs.009.001.07CORE)	head.001	admi.007			The message definition identifier is not allowed
	Payment type (pacs.008) is not allowed in CLM	pacs.008	admi.007			Customer payments (pacs.008) are not allowed in CLM
	The syntax of the instruction has to conform to the scheme	Any message	admi.007			The syntax of the message does not conform to the scheme

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
						//Daynamic error including element name//
	An instruction with the same instruction ID sent by the same instructing agent to the same instructed agent will be detected as a duplicate	pacs.008	pacs.002			An instruction with the same instruction ID sent by the same instructing agent to the same instructed agent is detected as a duplicate
	An instruction where all the following fields are duplicated will be detected as a duplicate: <ul style="list-style-type: none"> - instructing agent - message type - instructed agent - transaction ID - EndtoEnd ID - Settlementdate - Settlement amount 	pacs.008	pacs.002			The instruction was detected as a duplicate.
	The business sender must be authorized to debit the account: <ul style="list-style-type: none"> • as the owner of the account to be debited or • mandated by the owner of the account or • allowed by contractual arrangement between the third party and both instructing agent and instructed agent to do so (multi-addressee) 	pacs.008	pacs.002			The business sender is not authorized to debit the account
	Payments can be sent for the current	pacs.008	pacs.002			Payments can be sent

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	business day and up to 10 calendar days in advance.					for the current business day and up to 10 calendar days in advance.
	If the settlement date is on weekend or a RTGS holiday the payment will be rejected	pacs.008	pacs.002			As the settlement date is on weekend or a RTGS holiday the payment is rejected
	When the value date check is switched off, back valued payments are possible. Payments with more than 10 days in advance are not possible anyway.	pacs.008	pacs.002			When the value date check is switched off, back valued payments are possible. Payments with more than 10 days in advance are not possible anyway.
	The cut-off time for customer payments has to be observed	pacs.008	pacs.002			The cut-off time for customer payments has not been observed
	The instructing agent of the instruction has to be a RTGS DCA account holder	pacs.008	pacs.002			The instructing agent of the instruction is no RTGS DCA account holder
	The instructed agent of the instruction has to be a RTGS DCA account holder	pacs.008	pacs.002			The instructed agent of the instruction is no RTGS DCA account holder
	Currency must denominate the same currency as accounts indicated for posting.	pacs.008	pacs.002			Currency does not denominate the same currency as accounts indicated for posting.
	SettlementTimeRequest: Local time	pacs.008	pacs.002			SettlementTimeRequest:

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	(hh:mm) must be before the cut-off time for customer payments.					Local time (hh:mm) is after the cut-off time for customer payments.
	If a field of party is present and if the BIC is not provided then the Name and TownName and the Country must be provided.	pacs.008	admi.007			If a field of party is present and if the BIC is not provided then the Name and TownName and the Country must be provided.
	If UltimateDebtor is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the UltimateDebtor.	pacs.008	admi.007			If UltimateDebtor is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the UltimateDebtor.
	If InitiatingParty is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the InitiatingParty.	pacs.008	admi.007			If InitiatingParty is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the InitiatingParty.
	If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the Debtor.	pacs.008	admi.007			If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the Debtor.
	If OrganisationIdentification/AnyBIC is absent then Name and TownName	pacs.008	admi.007			IF OrganisationIdentification/AnyBIC is absent

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	and Country are mandatory to identify the Creditor.					THEN Name and TownName and Country are mandatory to identify the Creditor.
	If UltimateCreditor is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the UltimateCreditor.	pacs.008	admi.007			If UltimateCreditor is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the UltimateCreditor.
	InstructionInformation can only be used if code is "PHOB" in order to indicate the phone number.	pacs.008	admi.007			InstructionInformation can only be used if code is "PHOB" in order to indicate the phone number.
	Structured and unstructured remittance information are mutually exclusive and both may be absent.	pacs.008	admi.007			Structured and unstructured remittance information are mutually exclusive
	If InstructedAmount is present and the currency is different from the currency in InterbankSettlementAmount, then ExchangeRate must be present.	pacs.008	admi.007			If InstructedAmount is present and the currency is different from the currency in InterbankSettlementAmount, then ExchangeRate must be present.
	If InstructedAmount is present and the currency is the same as the currency in InterbankSettlementAmount, then ExchangeRate is not allowed.	pacs.008	admi.007			If InstructedAmount is present and the currency is the same as the currency in InterbankSet-

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
						tlementAmount, then ExchangeRate is not allowed.
	If ChargesInformation/Agent is present and if BIC is not provided then the Name, the TownName and the Country must be provided.	pacs.008	admi.007			If ChargesInformation/Agent is present and if BIC is not provided then the Name, the TownName and the Country must be provided.
	If IntermediaryAgent2 is present, then IntermediaryAgent1 must be present.	pacs.008	admi.007			If IntermediaryAgent2 is present, then IntermediaryAgent1 must be present.
	If IntermediaryAgent2Account is present, then IntermediaryAgent1 must be present.	pacs.008	admi.007			If IntermediaryAgent2Account is present, then IntermediaryAgent1 must be present.
	If IntermediaryAgent2Account is present, then IntermediaryAgent1 must be present.	pacs.008	admi.007			If IntermediaryAgent2Account is present, then IntermediaryAgent1 must be present.
	If IntermediaryAgent3Account is present, then IntermediaryAgent3 must be present.	pacs.008	admi.007			If IntermediaryAgent3Account is present, then IntermediaryAgent3 must be present.
	If IntermediaryAgent3 is present, then	pacs.008	admi.007			If IntermediaryAgent3 is

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	IntermediaryAgent2 must be present.					present, then IntermediaryAgent2 must be present.
	If PreviousInstructingAgent1Account is present, then PreviousInstructingAgent1 must be present.	pacs.008	admi.007			If PreviousInstructingAgent1Account is present, then PreviousInstructingAgent1 must be present.
	If InstructedAmount is not present, then ExchangeRate is not allowed.	pacs.008	admi.007			If InstructedAmount is not present, then ExchangeRate is not allowed.
	If PreviousInstructingAgent2Account is present, then PreviousInstructingAgent2 must be present.	pacs.008	admi.007			If PreviousInstructingAgent2Account is present, then PreviousInstructingAgent2 must be present.
	If PreviousInstructingAgent3Account is present, then PreviousInstructingAgent3 must be present.	pacs.008	admi.007			If PreviousInstructingAgent3Account is present, then PreviousInstructingAgent3 must be present.
	If PreviousInstructingAgent2 is present, then PreviousInstructingAgent1 must be present.	pacs.008	admi.007			If PreviousInstructingAgent2 is present, then PreviousInstructingAgent1 must be present.
	If PreviousInstructingAgent3 is present, then PreviousInstructingAgent2 must be present.	pacs.008	admi.007			If PreviousInstructingAgent3 is present, then PreviousInstructingAgent2 must be present.

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
						ingAgent2 must be present.
	If Till time and Reject Time are both mentioned the message will be rejected.	pacs.008	admi.007			If Till time and Reject Time are both mentioned so the message is rejected.
	Debtor account is blocked	pacs.008	pacs.002			Debtor account is blocked
	Creditor account is blocked	pacs.008	pacs.002			Creditor account is blocked
	Creditor account is blocked	pacs.008	pacs.002			Creditor account is blocked
	Debtor account is blocked	pacs.009	pacs.002			Debtor account is blocked
	The technical sender in combination to the business sender must be allowed to debit the account of the instructing agent.	pacs.009	pacs.002			The technical sender in combination to the business sender is not allowed to debit the account of the instructing agent.
	Clearing system reference must not be used in inbound messages	pacs.009	admi.007			Clearing system reference must not be used in inbound messages
	Settlement time indication must not be used in inbound messages	pacs.009	admi.007			Settlement time indication must not be used in inbound messages
	An instruction with the same instruction ID sent by the same instructing agent to the same instructed agent	pacs.009	pacs.002			An instruction with the same instruction ID sent by the same instructing

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Non-exhaustive
	will be detected as a duplicate					agent to the same instructed agent was detected as a duplicate
	<p>An instruction where all the following fields are duplicated will be detected as a duplicate:</p> <ul style="list-style-type: none"> - instructing agent - message type - instructed agent - transaction ID - EndtoEnd ID - Settlementdate - Settlement amount 	pacs.009	pacs.002			The instruction was detected as a duplicate
	<p>The business sender must be authorized to debit the account as:</p> <ul style="list-style-type: none"> • The owner of the account to be debited or • mandated by the owner of the account or • allowed by contractual arrangement between the third party and both instructing agent and instructed agent to do so (multi-addressee) • A CB acting on behalf a credit institution. 	pacs.009	pacs.002			The business sender is not authorized to debit the account
	Payments can be sent for the current business day and up to 10 calendar days in advance.	pacs.009	pacs.002			Payments can be sent for the current business day and up to 10 calendar days in advance.
	If the settlement date is on weekend	pacs.009	pacs.002			As the settlement date is

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	or a RTGS holiday the payment will be rejected					on weekend or a RTGS holiday the payment is rejected
	When the value date check is switched off, back valued payments in RTGS are possible. Payments with more than 10 days in advance are not possible anyway. Back valued payments to AS or CLM are also not possible.	pacs.009	pacs.002			When the value date check is switched off, back valued payments in RTGS are possible. Payments with more than 10 days in advance are not possible anyway. Back valued payments to AS or CLM are also not possible.
	The cut-off time for bank to bank payments has to be observed	pacs.009	pacs.002			The cut-off time for bank to bank payments was not observed
	Instructed agent must be a BIC of a RTGS account holder or a BIC of an AS technical account.	pacs.009	pacs.002			Instructed agent is neither a BIC of a RTGS account holder nor a BIC of an AS technical account.
	Currency must denominate same currency as accounts indicated for posting.	pacs.009	pacs.002			Currency does not denominate the same currency as accounts indicated for posting.
	For COVER payments an End to End ID of the pacs.008 must be present.	pacs.009	admi.007			For COVER payments an End to End ID of the pacs.008 must be present.
	SettlementTimeRequest: Local time	pacs.009	pacs.002			SettlementTimeRequest:

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	(hh:mm) must be before the cut-off time for bank to bank payments.					Local time (hh:mm) is after the cut-off time for bank to bank payments.
	If Till time and Reject Time are both mentioned the message will be rejected.	pacs.009	admi.007			If Till time and Reject Time are both mentioned so the message is rejected.
	Till Time must be before the cut-off time for bank to bank payments.	pacs.009	pacs.002			Till Time is after the cut-off time for bank to bank payments.
	From Time must be before the cut-off time for bank to bank payments.	pacs.009	pacs.002			From Time is after the cut-off time for bank to bank payments.
	If a field of party is present and If the BIC is not provided then the Name and TownName and the Country must be provided.	pacs.009	admi.007			If a field of party is present and If the BIC is not provided then the Name and TownName and the Country must be provided.
	InitiatingParty in UnderlyingCustomerCreditTransfer if used, must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country. Both identifications may be present.	pacs.009	admi.007			InitiatingParty in UnderlyingCustomerCreditTransfer if used, must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country. Both identifications may be present.
	InstructionInformation can only be used if Code is "PHOB" in order to	pacs.009	pacs.002			InstructionInformation can only be used if Code

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	indicate the phone number.					is "PHOB" in order to indicate the phone number.
	In UnderlyingCustomerCreditTransfer, Structured and Unstructured remittance information are mutually exclusive and both may be absent.	pacs.009	admi.007			In UnderlyingCustomerCreditTransfer, Structured and Unstructured remittance information are mutually exclusive and both may be absent.
	If IntermediaryAgent1 is present, then CreditorAgent must be present.	pacs.009	admi.007			If IntermediaryAgent1 is present, then CreditorAgent must be present.
	If IntermediaryAgent2 is present, then IntermediaryAgent1 must be present.	pacs.009	admi.007			If IntermediaryAgent2 is present, then IntermediaryAgent1 must be present.
	If IntermediaryAgent1Account is present, then IntermediaryAgent1 must be present.	pacs.009	admi.007			If IntermediaryAgent1Account is present, then IntermediaryAgent1 must be present.
	If IntermediaryAgent2Account is present, then IntermediaryAgent2 must be present.	pacs.009	admi.007			If IntermediaryAgent2Account is present, then IntermediaryAgent2 must be present.
	If IntermediaryAgent3Account is present, then IntermediaryAgent3	pacs.009	admi.007			If IntermediaryAgent3Account is

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	must be present.					present, then IntermediaryAgent3 must be present.
	If IntermediaryAgent3 is present, then IntermediaryAgent2 must be present.	pacs.009	admi.007			If IntermediaryAgent3 is present, then IntermediaryAgent2 must be present.
	If IntermediaryAgent1Account is present, then IntermediaryAgent1 must be present.	pacs.009	admi.007			If IntermediaryAgent1Account is present, then IntermediaryAgent1 must be present.
	If DebtorAgentAccount is present, then DebtorAgent must be present.	pacs.009	admi.007			If DebtorAgentAccount is present, then DebtorAgent must be present.
	If CreditorAgentAccount is present, then CreditorAgent must be present.	pacs.009	admi.007			If CreditorAgentAccount is present, then CreditorAgent must be present.
	IntermediaryAgent2 is not allowed in cover payments	pacs.009	admi.007			IntermediaryAgent2 is not allowed in cover payments
	IntermediaryAgent2Account is not allowed in cover payments	pacs.009	admi.007			IntermediaryAgent2Account is not allowed in cover payments
	IntermediaryAgent3 is not allowed in cover payments	pacs.009	admi.007			IntermediaryAgent3 is not allowed in cover payments

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	IntermediaryAgent3Account is not allowed in cover payments	pacs.009	admi.007			IntermediaryAgent3Account is not allowed in cover payments
	Ultimate Dabtor is not allowed in cover payments	pacs.009	admi.007			Ultimate Dabtor is not allowed in cover payments
	UltimateCreditor is not allowed in cover payments	pacs.009	admi.007			UltimateCreditor is not allowed in cover payments
	Using of code word MANPAY is only allowed if a responsible CB of the participant quoted in field debtor is the business sender of the message.	pacs.009	pacs.002			The mandated payment is not allowed by this business sender
	Payments with urgent priority are only allowed, if payment is: • instructing agent = CB or • Code = ASTI or SBTI	pacs.009	pacs.002			Proirity urgent is not allowed for this payment
	If UltimateDebtor in UnderlyingCustomerCreditTransfer is used, it must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country. Both identifications may be present.	pacs.009	admi.007			If UltimateDebtor in UnderlyingCustomerCreditTransfer is used, it must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country. Both identifications may be present.
	Debtor in UnderlyingCustomerCreditTransfer must be identified either with OrganisationIdentifi-	pacs.009	admi.007			Debtor in UnderlyingCustomerCreditTransfer must be identified either

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	cation/AnyBIC or with Name and TownName and Country. Both identifications may be present.					with OrganisationIdentification/AnyBIC or with Name and TownName and Country. Both identifications may be present.
	Debtor in UnderlyingCustomerCreditTransfer must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country. Both identifications may be present.	pacs.009	admi.007			Debtor in UnderlyingCustomerCreditTransfer must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country. Both identifications may be present.
	If PreviousInstructingAgent1 in UnderlyingCustomerCreditTransfer is used and IF BIC is not provided then the Name and TownName and the Country must be provided for the PreviousInstructingAgent1.	pacs.009	admi.007			If PreviousInstructingAgent1 in UnderlyingCustomerCreditTransfer is used and IF BIC is not provided then the Name and TownName and the Country must be provided for the PreviousInstructingAgent1.
	If IntermediaryAgent1 in UnderlyingCustomerCreditTransfer is used and if BIC is not provided then the Name and TownName and the Country must be provided for IntermediaryAgent1.	pacs.009	admi.007			If IntermediaryAgent1 in UnderlyingCustomerCreditTransfer is used and if BIC is not provided then the Name and TownName and the Country must be provided for IntermediaryAgent1.

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	If IntermediaryAgent2 in UnderlyingCustomerCreditTransfer is used and if BIC is not provided then the Name and TownName and the Country must be provided for IntermediaryAgent2.	pacs.009	admi.007			If IntermediaryAgent2 in UnderlyingCustomerCreditTransfer is used and if BIC is not provided then the Name and TownName and the Country must be provided for IntermediaryAgent2.
	If IntermediaryAgent3 in UnderlyingCustomerCreditTransfer is used and if BIC is not provided then the Name and TownName and the Country must be provided for IntermediaryAgent3.	pacs.009	admi.007			If IntermediaryAgent3 in UnderlyingCustomerCreditTransfer is used and if BIC is not provided then the Name and TownName and the Country must be provided for IntermediaryAgent3.
	If CreditorAgent's BIC in UnderlyingCustomerCreditTransfer is absent then the Name and TownName and the Country must be provided for CreditorAgent.	pacs.009	admi.007			If CreditorAgent's BIC in UnderlyingCustomerCreditTransfer is absent then the Name and TownName and the Country must be provided for CreditorAgent.
	Creditor in UnderlyingCustomerCreditTransfer must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country. Both identifications may be present.	pacs.009	admi.007			Creditor in UnderlyingCustomerCreditTransfer must be identified either with OrganisationIdentification/AnyBIC or with Name and Town-

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
						Name and Country. Both identifications may be present.
	If UltimateCreditor in UnderlyingCustomerCreditTransfer is used, it must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country of the UltimateCreditor. Both identifications may be present.	pacs.009	admi.007			If UltimateCreditor in UnderlyingCustomerCreditTransfer is used, it must be identified either with OrganisationIdentification/AnyBIC or with Name and TownName and Country of the UltimateCreditor. Both identifications may be present.
	If PreviousInstructingAgent2Account is present, then PreviousInstructingAgent2 must be present.	pacs.009	admi.007			If PreviousInstructingAgent2Account is present, then PreviousInstructingAgent2 must be present.
	If PreviousInstructingAgent3Account is present, then PreviousInstructingAgent3 must be present.	pacs.009	admi.007			If PreviousInstructingAgent3Account is present, then PreviousInstructingAgent3 must be present.
	If PreviousInstructingAgent2 is present, then PreviousInstructingAgent1 must be present.	pacs.009	admi.007			If PreviousInstructingAgent2 is present, then PreviousInstructingAgent1 must be present.
	If PreviousInstructingAgent3 is present, then PreviousInstructingAgent2	pacs.009	admi.007			If PreviousInstructingAgent3 is present,

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	must be present.					then PreviousInstructionAgent2 must be present.
	If the message is a core payment, then UnderlyingCustomerCreditTransfer is forbidden.	pacs.009	admi.007			If the message is a core payment, then UnderlyingCustomerCreditTransfer is forbidden.
	If the message is a cover payment, then UnderlyingCustomerCreditTransfer is mandatory.	pacs.009	admi.007			If the message is a cover payment, then UnderlyingCustomerCreditTransfer is mandatory.
	In order to send pacs.009 debiting a settlement bank, the AS needs to be authorized in reference data to send pacs.009 for the given RTGS DCA by the account owner.	pacs.009	pacs.002			The AS is not authorized to send pacs.009 for the given RTGS DCA by the account owner.
	if Codeword = ASTI or SBTI then priority must be URGENT	pacs.009	admi.007			if Codeword = ASTI then priority must be URGENT
	if Codeword = ASTI or SBTI then execution date must be current business day	pacs.009	pacs.002			if Codeword = ASTI then execution date must be current business day
	if codeword = ASTI then the BIC of the AS has to be in field "DebtorAgent"	pacs.009	pacs.002			if codeword = ASTI then the BIC of the AS has to be in field "DebtorAgent"
	When using payments, it is only possible to debit and credit RTGS DCAs. The usage of other account types (ASTechnicalAccount, sub-account etc.) is not allowed.	pacs.009	pacs.002			When using payments, it is only possible to debit and credit RTGS DCAs. The usage of other account types (ASTech-

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
						nicalAccount, sub-account etc.) is not allowed.
	If the DN of the sender is relative to a AS, the AS must be in the list delivered to RTGS by CRDM	AS Trans-ferinitiation	AS Initiation Status			Sender not allowed
	If the DN of the sender is relative to a CB or the OT, then, if the message is sent on behalf of the AS, the Tag Initiating Party must be filled with a BIC of an authorised AS. If the sender is a CB then the AS must be a member of this CB.	AS Trans-ferinitiation	AS Initiation Status			AS missing or not allowed in InitiatingParty / SubjectDetails
	RequestedExecutionDate must be the current business day (Warehouse payments are not allowed)	AS Trans-ferinitiation	AS Initiation Status			Invalid date
	ControlSum if filled, the sum of the individual amounts in PaymentTransaction are calculated and checked if it is equal to the ControlSum	AS Trans-ferinitiation	AS Initiation Status			Invalid ControlSum
	NumberOfTransactions if filled, is checked that this number is equal to the number of occurrences of PaymentTransaction in the message	AS Trans-ferinitiation	AS Initiation Status			Invalid NumberOfTransactions
	PriorityType must be "urgent"	AS Trans-ferinitiation	AS Initiation Status			Invalid PriorityType
	SettlementModelType must be in the list of procedures	AS Trans-ferinitiation	AS Initiation Status			Invalid SettlementModelType
	Initiating Party (or the sender of the	AS Trans-	AS Initiation			SettlementModelType

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	message if Initiating party is not present) must be authorised to send this SettlementModelType	ferinitiation	Status			not allowed for the sender
	Procedures A & B: The possible code is : "AGRE"	AS Trans-ferinitiation	AS Initiation Status			Invalid InformationPeriodType
	FromTime must be later that the current time and earlier than the cut-off time	AS Trans-ferinitiation	AS Initiation Status			Invalid FromTime
	ToTime must be later that the current time and earlier than the cut-off time It must be later than the scheduled time if present	AS Trans-ferinitiation	AS Initiation Status			Invalid ToTime
	Field must not be filled with blanks only	AS Trans-ferinitiation	AS Initiation Status			Invalid GroupIdentification
	PaymentScheme/Code: Procedure A or B: The field is optional only code REP is allowed Procedure C or D: The field is mandatory and only code CDS, CUO and SET are allowed	AS Trans-ferinitiation	AS Initiation Status			Invalid PaymentScheme code
	FirstAgent not allowed	AS Trans-ferinitiation	AS Initiation Status			FirstAgent not allowed
	FirstAgent domestic account not allowed	AS Trans-ferinitiation	AS Initiation Status			FirstAgent domestic account not allowed
	FinalAgent not allowed	AS Trans-ferinitiation	AS Initiation Status			FinalAgent not allowed
	FinalAgent domestic account not allowed	AS Trans-ferinitiation	AS Initiation Status			FinalAgent domestic account not allowed

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	FirstAgent and FinalAgent accounts must be different	AS Trans-ferinitiation	AS Initiation Status			FirstAgent and FinalAgent accounts must be different
	GroupIdentification must be unique over a period of 5 business days	AS Trans-ferinitiation	AS Initiation Status			GroupIdentification must be unique over a period of 5 business days
	MessageIdentification must be unique over a period of 5 business days	AS Trans-ferinitiation	AS Initiation Status			MessageIdentification must be unique over a period of 5 business days
	InstructedAmount: Procedure A, B: The total amount of debited payments from AS Technical Account must be equal to the total amount of credited payments to the Technical Account	AS Trans-ferinitiation	AS Initiation Status			Sum of debit from technical account is not equal to sum of credit to technical account
	DebitAccountOwner not allowed	AS Trans-ferinitiation	AS Initiation Status			DebitAccountOwner not allowed
	CreditAccountOwner not allowed	AS Trans-ferinitiation	AS Initiation Status			CreditAccountOwner not allowed
	Decrease: DomesticAccount; the validity of the sub-account will be checked. The LiquidityCreditTransfer between two sub-accounts is not allowed.	AS Trans-ferinitiation	AS Initiation Status			Debit DomesticAccount not allowed
	Increase: DomesticAccount; the validity of the sub-account will be checked. The LiquidityCreditTransfer between two sub-accounts is not	AS Trans-ferinitiation	AS Initiation Status			Credit DomesticAccount not allowed

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	allowed.					
	IsFormatted: Check that the indicator is "true"	AS Trans-ferinitiation	AS Initiation Status			The qualifier must be formatted
	Subject: The code has to be valid	AS Trans-ferinitiation	AS Initiation Status			Code unknown
	Invalid StatusCode	AS Trans-ferinitiation	AS Initiation Status			Invalid StatusCode
	Inconsistency between Settlement-ModelType, FirstAgent and FinalAgent	AS Trans-ferinitiation	AS Initiation Status			Inconsistency between SettlementModelType, FirstAgent and FinalAgent
	Order or message out of sequence	AS Trans-ferinitiation	AS Initiation Status			Order or message out of sequence
	Procedure already open	AS Trans-ferinitiation	AS Initiation Status			Procedure already open
	Cycle already open	AS Trans-ferinitiation	AS Initiation Status			Cycle already open
	Cycle already closed	AS Trans-ferinitiation	AS Initiation Status			Cycle already closed
	AS excluded	AS Trans-ferinitiation	AS Initiation Status			AS excluded
	Debtor BIC is not a published SWIFT BIC	AS Trans-ferinitiation	AS Initiation Status			Debtor BIC is not a published SWIFT BIC
	Creditor BIC is not a published SWIFT BIC	AS Trans-ferinitiation	AS Initiation Status			Creditor BIC is not a published SWIFT BIC
	Number of transactions is limited to 20000	AS Trans-ferinitiation	AS Initiation Status			Number of transactions is limited to 20000

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	Requested time for end of settlement period is not possible before end of information period.	AS Transfer initiation	AS Initiation Status			Requested time for end of settlement period is not possible before end of information period.
	Requested time for end of settlement period is not possible in the past.	AS Transfer initiation	AS Initiation Status			Requested time for end of settlement period is not possible in the past.
	Related AS does not participate in settlement procedure D	AS Transfer initiation	AS Initiation Status			Related AS does not participate in settlement procedure D
	FinalAgent must be a technical account - procedure D.	AS Transfer initiation	AS Initiation Status			FinalAgent must be a technical account - procedure D.
	Settlement bank is not allowed to address this technical account - procedure D	AS Transfer initiation	AS Initiation Status			Settlement bank is not allowed to address this technical account - procedure D
	CounterpartAS: If the PaymentSchemeCode is "CDS", this tag is mandatory and should contain a valid AS BIC with which the sending AS is in relation	AS Transfer initiation	AS Initiation Status			Counterpart AS does not contain a valid AS BIC in relation with the sender
	Counterpart AS forbidden for transactions other than cross-AS settlement	AS Transfer initiation	AS Initiation Status			Counterpart AS forbidden for transactions other than cross-AS settlement
	Procedure D code CDS: If the FirstAgent is an AS technical account - procedure D, the Debtor BIC is	AS Transfer initiation	AS Initiation Status			Procedure D code CDS: If the FirstAgent is an AS technical account - pro-

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Non-exhaustive
	mandatory					cedure D, the Debtor BIC is mandatory
	Procedure D code CDS: If the FinalAgent is an AS technical account - procedure D, the Creditor BIC is mandatory	AS Transferinitiation	AS Initiation Status			Procedure D code CDS: If the FinalAgent is an AS technical account - procedure D, the Creditor BIC is mandatory
	Payment reject at end of day	AS Transferinitiation	AS Initiation Status			Payment reject at end of day
	The payment is rejected because the settlement period time is reached	AS Transferinitiation	AS Initiation Status			The payment is rejected because the settlement period time is reached
	The payment has been revoked	AS Transferinitiation	AS Initiation Status			The payment has been revoked
	Rejection after reversing procedure	AS Transferinitiation	AS Initiation Status			Rejection after reversing procedure
	Payment initiation or individual transaction included in the payment initiation has been rejected	AS Transferinitiation	AS Initiation Status			Payment initiation or individual transaction included in the payment initiation has been rejected
	AS using procedure D cannot close the procedure	AS Transferinitiation	AS Initiation Status			
	Debtor account is blocked	pacs.010	pacs.002			Debtor account or creditor account is blocked
	Creditor account is blocked	pacs.010	pacs.002			Debtor account or creditor account is blocked
	The technical sender in combination	pacs.010	pacs.002			The technical sender in

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Non-exhaustive
	to the business sender must be allowed to credit the account of the instructing agent.					combination to the business sender is not allowed to credit the account of the instructing agent.
	An instruction with the same instruction ID sent by the same instructing agent to the same instructed agent will be detected as a duplicate	pacs.010	pacs.002			An instruction with the same instruction ID sent by the same instructing agent to the same instructed agent was detected as a duplicate
	An instruction where all the following fields are duplicated will be detected as a duplicate: <ul style="list-style-type: none"> - instructing agent - message type - instructed agent - transaction ID - EndtoEnd ID - Settlement Date - Settlement amount 	pacs.010	pacs.002			The instruction was detected as a duplicate
	The business sender must be authorized to credit the account as: <ul style="list-style-type: none"> • The owner of the account to be credited or • allowed by contractual arrangement between the third party and both instructing agent and instructed agent to do so (multi-addressee) 	pacs.010	pacs.002			The business sender is not authorized to credit the account
	Payments can be sent for the current	pacs.010	pacs.002			Payments can be sent

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	business day and up to 10 calendar days in advance.					for the current business day and up to 10 calendar days in advance.
	If the settlement date is on weekend or a RTGS holiday the payment will be rejected	pacs.010	pacs.002			If the execution date is on weekend or a RTGS holiday the payment will be rejected
	When the value date check is switched off, back valued payments in RTGS are possible. Payments with more than 10 days in advance are not possible anyway. Back valued payments to AS or CLM are also not possible.	pacs.010	pacs.002			When the value date check is switched off, back valued payments in RTGS are possible. Payments with more than 10 days in advance are not possible anyway. Back valued payments to AS or CLM are also not possible.
	The cut-off time for bank to bank payments has to be observed	pacs.010	pacs.002			The cut-off time for Bank to Bank payments was not observed
	The instructing agent of the instruction has to be a RTGS DCA account holder	pacs.010	pacs.002			The instructing agent of the instruction is no RTGS DCA account holder
	The instructed agent of the instruction has to be a RTGS DCA account holder	pacs.004	pacs.002			The instructed agent of the instruction is no RTGS DCA account holder
	The instructed agent of the instruction has to be a RTGS DCA account	pacs.010	pacs.002			The instructed agent of the instruction is no

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	holder					RTGS DCA account holder
	Currency must denominate the same currency as accounts indicated for posting.	pac.010	pac.002			Currency does not denominate same currency as accounts indicated for posting.
	SettlementTimeRequest: Local time (hh:mm) must be before the cut-off time for bank to bank payments.	pac.010	pac.002			SettlementTimeRequest: Local time (hh:mm) is after the cut-off time for bank to bank payments.
	If Till time and Reject Time are both mentioned the message will be rejected.	pac.010	admi.007			If Till time and Reject Time are both mentioned so the message is rejected.
	If CreditorAgentAccount is present, then CreditorAgent must be present.	pac.010	admi.007			If CreditorAgentAccount is present, then CreditorAgent must be present.
	Direct debits are not allowed with a BIC related to an AS technical account to be debited.	pac.010	pac.002			Direct debits are not allowed with a BIC related to an AS technical account to be debited.
	The instructing agent must be authorised by the instructed agent (mandate)	pac.010	pac.002			The instructing agent is not authorised by the instructed agent (Mandate)
	The amount of the direct debit must not be higher than the defined maximum amount for a single direct debit of the	pac.010	pac.002			The amount of the direct debit must is higher than the defined maximum amount for a single direct

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Will be filled in the next iterations
	sender					debit of the sender
	The amount of the direct debit including the total amount of already processed (means validated) direct debits of the current receiver must not be higher than the maximum amount for direct debits per day defined in the RTGS account of the receiver.	pac.010	pac.002			The amount of the direct debit including the total amount of already processed (means validated) direct debits of the current receiver must be higher than the maximum amount for direct debits per day defined in the RTGS account of the receiver.
	The sum of amounts of the current direct debit including the total amount of already processed (means validated) direct debits of the current sender for the current receiver must not be higher than the maximum amount for direct debits defined for this sender.	pac.010	pac.002			The sum of amounts of the current direct debit including the total amount of already processed (means validated) direct debits of the current sender for the current receiver is higher than the maximum amount for direct debits defined for this sender.
	Urgent priority is allowed for: CB	pac.010	pac.002			Urgent priority is only allowed for central banks
	Payments with urgent priority are only allowed, if payment is: <ul style="list-style-type: none"> • instructing agent = CB or • Code = ASTI or SBTI 	pac.010	pac.002			Priority urgent is not allowed for this payment
	Debtor account is blocked	pac.004	pac.002			Debtor account is blocked

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	settlement time indication must not be used in inbound messages	pac.004	admi.007			settlement time indication must not be used in inbound messages
	An instruction with the same instruction ID sent by the same instructing agent to the same instructed agent will be detected as a duplicate	pac.004	pac.002			An instruction with the same instruction ID sent by the same instructing agent to the same instructed agent was detected as a duplicate
	An instruction where all the following fields are duplicated will be detected as a duplicate: <ul style="list-style-type: none"> - instructing agent - message type - instructed agent - transaction ID - EndtoEnd ID - Settlementdate - Settlement amount 	pac.004	pac.002			The instruction was detected as a duplicate
	Creditor account is blocked	pac.004	pac.002			Creditor account is blocked
	The business sender must be authorized to credit/debit the account as: <ul style="list-style-type: none"> • The owner of the account to be credited/debited or • allowed by contractual arrangement between the third party and both instructing agent and instructed agent to do so (multi-addressee) 	pac.004	pac.002			The business sender is not authorized to credit the account

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	Payments can be sent for the current business day and up to 10 calendar days in advance.	pac.004	pac.002			Payments can be sent for the current business day and up to 10 calendar days in advance.
	If the execution date is on weekend or a RTGS holiday the payment will be rejected	pac.004	pac.002			If the execution date is on weekend or a RTGS holiday the payment will be rejected
	When the value date check is switched off, back valued payments in RTGS are possible. Payments with more than 10 days in advance are not possible anyway. Back valued payments to AS or CLM are also not possible.	pac.004	pac.002			When the value date check is switched off, back valued payments in RTGS are possible. Payments with more than 10 days in advance are not possible anyway. Back valued payments to AS or CLM are also not possible.
	The cut-off time for Customer payments has to be observed	pac.004	pac.002			The cut-off time for Customer payments has not been observed
	The instructing agent of the instruction has to be a RTGS DCA account holder	pac.004	pac.002			The instructing agent of the instruction is no RTGS DCA account holder
	Currency must denominate same currency as accounts indicated for posting.	pac.004	pac.002			Currency does not denominate same currency as accounts indicated for posting.
	IF Charges Agent is present and IF	pac.004	admi.007			IF Charges Agent is

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Non-exhaustive
	BIC is not provided THEN the Name and TownName and the Country must be provided.					present and IF BIC is not provided THEN the Name and TownName and the Country must be provided.
	If a field of party is present and If the BIC is not provided then the Name and TownName and the Country must be provided.	pacs.010	admi.007			If a field of party is present and If the BIC is not provided then the Name and TownName and the Country must be provided.
	If a field of party is present and If the BIC is not provided then the Name and TownName and the Country must be provided.	pacs.004	admi.007			If a field of party is present and If the BIC is not provided then the Name and TownName and the Country must be provided.
	If UltimateDebtor is present and If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the UltimateDebtor.	pacs.004	admi.007			If UltimateDebtor is present and If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the UltimateDebtor.
	If InitiatingParty is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the InitiatingParty.	pacs.004	admi.007			If InitiatingParty is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the InitiatingParty.

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Will be filled in the next iterations
	If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the Debtor.	pacs.004	admi.007			If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the Debtor.
	If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the Creditor.	pacs.004	admi.007			If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the Creditor.
	If UltimateCreditor is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the UltimateCreditor.	pacs.004	admi.007			If UltimateCreditor is present and if OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the UltimateCreditor.
	If ReturnedInstructedAmount is present and the currency is different from the currency in ReturnedInterbankSettlementAmount, then ExchangeRate must be present.	pacs.004	admi.007			If ReturnedInstructedAmount is present and the currency is different from the currency in ReturnedInterbankSettlementAmount, then ExchangeRate must be present.
	If ReturnedInstructedAmount is present and the currency is the same as the currency in ReturnedInterbankSettlementAmount, then Ex-	pacs.004	admi.007			If ReturnedInstructedAmount is present and the currency is the same as the currency

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	changeRate is not allowed.					in ReturnedInterbankSettlementAmount, then ExchangeRate is not allowed.
	If IntermediaryAgent2 is present, then IntermediaryAgent1 must be present.	pacs.004	admi.007			If IntermediaryAgent2 is present, then IntermediaryAgent1 must be present.
	If IntermediaryAgent3 is present, then IntermediaryAgent2 must be present.	pacs.004	admi.007			If IntermediaryAgent3 is present, then IntermediaryAgent2 must be present.
	If PreviousInstructingAgent2 is present, then PreviousInstructingAgent1 must be present.	pacs.004	admi.007			If PreviousInstructingAgent2 is present, then PreviousInstructingAgent1 must be present.
	If PreviousInstructingAgent3 is present, then PreviousInstructingAgent2 must be present.	pacs.004	admi.007			If PreviousInstructingAgent3 is present, then PreviousInstructingAgent2 must be present.
	If Originator is present and If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the Originator.	pacs.004	admi.007			If Originator is present and If OrganisationIdentification/AnyBIC is absent then Name and TownName and Country are mandatory to identify the Originator.
	If Reason/Code is equal to NARR,	pacs.004	admi.007			If Reason/Code is equal

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	then AdditionalInformation must be present.					to NARR, then AdditionalInformation must be present.
	If ChargesInformation is present, then ReturnedInstructedAmount must be present.	pac.004	admi.007			If ChargesInformation is present, then ReturnedInstructedAmount must be present.
	Account Id or Account Owner must be present, but not both.	camt.003	camt.004			Account Id or Account Owner must be present, but not both.
	If Balance Type is AVLB (available liquidity), OPNG (start balance) or CRRT (current balance), then status must be STLD.	camt.004	admi.007			If Balance Type is AVLB (available liquidity), OPNG (start balance) or CRRT (current balance), then status must be STLD.
	If the Balance Type is NOTE (timed payments) or XPCD (projected liquidity), then Status must be PNDG.	camt.004	admi.007			If the Balance Type is NOTE (timed payments) or XPCD (projected liquidity), then Status must be PNDG.
	A transaction may only have one status value for any one DateTimestamp.	camt.005	camt.006			A transaction may only have one status value for any one DateTimestamp.
	At least QueryType or TransactionCriteria must be present. Both can be present together.	camt.005	camt.006			At least QueryType or TransactionCriteria must be present. Both can be present together.
	If NewCriteria is used, at least SearchCriteria or ReturnCriteria must	camt.005	camt.006			If NewCriteria is used, at

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
	be present. Both can be present					least SearchCriteria or ReturnCriteria must be present. Both can be present
	If PaymentTo is used and if MemberIdentification is absent then Country is mandatory.	camt.005	admi.007			If PaymentTo is used and if MemberIdentification is absent then Country is mandatory.
	If PaymentFrom is used and if MemberIdentification is absent then Country is mandatory.	camt.005	admi.007			If PaymentFrom is used and if MemberIdentification is absent then Country is mandatory.
	If PaymentInstructionSatusDateTime and FinalStatus are present, then Final settled (STLD) is the only allowed code.	camt.005	admi.007			If PaymentInstructionSatusDateTime and FinalStatus are present, then Final settled (STLD) is the only allowed code.
	IF PaymentInstructionSatusDateTime is present Then PendingStatus and PendingAndFinalStatus are not allowed.	camt.005	admi.007			IF PaymentInstructionSatusDateTime is present Then PendingStatus and PendingAndFinalStatus are not allowed.
	Only codes STLE= earmarked, PSTL = Pending, ACPD = warehoused and STLM = information period (AS payment during information period) are valid.	camt.005	admi.007			Only codes STLE= earmarked, PSTL = Pending, ACPD = warehoused and STLM = information period (AS payment during

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations	Non-exhaustive	
						information period) are valid.
	If the requestor asks for debit/credit indication he has to query payments for a single account. A multiple occurrence of <AcctId> is not allowed in that case.	camt.005	admi.007			If the requestor asks for debit/credit indication he has to query payments for a single account. A multiple occurrence of <AcctId> is not allowed in that case.
	The Debtor can only be used when related to Debtor BIC of AS XML.	camt.005	camt.006			The Debtor can only be used when related to Debtor BIC of AS XML.
	If AccountEntrySearch is used and AccountIdentification is not present then EntryDate is mandatory.	camt.005	admi.007			If AccountEntrySearch is used and AccountIdentification is not present then EntryDate is mandatory.
	For each [ResolutionOfInvestigationV08], if every occurrence of [ResolutionOfInvestigationV08/Status/Confirmation] has value included in the following list 'PDCR' or 'RJCR' , then at least one occurrence of the following element(s) [ResolutionOfInvestigationV08/CancellationDetails/Transacti	camt.029	admi.007			Element for reason is missing

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Non-exhaustive
	onInforma- tionAndStatus/CancellationStatusReasonInformation/Reason] must be present					
	If at least one occurrence of the following element(s) [ReturnReservationV05/ReportOrError/BusinessReport] is (are) present , then the following element(s) [ReturnReservationV05/ReportOrError/OperationalError] must be absent	camt.047	admi.007			Element for error is missing
	If at least one occurrence of the following element(s) [ReturnReservationV05/ReportOrError/OperationalError] is (are) present , then the following element(s) [ReturnReservationV05/ReportOrError/BusinessReport] must be absent	camt.047	admi.007			Element for report is missing
	RelatedAgents can only be used if the notification is about a payment. RelatedParties can only be used if the notification is about a liquidity transfer.	camt.054	admi.007			RelatedAgents can only be used if the notification is about a payment. RelatedParties can only be used if the notification is about a liquidity transfer.
	Used only for Connected Payments. This element has to be used in combination with TransactionDe-	camt.054	admi.007			Used only for Connected Payments. This element has to be used in combi-

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive			Will be filled in the next iterations		Non-exhaustive
	tails/LocalInstrument.					nation with TransactionDe-tails/LocalInstrument.
	Depending on the service, different references will be used. Payments : (UETR, TransactionID, InstructionId) Services and liquidity management : MsgId Ancillary systems : EndToEndId	camt.054	admi.007			Depending on the service, different references will be used. Payments : (UETR, TransactionID, InstructionId) Services and liquidity management : MsgId Ancillary systems : EndToEndId
	RelatedAgents can only be used if the notification is about a payment. RelatedParties can only be used if the notification is about a liquidity transfer.	camt.054	admi.007			RelatedAgents can only be used if the notification is about a payment. RelatedParties can only be used if the notification is about a liquidity transfer.
	Debtor account is blocked	camt.050	camt.025			Debtor account or creditor account is blocked
	Creditor account is blocked	camt.050	camt.025			Debtor account or creditor account is blocked

Table 267 - RTGS validation rules

Description	User function
When performing a Cash Account create request, the Party Type of the Requestor must be NCB or Payment Bank.	Create Cash Account
Users belonging to NCBs can only create Cash Accounts for Parties that fall under their responsibility according to the Hierarchical Party Model, or TIPS Credit Memorandum Balances linked to Cash Accounts that fall under their responsibility.	Create Cash Account
Users belonging to Payment Banks can only create TIPS Credit Memorandum Balances linked to Cash Accounts that fall under their responsibility.	Create Cash Account
When performing a Cash Account create request, the Restriction Type must refer to an existing Restriction Type with Object Restriction Type equal to Cash Account and belonging to the same system entity of the Cash Account or of the Service Operator.	Create Cash Account
When performing a Cash Account create request, the Valid From specified in the Cash Account Restriction section must be equal to or greater than the current timestamp.	Create Cash Account
When performing a Cash Account create request the Currency Code must refer to an existing instance in CRDM with Settlement Currency set to True or a Currency-Service Link in place with the relevant Service.	Create Cash Account
When performing a Cash Account create request the Floor Notification Amount specified must be less than the Ceiling Notification Amount.	Create Cash Account
When performing a Cash Account create request, the Cash Account Number must be compliant with ISO 20022 standards and it must not be already assigned to any other Cash Account in CRDM.	Create Cash Account
When performing a Cash Account create request the Opening Date must be equal to or greater than the current date and be equal or greater than the Account Holder Opening Date. Furthermore it must be equal to or less than the Account Holder Closing Date.	Create Cash Account
When performing a Cash Account create request to create a T2S DCA, T2S Dedicated Transit Account or T2S CB account, the Linked Account must refer to an existing and open External RTGS Account instance in CRDM.	Create Cash Account
When performing a Cash Account create request, if the Linked Account references an External RTGS Account it must have the same currency code of the Cash Account.	Create Cash Account
When performing a Cash Account create request, in case of request of creation of Cash Account Restriction, the Valid From of the Cash Account Restriction must be equal or greater than the Valid From of the Restriction Type entity.	Create Cash Account
When performing a Cash Account create request, in case of request of creation of Cash Account Restriction, the Valid To of the Cash Account Restriction must be equal or less than the Valid To of the Restriction Type entity.	Create Cash Account

Description	User function
When performing a Cash Account create request the Closing Date specified in the request must be equal to or greater than the Opening Date. Furthermore it must be equal to or less than the Account Holder Closing Date.	Create Cash Account
When performing a Cash Account create request, the Valid To specified in the Cash Account Restriction section must be equal to or greater than the Valid From.	Create Cash Account
When performing a Cash Account create request to create a TIPS Credit Memorandum Balance the Linked Account must refer to an existing Cash Account instance in CRDM with type "TIPS Account" which is open throughout the specified opening period of the TIPS CMB being created.	Create Cash Account
When performing a Cash Account Create request, in case of request for creation of a Cash Account Restriction, the created restriction must not overlap with any other Cash Account Restriction in input having the same Restriction Type.	Create Cash Account
When performing a Cash Account create request, the account holding Party must refer to an existing active and open instance in CRDM with Party Type equal to NCB or Payment Bank.	Create Cash Account
When performing a Cash Account create request, when creating a T2S Dedicated Transit Account, no other account of the same type must be already associated to the relevant currency.	Create Cash Account
When performing a Cash Account create request, when creating a T2S DCA or a T2S CB account, there must be a T2S Dedicated Transit Account related to the relevant currency.	Create Cash Account
When performing a Cash Account create request, when creating a TIPS Account, there must be a TIPS Transit Account related to the relevant currency.	Create Cash Account
When performing a Cash Account create request, when creating a TIPS Transit Account or RTGS Dedicated Transit Account, no other account of the same type must be already associated to the relevant currency.	Create Cash Account
When performing a Cash Account create request, when creating an RTGS DCA, Ancillary System Guarantee funds account, RTGS sub-account, RTGS CB account or RTGS Technical Account, there must be an RTGS Dedicated Transit Account related to the relevant currency.	Create Cash Account
When performing a Cash Account create request check the relation between the Account Type to be created and the Party Type of the account holder.	Create Cash Account
When performing a Cash Account Create request, in case of immediate setup of Cash Account Restriction, the timestamp to be used must take a conventional value which the system will interpret as the current timestamp. Furthermore, no check must be per-	Create Cash Account

Description	User function
formed on such a conventional value in case of four eyes second step or processing of retrieved queued requests.	
When performing a Cash Account Create request, the number of decimals in the values provided for Floor Notification Amount and Ceiling Notification Amount must be compliant with the number of decimals foreseen for the relevant currency.	Create Cash Account
A Standing and Predefined Liquidity Transfer Order can only be created by the NCB or Payment Bank responsible for the account to be debited.	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the specified Cash account to be debited must refer to an existing, active and open instance in T2S.	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the specified Standing and Predefined Liquidity Transfer Order Reference must not be already assigned to an existing and active instance for the same Cash Account.	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the specified Credited Cash account must refer to an existing, active and open instance in CRDM. Furthermore, it must have the same currency as the debited Cash Account.	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the Event Type Code, when specified in the create request, must refer to an active and existing instance in Event Type.	Create Liquidity Transfer Order
Only one predefined liquidity transfer order can be defined to be executed at the same timestamp and/or business event for each Cash account.	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the Dedicated Amount field and the All Cash field cannot be set both to True.	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the Amount must be set to zero if the Dedicated Amount field or the All Cash field are set to True.	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the Amount cannot be set to zero if the Dedicated Amount field and the All Cash field are set to False.	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the Valid To specified in a Liquidity Transfer Order maintenance request must be equal to or greater than the current date, equal to or greater than the Valid From, and not greater than the debited account's closing date (if applicable).	Create Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Create request, the Valid To specified in a Standing and Predefined Liquidity Transfer Order maintenance request must be equal to or greater than the current date, equal to or greater than the Valid From, and not greater than the debited account's closing date (if applicable).	Create Liquidity Transfer Order

Description	User function
<p>finance request must be equal to or greater than the current date, equal to or greater than the Valid From, and not greater than the debited account's closing date (if applicable).</p>	
<p>When performing a Standing and Predefined Liquidity Transfer Order Create request, there cannot be more than one instance for each Cash Account linked to Event Type Code 'CARL' in a given time period.</p>	<p>Create Liquidity Transfer Order</p>
<p>When performing a Liquidity Transfer Order Create request, the number of decimals in the value provided for Amount must be compliant with the number of decimals foreseen for the relevant currency.</p>	<p>Create Liquidity Transfer Order</p>
<p>When performing a Cash Account delete or restore request, the Party Type of the Requestor must be NCB or Payment Bank.</p>	<p>Delete Cash Account</p>
<p>Users belonging to NCBs can only delete or restore Cash Accounts for Parties that fall under their responsibility according to the Hierarchical Party Model, or TIPS Credit Memorandum Balances linked to Cash Accounts that fall under their responsibility.</p>	
<p>Users belonging to Payment Banks can only delete or restore TIPS Credit Memorandum Balances linked to Cash Accounts that fall under their responsibility.</p>	
<p>The delete requests of Cash Accounts must refer to an existing and active instance. The account to be deleted must be already closed or must have Opening Date greater than the current date.</p>	<p>Delete Cash Account</p>
<p>When performing a Cash Account restore request it must refer to an existing and deleted Cash Account. The account to be restored must have Closing date equal to or earlier than the Current Business date or Opening date equal to or later than the Current Business date; in addition, the Opening date must be equal to or later than the Account Holder Opening Date and the Closing Date must be equal to or earlier than the Account Holder Closing Date.</p>	<p>Delete Cash Account</p>
<p>When performing a Cash Account restore request, when restoring a T2S Dedicated Transit Account, RTGS Dedicated Transit Account or a TIPS Transit Account, no other Transit Account must be already associated to the relevant currency in the same validity period.</p>	<p>Delete Cash Account</p>
<p>When performing a Cash Account delete request, in case of deletion of a future T2S Dedicated Transit Account, RTGS Dedicated Transit Account or TIPS Transit Account, no active Cash Accounts with the same currency for T2S, RTGS or TIPS respectively must exist in CRDM.</p>	<p>Delete Cash Account</p>
<p>A Cash Account cannot be deleted if there still are valid instances of the following entities linked to it: Liquidity Transfer Order, Liquidity Transfer Order Link Set, Credit Memorandum Balance, TIPS Credit Memorandum Balance-type Cash Account.</p>	<p>Delete Cash Account</p>

Description	User function
When performing a Cash Account restore request the currency code of the Cash Account to be restored must refer to an existing currency code in CRDM with Settlement Currency set to True or a Currency-Service Link in place with the relevant Service.	Delete Cash Account
When performing a Cash Account restore request the account holder must be an existing and active Party in CRDM with Party Type equal to NCB or Payment Bank.	Delete Cash Account
When performing a Cash Account restore request, all restrictions associated to the Cash Account to be restored must refer to existing Restriction Types whose Object Restriction Type is Cash Account.	Delete Cash Account
When performing a Cash Account restore request the Linked Account of the T2S DCA, T2S CB account or T2S Dedicated Transit Account to be restored must refer to an existing External RTGS Account in T2S.	Delete Cash Account
When performing a Cash Account restore request the Linked Account of the TIPS Credit Memorandum Balance to be restored must refer to an existing and open TIPS Account in CRDM.	Delete Cash Account
When performing a Cash Account restore request, if the Cash Account to be restored is linked to an External RTGS Account, they must have the same currency code.	Delete Cash Account
When performing a Cash Account restore request, the validity period of the Cash Account to be restored must be consistent with the validity period of the relevant Transit Account.	Delete Cash Account
When performing a Cash Account restore request the relation between the Account Type to be restored and the Party Type of the account holder is checked.	Delete Cash Account
A Standing and Predefined Liquidity Transfer Order can only be deleted by the NCB or Payment Bank responsible for the account to be debited.	Delete Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order delete request, it must refer to an existing and active instance.	Delete Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order restore request, it must refer to an existing and deleted Standing and Predefined Liquidity Transfer Order.	Delete Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order restore request, the restored credited Cash Account must refer to an existing and open account in CRDM.	Delete Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order restore request, the restored debited Cash Account must refer to an existing and open account in CRDM.	Delete Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order restore request the Standing and Predefined Liquidity Transfer Order Reference to be restored must not	Delete Liquidity Transfer Order

Description	User function
be already assigned to an existing and active instance for the same Cash Account.	
When performing a Standing and Predefined Liquidity Transfer Order restore request, only one predefined liquidity transfer order can be defined to be executed at the same timestamp and/or business event for each cash account.	Delete Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order restore request, the Event Type Code to be restored must refer to an existing code in Event type.	Delete Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Restore request, there cannot be more than one instance for each Cash Account linked to Event Type Code 'CARL' in a given time period.	Delete Liquidity Transfer Order
When performing a Cash Account update request the Party Type of the Requestor must be NCB or Payment Bank.	Update Cash Account
Users belonging to NCBs can only update Cash Accounts for Parties that fall under their responsibility according to the Hierarchical Party Model, or TIPS Credit Memorandum Balances linked to Cash Accounts that fall under their responsibility.	Update Cash Account
Users belonging to Payment Banks can only update TIPS Credit Memorandum Balances linked to Cash Accounts that fall under their responsibility.	Update Cash Account
The update requests of a Cash Account must refer to an existing and active account. Furthermore, the Closing Date must be equal to or greater than the current date.	Update Cash Account
When performing a Cash Account update request, in case of request of creation of Cash Account Restriction, the Restriction Type must refer to an existing Restriction Type with Object Restriction Type equal to Cash Account and belonging to the same system entity of the Cash Account or of the Service Operator.	Update Cash Account
A Cash Account cannot be closed if there still are valid instances of the following entities linked to it: Liquidity Transfer Order, Liquidity Transfer Order Link Set.	Update Cash Account
When performing a Cash Account update request, any update of the Opening Date and Closing Date must be consistent with the validity periods of other existing Cash Accounts with type 'TIPS Credit Memorandum Balance' linking to it.	Update Cash Account
When performing a Cash Account update request, the Floor Notification Amount must be less than the Ceiling Notification Amount	Update Cash Account
When performing a Cash Account update request, the Linked Account can be specified only for TIPS Credit Memorandum Balances, T2S Dedicated Transit Accounts, T2S CB Accounts and T2S DCAs.	Update Cash Account
When performing a Cash Account update request, the Linked Account, when it refers to an External RTGS Cash Account, must refer to an existing and open instance in CRDM.	Update Cash Account

Description	User function
When performing a Cash Account update request, if the Linked Account references an External RTGS Account it must have the same currency code of the Cash Account.	Update Cash Account
When performing a Cash Account update request, the Closing Date must be equal to or greater than the current date and equal to or greater than the Cash Account Opening Date. Furthermore it must be equal to or less than the Account Holder Closing Date.	Update Cash Account
When performing a Cash Account update request, in case of request of creation of Cash Account Restriction, the Valid From must be equal to or greater than the current timestamp.	Update Cash Account
When performing a Cash Account update request, in case of request of creation/update of Cash Account Restriction, the Valid To specified in the Cash Account Restriction section must be equal to or greater than the current timestamp and must be equal to or greater than the Valid From.	Update Cash Account
When performing a Cash Account update request, the Opening Date can be updated only if the existing one is greater than the current date and the new one must be equal to or greater than the current date. Furthermore it must be equal to or greater than the Account Holder Opening Date and equal to or less than the Account Holder Closing Date.	Update Cash Account
When performing a Cash Account update request on the Linked Account, Opening Date and/or Closing Date of a TIPS Credit Memorandum Balance, the Linked Account must refer to an existing Cash Account instance in CRDM with type "TIPS Cash Account" which is open throughout the specified validity period of the TIPS CMB being updated.	Update Cash Account
When performing a Cash Account update request, in case of request of deletion of Cash Account Restriction, the Valid From must be greater than the current timestamp or the Cash Account Restriction must be closed.	Update Cash Account
When performing a Cash Account update request, case of request of update of Cash Account Restriction, it must refer to an existing Cash Account Restriction with a non-past Valid To.	Update Cash Account
When performing a Cash Account update request, the specified Currency Code must refer to the one already linked to the existing Cash Account.	Update Cash Account
When performing a Cash Account update request, in case of request of creation of Cash Account Restriction, the Valid From of the Cash Account Restriction must be equal or greater than the Valid From of the Restriction Type.	Update Cash Account
When performing a Cash Account update request, in case of request of creation of Cash Account Restriction, the Valid To of the Cash Account Restriction must be equal or less than the Valid To of the Restriction Type.	Update Cash Account

Description	User function
When performing a Cash Account Update request, in case of request for creation/update of Cash Account Restriction, the new or updated restriction must not overlap with any other Cash Account Restrictions having the same Restriction Type on the same Cash Account.	Update Cash Account
When performing a Cash Account update request, in case of update of the Opening or Closing Date of an RTGS Dedicated Transit Account, T2S Dedicated Transit Account or TIPS Transit Account, no active Cash Account with the same currency for RTGS, T2S and TIPS respectively must be open outside of the Transit Account validity period.	Update Cash Account
When performing a Cash Account Update request, the validity period of the Cash Account must be contained within the validity period of the relevant Transit Account.	Update Cash Account
When performing a Cash Account Update request, Cash Accounts for TIPS, RTGS and CLM require an existing and active Party-Service Link to be in place between the Owner Party and the relevant Service for the relevant validity period.	Update Cash Account
When performing a Cash Account Update request, in case of immediate setup or removal of Cash Account Restriction, the timestamp to be used must take a conventional value which the system will interpret as the current timestamp. Furthermore, no check must be performed on such a conventional value in case of four eyes second step or processing of retrieved queued requests.	Update Cash Account
When performing a Cash Account Update request, the number of decimals in the values provided for Floor Notification Amount and Ceiling Notification Amount must be compliant with the number of decimals foreseen for the relevant currency.	Update Cash Account
A Standing and Predefined Liquidity Transfer Order can only be updated by the NCB or Payment Bank responsible for the account to be debited.	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order update request, it must refer to an existing and active instance in CRDM.	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Update request, if the Order Type is 'Predefined', the Valid From and Valid To must contain identical values.	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order update request, the Event Type Code, when specified in the update request, must refer to an active and existing instance in Event Type.	Update Liquidity Transfer Order
Only one predefined liquidity transfer order can be defined to be executed at the same timestamp and/or business event for each cash account.	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order update request, the Dedicated Amount field and the All Cash field cannot be set both to True.	Update Liquidity Transfer Order

Description	User function
When performing a Standing and Predefined Liquidity Transfer Order update request, the Amount must be set to zero if the Dedicated Amount field or the All Cash field are set to True.	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order update request, the Amount cannot be set to zero if the Dedicated Amount field and the All Cash field are set to False.	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order update request, the Valid To must be equal to or greater than the current date, greater than the valid from and not greater than the Cash account's closing date (if applicable).	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order update request, the Valid From specified must be equal to or greater than the current date and not greater than the Cash account's closing date (if applicable).	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order update request, the Valid From can be modified only if the existing one is greater than the current date.	Update Liquidity Transfer Order
When performing a Standing and Predefined Liquidity Transfer Order Update request, there cannot be more than one instance for each Cash Account linked to Event Type Code 'CARL' in a given time period.	Update Liquidity Transfer Order
When performing a Liquidity Transfer Order Update request, the number of decimals in the value provided for Amount must be compliant with the number of decimals foreseen for the relevant currency.	Update Liquidity Transfer Order
A Party can be created only by Service Operator, CSD or NCB. A user belonging to a CSD or NCB can only create parties that fall under their responsibility according to the Hierarchical Party Model.	Create Party
When performing a Party Create request, the 'System Entity' specified in input must refer to an existing instance in CRDM, and its type must be consistent with the 'Party Type' specified in input.	Create Party
When performing a Party Create request, the Party Type cannot be 'CSD' or 'NCB' if there is already a CSD or NCB defined within the System Entity.	Create Party
When performing a Party Create request, the 'Party Mnemonic' specified in the Party Code section must not be already assigned to another active Party belonging to the same System Entity and having the same Parent BIC.	Create Party
When performing a Party Create request, the 'Country Code' specified in the Party Address section must refer to an existing Country Code in CRDM.	Create Party
When performing a Party Create request, In case of request for creation of Party Restriction, the created restriction type must refer to an existing type in [Restriction Type]	Create Party

Description	User function
entity with Object Restriction Type 'Party'.	
When performing a Party Create request, In case of request for creation of Party Restriction, the created restriction type must not overlap with any other Party Restriction in input having the same [Restriction Type].	Create Party
When performing a Party Create request, the 'Party Mnemonic' specified in the Party Code section (when its type is BIC) must exist in the BIC Directory.	Create Party
When performing a Party Create request, the Party Opening Date specified in the request must be equal to or greater than the current date.	Create Party
When performing a Party Create request, the Party Closing Date, if specified, must be equal to or greater than the current date and greater than the Opening Date.	Create Party
When performing a Party Create request, the Party Restriction 'Valid To', when specified, must be equal to or greater than the current timestamp, equal to or greater than the Party Restriction Valid From and equal to or less than the Valid To of the relevant Restriction Type entity.	Create Party
When performing a Party Create request, the Party Restriction 'Valid From', when specified, must be equal to or greater than the current timestamp and equal to or greater than the Valid From of the relevant Restriction Type entity and equal to or less than the Valid To of the relevant Restriction Type entity.	Create Party
When performing a Party Create request, in case of request for creation of Market-Specific Party Attribute Value, it must refer to an existing Market-Specific Attribute with Type "Party" and it must belong to the relevant System Entity.	Create Party
When performing a Party Create request, in case of request for creation of Market-Specific Party Attribute Value, it must be unique within its System Entity in case it is defined as such in CRDM.	Create Party
When performing a Party Create request, in case of request for creation of a Market-Specific Party Attribute, the Market-Specific Attribute Value must be present if the relevant Market-Specific Attribute is defined as mandatory.	Create Party
When performing a Party create request the Market-Specific Party Attribute Value must be compliant with the values or rules defined in the relevant Attribute Domain.	Create Party
When performing a Party Create request, the 'Valid From' specified in the Party Code section, must be equal to the current business date.	Create Party
When performing a Party Create request, the 'Valid From' specified in the Party Address section, must be equal to the current business date.	Create Party
When performing a Party Create request, the 'Valid From' specified in the Party Name	Create Party

Description	User function
section, must be equal to the current business date.	
When performing a Party Create request, the Collateralisation Procedure specified in Autocollateralisation Rule section, must be equal to Repo in case the Party Type is not NCB.	Create Party
When performing a Party Create request, the Party Address section must not be filled in if the Party Type is CSD Participant.	Create Party
When performing a Party Create request, the Autocollateralisation Rule section must not be filled in if the Party Type is not NCB or Payment Bank.	Create Party
When performing a Party Create request, in case of immediate setup of Party Restriction, the timestamp to be used must take a conventional value which the system will interpret as the current timestamp. Furthermore, no check must be performed on such a conventional value in case of four eyes second step or processing of retrieved queued requests.	Create Party
Party can only be deleted or restored by the Service Operator, CSD or NCB. A user belonging to a CSD or NCB can only delete or restore parties that fall under their responsibility according to the Hierarchical Party Model.	Delete Party
When performing a Party Delete request, it must refer to an existing, active and closed Party or with a future Opening date.	Delete Party
When performing a Party Restore request, it must refer to an existing and deleted Party already closed or with an Opening date equal to or greater than the current business date.	Delete Party
When performing a Party Restore request, the Party Type cannot be 'CSD' or 'NCB' if there is already a CSD or NCB defined within the System Entity.	Delete Party
When performing a Party Restore request, the 'PartyMnemonic specified in the PartyCode section must not be already assigned to an active party having the same Party Type and belonging to the same System Entity and having the same Parent BIC in case the Party to be restored is not closed.	Delete Party
When performing a Party Restore request, the 'Country Code' specified in the Party Address section must refer to an existing Country Code in CRDM.	Delete Party
When performing a Party Restore request, the 'Restriction Type' specified in the Party Restriction section must refer to an existing type in CRDM available for the relevant System Entity.	Delete Party
In case of request to delete a Party, all the linked instances in a higher position within the deletion hierarchy (i.e. Securities Account, Cash Account, External RTGS Account, Security CSD Link, CSD Account Link and Party) must be deleted.	Delete Party

Description	User function
When performing a Party Restore request, the 'Technical Address' specified in the Party Technical Address section must exist in the BIC Directory, when its type is BIC.	Delete Party
When performing a Party Restore request, the 'Party Mnemonic' specified in the Party Code section (when its type is BIC) must exist in the BIC Directory.	Delete Party
When performing a Party restore request, the Party Restriction 'Valid To', when specified, must be equal to or less than the Valid To of the relevant Restriction Type entity.	Delete Party
When performing a Party restore request, the Party Restriction 'Valid From', when specified, must be equal to or greater than the Valid From of the relevant Restriction Type entity and equal to or less than the Valid To of the relevant Restriction Type entity.	Delete Party
In case of restore of Market-Specific Party Attribute Value, it must refer to an existing Market-Specific Attribute with Type "Party" and it must belong to the relevant System Entity.	Delete Party
In case of request for restore of Market-Specific Party Attribute Value, the Value must be unique (within its System Entity) if it is defined as "unique" in [Market-Specific Attribute] entity.	Delete Party
When performing a Party Restore request, the Market-Specific Attribute Value must be present if the relevant Market-Specific Attribute is defined as mandatory.	Delete Party
When performing a Party restore request the Market-Specific Party Attribute Value must be compliant with the values or rules defined in the relevant Attribute Domain.	Delete Party
Party can only be updated by the Service Operator, CSD or NCB. A user belonging to a CSD or NCB can only update parties that fall under their responsibility according to the Hierarchical Party Model.	Update Party
When performing a Party Update request, it must refer to an existing and active Party whose Closing Date is equal to or greater than the current business date.	Update Party
When performing a Party Update request, the update request of a "minor" entity (such as Party Name, Party code, Party Address, Market-Specific Attribute, Party Restriction, AutoCollateralisation Rule) must refer to an existing and active instance with a non-past Valid To, where applicable.	Update Party
Each party must have at least one party technical address.	Update Party
When performing a Party Update request, in case of request for creation of Party Technical Address, the PTA specified cannot be identical to a PTA already linked to the relevant Party.	Update Party
When performing a Party Update request, the create request of a historical (i.e. which has the validity date) "minor" entity (such as Party Name Party code, Party Address)	Update Party

Description	User function
cannot have a past validity date.	
When performing a Party Update request, the delete request of a historical (i.e. which has the validity date) "minor" entity (such as Party Name, Party Address) cannot refer to an entity having a past validity date. This does not apply to the Party Code, for which only the currently active entity cannot be deleted.	Update Party
When performing a Party Update request, the 'Party Mnemonic' specified in the Party Code section must not be already assigned, as an active instance, to another active Party belonging to the same System Entity and having the same Parent BIC.	Update Party
When performing a Party Update request, the 'Country Code' specified in the Party Address section must refer to an existing Country Code in CRDM.	Update Party
When performing a Party Update request, in case of request for creation of Party Restriction, the created restriction type must refer to an existing type in [Restriction Type] entity with Object Restriction Type 'Party'.	Update Party
When performing a Party Update request, in case of request for deletion of Party Restriction, it must refer to a closed instance or its Valid From must be greater than the current timestamp.	Update Party
When performing a Party Update request, in case of request to close a Party, all the linked instances in a higher position within the deletion hierarchy (i.e. Securities Account, Cash Account, External RTGS Account, Security CSD Link and CSD Account link, Party) must be closed or deleted.	Update Party
When performing a Party Update request, the 'Party Mnemonic' specified in the Party Code section (when its type is BIC) must exist in the BIC Directory.	Update Party
When performing a Party Update request, in case of Closing of [Party], the specified 'Closing Date' must be equal to or greater than the current business date.	Update Party
When performing a Party Update request, it is only possible to update the 'Opening Date' if it is greater than the current business date. The new specified value must be equal to or greater than the current business date and it must not be greater than the opening date of the Cash Account for which the party is the Account holder.	Update Party
When performing a Party Update request, the specified Party Restriction 'Valid To' must be equal to or greater than the current timestamp, greater than the relevant Valid From, equal to or greater than the Valid From of the relevant Restriction Type and equal to or less than the Valid To of the relevant Restriction Type.	Update Party
When performing a Party update request, the Valid From specified in a Party Restriction create request must be equal to or greater than the current timestamp, equal to or greater than the Valid From of the relevant Restriction Type and equal to or less than the	Update Party

Description	User function
Valid To of the relevant Restriction Type.	
When performing a Party Update request, in case of request for creation/update of Market-Specific Party Attribute Value, it must refer to an existing Market-Specific Attribute with Type "Party" and it must belong to the relevant System Entity.	Update Party
When performing a Party Update request, in case of request for creation/update of Market-Specific Party Attribute Value, it must be unique within its System Entity in case it is defined as such in CRDM.	Update Party
When performing a Party Update request, in case of request for deletion of a Market-Specific Party Attribute, the relevant [Market-Specific Attribute] entity must not be defined as "mandatory".	Update Party
When performing a Party Update request, in case of request for update of a Market-Specific Party Attribute, the Market-Specific Attribute Value must be present if the relevant [Market-Specific Attribute] is defined as mandatory.	Update Party
When performing a Party update request the Market-Specific Party Attribute Value must be compliant with the values or rules defined in the relevant Attribute Domain.	Update Party
When performing a Party Update request, each Market-Specific Attribute can have no more than one value for a given Party.	Update Party
When performing a Party Update request, in case of request for creation/update of Party Restriction, the new or updated restriction must not overlap with any other Party Restriction having the same Restriction Type on the same Party.	Update Party
When performing a Party update request, the Collateralisation Procedure specified in Autocollateralisation Rule section, must be equal to Repo in case the Party Type is not NCB.	Update Party
When performing a Party update request, the Party Address section must not be filled in if the Party Type is CSD Participant.	Update Party
When performing a Party update request, the Autocollateralisation Rule section must not be filled in if the Party Type is not NCB or Payment Bank.	Update Party
When performing a Party update request, the request of creation of the Autocollateralisation Rule is not allowed in case Rules have already been defined.	Update Party
When performing a Party Update request to change the Party BIC, there cannot be more than one Party, besides the CB, with the same BIC linked to the same Service (TIPS, RTGS or CLM).	Update Party
When performing a Party Update request to change the Party BIC, there cannot be more than one User flagged as Main User for the same Certificate DN and the same Party	Update Party

Description	User function
BIC.	
When performing a Party Update request, the update request of a historical "minor" entity (such as Party Name, Party Address) must refer to an instance currently in use or having a future validity.	Update Party
When performing a Party Update request, the update request of Party Code must refer to an instance having a future validity.	Update Party
When performing a Party Update request, in case of immediate setup or removal of Party Restriction, the timestamp to be used must take a conventional value which the system will interpret as the current timestamp. Furthermore, no check must be performed on such a conventional value in case of four eyes second step or processing of retrieved queued requests.	Update Party
When performing a request to create a Limit, the requestor must be authorised to create the requested data according to the following: A Service Operator user can create all data A NCB user can create only Limits for RTGS DCAs, T2S CMBs and TIPS CMBs belonging to its own System Entity A Payment Bank user can create only Limits for RTGS DCAs, non-primary T2S CMBs and TIPS CMBs linked to its own Cash Account	Create Limit
When performing a Limit create request, the Cash Account specified must refer to an existing and active instance in CRDM.	Create Limit
When performing an autocollateralisation, external guarantee or unsecured credit Limit create request, the BIC+BIC Branch Code specified must refer to an existing and active BIC+BIC Branch Code in BIC directory.	Create Limit
When performing an autocollateralisation, external guarantee or unsecured credit limit create request , the Limit Type must be Autocollateralisation if the relevant CMB is a primary one.	Create Limit
When performing an autocollateralisation, external guarantee or unsecured credit limit create request, the Limit Value must be set to zero for Primary CMB if the Regular Securities Account or the NCB Cash Account for the relevant CMB are not defined.	Create Limit
When performing an autocollateralisation, external guarantee or unsecured credit limit create request, the Limit Value must be set to zero if the Receiving Securities Account for the relevant CMB are not defined for Repo and Pledge countries.	Create Limit
When performing an autocollateralisation, external guarantee or unsecured credit limit create request, the BIC+BIC Branch Code specified must be authorised to use the Cash Account provided in input.	Create Limit

Description	User function
When performing a limit create request, if the limit type is TIPS CMB Limit then the Cash Account type must be TIPS CMB; if the limit type is RTGS DCA Limit, the Cash Account type must be RTGS DCA; if the limit type is autocollateralisation, external guarantee or unsecured credit the Cash Account type must be T2S DCA or T2S CB account.	Create Limit
When performing a Limit create request, it must be verified that no Limit has already been defined for the BIC+BIC Branch Code (if present), Cash Account, Valid From and Limit Type provided in input.	Create Limit
When performing a Limit create request, the Valid From date must be equal to or greater than the current date.	Create Limit
When performing a Limit Create request, the number of decimals in the value provided for Limit Amount must be compliant with the number of decimals foreseen for the relevant currency.	Create Limit
When performing a request to delete a Limit, the requestor must be authorised to delete the requested data according to the following: A System Operator user can delete all data A NCB user can delete only Limits for RTGS DCAs, T2S CMBs and TIPS CMBs belonging to its own System Entity A Payment Bank user can delete only Limits for RTGS DCAs, non-primary T2S CMBs and TIPS CMBs linked to its own Cash Account	Delete Limit
The delete requests of an autocollateralisation, external guarantee or unsecured credit Limit must refer to an existing and active instance whose Limit Amount is equal to zero.	Delete Limit
The restore requests of a Limit must refer to an existing and deleted instance.	Delete Limit
When performing an autocollateralisation, external guarantee or unsecured credit Limit restore request, the Credit Memorandum Balance Identifier must refer to an existing and active CMB instance in CRDM.	Delete Limit
When performing a TIPS CMB Limit restore request, the Credit Memorandum Balance Identifier must refer to an existing and active Cash Account instance in CRDM with Account Type equal to TIPS CMB.	Delete Limit
When performing an RTGS DCA Limit restore request, the Cash Account Identifier must refer to an existing and active Cash Account instance in CRDM with Account Type equal to RTGS DCA.	Delete Limit
When performing a Limit restore request, the Valid From date must be equal to or greater than the current date.	Delete Limit
When performing a request to update a Limit, the requestor must be authorised to update the requested data according to the following:	Update Limit

Description	User function
<p>A Service Operator user can update all data</p> <p>A NCB user can update only Limits for RTGS DCAs, T2S CMBs and TIPS CMBs belonging to its own System Entity</p> <p>A Payment Bank user can update only Limits for RTGS DCAs, non-primary T2S CMBs and TIPS CMBs linked to its own Cash Account</p>	
The update requests of a Limit must refer to an existing and active instance.	Update Limit
When performing an autocollateralisation, external guarantee or unsecured credit Limit update request, the Limit Value must be set to zero for Primary CMB if the Regular Securities Account or the NCB Cash Account for the relevant CMB are not defined.	Update Limit
When performing an autocollateralisation, external guarantee or unsecured credit Limit update request, the Limit Value must be set to zero if the Receiving Securities Account for the relevant CMB are not defined for Repo and Pledge countries.	Update Limit
When performing a Limit Update request, the number of decimals in the value provided for Limit Amount must be compliant with the number of decimals foreseen for the relevant currency.	Update Limit
Direct Debit Mandate can be created only by Service Operator, NCBs or Payment Banks.	Create Direct Debit Mandate
Users belonging to NCBs can only create Direct Debit Mandates on Cash Accounts within their System Entity.	Create Direct Debit Mandate
Users belonging to Payment Banks can only create Direct Debit Mandates on Cash Accounts they are defined as owners of.	Create Direct Debit Mandate
The From Account must be an existing and active Cash Account in the data scope of the requestor with account type equal to "RTGS Dedicated Cash Account".	Create Direct Debit Mandate
The Payee Party Identifier must refer to an existing Party in CRDM with party type equal to "Payment Bank".	Create Direct Debit Mandate
The Valid From date must be equal to or later than the current business date and equal to or later than the Opening Date of the specified From Account.	Create Direct Debit Mandate
The Valid To must be equal to or later than the current business date, equal to or later than the Valid From, and equal to or earlier than the Closing Date of the specified From Account.	Create Direct Debit Mandate
At any given moment, there can be no more than one Direct Debit Mandate between the same From Account – Payee Party pair.	Create Direct Debit Mandate
Direct Debit Mandate can be deleted and restored only by Service Operator, NCBs or Payment Banks.	Delete Direct Debit Mandate

Description	User function
Users belonging to NCBs can only delete/restore Direct Debit Mandates on Cash Accounts within their System Entity.	Delete Direct Debit Mandate
Users belonging to Payment Banks can only delete/restore Direct Debit Mandates on Cash Accounts they are defined as owners of.	Delete Direct Debit Mandate
In a delete operation, the Direct Debit Mandate identifier must refer to an existing and active Direct Debit Mandate with future Valid From or past Valid To.	Delete Direct Debit Mandate
In a restore operation, the Direct Debit Mandate identifier must refer to an existing and deleted Direct Debit Mandate with future Valid From or past Valid To.	Delete Direct Debit Mandate
In a restore operation, the From Account must be an existing and active Cash Account in the data scope of the requestor with account type equal to "RTGS Dedicated Cash Account".	Delete Direct Debit Mandate
In a restore operation, the Payee Party Identifier must refer to an existing Party in CRDM with party type equal to "Payment Bank".	Delete Direct Debit Mandate
At any given moment, there can be no more than one Direct Debit Mandate between the same From Account – Payee Party pair.	Delete Direct Debit Mandate
Direct Debit Mandate can be updated only by Service Operator, NCBs or Payment Banks.	Update Direct Debit Mandate
Users belonging to NCBs can only update Direct Debit Mandates on Cash Accounts within their System Entity.	Update Direct Debit Mandate
Users belonging to Payment Banks can only update Direct Debit Mandates on Cash Accounts they are defined as owners of.	Update Direct Debit Mandate
The Direct Debit Mandate identifier must refer to an existing and active Direct Debit Mandate with future Valid To.	Update Direct Debit Mandate
The Valid From date must be equal to or later than the current business date and equal to or later than the Opening Date of the specified From Account.	Update Direct Debit Mandate
The Valid To must be equal to or later than the current business date, equal to or later than the Valid From, and equal to or earlier than the Closing Date of the specified From Account.	Update Direct Debit Mandate
At any given moment, there can be no more than one Direct Debit Mandate between the same From Account – Payee Party pair.	Update Direct Debit Mandate
When performing a Standing and Predefined Liquidity Transfer Order Create request, a From Date has to be defined.	Update Liquidity Transfer Order
For the user query "Liquidity Transfer Order Detail Query (STDL)" the following search	Query Liquidity Transfer Order

Description	User function
<p>criteria are allowed:</p> <ul style="list-style-type: none"> - Account Identification - Party BIC 	
In case a Cash Account Identification is specified, it has to be known in CRDM.	Query Liquidity Transfer Order
The party has to be known in CRDM.	Query Liquidity Transfer Order/Query Account
Content for element 'Id' must match 'MsgId/Id'	Query Cash Account/Create Cash Account
Content for element 'CreDtTm' must match 'MsgId/CreDtTm'	Query Cash Account/Create Cash Account
Content for element "Org/FullGlnm" must match "OrgId/BIC"	Query Cash Account/Create Cash Account
Content for element 'Org/CtryOfOpr' must match 5th and 6th chars of element 'OrgId/BIC'	Query Cash Account/Create Cash Account
<p>For the Cash Account Reference Data Query the following search criteria are allowed:</p> <ul style="list-style-type: none"> - Cash Account Identifier - Account Type - Currency - Opening Date - Account Owner BIC - Closing Date 	Query Cash Account
<p>For the Cash Account Reference Data Query, at least one of the following search criteria fields should be present:</p> <ul style="list-style-type: none"> - Cash Account Identifier - Account Type - Currency - Opening Date - Account Owner BIC - Closing Date 	Query Cash Account
In case a Cash Account Identification is specified, it has to be known in CRDM.	Query Cash Account
In case a Currency is specified, it has to be known in CRDM.	Query Cash Account
In case the Date From and Date To are specified as ranges, the From value of the Date From has to be before or equal to the To value of the Date To.	Query Cash Account/Query Party
In case the To Date is stated as a range, the From value of the Date To has to be before or equal to the To value of the Date To.	Query Cash Account

Description	User function
In case the Date From is stated as a range, the From value of the Date From has to be before or equal to the To value of the Date From.	Query Cash Account/Query Party
When performing a request to read an Audit Trail, the requestor must be authorised to access the requested data.	Query Cash Account Audit Trail/Query Party Audit Trail
A request to read an Audit Trail must refer to existing data in CRDM.	Query Cash Account Audit Trail/Query Party Audit Trail
At least one of the following search criteria fields should be present, if the tag search criteria is specified for query Cash Account Audit Trail:	Query Cash Account Audit Trail
- CashAccountId	
- DatePeriod	
In case the Date is stated as a range, the Date From has to be before or equal to the Date To.	Query Cash Account Audit Trail/Query Party/Query Party Audit Trail
When performing a request to read a Party, the requestor must be authorised to access the requested data.	Query Party
For the user query 'Party Reference Data Query (PYRD)' the following search criteria are allowed:	Query Party
- BIC of the Party	
- Parent BIC of the Party (NCB BIC or CSD BIC)	
- Party Type	
- Opening Date	
- Closing Date	
At least one of the following search criteria fields should be present, if the tag search criteria is specified for query 'Party Reference Data Query (PYRD) ':	Query Party
- BIC of the Party	
- BIC of the CSD	
- BIC of the NCB	
- Party Type	
- Opening Date	
- Closing Date	
In case a Party BIC is specified, it has to be known in CRDM.	Query Party
In case the Date is stated as a range, the Date From has to be before or equal to the Date To.	Query Party
At least one of the following search criteria fields should be present, if the tag search criteria is specified for query Party Audit Trail:	Query Party Audit Trail

Description	User function
- PartyId - DatePeriod	
When performing a Party Create request, the Party Address section must be filled in if the Party Type is different than CSD Participant.	Create Party
In case a Cash Account Identification is specified, it has to be known in CRDM.	Query Direct Debit Mandate
The Creditor Party has to be known in CRDM.	Query Direct Debit Mandate

Table 268 - CRDM validation rules

rule id	description	inbound message	outbound message	code field	reason code	error text
ICSA010	The digital signature has to be valid.	head.001	admi.007		I071	Digital signature is not valid.
ICSA010	The digital signature has to be valid.	head.002	admi.007		I071	Digital signature is not valid.
ICAA001	The invoked TARGET Service responds to the query request within the timeout limit. Message based or file based store and forward network service will be used.	any query message	admi.007		I074	The invoked TARGET Service cannot respond to the query request within the timeout limit. Store and forward network service will be used.
ICAA002	The invoked TARGET Service responds to the query request via file store and forward network service as the query response exceeds the real time message based network service size (oversize handling).	any query message	admi.007		I076	The invoked TARGET Service cannot respond via message based network service due to size restriction. File store and forward network service will be used.

Table 269 - ESMIG validation rules

15.2 Digital signature on business layer

Will be completed in v2.0.

15.3 Mechanism and introduction for signature constructions

This annex outlines how signatures are constructed for the business messages. The following business message types have been identified.

message type 1: file with multiple ISO 20022 messages

message type 2: single ISO20022 BAH and message

The design goal for the proposed construction of signatures in the following sections is that as much as possible is handled by standard XML Digital Signature processing specifications and as little as possible by specific processing. This makes it less likely that errors and/or discrepancies occur in the different implementations, and therefore improve the overall security of the solution.

15.4 Use of XML and canonicalisation algorithm

Exclusive XML canonicalization³¹ has to be performed for above mentioned business messages on extracted data. It is important to ensure a context free extraction otherwise the signatures will be broken if either the message or the signature itself is modified due to inherited namespaces.

This implies that the canonicalization algorithm specified in the "SignedInfo" element and in all the references should be in line with following information:

<http://www.w3.org/2001/10/xml-exc-c14n#>

15.5 Message type 1: file with multiple ISO 20022 messages

For message type 1) the requirement in the UDFS chapter [Digital Signature managed within the business layer](#) [▶ 378] .

"The NRO³² signature is stored in the BAH in case of individual messages or in the file header in case of messages grouped into a file. In case messages grouped into a file, the BAH of the included individual messages does not include a signature. File (meaning multi-message):

31 Exclusive XML Canonicalization <http://www.w3.org/TR/xml-exc-c14n/>

The signature is part of the file header. It is over the list of BAH's and ISO 20022 messages and covers the whole <XChg> element of the Business File (head.002), except for the signature itself.”

The signature, in particular, covers the whole “BusinessFileHeader <XChg>” element, except for the signature itself. So consequently the following field will be not taken into account for signature calculation:

Xchg/PyldDesc/ApplSpfcInf/Sgntr/ds:Signature³³

Hence a signature will then be constructed as follows:

- I One reference (in blue below) points out the XChg itself. This is done using the same document reference URI = "", which means the entire document. To leave the signature element itself out of the digest calculation, the transform "#enveloped-signature" is used.
- I One reference (in yellow below) points to the keyinfo element of the signature itself. This is a XAdES³⁴ requirement.

32 Non-repudiation of origin is intended to protect against the originator's false denial of having sent the message.

33 Due to the XAdES requirement the ds:keyinfo element inside the ds:signature is covered/protected by the signature.

34 ETSI TS 101 903 V1.4.2 (2010-12) XML advanced electronic signatures

1) A message type 1³⁵ signature example is reported in the below picture:

```
<ds:Signature Id="_8aaee938-014d-489e-a385-b72155000474" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
    <ds:Reference URI="">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig#sha256" />
      <ds:DigestValue>GUTJy22YxtDXe7yZvdYfJ/GYM+pGH4h5dgWe7c+2gXU=</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="#_4eaf74f7-086b-410e-b214-45136a615bac">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig#sha256" />
      <ds:DigestValue>8CepFq00h78WgVHh23B16RFQZWhdFM6AjY+b0texoSx=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>QzvbmDLi8QlEmsfKz...HNgew=</ds:SignatureValue>
  <ds:KeyInfo Id="_4eaf74f7-086b-410e-b214-45136a615bac">
    <ds:X509Data>
      <ds:X509Certificate>MIIEXTCCA8ag...IY5uXkO8IGZ3XUsw=</ds:X509Certificate>
    </ds:X509Data>
  </ds:KeyInfo>
</ds:Signature>
```

Reference to the whole document, less the signature

Reference to KeyInfo (a XAdES requirement)

Figure 94 - Message type 1 signature example

Reference to the message (head.002):

```
<Xchg xmlns="urn:iso:std:iso:20022:tech:xsd:head.002.001.01">
  <PyldDesc>
    <PyldDtIs>
      <PyldIdr>FILEREf1</PyldIdr>
      <CreDtAndTm>2014-12-17T09:30:47Z</CreDtAndTm>
    </PyldDtIs>
    <App1SpfcInf>
      <SysUsr>SystemUserX1</SysUsr>
      <Sgntr>...</Sgntr>
      <TtlNbOfDocs>1</TtlNbOfDocs>
    </App1SpfcInf>
    <PyldTpDtIs>
      <Tp>ISO20022</Tp>
    </PyldTpDtIs>
    <MnfstDtIs>
      <DocTp>camt.003.001.05</DocTp>
      <NbOfDocs>1</NbOfDocs>
    </MnfstDtIs>
  </PyldDesc>
  <Pyld>
    <BizData xmlns="urn:iso:std:iso:20022:tech:xsd:head.003.001.01">
      <AppHdr xmlns="urn:iso:std:iso:20022:tech:xsd:head.001.001.01">...</AppHdr>
      <Document xmlns="urn:swift:xsd:DRAFT7camt.003.001.05">...</Document>
    </BizData>
  </Pyld>
</Xchg>
```

35 ESMIG digital signature services are configured to produce and generate rsa-sha256 signatures, and use sha256 digest.

2) A message type 1 structure example (including signature) is provided in XML format as described below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Xchg xmlns="urn:iso:std:iso:20022:tech:xsd:head.002.001.01">
  <PyldDesc>
  <PyldDtIs>
  <PyldIdr>FILEREFF1</PyldIdr>
  <CreDtAndTm>2014-12-17T09:30:47Z</CreDtAndTm>
  </PyldDtIs>
  <ApplSpfcInf>
  <SysUsr> SystemUserX1</SysUsr>
  <Sgntr>
  <ds:Signature Id="_8af629dd-bb2c-4207-b0b4-c3edb7d17444"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
  <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
  sha256" />
  <ds:Reference URI="#_f6fa91c7-ee9f-4702-8f08-820bd7a86ac2">
  <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
  <ds:DigestValue>wF0mYpRxS6RA0x0dr1ZKfmV3Tza4jVWw8Afg0efdogU=</ds:DigestValue>
  </ds:Reference>
  <ds:Reference URI="">
  <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
  <ds:DigestValue>LQSkT1Mksb6iSiyqwCmAAs/ZKd9NkwI068Kukx9JP/U=</ds:DigestValue>
  </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>rLCX6pUzTEYGAHMNu/NczFwbXVgncgVsjmhCNnNsXjbU8CqJeytFM3XJFvPocqq
  TX2ZsPg+GAE89xFBb2xe7j8Z1mgTweEuU3uvofKjN7Lo4ZnIaUQxPUBStY6cp7K+YtAwQ31bfq2a/mWPQb
  b0C5fUsCwrn/Nxf/6q6PpO+MiMwBwOj4mgFnkqv3pFvhmFPPWC1AuReS/RMLjZrGYVSBiBgxkv71D7ijTb
  bbZJzWfwlHK0z7fdzIA10wUzi+9mst858kIEcVX7QhbBdK8PxBSvRGau1lbMIGlRHWEe9fgN6y15rSvpfR
  ODewUS1GU+LgV9SuL3g+GxpwhYT5+MJ/A==</ds:SignatureValue>
  <ds:KeyInfo Id="_f6fa91c7-ee9f-4702-8f08-820bd7a86ac2">
  <ds:X509Data>
```

```

<ds:X509Certificate>MIID0DCCArigAwIBAgIBBTANBgkqhkiG9w0BAQsFADBMMQswCQYDVQQGEWJGUj
EcMBoGA1UECgwTS2V5bmVjdG1zLU9wZW5UcnVzdDEfMB0GA1UEAwwWT3B1b1RydXN0IFRlc3QgQ0EgU0hB
MjAeFw0xMjExMTUwMDU3MzVaFw0xNDExMTUwMDU3MzVaMFgxCAJBgNVBAYTAk1UMQ8wDQYDVQQKDAZPIF
RFU1QXExjAQBgNVBAsMCU9VIFRFU1QgMjESMBAGA1UECwwJT1UgVEVTVCAxMRAwDgYDVQQDDAdUZXR0IEN0
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtNB/1lzF05cVqDIlzQJRszZ9TK7AhlnxnR2E
P1hRnP7GRnnksqyYMJECiL/4NnTEhftQe7AGSaWeX7x0sGHJGd72NwmFQazVjHyaT8XSxaxUoG4kc1F5Qa
DOvxxUAHTtM2qYNjppqFyKkTGbA5D7IqS36zTBYawCE40k9hU2/pvInG3jiKA60U4of9oqEQe4+hw2Ixn0
1mRmxPunKYozWVn3ggL/QQ1H/yggkBdplG2qmIU09cvyVdycABW+5R56NyR42xVRcb56rvI5Qcbnbsrvk
cbmslGdo/qnKvxcThXstt3TqGq+kZ1CIHDoJsF8ZDQKuIjXMEgsurt/OHQIDAQABo4GwMIGtMB0GA1UdDg
QWBRRsJehOf8/t06YtF04hEYcc1C0zoTAFBgNVHSMEGDAWgBRRcv9bAGffzbq1TCZ0MpE7ji+fpTARBglg
hkgBhvhCAQEEBAMCB4AwDgYDVR0PAAQH/BAQDAgBAMEgGA1UdHwRBMD8wPaA7oDmGN2h0dHA6Ly9wa210ZX
N0Lm9wZW50cnVzdC5jb20vT3B1b1RydXN0X1Rlc3RfQ0FfU0hBMTU5cmwwDQYJKoZIhvcNAQELBQADggEB
AGMAu3Yo2Z9Ff1FLX/DHvcw8T5otZlaYtJiHdYcEtvhjY24vcXJzwBuHbfopVu91XZFuxXjG12SSyKsK4s
RHfUVPQdryAMGzMUW+0gVfjupV54jr6vkaELq2t6oyE52CHqvvlHyLJz5CIW6jDEmAzGNJZ2wdRr4fu9z
M2lm4X5JITsZGxY/JH02f1155QJuVn7NSffx8PxRsIKYNZ+Z7kcZNTSL9zDwYXob5PUBv60fXMhWPJtngz
80I8NGqDVQIjtnbgcsSgdchRMV4JOUb8UK7RAJpG4aR/5RkaMk06DLHXJteXfmsKfLyDq3H8B+eHgFJJW
CeYmNvqk755EVNE=</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
</Sgntr>
<TtlNbOfDocs>1</TtlNbOfDocs>
</ApplSpfcfInf>
<PyldTpDtls>
<Tp>ISO20022</Tp>
</PyldTpDtls>
<MnfstDtls>
<DocTp>camt.003.001.05</DocTp>
<NbOfDocs>1</NbOfDocs>
</MnfstDtls>
</PyldDesc>
<Pyld>
<BizData xmlns="urn:iso:std:iso:20022:tech:xsd:head.003.001.01">
<AppHdr xmlns="urn:iso:std:iso:20022:tech:xsd:head.001.001.01">
<Fr>
<FIId>
<FinInstnId>
<BICFI>CSDPARTCPNT</BICFI>
<Othr>
<Id>CSDBICIDXXX</Id>
| </Othr>

```

```
</FinInstnId>
</FIId>
</Fr>
<To>
<FIId>
<FinInstnId>
<BICFI>SYSTEMIDT2S</BICFI>
<Othr>
<Id>CSDBICIDXXX</Id>
</Othr>
</FinInstnId>
</FIId>
</To>
<BizMsgIdr>REF3 </BizMsgIdr>
<MsgDefIdr>camt.003.001.05</MsgDefIdr>
<CreDt>2014-12-17T09:30:47Z</CreDt>
</AppHdr>
<Document xmlns="urn:swift:xsd:DRAFT7camt.003.001.05">
<GetAcct>
<MsgHdr>
<MsgId>REF3</MsgId>
<ReqTp>
<Prtry>
<Id>CASB</Id>
</Prtry>
</ReqTp>
</MsgHdr>
<AcctQryDef>
<AcctCrit>
<NewCrit>
<SchCrit>
<AcctId>
<EQ>
<Othr>
<Id>T2SDEDICATEDCASHACCOUNT1</Id>
</Othr>
</EQ>
</AcctId>
<Ccy>EUR</Ccy>
<AcctOwr>
<FinInstnId>
<BIC>ACCTOWNRXXX</BIC>
</FinInstnId>
</AcctOwr>
<AcctSvcr>
<FinInstnId>
<BIC>ACCTSVCRRXX</BIC>

</FinInstnId>
</AcctSvcr>
</SchCrit>
</NewCrit>
</AcctCrit>
</AcctQryDef>
</GetAcct>
</Document>
</BizData>
</Pyl>
</Xchg>
```

15.6 Message type 2: single ISO 20022 message

For message type 2) the requirement in UDFS chapter [Digital Signature managed within the business layer](#) [▶ 378] states ³⁶:

"Single message: The signature is over the ISO 20022 message and takes into account the business processing relevant information specified within the BAH (e. g. pair of BICs for definition of the instructing party), except for the signature itself. The digital signature grouped in the BAH itself is not part of this signature calculation."

So consequently the following field will be not taken into account for signature calculation:

AppHdr/Sgntr/ds:Signature ³⁷

In this case the BAH and the ISO 20022 message are considered not to be in the same document.

"Technically speaking, the Application Header is a separate XML document standing apart from the XML documents which represent the business message instance itself."

Since the documents that are referenced do not carry an ID attribute ³⁸ that could be used for identifying the specific document, it has been decided to use a specific reference for the business message, ESMIG ensures that the BAH and the corresponding ISO message are always stored together.

TARGET Service specific reference for document signature

In the XML digital signature standard there is the possibility to use a reference with no URI i.e. omitting the URI attribute entirely. However there can be at most one such reference in a signature, and handling of it is specific, and not covered by the XML digital signature standard ³⁹. Hence the reference to the message must be given by the context and known by the application.

The signature will then be constructed as follows.

- I One reference (in blue below) points out the BAH (AppHdr) itself. This is done using the same document reference URI = "", which means the entire document. To leave the signature element itself out of the digest calculation, the transform "#enveloped-signature" is used;
- I One reference (in green below) is application specific and refers to the business message (no URI). The application will provide the signature API with the relevant message. The signature API is customised to resolve the no URI reference to this message;

36 See also MUG (Message user guide) for BAH; <http://www.iso20022.org/bah.page>

37 Due to the XAdES requirement the ds:keyinfo element inside the ds:signature is covered/protected by the signature.

38 ISO20022 do not support and specify an ID attribute, that can be used to uniquely identify BAH and ISO message.

39 XML signature syntax and processing (Second Edition), W3C Recommendation 10 June 2008, "<http://www.w3.org/TR/xmlsig-core/>"

One reference (in yellow below) points to the keyinfo element of the signature itself (XAdeS requirements).

1) A message type 2⁴⁰ signature example (with application specific reference) is reported in the below picture:

```

<ds:Signature Id="_003adca5-654a-473d-b1cf-3e926cd5d3f7" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
    <ds:Reference URI="">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256" />
      <ds:DigestValue>Ffg8hActTHIR9tyj8BOP2/7EMyECb9wb7CKQvhGSz/A=</ds:DigestValue>
    </ds:Reference>
    <ds:Reference>
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256" />
      <ds:DigestValue>hEKN3t4XgQt2fkJE7WH4xgg/21cKPaAUnFDII7vIdcQ=</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="#_05dda060-fd01-4538-9db0-56c8e5d3dfc1">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256" />
      <ds:DigestValue>bcF4Ty77sjjGLXSd5Yb8QqJijbvy4RRbJkh8zFEFbco=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>Ft1F0n3hzk5Y78Im/...newun=</ds:SignatureValue>
  <ds:KeyInfo Id="_05dda060-fd01-4538-9db0-56c8e5d3dfc1">
    <ds:X509Data>
      <ds:X509Certificate>MIIEEXTCCAsG...IY5uXk03IGZ3XUsw=</ds:X509Certificate>
    </ds:X509Data>
  </ds:KeyInfo>
</ds:Signature>

```

Reference to the BAH, less the signature

Application specific Reference (to the message)

Reference to KeyInfo (a XAdES requirement)

Figure 95 - Message type 2 signature example

General remark: The signature is over the ISO 20022 message and takes into account the business processing relevant information specified within the message header (BAH), except the signature itself. The digital signature in the BAH itself is NOT part of this signature calculation.

40 ESMIG digital signature services are configured to produce and generate rsa-sha256 signatures, and use sha256 digest.

Reference to the BAH (AppHdr):

```
<AppHdr xmlns="urn:iso:std:iso:20022:tech:xsd:head.001.001.01" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Fr>
    <FIId>
      <FinInstnId>
        <BICFI>FACHFRP1000</BICFI>
        <Othr>
          <Id>FAAHFRP1000</Id>
        </Othr>
      </FinInstnId>
      <ClrSysMmbId>
        <ClrSysId>
          <Prtry>T2S</Prtry>
        </ClrSysId>
        <MmbId>SystemUserX1</MmbId>
      </ClrSysMmbId>
    </FIId>
  </Fr>
  <To>
    <FIId>
      <FinInstnId>
        <BICFI>SETTLSYST2S</BICFI>
        <Othr>
          <Id>FAAHFRP1000</Id>
        </Othr>
      </FinInstnId>
    </FIId>
  </To>
  <BizMsgId>18ROS24SEC500101</BizMsgId>
  <MsgDefId>semt.013.001.02</MsgDefId>
  <MsgId>05-24T14:25:11Z</MsgId>
  <Sgnt>...</Sgnt>
</AppHdr>
```

Reference to the BAH less the signature

Figure 96 - Reference to the BAH (AppHdr)

Reference to the message (e.g. semt.013):

```
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:swift:xsd:semt.013.001.02">
  <IntraPosRvmt Instr>
    <TxId>18RS01801ALM1</TxId>
    <SkppAcct>
      <Id>000370550</Id>
    </SkppAcct>
    <FinInstnId>
      <ISIN>FC0003620449</ISIN>
    </FinInstnId>
    <IntraPosDtIs>
      <StImQty>
        <FaceAmt>6000</FaceAmt>
      </StImQty>
      <StImDt>
        <Dt>2012-09-28</Dt>
      </StImDt>
      <BalFr>
        <Cd>AWAS</Cd>
      </BalFr>
      <BalTo>
        <Prtry>
          <Id>FFE1</Id>
          <Issr>T2S</Issr>
          <SchmelNm>RT</SchmelNm>
        </Prtry>
      </BalTo>
    </IntraPosDtIs>
  </IntraPosRvmt Instr>
</Document>
```

The application will provide the signature API with the relevant message.

Figure 97 - Reference to the message (e.g. semt.013)

2) A message type 2 structure example (including signature) is provided in XML format as described below:

```
<?xml version="1.0" encoding="UTF-8"?>
<AppHdr xmlns="urn:iso:std:iso:20022:tech:xsd:head.001.001.01">
  <Fr>
    <FIId>
      <FinInstnId>
        <BICFI>CSDPARTCPNT</BICFI>
        <ClrSysMmbId>
          <ClrSysId>
            <Prtry>T2S</Prtry>
          </ClrSysId>
          <MmbId>SystemUserX1</MmbId>
        </ClrSysMmbId>
        <Othr>
          <Id>CSDBICIDXXX</Id>
        </Othr>
      </FinInstnId>
    </FIId>
  </Fr>
  <To>
    <FIId>
      <FinInstnId>
        <BICFI>SETTLSYST2S</BICFI>
        <Othr>
          <Id>CSDBICIDXXX</Id>
        </Othr>
      </FinInstnId>
    </FIId>
  </To>
  <BizMsgIdr>SENDERREFERENCE</BizMsgIdr>
  <MsgDefIdr>sese.023.001.02</MsgDefIdr>
  <CreDt>2001-12-17T09:30:47Z</CreDt>
  <Sgntr>
    <ds:Signature Id="_be4dd7de-c63a-43a6-9b62-f69290939eb6"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
        <ds:Reference URI="#_98742d60-2afc-4fa7-a731-828756ce47b1">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </ds:Transforms>
        </ds:Reference>
      </ds:SignedInfo>
    </ds:Signature>
  </Sgntr>
</AppHdr>
```

```
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>vB/xxu+qkEVUH5i9uVdBHOXOp6+XDsan/iHxH+UiMGo=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>hWGkHPu5IMYxe4KFYyaMOWFYq0w2pi+BYNyVHEwm/Z8=</ds:DigestValue>
</ds:Reference>
<ds:Reference>
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>10eHeNdJM1v177M0HzFsmP0IBMYvdPXVuRcR77hAgUg=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>HllitYLicuu5drRrzu5CFxk5GZ3LD00nEPCrXkfWiu54y0zA3P2r6AIE1cYIdue
Y8nioLEvcZcvKVS4zt6bbHv8RRaWmU+Jf13x4vTH5g8W6RY10LPErRbTncn9r3Nb/hxeBj6Rztv3vR+gW+
JY2ly3pkTIAb80JhQ9kcauarcwqG6MAWM3UjK31j796Ldi7ddvHohgW1qHXzdidiBfcONatYnIXZrw/77DU
nBecimz4yqJvCo1Sri1asC0LHFdbeudgBivJtQ/CD1/So9Mkrw6VNUXohv5L3i3J3fNI9gmM1oC/ZJGL1H
LfOsyJ7GokRsydpd1YWFQvNNhu1OupanRA==</ds:SignatureValue>
<ds:KeyInfo Id="_98742d60-2afc-4fa7-a731-828756ce47b1">
<ds:X509Data>

<ds:X509Certificate>MIID0DCCArigAwIBAgIBBTANBgkqhkiG9w0BAQsFADBMMQswCQYDVQQGEWJGUj
EcMBoGA1UECgwTS2V5bmVjdG1zLU9wZW5UcnVzdDEfMB0GA1UEAwwW3B1b1RydXN0IFRlc3QgQ0EgU0hB
MjAeFw0xMjExMTUwMDU3MzVaFw0xNDExMTUwMDU3MzVaMFgxCzAJBgNVBAYTAklUMQ8wDQYDVQQKDAZPIF
RFU1QxEjAQBgnVBAsMCU9VIFRFU1QgMjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtNB/1lzF05cVqDI1zQJRszZ9TK7Ah1nxxnR2E
P1hrnP7GRnksqyYmJECiL/4NnTEhftQe7AGSawEX7x0sGHJGd72NwmFQazVjHyaT8XSxaxUoG4kc1F5Qa
D0vVxUAHTtM2qYNjppfYkktGbA5D7IqS36zTBYawCE40k9hU2/pvInG3jiKA60U4of9oqEQe4+hW2IxnN0
1mRmxPunkYoZWVn3ggL/QQ1H/yggkBDpLG2qmIU09cvyVdyCABW+5R56NyR42xVRcb56rvI5Qcbbnsrvk
cbms1Gdo/qnKvxcHxstt3TqGq+kZ1CIHDoJsF8ZDQKuIjXMEgsurt/OHQIDAQABo4GwMIgtMB0GA1UdDg
QWBBRsJeh0f8/t06YtF04hEYcc1C0zoTafBgNVHSMEGDAWgBRRcv9bAGffzBq1TCZ0MPE7ji+fpTARBglg
hkgBhvCAQEABAMCB4AwDgYDVR0PAAQH/BAQDAgBAQEgA1UdHwRBMD8wPaA7oDmGN2h0dHA6Ly9wa210ZX
N0Lm9wZW50cnVzdC5jb20vT3B1b1RydXN0X1Rlc3RfQ0FFU0hBMi5jcmmwDQYJKoZIhvcNAQELBQADggEB
AGMAu3Yo2Z9Ff1FLX/DHVcw8T5otZ1aYtJiHdYcEtvhjY24vcXJzwBuHbFopVu91XZFuXjG12SSyKsK4s
RHfUVPQdryAMGzMUW+OgjVFjupV54jr6vkaELq2t6oyE52CHqvVlHyLJz5CIW6jDEmAzGNJZ2wdr4f9z
M21m4X5JITsZGxY/JHO2f1155QJuVn7NSffx8PxRsIKYNZ+Z7kczNTSL9zDwYXob5PUBv60fXmHWPJtngz
80I8NGqDVQIjtnbgcsSgDchRMVY4J0ub8UK7RAJpG4aR/5RkAmk06DLHXJteXfmsKfLyDq3H8B+eHgFJJW
CeYmVnqk755EVNE=</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
</Sgntr>
</AppHdr>
```

15.7 ESMIG digital signature services usage of “ds:object”, attribute ID of the “signature” and “keyinfo”, anchor of trust

Usage of block “Object”:

In message type 1 and 2 the “ds:object” element is not used when constructing the signature. The ESMIG digital signature API (Application Programming Interface) follows standard XML signature processing which defines what happens when a “ds:object” element is encountered:

if the “ds:object” (or its content) is referenced in “ds:signedInfo”, then the API will verify this reference as part of the signature verification;

if the “ds:object” is not referenced in “ds:signedInfo”, then the API will ignore it, when performing the cryptographic check of the signature.

However if the “ds:object” contains e.g. XAdES Qualifying properties, these will be examined in order to determine the signature format, i.e. is the signature a XAdES-BES or XAdES-T or XAdES-C.

Note: ESMIG recommendation is to not use in message type 1 and 2 the “ds:object” element.

Usage of Attribute ID of the block “signature”:

ESMIG will generate the ID attribute of the “signature” element when building a signature to be sent to counterparts. The ID attribute is optional for signatures sent to ESMIG. If present the value of the ID attribute must be an underscore (“_”) followed by a universally unique identifier (UUID), that is either timebased (UUID version 1) or random (UUID version 4). The UUID generating system is responsible for ensuring that all the UUID’s in a single document are unique.

Usage of block “keyinfo”:

The XAdES standard allows two different methods to comply with the XAdES-BES requirement. In ESMIG signature services implementation it has been decided to use the one that includes the signer certificate in the “keyinfo” element:

Element “keyinfo” must be present and must include the “ds:X509Data/ds:X509Certificate” containing the signing certificate.

The ID attribute on the “keyinfo” element is mandatory and the value of the ID attribute must be an underscore (“_”) followed by a universally unique identifier (UUID), that is either timebased (UUID version 1) or random (UUID version 4).

The “signedinfo” element must reference the “keyinfo” element using the ID attribute.

Usage of the alternative “ds:Object/QualifyingProperties/SignedProperties/SignedSignatureProperties/SigningCertificate” element is not allowed.

Anchor of trust

It is necessary that the parties have enough information to validate the signatures. This is ensured by having the same anchor of trust in both ends and providing certificates in “keyinfo”. Depending on the Certification Authority (CA) structure and the chosen anchor of trust, the number of certificates included in the “keyinfo” element may vary:

In case of a root CA that issues intermediate CA certificates that in turn issue the signer certificates, the chain in the “keyinfo” element depends on the chosen anchor of trust:

- if the anchor of trust is the intermediate CA, then the chain in the “keyinfo” element need only to contain the signer certificate;
 - if the anchor of trust is the root CA, the chain in the “keyinfo” element must include both the signer certificate and the intermediate CA certificate.
- In case of a root CA that issues signer certificates directly, the root CA is the anchor of trust: The chain in the “keyinfo” element needs only to contain the signer certificate.

The parties communicating must use the same certificates as anchor of trust. It is up to ESMIG signature services for each CA to choose the certificate (root or intermediate) that constitutes the anchor of trust.

16 Glossary

Term	Definition	Source
4CB	The Deutsche Bundesbank (BBk), the Banco de España (BdE), the Banque de France (BdF) and the Banca d'Italia (Bdl), collectively, in their capacity as the national central banks responsible for building, maintaining and running T2 Service and common components, in accordance with the relevant contractual arrangements and with decisions of the ECB's Governing Council.	CLM/RTGS
4CB Network	The 4CB Network is the common internal technical network used by the providers of the market infrastructure services.	CLM/RTGS
account	An account is a record of debit and credit entries to cover transactions involving a particular item or a particular person or concern.	CLM/RTGS
account holder	Individual or entity which owns an account	CLM/RTGS
Account Monitoring Group	An optional clustering of accounts for consolidated liquidity monitoring purposes.	CLM/RTGS
act on behalf	Corresponds to the situation when a participant has been granted the authority to perform actions on behalf of one or more other account holders. Central banks are allowed to act on behalf of their participants.	CLM/RTGS
addressable BIC	These BICs can only send and receive payments to/from the system via the CLM Account Holder. Their payments are settled in the account of the respective CLM Account Holder.	RTGS
algorithm	An algorithm is a mathematical method to provide a smooth, fast and liquidity saving resolution of the payment queue, for example by taking offsetting payment flows into account.	RTGS
ancillary system	A system in which payments or securities are exchanged and/or cleared, while the ensuing monetary obligations are settled in another	RTGS

Term	Definition	Source
	<p>system, typically a RTGS system. Ancillary systems are, e.g.:</p> <ul style="list-style-type: none"> retail payment systems (RS) large value payment systems (LVPS) foreign exchange (FX) systems money market systems clearing houses Securities Settlement Systems (SSS) 	
application-to-application	A connectivity mode that enables the exchange of information between the application of the service provider and the software application(s) of the actors.	CLM/RTGS
AS batch message	Message used to instruct several AS payments for generic procedures A-D	RTGS
AS payment instruction	Single instructions ordered by ancillary systems using one of the generic procedures (A-D) sent via ASTransferInitiation	RTGS
AS settlement bank	CLM Account Holder who pertains to one or more ancillary system. The participant may manage the ancillary system settlement process (e.g. the determination of settlement positions, monitoring of the exchange of payments, etc.) not only for own purposes but also for other ancillary system participants on its RTGS dedicated cash account.	RTGS
AS technical account	An AS technical account is used and held by the ancillary system in order to collect money for the settlement of AS payments.	RTGS
authentication	The methods used to verify the origin of a message or to verify the identity of a participant connected to a system.	CLM/RTGS
automated liquidity transfer	In case of insufficient liquidity on a CLM Account Holder's main cash account to settle a payment linked to a central bank operation or cash withdrawal, CLM will automatically trigger	CLM

Term	Definition	Source
	an inter-service liquidity transfer with the missing amount from the CLM Account Holder's RTGS dedicated cash account used for payments.	
availability	The ability of a configuration item or service/component to perform its agreed function when required	CLM/RTGS
available liquidity	Credit balance on the account plus collateralised credit line for overdraft (if available)	CLM/RTGS
backup payments	In the event of a technical system outage a CLM Account Holder (affected participant) may lose its ability to send payments to and receive payments from RTGS. In order to give the affected participant the possibility to reduce the business impact of the technical failure, functionality is offered to generate payments via U2A, the so-called backup payments functionality.	RTGS
Banking Group	A Banking Group allows a number of parties (belonging to one or multiple central banks) to be viewed collectively for certain business purposes, such as oversight and regulation.	CLM/RTGS
beneficiary	A recipient of funds (payee) or securities. Depending on the context, a beneficiary can be a account holder in CLM or RTGS and/or a final recipient.	CLM/RTGS
BIC directory	Directory published by SWIFT. It contains the business identifier codes (BIC) that SWIFT has registered according to the ISO 9362 standard, and the names and addresses of the corresponding entities.	RTGS
BIC11	In addition to the first eight characters of the BIC, an optional branch code of three characters is used to identify any branch or reference of an institution.	CLM/RTGS
Bilateral/multilateral limit	Please see term "limit".	RTGS

Term	Definition	Source
blocking	Blocking means the exclusion of parties (party blocking) or accounts (account blocking). Blocked parties cannot interact with the services. Blocking of accounts prevents any transfers addressing this/these account/s.	CLM/RTGS
broadcast	Information message simultaneously available to all or a selected group of participants in CLM and RTGS.	CLM/RTGS
Business Application Header	The message envelope for business application data that determines which business application the data are routed to and identifies the type of content	CLM/RTGS
business day	The business day comprises and defines the opening times and specific phases per T2 component.	CLM/RTGS
business identifier code	Identification of financial or non-financial institutions within the financial services industry according to the International Organization for Standardization (ISO) Standard 9362.	CLM/RTGS
bypass FIFO	See FIFO by-passing.	RTGS
cancellation	The term "cancellation" refers to an action that shall annul the process (e.g. cancellation of a payment by a participant). The respective payment status is "cancelled".	CLM/RTGS
cash account(s)	All types of accounts available within a service or component	CLM/RTGS
cash withdrawal	Cash withdrawal is an offered option that belongs to the wide range of central bank operations. The customer has the possibility to have money he received in cash from his central bank to be debited on his main cash account.	CLM/RTGS
ceiling	An upper threshold of an account balance defined by the participant for initiating a specific action (sending of notification or liquidity transfer order).	CLM/RTGS

Term	Definition	Source
central bank's ECB account	A central bank's ECB account is an account that records the central bank's asset/liability position towards the ECB in respect of cross-central bank community transactions.	CLM
central bank	A central bank is the institution responsible for monetary policy and the proper functioning of the monetary system in a country or area.	CLM/RTGS
central bank account	A central bank account in RTGS is a cash account owned by a central bank that is allowed to have negative balance.	CLM/RTGS
central bank operations	Operations initiated by central banks in their capacity as central bank of issue, e.g. tender policy operations, changes of the credit line.	CLM/RTGS
central bank services	Business service managing central bank operations and meeting monetary policy requirements.	CLM/RTGS
central counterparty	An entity that interposes itself between the counterparties to the contracts traded in one or more financial markets, becoming buyer to every seller and the seller to every buyer	RTGS
Central European Time	Standard time which is one hour ahead of Coordinated Universal Time (UTC).	CLM/RTGS
Central Liquidity Management (CLM)	Business component of the T2 Service managing and showing funds and credit lines for CLM Account Holders and central bank operations. In addition, central component for funding the RTGS component and T2S and TIPS.	CLM/RTGS
clearing	The process of transmitting, reconciling and, in some cases, confirming payment or securities transfer orders prior to settlement, possibly including the netting of orders and the establishment of final positions for settlement.	CLM/RTGS
clearing house	A central entity (or central processing mechanism) through which financial institutions agree to exchange transfer instructions for funds or securities. In some cases, the clearing house	CLM/RTGS

Term	Definition	Source
	may act as central counterparty for the participants and therefore assume significant financial risks.	
CLM co-manager	The aim of the co-management function is to allow small banks to fulfil their reserve requirement directly but delegate cash flow management to other banks.	CLM
CLM Liquidity Transfer Group	A Liquidity Transfer Group is an optional group of main cash accounts. Central banks can setup Liquidity Transfer Groups for the purpose of arranging intra-CLM liquidity transfers between them.	CLM
CLM Account Holder	A participant in the CLM component that opted for using the functions CLM offers.	CLM
collateral	An asset or third-party commitment that is used by the collateral provider to secure an obligation vis-à-vis the collateral taker	CLM/RTGS
common reference data	Reference data used by all services or components	CLM/RTGS
Common Reference Data Management	Business component managing centrally the reference data for all services or components	CLM/RTGS
connected payment	Payments by a central bank to a participant that trigger a change in the credit line of this participant and an immediate debit/credit of its account to compensate the change in this credit line.	CLM/RTGS
contingency settlement	Common component for the management of the emergency situations.	CLM/RTGS
Continuous Linked Settlement	Payment-versus-payment (PvP) mechanism offered by CLS bank, meaning that a foreign exchange operation is settled only if both counterparties simultaneously have an adequate position in the currency they are selling.	RTGS
credit line	A commitment to grant intra-day credit on demand based on collateral provided to a central bank.	CLM/RTGS

Term	Definition	Source
credit transfer	A payment or, sometimes, a sequence of payments made for the purpose of placing funds at the disposal of the beneficiary. Both the payment instructions and the funds described therein move from the bank of the payer/originator to the bank of the beneficiary, possibly via several other banks as intermediaries and/or more than one credit transfer system.	CLM/RTGS
customer	Entity which is not a participant (direct or indirect) and which uses the service of a participant to exchange transactions in the system	CLM/RTGS
cut-off time	The deadline defined by a system (or an agent bank) to accept transfer orders	CLM/RTGS
data migration tool	Tool to migrate data into CLM, RTGS and CRDM. It supports in case of the initial loading of data.	CLM/RTGS
data propagation	Data Propagation is the distribution of data from one or more source databases to one or more local access databases.	CLM/RTGS
Data Warehouse	Centralised collection of data from operational business applications in which data are aggregated and optimised for reporting and analysis.	CLM/RTGS
dedicated cash account	An account dedicated for a single service/component e.g. TIPS, T2S, RTGS.	CLM/RTGS
dedicated transit account	Dedicated transit accounts are accounts that are owned by central banks which may have either zero or positive balance as they reflect any movement of liquidity from/to the various settlement components and services (i.e. RTGS, T2S and TIPS). They are technical accounts involved in the liquidity transfer process and cannot be involved in the settlement of central bank operations.	CLM/RTGS
deposit facility	A standing facility of the Eurosystem which counterparties may use to make overnight deposits at a national central bank, which are	CLM

Term	Definition	Source
	remunerated at a pre-specified interest rate.	
direct debit	A direct debit is a payment which allows debiting the payer's account by the amount specified in the instruction on the basis of a direct debit authorisation.	CLM/RTGS
direct debit mandate	A direct debit mandate is the authorisation defined in CRDM by a payer to debit the payer's account upon a direct debit instruction from a payee.	CLM/RTGS
distinguished name	A name that uniquely identifies an entry in a directory or network. Usually it is a sequence of attribute-value assertions (e.g. "cn=smith") separated by commas, e.g.: <cn=smith,ou=t2s-ops,o=bnkacctt,o=nsp-1>.	CLM/RTGS
DWH query	A real-time function to retrieve information from the data warehouse using selection criteria to fulfil ad hoc information demands. This function can be used only via U2A.	CLM/RTGS
DWH report	A time-triggered retrieval of information from the data warehouse which is sent as attachment to specific recipients via e-mail.	CLM/RTGS
earliest execution time	Parameter in a payment which defines the earliest point in time a settlement can take place, i.e. before this defined time no settlement attempt will be carried out (see also "from time").	RTGS
earmarked	Earmarked is a status of a payment which is ready for settlement but not taken into account for various reasons.	CLM/RTGS
ECB mirror account	An account in CLM owned by the ECB for each central bank on which the bookings done on the central bank's ECB accounts are "mirrored"	CLM
eligible monetary policy counterparty	This is a subset of the institutions which are subject to the Eurosystem's minimum reserve system. It includes those credit institutions established in the euro area which are required	CLM

Term	Definition	Source
	to hold minimum reserves with a Eurosystem central bank and have access to the Eurosystem monetary policy operations (open market operations and standing facilities).	
end of day	End of the defined business day	CLM/RTGS
entry disposition	A broad set of liquidity management features achieving a flexible and need-based control of the payment flows, thereby limiting possible liquidity risks	CLM/RTGS
Eurosystem Single Market Infrastructure Gateway	The common entry point for all interaction with T2, T2S and TIPS. Based on common technical specifications, ESMIG is network agnostic. It allows participants to connect through one or multiple service providers for both A2A and U2A interfaces.	CLM/RTGS
Extensible Mark-up Language	An open standard developed and maintained by World Wide Web Consortium (W3C), for describing and structuring data for the transmission and exchange of information between computer applications and organisations/humans.	CLM/RTGS
FIFO	First in, first out.	CLM/RTGS
FIFO by-passing	The system tries to process the first transfer in the queue, but if that cannot be executed owing to lack of funds it then tries to settle the next transfer instead; also called Bypass FIFO.	RTGS
file	A file is identified via the file header (BFH). It may include zero, one or many single individual messages.	CLM/RTGS
file-based network channel	A channel through which data is exchanged and which DEP data structure is defined with minimum size 0 and maximum size 32 MB. The channel can be used for exchange of messages and files.	CLM/RTGS
final	Irrevocable, unconditional, or not annulable	CLM/RTGS
final settlement	Settlement which is irrevocable, unconditional,	CLM/RTGS

Term	Definition	Source
	or not annulable	
floor	A lower threshold of an account balance defined by the participant for initiating a component-specific action	CLM/RTGS
from time	Parameter in a payment which defines the earliest point in time a settlement can take place, i.e. before this defined time no settlement attempt will be carried out (see also "earliest execution time").	CLM/RTGS
general ledger	The general ledger sometimes known as nominal ledger, is the main accounting record of a business which uses double-entry bookkeeping. The general ledger file exists for all services on service level except for CLM. CLM provides its own data and the other services data in several general ledger files, i.e. one general ledger file per service and central bank.	CLM/RTGS
Graphical User Interface	The interface that allows a user to interact with a software application through the use of graphical elements (e.g. windows, menus, buttons and icons) on a computer screen, using the keyboard and mouse	CLM/RTGS
gridlock	A situation that can arise in a funds or securities transfer system in which the failure of some transfer orders to be executed (because the necessary funds or securities are unavailable) prevents a substantial number of other orders from other participants from being executed.	RTGS
gross settlement system	A transfer system in which the settlement of funds or securities occurs individually (on an instruction-by-instruction basis).	CLM/RTGS
guarantee fund mechanism	Mechanism to provide the complementary liquidity needed according to pre-defined rules in case an ancillary system cannot settle using the settlement banks liquidity only	RTGS
guarantee funds account	Account used in case the optional guarantee mechanism has to be activated by an ancillary	RTGS

Term	Definition	Source
	system or a central bank on its behalf	
guarantor	Owner of the guarantee funds account	RTGS
immediate liquidity transfer order	Liquidity transfers initiated by the participant via U2A or A2A during the business day and with immediate impact	CLM/RTGS
incident	An event which is not part of the standard operation of the service and which causes, or may cause, an interruption or a reduction of the quality of the TARGET Services	CLM/RTGS
Indirect Participant	An Indirect Participant is a credit institution, which has entered into an agreement with a CLM Account Holder to submit payments and receive payments via such direct RTGS Participant's RTGS dedicated cash account, and which has been recognised by RTGS component as an indirect participant.	RTGS
information period	Information period is an optional connected mechanism used in the settlement of ancillary systems. If used by the AS the settlement bank will receive information about the specified time of the settlement of the AS and the needed liquidity. The settlement bank will gain the possibility to disagree on the specified amount.	CLM/RTGS
instructed party	Party that is instructed by the previous party in the chain to carry out the (set of) instruction(s)	CLM/RTGS
instructing party	Party that instructs the next party in the chain to carry out the (set of) instruction(s)	CLM/RTGS
instructions	Orders for a service/component e.g. payment, liquidity transfer order, tasks	CLM/RTGS
inter-service liquidity transfer	Transfer of funds between accounts of two components/services	CLM/RTGS
intraday liquidity	Funds which can be accessed during the business day, usually to enable financial institutions to make payments on an intraday basis	CLM/RTGS
ISO 20022	The international standard for financial services	CLM/RTGS

Term	Definition	Source
	messaging, maintained by the International Organization for Standardization (ISO).	
latest execution time	Parameter in a payment which defines the latest point in time a settlement can take place, i.e. after this defined time no settlement attempt will be carried out (see also "till time")	CLM/RTGS
legal entity identifier	The legal entity identifier is a 20-digit, alpha-numeric code based on the ISO 17442 standard. It connects to key reference information that enables clear and unique identification of legal entities participating in financial transactions.	CLM/RTGS
limit	Amount for payments a CLM Account Holder is willing to pay to another participant/account (bilateral limit) or to the other participants/accounts (multilateral - limit towards whom no bilateral limit is defined), without having received payments (that are credits) first. For a CLM Account Holder it is possible to establish standing orders or current bilateral (respectively multilateral) limits.	RTGS
liquidity transfer	Liquidity transfer is a cash transfer order, the main purpose of which is to transfer liquidity between different accounts of the same participant.	CLM/RTGS
Liquidity Transfer Group	Liquidity Transfer Group refers to an optional grouping of cash accounts defined by a central bank for the purpose of arranging liquidity transfers.	CLM/RTGS
liquidity transfer order	Liquidity transfer order is a cash transfer order, the main purpose of which is to transfer liquidity between different accounts of the same participant.	CLM/RTGS
local reference data	Reference data used by specific services/components and stored in this service/component in a local database. Data used to fit the unique requirements of the	CLM/RTGS

Term	Definition	Source
	single T2 component (e.g. direct debit authorisation in RTGS).	
main cash account	Account kept in CLM for provision of credit lines, central bank operations and liquidity management incl. sourcing of dedicated cash accounts	CLM
mandated payment	Payment initiated by an entity that is not party to the transaction (typically by a central bank) on behalf of another entity. A central bank sends a credit transfer (with specific message structure) on behalf of the failed CLM Account Holder (only in case of contingency situations).	CLM/RTGS
marginal lending account	A marginal lending account is owned by the relevant central bank but is opened in the name of the CLM Account Holder. There is one marginal lending account for each monetary policy counterparty or CLM Account Holder subject to standing facilities.	CLM
marginal lending facility	<p>A standing facility of the Eurosystem which counterparties may use to receive overnight credit from a CB at a pre-specified interest rate against eligible assets.</p> <p>Two kinds of marginal lending are available.</p> <ul style="list-style-type: none"> Marginal lending on request: requested by the participant, e.g. to cover a shortage of liquidity. Automated marginal lending: automatic transformation of intraday credit in overnight credit at the end of the day. 	CLM/RTGS
message subscription	A service that allows authorised interested party with direct connectivity to the service to subscribe, based on a set of predefined parameters, to copies of messages sent between a directly connected party and the service in real time using push-mode messaging	CLM/RTGS
message-based network channel	A channel through which data is exchanged and which DEP data structure is defined with mini-	CLM/RTGS

Term	Definition	Source
	imum size 0 and maximum size 32 KB. The channel can be used for exchange of messages and files.	
messages	A message is a data structure containing a business application header and a payload.	CLM/RTGS
monetary financial institution	Monetary financial institutions include the Eurosystem (ECB and the NCBs of those countries that have adopted the euro), credit institutions and non-credit institutions (mainly money market funds) whose business is to receive deposits from entities other than MFIs and to grant credit and/or invest in securities.	CLM
multi-addressee	CLM Account Holders are able to authorise their branches and credit institutions belonging to their group to channel payments through the account of the CLM Account Holder without its involvement by submitting/receiving payments directly to/from the system.	RTGS
network service	A modus for connection, four possible options: <ul style="list-style-type: none"> store-n-forward Message-based store-n-forward File-based real-time Message-based real-time File-based 	CLM/RTGS
Network Service Provider	A business entity, licensed – in this case - by the Eurosystem, which provides the technical infrastructure, including hardware and software, to establish a secure and encrypted network connection permitting the exchange of information between actors	CLM/RTGS
non repudiation of origin	Protection against the originator's false denial of having sent the message	CLM/RTGS
notification	A confirmation of a change in the business status of a payment, liquidity transfer, other instruction, account floor or account ceiling	CLM/RTGS
offsetting	Offsetting in the RTGS aims at increasing the capacity of the system to settle payments,	RTGS

Term	Definition	Source
	thereby reducing queues, speeding up the settlement process and reducing the need of intraday liquidity. A bilateral or multilateral off-setting mechanism considers payments in the queues of participants and tries to settle them simultaneously on a gross basis within one legal and logical second.	
opening day	See "TARGET opening day".	CLM/RTGS
overnight credit	See "marginal lending facility".	CLM
overnight deposit	See "deposit facility".	CLM
overnight deposit account	An overnight deposit account is owned by the relevant central bank but is opened in the name of the CLM Account Holder. There is one overnight deposit account for each monetary policy counterparty or CLM Account Holder subject to standing facilities.	CLM
partial settlement	The settlement of only part of a settlement instruction's original amount, when full settlement is not possible owing to lack of cash or securities.	CLM/RTGS
party	Any legal entity or organisation interacting with the T2 Service either directly or indirectly and which is identified in CRDM as party	CLM/RTGS
party type	The party type identifies the different kinds of contractual agreements which allow processing with the services/components.	CLM/RTGS
payee	See "beneficiary".	CLM/RTGS
payer	The party to a payment transaction which issues the payment or agrees to the transfer of funds to a payee.	CLM/RTGS
payment	A payment is a transfer of funds which discharges an obligation on the part of a payer vis-à-vis a payee.	CLM/RTGS
payment message	A message which provides all payment information to a service/component in A2A mode.	CLM/RTGS

Term	Definition	Source
payment	A payment is an order to initiate a payment .The order may relate either to a credit transfer or to a direct debit.	CLM/RTGS
payment instruction	A message (pacs.009) sent by the ancillary system to instruct a debit on a settlement banks RTGS DCA. Can be sent as single message or bundled in a file.	CLM/RTGS
payment system	A payment system consists of a set of instruments, banking procedures and, typically, inter-bank funds transfer systems which facilitate the circulation of money.	CLM/RTGS
payment versus payment	A mechanism in a foreign exchange settlement system which ensures that a final transfer of one currency occurs if, and only if, a final transfer of the other currency or currencies takes place (e.g. CLS).	RTGS
pending value	Remaining amount of an order (e.g. reservation) which cannot be executed due to lack of liquidity. This amount will be queued and be processed in an event-oriented manner, i.e. in case of incoming liquidity the pending amount will be decreased.	RTGS
predefined DWH queries and reports	Predefined means that there is a fixed number of parameters and the parameter values are selectable.	CLM/RTGS
priority	In general, payments are settled immediately, if sufficient liquidity is available on the account of the participant. Considering their urgency, they can be submitted by the sender using priorities: urgent, high, normal. Payments which cannot be settled immediately are queued according to their priority.	CLM/RTGS
privilege	A right, either granted or denied, to execute certain functions within an application or to access and/or update certain data.	CLM/RTGS
problem	An abnormal state or condition at the compo-	CLM/RTGS

Term	Definition	Source
	ment, equipment, or sub-system level, which may lead to a failure that produces incorrect or unexpected results, showing a discrepancy between the relevant specifications and the actual results.	
pull mode	A communication model using the request/response (and query/response) message exchange pattern. A service consumer requests specific information from a service provider and then waits to receive the response.	CLM/RTGS
push mode	A communication model in which the service provider actively passes event-driven or time-triggered messages to a service consumer based on a subscription by the consumer to the information.	CLM/RTGS
query	A function to retrieve information from a database using selection criteria to fulfil ad hoc information demands	CLM/RTGS
queue	Location where transfer orders are held pending by the sending participant or by the system until it can be processed according the rules of the system	CLM/RTGS
real-time	At the same time as event actually happens	CLM/RTGS
real-time gross settlement	The continuous (real-time) settlement of funds or securities transfers individually on an order-by-order basis with intraday finality	CLM/RTGS
real-time gross settlement system	A settlement system in which processing and settlement take place on a transaction-by-transaction basis in real-time	CLM/RTGS
real-time network channel	A network channel that requires both parties to be available and reachable when the message is sent. In case the message cannot be delivered, no retry mechanism is foreseen.	CLM/RTGS
receiver	A participant who obtains the respective message.	CLM/RTGS
reject time	Parameter in a payment which defines a point	CLM/RTGS

Term	Definition	Source
	in time a payment will be rejected if it is not settled by then.	
rejection	The term “rejection” refers to a process of the system or to an action operator which refuses to continue processing (e.g. failed validations, the end of day procedures).	CLM/RTGS
report	An event-driven or time-triggered publishing of information in a defined standard format to specific recipients	CLM/RTGS
report configuration	Action of setting up preferences about which reports the user would like to receive by when.	CLM/RTGS
reservation	Possibility to dedicate/reserve liquidity for special transactions. Liquidity can be reserved for central bank operations, high priority payments and urgent priority payments. Liquidity which is reserved for a particular purpose will not be used for other transactions.	CLM/RTGS
revocation	The term “revocation” refers to an action which aim is to invalidate the operation by withdrawing or reversing (e.g. during the four-eye-processing). The respective task queue order status is "revoked".	CLM/RTGS
role	A role is a set of privileges which identifies the capability of triggering one or several user functions and it is an element to assign access rights to users.	CLM/RTGS
RTGS component	Comprises the processing of high-value payments and ancillary system settlement.	RTGS
RTGS dedicated cash account	An RTGS dedicated cash account is a type of cash account managed within the RTGS and maintained by a CLM Account Holder to settle all transactions submitted to and processed by the RTGS.	RTGS
RTGS directory	The RTGS directory provides information on all participants that are reachable for payments via the RTGS component of the T2 Service.	RTGS

Term	Definition	Source
RTGS Liquidity Transfer Group	A Liquidity Transfer Group is an optional group of dedicated cash accounts. Central banks can setup Liquidity Transfer Groups for the purpose of arranging intra-RTGS liquidity transfers between them.	RTGS
RTGS sub-account	Specific account, belonging to an RTGS dedicated cash account, holding dedicated liquidity for ancillary system settlement.	RTGS
rule-based liquidity transfer	In case of a breach of the floor or ceiling threshold the RTGS or CLM component creates an inter service liquidity transfer, that is triggered by the system based on a rule, that a participant has configured. In case of liquidity transfers pending due to an urgent/high payment configuration rule.	RTGS
Securities Settlement System	A transfer system for settling securities transactions. It comprises all of the institutional arrangements required for the clearing and settlement of securities trades and the provision of custody services for securities.	RTGS
sender	A participant who initiates the process by sending the respective message to the T2 Service	CLM/RTGS
service	A set of business functions and provisions	CLM/RTGS
service level	The measured and reported achievement against one or more service level targets	CLM/RTGS
service level management	The framework of the Eurosystem for specifying services, and monitoring the agreed service levels	CLM/RTGS
service level target	A commitment that is documented in the service level agreement. Service level targets are based on the service levels required to meet business objectives.	CLM/RTGS
settlement period	The settlement period is an optional feature. If used by the AS it will indicate the pre-defined period of time for settlement. If settlement is not completed until the end of settlement period the	CLM/RTGS

Term	Definition	Source
	transactions will be rejected.	
standing liquidity transfer order	Instruction of a CLM Account Holder to transfer regularly a fixed amount (event triggered) between different accounts (main cash accounts, dedicated cash accounts) of the same participant.	CLM/RTGS
standing order for limits	A standing order for limit is an instruction of a participant to define bilateral and/or multilateral limits of a fixed amount on a regular basis	RTGS
standing order for reservation	A standing order for reservation is an instruction to set up a reservation for urgent/high payments or for central bank operations. This reservation has a fixed amount for a business day without a predefined end date.	CLM/RTGS
store-and-forward network channel	A network channel that does not require both parties to be available and reachable when the message is sent. In case the message cannot be delivered, a retry mechanism is foreseen.	CLM/RTGS
straight-through processing	The automated end-to-end processing of trades/payment transfers, including the automated completion of generation, confirmation, clearing and settlement of instructions.	RTGS
system entity	Either the T2 operator or a CSD or NCB for which a segregation of processing capabilities and data are required	CLM/RTGS
system user	A system user can be an individual person or technical user interacting with the TARGET Services.	CLM/RTGS
T2 Actor	Either a central bank, whose currency is available for settlement-related processing in T2, or a client of a central bank having a contractual relationship with the central bank for the processing of its settlement-related cash-processing activities in T2.	CLM/RTGS
T2 operator	The legal and/or organisational entity/entities that operates/operate the T2 platform. As part	CLM/RTGS

Term	Definition	Source
	of an internal distribution of work within the Eurosystem, the Governing Council entrusted the 4CB with operating T2 on behalf of the Eurosystem.	
T2 Service	T2 Service contains CLM and RTGS	CLM/RTGS
TARGET	Trans-European Automated Real-time Gross settlement Express Transfer: the Eurosystem's real-time gross settlement system for the euro. The first-generation TARGET system was replaced by TARGET2.	CLM/RTGS
target amount	The amount up to which the balance (available liquidity) of a main cash account is reduced -in case of ceiling breach- or increased -in case of floor breach. The target amount is an optional feature and can be defined in CRDM by the account holder.	CLM/RTGS
Target Instant Payment Settlement	A real-time settlement system for retail payments settled in central bank money	CLM/RTGS
TARGET opening day	A day on which settlement takes place according to the daily processing schedule and according to the published calendar of opening days	CLM/RTGS
TARGET2	The Trans-European Automated Real-time Gross settlement Express Transfer system, which functions in accordance with Guideline ECB/2007/2 of 26 April 2007 (OJ L 237, 8.9.2007, p. 1).	CLM/RTGS
TARGET2-Securities	The set of hardware, software and other technical infrastructure components through which the Eurosystem provides the services for central securities depositories and central banks that allow core, neutral and borderless settlement of securities transactions on a delivery versus payment basis in central bank money.	CLM/RTGS
tasks	Tasks are activities in a task queue which need to be performed.	CLM/RTGS

Term	Definition	Source
technical account	Account used in the context of ancillary systems settlement as intermediary account for the collection of debits/credits.	CLM/RTGS
till time	Parameter in a payment which defines the latest point in time a settlement can take place, i.e. after this defined time no settlement attempt will be carried out (see also "latest execution time").	CLM/RTGS
transit account	(Technical) account maintained in CLM and RTGS component, T2S and TIPS for the processing of liquidity transfers	CLM/RTGS
user interaction	Activity by a user undertaken whilst interacting with the market infrastructure services, either through a Graphical User Interface or via a local software application	CLM/RTGS
user requirement	A condition or capability needed by a stakeholder to solve a problem or achieve an objective	CLM/RTGS
user requirements document	The document setting out the user requirements	CLM/RTGS
user-to-application	A connectivity mode for the exchange of information through a Graphical User Interface.	CLM/RTGS
V-shape	Type of transmission of messages meaning the addressed platform takes care of the further routing of messages	CLM/RTGS
warehoused payment	Payments submitted up to ten calendar days in advanced. In this case, the payment message is warehoused until the day –time settlement phase with the respective date starts.	CLM/RTGS

17 List of abbreviations

Abbreviation	Meaning
4CB	The Deutsche Bundesbank (BBk), the Banco de España (BdE), the Banque de France (BdF) and the Banca d'Italia (BdI), collectively
A2A	application-To-application
API	Application Programming Interface
AS	ancillary system
BAH	Business Application Header
BIC	Business Identifier Code
CA	certification authority
CB	central bank
CB account	central bank account
CBO	central bank operation
CCP	central counterparty
CET	Central European Time
CEST	Central European Summer Time
CLM	Central Liquidity Management
CLS	Continuous Linked Settlement
CMB	credit memorandum balance
CMS	collateral management system
CRDM	Common Reference Data Management
DCA	dedicated cash account
DEP	Data Exchange Protocol
DMT	Data Migration Tool
DN	distinguished name
DWH	Data Warehouse
EBA	Euro Banking Association
ECB	European Central Bank

Abbreviation	Meaning
ECMS	Eurosystem Collateral Management System
EoD	end of day
ESMIG	Eurosystem Single Market Infrastructure Gateway
FIFO	first in first out
FILERT	file real-time
FILESNF	file store-and-forward
GUI	Graphical User Interface
ISO	International Organization for Standardization
LEA	Legal Archiving
LEI	legal entity identifier
LT ⁴¹	liquidity transfer
LTO ⁴²	liquidity transfer order
MCA	main cash account
MFI	monetary financial institution
MSGRT	message real-time
MSGSNF	message store-and-forward
NCB	National Central Bank
NRO	Non Repudiation of Origin
NSP	Network Service Provider
NTS	night-time settlement
PvP	payment versus payment
RTGS	Real-Time Gross Settlement
SoD	start of day
SSS	Securities Settlement System
STP	straight-through processing

41 Only used in figures

42 Only used in figures

Abbreviation	Meaning
T2S	TARGET2-Securities
TARGET	Trans-European Automated Real-Time Gross Settlement Express Transfer
TIPS	Target Instant Payment Settlement
U2A	user-To-application
URD	user requirements document
URI	Universal Resource Identifier