

Central liquidity management

User detailed functional specifications

Author	4CB
Version	0.4
Date	28 September 2018

Table of contents

Introduction	29
Reader's guide.....	30
Part I - General features of the CLM component.....	32
1 Overview of CLM component.....	32
2 Access to CLM.....	34
2.1 Connectivity (U2A/A2A)	34
2.2 Authentication and authorisation process.....	35
2.3 Authentication and authorisation in ESMIG	35
2.3.1 Authentication and authorisation concepts	36
2.3.1.1 User.....	36
2.3.1.2 Certificate.....	36
2.3.1.3 DN.....	36
2.3.1.4 Technical sender	37
2.3.1.5 Business sender	37
2.3.2 Authentication process	37
2.3.2.1 Authentication of the technical sender	37
2.3.3 Authorisation process	38
2.3.3.1 Authorisation of the technical sender	38
2.4 Security	38
2.4.1 Confidentiality	38
2.4.2 Integrity.....	39
2.4.3 Availability.....	39
2.4.4 Monitoring.....	39
2.4.5 Auditability.....	39
2.5 Graphical User Interface.....	40
2.6 Routing.....	40
2.6.1 ESMIG routing functions.....	43
2.6.1.1 Inbound routing.....	43
2.6.1.2 Outbound routing.....	45
3 Parties and accounts	47

3.1	Parties.....	47
3.1.1	Setup of parties.....	47
3.1.2	Concept of party in CLM.....	48
3.1.3	Reference data for parties used by CLM.....	49
3.1.4	Participation types.....	50
3.1.5	Blocking/unblocking party.....	50
3.2	Accounts structure and functionalities.....	51
3.2.1	Account types.....	52
3.2.2	Functionalities.....	56
3.2.3	Messaging.....	59
3.2.4	Blocking/unblocking account.....	61
3.3	Types of groups.....	62
3.4	Shared reference data.....	64
3.5	Interaction between CLM and CRDM.....	67
4	Business day.....	69
4.1	Business calendar.....	69
4.2	Overview.....	69
4.3	Detailed description of CLM business day phases.....	71
4.3.1	Start of day.....	71
4.3.2	Processing times/windows.....	72
4.3.3	Maintenance window.....	73
4.3.4	End of day.....	73
4.4	Dependencies to other services or components.....	75
4.5	Calculating the positions of CBs vis-à-vis other CBs.....	76
5	Business and features description.....	81
5.1	Settlement of payments linked to CBOs.....	81
5.1.1	Overview.....	81
5.1.2	Definition of execution time.....	82
5.1.3	Revalidate warehoused payments at SoD.....	83
5.1.4	Flow of payments.....	84
5.1.4.1	Payments initiated by CB - credit transfer.....	84
5.1.4.2	Payments initiated by CB - direct debit.....	89
5.1.5	Rejection of payments.....	93
5.1.6	Amendment of payments.....	94
5.1.7	Revocation of payments.....	100

5.1.8	Processing of payment orders	104
5.1.8.1	Entry disposition	106
5.1.8.2	Comprehensive queue management	111
5.1.8.3	Dissolution of the payment queue	113
5.2	Liquidity management.....	113
5.2.1	Available liquidity	114
5.2.2	Liquidity transfer.....	115
5.2.2.1	Overview.....	115
5.2.2.2	Initiation of liquidity transfers	116
5.2.2.3	Liquidity transfer process.....	117
5.2.2.3.1	Immediate liquidity transfer from MCA to DCA	118
5.2.2.3.2	Immediate liquidity transfer from DCA to MCA	120
5.2.2.3.3	Immediate liquidity transfer between two MCAs	122
5.2.2.3.4	Immediate liquidity transfer between two DCAs in different settlement services/components	124
5.2.2.3.5	System-generated liquidity transfer orders	126
5.2.2.3.6	Rejection of liquidity transfer orders	129
5.2.3	Liquidity management features.....	131
5.2.3.1	Liquidity reservation.....	131
5.2.3.1.1	Overview.....	131
5.2.3.1.2	Liquidity reservation process.....	132
5.2.3.1.3	Effect and tapping of liquidity reservation	138
5.2.3.2	Floor/ceiling.....	140
5.2.3.2.1	Definition of floor/ceiling threshold	140
5.2.3.2.2	Breach of floor/ceiling threshold - notification	141
5.2.3.2.3	Breach of floor/ceiling threshold - automatic liquidity transfer.....	142
5.3	Reserve management.....	143
5.3.1	Overview.....	143
5.3.2	Collection of EoD balances.....	145
5.3.3	Daily calculations	145
5.3.4	Periodic calculations	146
5.3.5	Generate payment orders	147
5.4	Standing facilities management.....	147
5.4.1	Overnight deposit.....	147
5.4.1.1	Overview.....	147
5.4.1.2	Overnight deposit process.....	150
5.4.1.2.1	Setup overnight deposit.....	150
5.4.1.2.2	Overnight deposit reverse transaction	152
5.4.1.2.3	Overnight deposit reimbursement and interest calculation	153
5.4.2	Marginal lending "on request"	155

5.4.2.1	Overview.....	155
5.4.2.2	Setup marginal lending "on request"	156
5.4.2.2.1	Before launch of ECMS.....	158
5.4.2.2.2	After launch of ECMS.....	158
5.4.2.3	Marginal lending reimbursement and interest calculation	159
5.4.2.3.1	Before launch of ECMS.....	160
5.4.2.3.2	After launch of ECMS.....	160
5.4.3	Automated marginal lending	161
5.4.3.1	Overview.....	161
5.4.3.2	Automated marginal lending process	162
5.4.3.2.1	Process automated marginal lending	162
5.4.3.2.2	Marginal lending reimbursement and interest calculation	162
5.5	Information management for CLM	163
5.5.1	CLM status management.....	163
5.5.1.1	Concept.....	163
5.5.1.2	Overview.....	164
5.5.1.3	Status management process.....	164
5.5.2	CLM report generation	174
5.5.2.1	Concept.....	174
5.5.2.2	Overview.....	175
5.5.2.3	Report generation process	175
5.5.3	Query management for CLM, CRDM, scheduler and billing	177
5.5.3.1	Concept for CLM	177
5.5.3.2	Overview for CLM.....	177
5.5.3.3	Query management process for CLM	178
6	Overview of used common components in CLM component	180
6.1	CRDM features	180
6.1.1	Concept.....	180
6.1.2	Overview.....	180
6.1.3	Access rights.....	181
6.1.3.1	Access rights concepts.....	181
6.1.3.1.1	User function	181
6.1.3.1.2	Privilege.....	181
6.1.3.1.3	Role.....	193
6.1.3.1.4	User.....	193
6.1.3.1.5	Common reference data objects and the hierarchical party model....	193
6.1.3.1.6	Data scope.....	194
6.1.3.2	Access rights configuration	196
6.1.3.2.1	Configuration of users	196

6.1.3.2.2	Configuration of privileges	196
6.1.3.2.3	Configuration of roles	203
6.1.3.3	Access rights configuration process	205
6.1.3.3.1	Configuration of access rights at party level	207
6.1.3.3.2	Configuration of access rights at user level	209
6.1.4	Message subscription	209
6.1.4.1	Message subscription configuration	209
6.1.4.2	Message subscription parameter types	210
6.1.4.3	Message subscription examples	210
6.1.5	Common reference data maintenance process	211
6.1.5.1	Common reference data objects	211
6.1.5.2	Reference data maintenance types	214
6.1.5.3	Validity of common reference data objects	215
6.1.5.4	Common reference data archiving and purging	219
6.1.5.5	Lifecycle of common reference data objects	220
6.1.5.6	Common reference data propagation	223
6.2	Data warehouse	228
6.2.1	Introduction	228
6.2.2	Scope of the data warehouse	228
6.2.3	Access	228
6.2.3.1	Connectivity	228
6.2.3.2	Authentication and authorisation	228
6.2.4	User roles and access rights	228
6.2.4.1	Overview	228
6.2.4.2	User rights	228
6.2.4.3	User profiles	228
6.2.5	Data warehouse queries and reports	228
6.2.5.1	Overview	228
6.2.5.2	Types of queries and reports	228
6.2.5.3	Predefined queries and reports	228
6.3	Billing	229
6.4	Legal archiving	229
6.5	ESMIG features	229
6.5.1	ESMIG features overview	229
6.5.1.1	Authentication of the message sender	229
6.5.1.2	Participation to the Closed Group of Users	230
6.5.1.3	Validation of the received messages	230
6.5.1.4	Message forwarding	230
6.5.2	Access to ESMIG	230

6.5.2.1	Single access point for the external communication	230
6.5.2.2	Network agnostic communication	230
6.5.3	ESMIG Portal	230
7	Contingency services	232
8	Operations and support.....	233
8.1	Business application configuration	233
8.2	Calendar management	233
8.3	Business day management	233
8.4	Business and operations monitoring.....	233
8.5	Possible actions of operator service desk in ESMIG.....	233
8.5.1	Technical monitoring.....	233
8.6	Archiving management	233
8.7	Trouble management.....	233
9	Additional information for CBs	234
9.1	Role of CBs in CLM	234
9.2	Settlement of payments - specific functions for CBs	235
9.2.1	Payments linked to monetary policy operations	235
9.2.2	Cash withdrawals.....	236
9.3	Credit line management.....	236
9.3.1	Credit line update.....	236
9.3.1.1	Overview.....	236
9.3.1.2	Credit line update process.....	237
9.3.1.2.1	Increase of credit line	237
9.3.1.2.2	Decrease of credit line.....	240
9.3.2	Connected payment.....	242
9.3.2.1	Overview.....	242
9.3.2.2	Connected payment process.....	242
9.4	General ledger handling	246
9.4.1	CLM general ledgers	246
9.4.1.1	CLM general ledgers production	246
9.4.1.2	CLM general ledgers content	248
9.4.2	General ledger provision.....	248
9.5	Query management - CB specific queries	249

9.6	Business/liquidity monitoring for CBs	250
9.7	Reserve management - specific functions for CBs.....	251
9.7.1	Input minimum reserve requirement.....	251
9.7.2	Input additional balances	252
9.7.3	Authorise penalty	252
9.8	Standing facilities - specific functions for CBs	252
9.9	Billing - specific functions for CBs.....	253
9.10	Contingency services - specific functions for CBs.....	253
9.11	Specific requirements for CBs of "out" countries	253
Part II - Dialogues with the CLM participant.....		254
10 Processes with CLM		254
10.1	Send file	257
10.2	Cash transfer order processing	259
10.2.1	Send cash transfer order	259
10.2.2	Revoke/cancel payment	261
10.2.3	Amend payment.....	262
10.2.4	Execute CLM standing order	264
10.3	Settle CLM cash transfer orders	266
10.3.1	Standard CLM settlement.....	266
10.3.1.1	Process floor and ceiling	270
10.3.1.2	Process automated liquidity transfer	272
10.3.2	Process reject time instructions	273
10.3.3	Settle connected payments	274
10.3.4	Modify credit line	276
10.3.5	Settle marginal lending on request	281
10.3.6	Reservation management.....	284
10.4	Run CLM EoD processing	286
10.4.1	Reject payments (EoD).....	286
10.4.2	Process automated marginal lending	287
10.4.3	Process minimum reserve	289
10.5	CLM SoD processing	292
10.5.1	Reimburse marginal lending	292
10.5.2	Process overnight deposit	293
10.5.3	Revalidate warehoused payments at SoD	295
10.6	Reference data management	297

10.6.1	Maintain local reference data object - maintain reservation	297
10.7	Information services.....	299
10.7.1	Execute query.....	299
10.7.2	Receive report.....	303
10.7.3	Receive system notification	304
10.7.4	Enter broadcast.....	305
11	Dialogues and processes	306
11.1	Dialogues and processes between CRDM and CRDM Actor.....	306
11.1.1	A2A Common reference data maintenance and query process.....	306
11.1.1.1	Reference data maintenance process.....	306
11.1.1.1.1	Reference data objects	307
11.1.1.2	Common reference data query.....	308
11.1.1.2.1	Reference data query message coverage	310
11.1.2	DMT file upload.....	311
11.1.2.1	Activity diagram	311
11.1.2.1.1	Upload DMT file.....	312
11.1.2.1.2	DMT file validation	312
11.1.2.1.3	DMT file release	312
11.1.2.1.4	DMT file processing.....	312
11.1.2.1.5	DMT file results provisioning	313
11.1.2.1.6	Download DMT file results	313
11.2	Dialogues and processes between ESMIG and participant	314
11.2.1	Communication processing	314
11.2.1.1	Introduction.....	314
11.2.1.2	Schema validation	314
11.2.1.3	Technical message validation	314
11.2.1.4	Inbound and Outbound messages	314
11.2.1.4.1	Inbound messages	314
11.2.1.4.2	Outbound Messages	314
11.2.1.4.3	ReceiptAcknowledgement (admi.007.001.01)	315
11.3	Dialogues and processes with data warehouse	316
11.4	Dialogues and processes with billing.....	316
	Part III - Catalogue of messages.....	317
	12 Messages – introduction	317

13	Messages - general information	320
13.1	Message validation	320
13.1.1	Structure of ISO 20022 messages.....	320
13.1.2	CLM-specific schema customisation	322
13.1.3	XML character set.....	324
13.1.3.1	Schema validation	325
13.1.3.1.1	Business validation.....	328
13.2	Communication infrastructure.....	329
13.2.1	Envelope messages	329
13.2.1.1	Business Application Header	329
13.2.1.2	Business file header	331
13.2.1.3	Digital Signature managed within the business layer.....	333
13.2.1.4	Time zones.....	334
13.2.1.5	Outbound traffic exceeding given size limitations	335
13.2.1.6	Re-sending of messages.....	336
14	List of messages	337
14.1	Account management (acmt).....	339
14.1.1	AccountQueryList (acmt.025)	339
14.1.1.1	Overview and scope of the message	339
14.1.1.2	Schema.....	340
14.1.1.3	The message in business context	340
14.1.2	AccountListReport (acmt.026)	341
14.1.2.1	Overview and scope of the message	341
14.1.2.2	Schema.....	342
14.1.2.3	The message in business context.....	342
14.2	Administration (admi).....	345
14.2.1	ReportQueryRequest (admi.005).....	345
14.2.1.1	Overview and scope of the message	345
14.2.1.2	Schema.....	345
14.2.1.3	The message in business context	346
14.2.2	ResendRequest (admi.006).....	346
14.2.2.1	Overview and scope of the message	346
14.2.2.2	Schema.....	347
14.2.2.3	The message in business context	347
14.2.3	ReceiptAcknowledgement (admi.007)	347
14.2.3.1	Overview and scope of the message	347
14.2.3.2	Schema.....	348

14.2.3.3	The message in business context	348
14.3	Cash management (camt)	350
14.3.1	GetAccount (camt.003)	350
14.3.1.1	Overview and scope of the message	350
14.3.1.2	Schema	350
14.3.1.3	The message in business context	351
14.3.2	ReturnAccount (camt.004)	355
14.3.2.1	Overview and scope of the message	355
14.3.2.2	Schema	355
14.3.2.3	The message in business context	356
14.3.3	GetTransaction (camt.005)	368
14.3.3.1	Overview and scope of the message	368
14.3.3.2	Schema	369
14.3.3.3	The message in business context	370
14.3.4	ReturnTransaction (camt.006)	378
14.3.4.1	Overview and scope of the message	378
14.3.4.2	Schema	378
14.3.4.3	The message in business context	379
14.3.5	ModifyTransaction (camt.007)	384
14.3.5.1	Overview and scope of the message	384
14.3.5.2	Schema	385
14.3.5.3	The message in business context	386
14.3.6	GetBusinessDayInformation (camt.018)	386
14.3.6.1	Overview and scope of the message	386
14.3.6.2	Schema	386
14.3.6.3	The message in business context	387
14.3.7	ReturnBusinessDayInformation (camt.019)	388
14.3.7.1	Overview and scope of the message	388
14.3.7.2	Schema	389
14.3.7.3	The message in business context	389
14.3.8	ModifyStandingOrder (camt.024)	392
14.3.8.1	Overview and scope of the message	392
14.3.8.2	Schema	393
14.3.8.3	The message in business context	393
14.3.9	Receipt (camt.025)	397
14.3.9.1	Overview and scope of the message	397
14.3.9.2	Schema	398
14.3.9.3	The message in business context	399
14.3.10	ResolutionOfInvestigation (camt.029)	406

14.3.10.1	Overview and scope of the message	406
14.3.10.2	Schema.....	407
14.3.10.3	The message in business context	407
14.3.11	GetReservation (camt.046).....	409
14.3.11.1	Overview and scope of the message	409
14.3.11.2	Schema.....	410
14.3.11.3	The message in business context	410
14.3.12	ReturnReservation (camt.047).....	411
14.3.12.1	Overview and scope of the message	411
14.3.12.2	Schema.....	411
14.3.12.3	The message in business context	412
14.3.13	ModifyReservation (camt.048).....	414
14.3.13.1	Overview and scope of the message	414
14.3.13.2	Schema.....	414
14.3.13.3	The message in business context	415
14.3.14	DeleteReservation (camt.049)	416
14.3.14.1	Overview and scope of the message	416
14.3.14.2	Schema.....	417
14.3.14.3	The message in business context	417
14.3.15	LiquidityCreditTransfer (camt.050)	418
14.3.15.1	Overview and scope of the message	418
14.3.15.2	Schema.....	419
14.3.15.3	The message in business context	420
14.3.16	BankToCustomerStatement (camt.053)	425
14.3.16.1	Overview and scope of the message	425
14.3.16.2	Schema.....	426
14.3.16.3	The message in business context	427
14.3.17	BankToCustomerDebitCreditNotification (camt.054).....	442
14.3.17.1	Overview and scope of the message	442
14.3.17.2	Schema.....	443
14.3.17.3	The message in business context	444
14.3.18	FIToFIPaymentCancellationRequest (camt.056)	452
14.3.18.1	Overview and scope of the message	452
14.3.18.2	Schema.....	452
14.3.18.3	The message in business context	453
14.3.19	GetStandingOrder (camt.069)	457
14.3.19.1	Overview and scope of the message	457
14.3.19.2	Schema.....	457
14.3.19.3	The message in business context	458
14.3.20	ReturnStandingOrder (camt.070)	459

14.3.20.1	Overview and scope of the message	459
14.3.20.2	Schema.....	459
14.3.20.3	The message in business context	460
14.3.21	DeleteStandingOrder (camt.071).....	466
14.3.21.1	Overview and scope of the message	466
14.3.21.2	Schema.....	466
14.3.21.3	The message in business context	466
14.3.22	BillingReportRequest (camt.076).....	468
14.3.22.1	Overview and scope of the message	468
14.3.22.2	Schema.....	468
14.3.22.3	The message in business context	468
14.3.23	BillingReport (camt.077)	468
14.3.23.1	The message in business context.....	468
14.3.23.2	Schema.....	468
14.3.23.3	The message in business context	468
14.3.24	AuditTrailQuery (camt.097).....	468
14.3.24.1	Overview and scope of the message	468
14.3.24.2	Schema.....	469
14.3.24.3	The message in business context	469
14.3.25	AuditTrailReport (camt.098).....	470
14.3.25.1	Overview and scope of the message	470
14.3.25.2	Schema.....	471
14.3.25.3	The message in business context	472
14.3.26	DirectDebitMandateQuery (camt.099)	474
14.3.26.1	Overview and scope of the message	474
14.3.26.2	Schema.....	474
14.3.26.3	The message in business context	475
14.3.27	DirectDebitMandateReport(camt.100)	476
14.3.27.1	Overview and scope of the message	476
14.3.27.2	Schema.....	476
14.3.27.3	The message in business context	477
14.3.28	ModifyCreditLine (camt.998).....	479
14.3.28.1	Overview and scope of the message	479
14.3.28.2	Schema.....	479
14.3.28.3	The message in business context	480
14.3.29	InsertBalance_RM (camt.998)	481
14.3.29.1	Overview and scope of the message	481
14.3.29.2	Schema.....	481
14.3.29.3	The message in business context	482

14.4	Headers (head)	482
14.4.1	BusinessApplicationHeader (head.001)	482
14.4.1.1	Overview and scope of the message	482
14.4.1.2	Schema.....	483
14.4.1.3	The message in business context	484
14.4.2	BusinessFileHeader (head.002)	485
14.4.2.1	Overview and scope of the message	485
14.4.2.2	Schema.....	485
14.4.2.3	The message in business context	486
14.5	Payments clearing and settlement (pacs).....	487
14.5.1	PaymentStatusReport (pacs.002).....	487
14.5.1.1	Overview and scope of the message	487
14.5.1.2	Schema.....	487
14.5.1.3	The message in business context	488
14.5.2	FinancialInstitutionCreditTransfer (COR) (pacs.009)	491
14.5.2.1	Overview and scope of the message	491
14.5.2.2	Schema.....	492
14.5.2.3	The message in business context	493
14.5.3	FinancialInstitutionDirectDebit (pacs.010)	496
14.5.3.1	Overview and scope of the message	496
14.5.3.2	Schema.....	496
14.5.3.3	The message in business context	497
14.6	Reference data (reda).....	499
14.6.1	PartyQuery (reda.015)	499
14.6.1.1	Overview and scope of the message	499
14.6.1.2	Schema.....	499
14.6.1.3	The message in business context	500
14.6.2	PartyReport (reda.017)	501
14.6.2.1	Overview and scope of the message	501
14.6.2.2	Schema.....	501
14.6.2.3	The message in business context	502
14.6.3	CashAccountAuditTrailQuery (reda.039).....	504
14.6.3.1	Overview and scope of the message	504
14.6.3.2	Schema.....	504
14.6.3.3	The message in business context	505
14.6.4	CashAccountAuditTrailReport (reda.040).....	506
14.6.4.1	Overview and scope of the message	506
14.6.4.2	Schema.....	506
14.6.4.3	The message in business context	507

14.6.5	PartyAuditTrailQuery (reda.042).....	509
14.6.5.1	Overview and scope of the message	509
14.6.5.2	Schema.....	509
14.6.5.3	The message in business context	510
14.6.6	PartyAuditTrailReport (reda.043).....	510
14.6.6.1	Overview and scope of the message	510
14.6.6.2	Schema.....	511
14.6.6.3	The message in business context	512
14.6.7	CalendarQuery(reda.064).....	514
14.6.7.1	Overview and scope of the message	514
14.6.7.2	Schema.....	514
14.6.7.3	The message in business context	515
14.6.8	CalendarReport(reda.065).....	515
14.6.8.1	Overview and scope of the message	515
14.6.8.2	Schema.....	515
14.6.8.3	The message in business context	516
15	Specific messages for CBs	518
15.1	ReturnAccount (camt.004) - specific for CBs	518
15.1.1	Overview and scope of the message.....	518
15.1.2	Schema.....	518
15.1.3	The message in business context	518
15.2	ModifyTransaction (camt.007) - specific for CBs.....	518
15.2.1	Overview and scope of the message.....	518
15.2.2	Schema.....	518
15.2.3	The message in business context	518
15.3	Receipt (camt.025) - specific for CBs	519
15.3.1	Overview and scope of the message.....	519
15.3.2	Schema.....	519
15.3.3	The message in business context	519
15.4	ResolutionOfInvestigation (camt.029) - specific for CBs	519
15.4.1	Overview and scope of the message.....	519
15.4.2	Schema.....	519
15.4.3	The message in business context	519
15.5	FIToFIPaymentCancellationRequest (camt.056) - specific for CBs	519
15.5.1	Overview and scope of the message.....	519
15.5.2	Schema.....	519
15.5.3	The message in business context	519

15.6	camt.998 - AuthorizePenalty_RM	520
15.6.1	Overview and scope of the message.....	520
15.6.2	Schema.....	520
15.6.3	The message in business context	521
15.7	camt.998 - GetPenalty_RM	521
15.7.1	Overview and scope of the message.....	521
15.7.2	Schema.....	522
15.7.3	The message in business context	522
15.8	camt.998 - ReturnPenalty_RM	523
15.8.1	Overview and scope of the message.....	523
15.8.2	Schema.....	523
15.8.3	The message in business context	524
15.9	camt.998 - InsertCompulsoryReserve	527
15.9.1	Overview and scope of the message.....	527
15.9.2	Schema.....	527
15.9.3	The message in business context	528
15.10	camt.998 - GetCompulsoryReserve_RM.....	528
15.10.1	Overview and scope of the message.....	528
15.10.2	Schema.....	529
15.10.3	The message in business context	529
15.11	camt.998 - ReturnCompulsoryReserve_RM.....	530
15.11.1	Overview and scope of the message.....	530
15.11.2	Schema.....	531
15.11.3	The message in business context	531
15.12	camt.998 - SendPeriodicFlow_RM	533
15.12.1	Overview and scope of the message.....	533
15.12.2	Schema.....	534
15.12.3	The message in business context	534
Part IV - Appendixes		537
16 Index and digital signature.....		537
16.1	Index of business rules and error codes.....	537
16.2	Digital signature on business layer	574
16.3	Mechanism and introduction for signature constructions	574
16.4	Use of XML and canonicalisation algorithm.....	575
16.5	Message type 1: file with multiple ISO 20022 messages	575

16.6	Message type 2: single ISO 20022 message	580
16.7	ESMIG digital signature services usage of “ds:object”, attribute ID of the “signature” and “keyinfo”, anchor of trust	584
17	Glossary.....	587
18	List of abbreviations	609

List of figures

Figure 1 - Structure of the CLM UDFS	30
Figure 2 - Connectivity (U2A-A2A)	34
Figure 3 - Technical sender authentication	37
Figure 4 - Inbound routing	44
Figure 5 - Outbound routing.....	46
Figure 6 - Processing times windows	72
Figure 7 - pacs.009 CBOs	85
Figure 8 - pacs.009 CBOs technical validation failed	87
Figure 9 - pacs.009 CBOs business validation failed	88
Figure 10 - pacs.010 CBOs	90
Figure 11 - pacs.010 CBOs - technical validation failed.....	91
Figure 12 - pacs.010 CBOs - validation failed	92
Figure 13 - Amend payment succeeded.....	97
Figure 14 - Amend payment failed	99
Figure 15 - Revoke payment via camt.056 – positive	101
Figure 16 - Revoke payment via camt.056 - negative.....	103
Figure 17 - Entry disposition of CBOs	108
Figure 18 - Entry disposition of liquidity transfers.....	110
Figure 19 - Available liquidity.....	114
Figure 20 - Example camt.050 liquidity transfer from MCA to RTGS DCA	119
Figure 21 - Example camt.050 liquidity transfer from RTGS DCA to MCA	121
Figure 22 - camt.050 liquidity transfer intra-CLM	123
Figure 23 - Example camt.050 liquidity transfer inter-service/component	125
Figure 24 - Liquidity transfer from CLM triggered by automatic processes.....	128
Figure 25 - One time reservation with immediate effect.....	133
Figure 26 - One time reservation with immediate effect.....	135
Figure 27 - Deletion reservation with immediate effect	137
Figure 28 - Breach of floor/ceiling threshold – notification	142
Figure 29 - Breach of floor/ceiling threshold - automatic liquidity transfer	143
Figure 30 - camt.050 - setup overnight deposit.....	150
Figure 31 - camt.050 - reverse overnight deposit.....	152
Figure 32 - Reimburse overnight deposit and booking of interest.....	154

Figure 33 - Marginal lending “on request”	157
Figure 34 - File state diagram.....	165
Figure 35 - Message state diagram.....	166
Figure 36 - Inbound CLM message state diagram	168
Figure 37 - Task queue order state diagram	171
Figure 38 - Common reference data objects and the hierarchical party model	194
Figure 39 - Data scopes	195
Figure 40 - Access rights configuration steps	197
Figure 41 - Access rights configuration process (A).....	206
Figure 42 - Access rights configuration process (B).....	207
Figure 43 - Example - configuration of access rights at party level by the operator	208
Figure 44 - Configuration of access rights at user level	209
Figure 45 - Example - archiving and purging after deletion of a common reference data object.....	219
Figure 46 - Lifecycle of common reference data objects with unlimited validity period	221
Figure 47 - Lifecycle of common reference data objects with limited validity period	222
Figure 48 - Modify credit line process - increase.....	238
Figure 49 - Modify credit line process - decrease	240
Figure 50 - pacs.009 connected payment	243
Figure 51 - pacs.010 connected payment	245
Figure 52 - UML conventions – example I.....	255
Figure 53 - UML conventions - example II.....	256
Figure 54 - UML conventions - example III.....	257
Figure 55 - Send file	258
Figure 56 - Send CLM cash transfer order	260
Figure 57 - Revoke/cancel payment order	262
Figure 58 - Amend payment order.....	263
Figure 59 - Execute CLM standing order.....	265
Figure 60 - Standard CLM settlement	268
Figure 61 - Floor and ceiling processing	271
Figure 62 - Automated liquidity transfer.....	273
Figure 63 - Process reject time instructions	274
Figure 64 - Settle connected payment.....	276
Figure 65 - Increase credit line	277

Figure 66 - Decrease credit line	279
Figure 67 - Setup marginal lending on request	281
Figure 68 - ECMS - setup marginal lending on request	283
Figure 69 - Reservation management	286
Figure 70 - Reject payments (EoD)	287
Figure 71 - Process automated marginal lending.....	289
Figure 72 - Process minimum reserve.....	291
Figure 73 - Reimburse marginal lending	293
Figure 74 - Process overnight deposit.....	295
Figure 75 - Revalidate warehoused payments at SoD	297
Figure 76 - Maintain reservation CLM	299
Figure 77 - CLM execute query	301
Figure 78 - Receive report	304
Figure 79 - CLM receive system notification	305
Figure 80 - Common reference data maintenance process	307
Figure 81 - Common reference data query process.....	310
Figure 82 - DMT file upload process	312
Figure 83 - Business file header	333
Figure 84 - Message type 1 signature example	576
Figure 85 - Message type 2 signature example	581
Figure 86 - Reference to the BAH (AppHdr).....	582
Figure 87 - Reference to the message (e.g. semt.013).....	582

List of tables

Table 1 - Services and components	45
Table 2 - Setup of parties for CLM	47
Table 3 - Party reference data attributes	50
Table 4 - Account reference data attributes	56
Table 5 - Direct debit mandate reference data attributes	57
Table 6 - Standing liquidity transfer order reference data attributes	58
Table 7 - Message subscription parameter types	60
Table 8 - Applicable parameter types for outgoing messages	60
Table 9 - Setup of groups for CLM	62
Table 10 - The MFI reference data attributes	63
Table 11 - Attributes of the CLM scheduled events	64
Table 12 - Attributes of the CLM currency	65
Table 13 - Attributes of the CLM rates	66
Table 14 - Attributes of the reserve management maintenance periods	66
Table 15 - Attributes of the duplicate check	67
Table 16 - Attributes of the warehoused payment period	67
Table 17 - Main windows during the business day	71
Table 18 - Settlement items resulting from credit transfers	78
Table 19 - Postings	78
Table 20 - Transactions on DCA side	80
Table 21 - Payments with set execution time indicators	83
Table 22 - CB credit transfer (technical and business validations passed)	86
Table 23 - CB credit transfer (technical validation failure)	87
Table 24 - CB credit transfer (business validation failure)	88
Table 25 - CB direct debit (technical and business validations passed)	90
Table 26 - CB direct debit (technical validation failure)	91
Table 27 - CB direct debit (business validation failure)	92
Table 28 - Possible amendment types in CLM	95
Table 29 - Successful amendment of payment	98
Table 30 - Failed amendment of payment	100
Table 31 - Successful revocation of pending payment	102
Table 32 - Failed revocation of payment	104

Table 33 - Effective settlement order.....	105
Table 34 - Entry disposition of liquidity transfers - example	111
Table 35 - Effect of changing the execution time	112
Table 36 - Effect of changing the order of queued CBOs	112
Table 37 - Origin of possible events	113
Table 38 - Underlying liquidity transfer characteristics	117
Table 39 - Liquidity transfer from MCA to RTGS DCA.....	120
Table 40 - Liquidity transfer from RTGS DCA to MCA	122
Table 41 - Liquidity transfer intra-CLM	124
Table 42 - Liquidity transfer inter-service/component	126
Table 43 - Liquidity transfer inter-service/component	129
Table 44 - Creation of a one-time liquidity reservation with immediate effect.....	134
Table 45 - Modification of a one-time liquidity reservation with immediate effect	136
Table 46 - "Resetting to zero" reservation.....	138
Table 47 - Liquidity tapping in CLM	139
Table 48 - Usage of reserve for CBOs – numeric example.....	140
Table 49 - Setup overnight deposit.....	151
Table 50 - Reverse overnight deposit.....	153
Table 51 - Reimburse overnight deposit.....	155
Table 52 - Marginal lending “on request”	158
Table 53 - Process steps of marginal lending reimbursement and interest calculation	160
Table 54 - Process steps of marginal lending reimbursement and interest calculation	161
Table 55 - CLM file statuses	166
Table 56 - CLM message statuses	167
Table 57 - Cash transfer statuses	170
Table 58 - Task queue order statuses	174
Table 59 - Parameters for the set-up of a report	176
Table 60 - Initiating queries for CLM	178
Table 61 - Access rights management	183
Table 62 - Party data management	184
Table 63 - Cash account data management	187
Table 64 - Message subscription configuration	188
Table 65 - Report configuration	188

Table 66 - Reference data queries	192
Table 67 - TIPS functions	193
Table 68 - Other.....	193
Table 69 - User privileges (data scope).....	195
Table 70 - Privilege assignment options.....	198
Table 71 - Assignment of privileges to roles.....	198
Table 72 - Assignment of privileges to users.....	200
Table 73 - Assignment of privileges to parties	201
Table 74 - Cascade process when revoking privileges	202
Table 75 - Assignment of privilege to party and users	203
Table 76 - Cascade process when revoking roles	204
Table 77 - Assignment of privileges to roles.....	205
Table 78 - Assignment of roles to users	205
Table 79 - Assignment of privileges to users.....	205
Table 80 - Message subscription parameter types	210
Table 81 - Definition of a new message subscription rule set.....	211
Table 82 - Definition of the rules for a new message subscription rule set.....	211
Table 83 - Common reference data objects	213
Table 84 - Management of reference data objects in DMT	215
Table 85 - Management of reference data objects in A2A mode	215
Table 86 - Common reference data objects with unlimited validity period	216
Table 87 - Common reference data objects with limited validity period	219
Table 88 - CRDM data segregation per service/component	228
Table 89 - Tasks of the CBs	234
Table 90 - Process description for increasing credit line.....	239
Table 91 - Process description for decreasing credit line.....	241
Table 92 - Connected payment (pacs.009) – decrease of a credit line.....	244
Table 93 - Connected payment (pacs.010) – increase of a credit line.....	246
Table 94 - List of CB specific queries	249
Table 95 - Message sent by the CB to the CLM component.....	252
Table 96 - Message sent by the CB to the CLM component.....	252
Table 97 - Message sent by the CB to the CLM component.....	252
Table 98 - Messages sent by the submitting actor to CLM	259

Table 99 - Process description amend payment	264
Table 100 - Process description execute CLM standing order.....	266
Table 101 - Notification message sent by the CLM component to CB.....	292
Table 102 - A2A messages for query processing	303
Table 103 - Receive report	304
Table 104 - Receive system notification.....	305
Table 105 - Common reference data maintenance process	307
Table 106 - CRDM messages	308
Table 107 - Common reference data query process.....	310
Table 108 - Common reference data query messages	311
Table 109 - DMT files specifications.....	314
Table 110 - FIN X-Character set	325
Table 111 - List of messages	339
Table 112 - AccountQueryList (acmt.025) – usage case Cash account reference data query.....	341
Table 113 - AccountListReport (acmt.026) – usage case Cash account reference data query response....	344
Table 114 - AccountListReport (acmt.026) – usage case Error	344
Table 115 - ReportQueryRequest (admi.005) - usage case Account Statement Query	346
Table 116 - ReceiptAcknowledgement(admi.007) – usage case Negative Receipt Acknowledgement.....	349
Table 117 - GetAccount (camt.003) – usage case Available Liquidity CLM Query	352
Table 118 - GetAccount (camt.003) – usage case Available Liquidity CLM Query	352
Table 119 - GetAccount (camt.003) – usage case Available Liquidity CLM Query	353
Table 120 - GetAccount (camt.003) – usage case Query Minimum Reserve Requirements per Participant (CB only)	354
Table 121 - ReturnAccount (camt.004) – usage case Available Liquidity CLM Query (Data response)	358
Table 122 - ReturnAccount (camt.004) – usage case Available Liquidity CLM Query (Error response).....	359
Table 123 - ReturnAccount (camt.004) – usage case Minimum Reserve Query (Data response).....	361
Table 124 - ReturnAccount (camt.004) – usage case Query Minimum Reserve Requirements per Participant (Data response)	363
Table 125 - ReturnAccount (camt.004) – usage case Floor Notification.....	365
Table 126 - ReturnAccount (camt.004) – usage case Ceiling Notification.....	366
Table 127 - ReturnAccount (camt.004) – usage case Reimbursed Marginal Lending	368
Table 128 - GetTransaction (camt.005) – usage case CLM Payment Order Query	376
Table 129 - GetTransaction (camt.005) – usage case Query Standing Facilities Transaction.....	377
Table 130 - ReturnTransaction (camt.006) – usage case CLM Payment Order Query (Data response)	383

Table 131 - ReturnTransaction (camt.006) – usage case CLM Payment Order Query (Error response).....	383
Table 132 - ReturnTransaction (camt.006) – usage case Query Standing Facilities Transaction (Data response)	384
Table 133 - ModifyTransaction (camt.007) – usage case TBC1	386
Table 134 - GetBusinessDayInformation (camt.018) – usage case Event Query.....	387
Table 135 - GetBusinessDayInformation (camt.018) – usage case System Time Query.....	388
Table 136 - ReturnBusinessDayInformation (camt.019) – usage case System Time Query (Data response)	390
Table 137 - ReturnBusinessDayInformation (camt.019) – usage case System Time Query (Error response)	390
Table 138 - ReturnBusinessDayInformation (camt.019) – usage case Event Query (Data response).....	391
Table 139 - ReturnBusinessDayInformation (camt.019) – usage case Event Query (Error response)	392
Table 140 - ReturnBusinessDayInformation (camt.019) – usage case System Notification.....	392
Table 141 - ModifyStandingOrder (camt.024) – usage case RTGS modify standing order.....	394
Table 142 - ModifyStandingOrder (camt.024) – usage case ASI6 RTGS Modify standing order	396
Table 143 - ModifyStandingOrder (camt.024) – usage case CLM modify standing order	397
Table 144 - Receipt (camt.025) – usage category case Settlement – Rejected	400
Table 145 - Receipt (camt.025) – usage category case Settlement – Settled	400
Table 146 - Receipt (camt.025) – usage category case Settlement – Unsettled	401
Table 147 - Receipt (camt.025) – usage category case CLM Status – Rejected	402
Table 148 - Receipt (camt.025) – usage category case CLM Status – Confirmed.....	402
Table 149 - Receipt (camt.025) – usage category case CLM Status – Validation Error.....	403
Table 150 - Receipt (camt.025) – usage category case Liquidity Management – Approved.....	404
Table 151 - Receipt (camt.025) – usage category case CRDM – Rejected	405
Table 152 - Receipt (camt.025) – usage category case CRDM – Completed	405
Table 153 - Receipt (camt.025) – usage category case CRDM – Queued.....	406
Table 154 - ResolutionOfInvestigation (camt.029) – usage case Rejection of Payment Order Cancellation Request	409
Table 155 - GetReservation (camt.046) – usage case Current Reservations Query	411
Table 156 - ReturnReservation (camt.047) – usage case Current Reservations Query (Data response) ...	413
Table 157 - ReturnReservation (camt.047) – usage case Current Reservations Query (Error response) ...	414
Table 158 - ModifyReservation (camt.048) - usage case Modify Reservation Request	416
Table 159 - ModifyReservation (camt.048) - usage case Modify Standing Order for Reservation.....	416
Table 160 - DeleteReservation (camt.049) – usage case Delete Reservation Request.....	418

Table 161 - DeleteReservation (camt.049) – usage case Delete Standing Order for Reservation	418
Table 162 - LiquidityCreditTransfer (camt.050) – usage case Automated Interservice Liquidity Transfer Order	421
Table 163 - LiquidityCreditTransfer (camt.050) – usage case Payment Order Message	422
Table 164 - LiquidityCreditTransfer (camt.050) – usage case Inter-Service Liquidity Transfer (Floor Processing)	423
Table 165 - LiquidityCreditTransfer (camt.050) – usage case Inter-Service Liquidity Transfer (Ceiling Processing)	424
Table 166 - LiquidityCreditTransfer (camt.050) – usage case Liquidity Transfer Order (Overnight Deposit)	425
Table 167 - BankToCustomerStatement (camt.053) – usage case Query Response Message for Business Data	432
Table 168 - BankToCustomerStatement (camt.053) – usage case Statement of Accounts.....	437
Table 169 - BankToCustomerStatement (camt.053) – usage case CB General Ledger	442
Table 170 - BankToCustomerDebitCreditNotification (camt.054) – usage case Liquidity Transfer Settlement Notification	446
Table 171 - BankToCustomerDebitCreditNotification (camt.054) – usage case Payment Order Settlement Notification (Intra-Service LTs)	447
Table 172 - BankToCustomerDebitCreditNotification (camt.054) – usage case Payment Order Settlement Notification (Inter-Service LTs)	447
Table 173 - BankToCustomerDebitCreditNotification (camt.054) – usage case Payment Order Settlement Notification (CBOs)	448
Table 174 - BankToCustomerDebitCreditNotification (camt.054) – usage case Settlement Notification (CLM Standing Order SoD)	448
Table 175 - BankToCustomerDebitCreditNotification (camt.054) – usage case Connected Payment Settlement Notification	449
Table 176 - BankToCustomerDebitCreditNotification (camt.054) – usage case Modify Credit Line Settlement Notification	450
Table 177 - BankToCustomerDebitCreditNotification (camt.054) – usage case Marginal Lending Interest (on request) Settlement Notification.....	450
Table 178 - BankToCustomerDebitCreditNotification (camt.054) – usage case Marginal Lending Interest (automated) Settlement Notification.....	451
Table 179 - BankToCustomerDebitCreditNotification (camt.054) – usage case Overnight Deposit Interest Payment Settlement Notification.....	451
Table 180 - FIToFIPaymentCancellationRequest (camt.056) – usage case Cancel Payment Order	457
Table 181 - GetStandingOrder (camt.069) – usage case Get Standing order details	458

Table 182 - ReturnStandingOrder (camt.070) – usage case RTGS return standing order details	461
Table 183 - ReturnStandingOrder (camt.070) – usage case ASI6 RTGS return standing order details	463
Table 184 - ReturnStandingOrder (camt.070) – usage case CLM Return standing order details	465
Table 185 - ReturnStandingOrder (camt.070) – usage case Error	466
Table 186 - DeleteStandingOrder (camt.071) – usage case Delete standing order	467
Table 187 - AuditTrailQuery (camt.097) – usage case Audit Trail for CLM Query.....	470
Table 188 - AuditTrailReport (camt.098) – usage case Audit Trail for CLM Query (Data response).....	473
Table 189 - AuditTrailReport (camt.098) – usage case Audit Trail for CLM Query (Data response).....	473
Table 190 - DirectDebitMandateQuery (camt.099) – usage case Direct debit mandate query	475
Table 191 - DirectDebitMandateReport (camt.100) – usage case Direct debit mandate report	478
Table 192 - DirectDebitMandateReport (camt.100) – usage case Error	479
Table 193 - CashManagementProprietaryMessage ModifyCreditLine (camt.998) – usage case TBC	480
Table 194 - CashManagementProprietaryMessage InsertBalanceRM (camt.998) – usage case TBC.....	482
Table 195 - BusinessFileHeader (head.002) – usage case Consistency and completeness checks.....	486
Table 196 - PaymentStatusReport (pacs.002) – usage case Payment Order Settlement Notification.....	489
Table 197 - PaymentStatusReport (pacs.002) – usage case Payment Order Rejection Notification	490
Table 198 - PaymentStatusReport (pacs.002) – usage case Payment Processing Notification	491
Table 199 - FinancialInstitutionCreditTransfer (pacs.009) – usage case Payment Order Message.....	495
Table 200 - FinancialInstitutionDirectDebit (pacs.010) – usage case Payment Order Message.....	499
Table 201 - PartyQuery (reda.015) – usage case Party reference data query	501
Table 202 - PartyReport (read.017) – usage case Party reference data response	503
Table 203 - PartyReport (read.017) – usage case Error	504
Table 204 - CashAccountAuditTrailQuery (camt.039) – usage case Cash account audit trail query	505
Table 205 - CashAccountAuditTrailReport (reda.040) – usage case Cash account audit trail report	508
Table 206 - CashAccountAuditTrailReport (reda.040) – usage case Error.....	508
Table 207 - PartyAuditTrailQuery (reda.042) – usage case Party audit trail query.....	510
Table 208 - PartyAuditTrailReport (reda.043) – usage case Party audit trail report	513
Table 209 - PartyAuditTrailReport (reda.043) – usage case Error.....	513
Table 210 - CalendarQuery (reda.064) – usage case Calendar query	515
Table 211 - CalendarReport (reda.065) – usage case Calendar report.....	517
Table 212 - CalendarReport (reda.065) – usage case Error.....	517
Table 213 - CashManagementProprietary MessageAuthorisePenalty (camt.998) – usage case TBC	521

Table 214 - CashManagementProprietaryMessage GetPenaltyRM (camt.998) – usage case Query Penalty	523
Table 215 - CashManagementProprietaryMessage ReturnPenaltyRM (camt.998) – usage case Query Penalty (Data response)	526
Table 216 - CashManagementProprietaryMessage ReturnPenaltyRM (camt.998) – usage case Query Penalty (Error response)	527
Table 217 - CashManagementProprietaryMessage InsertCompulsoryReserve – usage case TBC	528
Table 218 - CashManagementProprietaryMessage GetCompulsoryReserveRM (camt.998) – usage case Query Minimum Reserve of a Banking Community	530
Table 219 - CashManagementProprietaryMessage ReturnCompulsoryReserveRM (camt.998) – usage case Query Minimum Reserve of a Banking Community (Data response)	532
Table 220 - CashManagementProprietaryMessage ReturnCompulsoryReserveRM (camt.998) – usage case Query Minimum Reserve of a Banking Community (Error response)	533
Table 221 - CashManagementProprietaryMessage SendPeriodicFlowRM (camt.998) – usage case Notify CBs on Minimum Reserve Fulfilment	536
Table 222 - CLM validation rules	552
Table 223 - CRDM validation rules	573
Table 224 - CRDM validation rules	574
Table 225 - ESMIG validation rules	574

Introduction

This document describes the Central Liquidity Management (CLM) as a component of T2 and CLM Account Holders' interactions with other components and services. CLM is a business component of T2 providing information on central bank (CB) liquidity and managing credit lines and central bank operations (CBOs). In addition, CLM is the central component for funding the Real-time Gross Settlement (RTGS) component and the TARGET2-Securities (T2S) and Target Instant Payment Settlement (TIPS) Services.

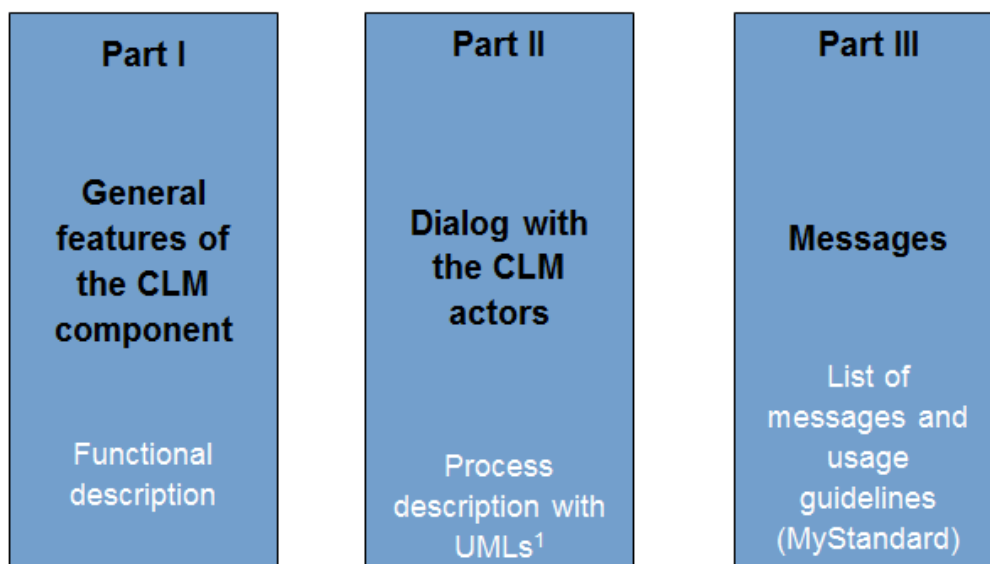
The document is intended to guide CLM Account Holders to the proper understanding of the component and to offer all the information needed for the implementation of the software interface on their side. Furthermore, the document focuses on the provision of information to CLM Account Holders to design and build the interface of their business application with CLM (application-to-application (A2A)/user-to-application (U2A)). The CLM UDFS is publicly available.

The document is divided into three parts.

- I The first part provides a full description of all the CLM features and the related accounts and application processes, functional details concerning access to CLM and connectivity, dependencies and interactions with other services/components, operations and support features. The background information provided in chapter [Overview of CLM component](#) [▶ 32] supports the understanding of the CLM component with its applications and the interaction with the common components described in the following chapters. Afterwards, it guides the reader through the CLM features (i.e. participation and accounts) and gives an overview of common components used by CLM (e.g. Common Reference Data Management (CRDM), Data Warehouse (DWH)). The contingency services are explained in chapter [Contingency services](#) [▶ 232] and CB specific information is provided in chapter [Additional information for CBs](#) [▶ 234].
- I The second part provides process descriptions, which allow CLM Account Holders to interact with CLM 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 CLM 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](#) [▶ 306] describes the dialogue between CRDM and participants via A2A. Subsequently, also the interaction with the Eurosystem Single Market Infrastructure Gateway (ESMIG) is outlined in chapter [Dialogues and processes](#) [▶ 306].
- I The third part provides a detailed description of all XML messages CLM Account Holders may use to interact in A2A mode with CLM. The description of the messages includes 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 CLM 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 CLM. On the other hand, technical readers involved in the specification and development of technical interfaces to CLM may not be interested in the complete description of the features CLM offers. They would probably search the necessary information to design and build the interface of the CLM Account Holders' business application with CLM. 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.

- I [Overview of CLM component](#) [▶ 32], which summarises the CLM features and functionalities;
- I [Access to CLM](#) [▶ 34], which focuses on how to connect to CLM including authentication and authorisation processes and explains the envisaged usage of access rights depending on the respective roles;
- I [Parties and accounts](#) [▶ 47], which provides a general description of the main reference data needed for CLM and the accounts maintained in CLM, specifying how they are used for the settlement of a liquidity transfer (e.g. which parties and related accounts are involved and how to set-up groups for monitoring the liquidity transfer activities);

-
- I [Contingency services](#) [▶ 232], which informs how to cope with a defined contingency situation.

Business oriented perspective

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

- I [Business day](#) [▶ 69], where the business reader finds an overview of respective processes and schedules;
- I [Business and features description](#) [▶ 81], which informs about the settlement process of payments as well as the liquidity-, reserve- and information-management;
- I [Overview of used common components in CLM component](#) [▶ 180] completes the view on the message transfers used in CLM;
- I [Cash transfer order processing](#) [▶ 259] to find a description of the processing of a payment order and useful information in order to understand the management of liquidity
- I [Dialogues and processes between CRDM and CRDM Actor](#) [▶ 306] wherein query information may be of relevance;
- I [Index of business rules and error codes](#) [▶ 537] including the relevant codes to perform functional checks.

Technical oriented perspective

- I [Processes with CLM](#) [▶ 254] respectively [Dialogue between CRDM and CRDM Actors](#) [▶ 306], where an overview of the possible A2A dialogue with CLM is defined. Each sub-chapter of this chapter describes the flows within and to and from CLM. The reader can focus on the functionality of note, analysing the procedures and main scenarios;
- I [Part III - Catalogue of messages](#) [▶ 317], where a detailed description of the content of a given XML message is provided.

This chapter is subject to further review depending on the subsequent maintenance of the UDFS document in the future.

Part I - General features of the CLM component

1 Overview of CLM component

CLM ensures the adequate provisioning and clear allocation of liquidity for the different settlement purposes across all TARGET Services and accounts in a currency.

The primary aim of CLM is to offer a centralised mechanism for the steering, monitoring and management of payment capacity. All credit institution's transactions with its CB are managed in CLM (including the ones related to CBOs such as minimum reserve management and standing facilities management). The interaction with CBs is segregated from the real-time interbank/customer payments as well as the ancillary system transactions in RTGS.

CLM provides:

- I instruments for the management of liquidity such as immediate/standing or automatic liquidity transfer orders and floor/ceiling definitions;
- I information tools, queries and reporting for the status monitoring of liquidity and processing results.

In order to reach these objectives, the CLM holds main cash accounts (MCAs) as the central source of liquidity; the MCA is used for all CBOs as well as the management of the credit line (if applicable). The available liquidity can be transferred to the dedicated cash accounts (DCAs) of RTGS, TIPS and T2S. The minimum reserve calculation and standing facilities take all balances on relevant accounts (MCAs, DCAs) into account. A MCA Holder is responsible for its own liquidity management and for monitoring the settlement process or grant access to another party to perform these tasks on its behalf. The A2A communication between credit institutions and all TARGET Services and common components is based on the ISO 20022 compliant messages.

CLM 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 parties the access to all TARGET Services through the connection with a single certified network service provider. All network service providers require compliance with the same communication interface specifications in A2A mode (in store-and-forward and real-time communication protocol) and U2A mode via Graphical User Interface (GUI).
- I The 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) are set-up and managed (or implemented) in the respective service. The access to all collected data allows to use a billing component as well as queries and reports.

-
- | The 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 the 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. This schedule defines the structure of the business day in the TARGET 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 TARGET Services and common components. To do so, relevant information for each cash account has 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 only. Those tools have interfaces to all applications. They support the monitoring and controlling of the CLM component.

2 Access to CLM

2.1 Connectivity (U2A/A2A)

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

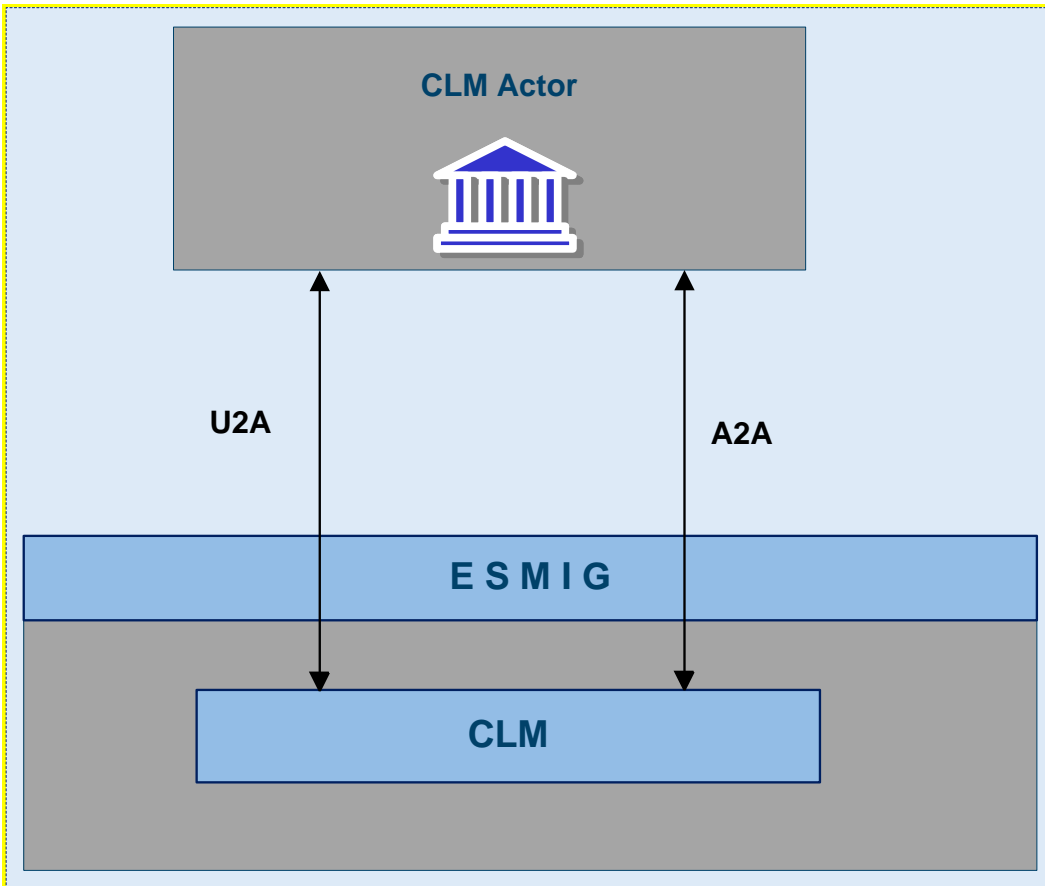


Figure 2 - Connectivity (U2A-A2A)

A2A

Software applications can communicate with CLM exchanging single messages and multi messages (only inbound to CLM). A2A communication relies on Extensible Mark-up Language (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.

U2A

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

2.2 Authentication and authorisation process

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

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

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

CLM then carries out the authorisation of the sender at application level. The DN that is used to sign the A2A message or used to access U2A is linked to one user. The user may have 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](#) [36] 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](#) [37] and [Authorisation process](#) [38] 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 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.

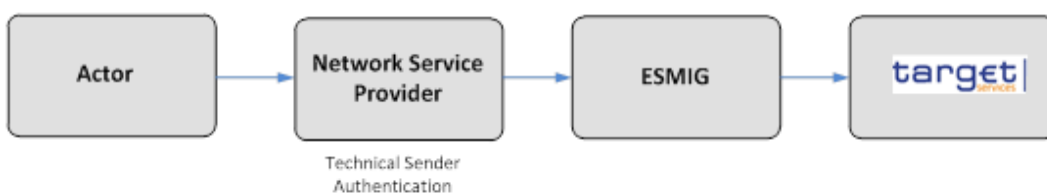


Figure 3 - Technical sender authentication

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](#) [▶ 38]).

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](#) [. 38]).

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 CLM in terms of principles applied to ensure CLM Actors can securely exchange information with CLM.

It means that the following security conditions are met:

- I **Confidentiality:** ensuring that information is accessible only to authenticated and authorised CLM 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

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

2.4.2 Integrity

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

In U2A mode, CLM offers users in addition the possibility to further ensure the data integrity via usage of a dual authorisation concept, the 4-eyes principle. In case this option is chosen for a specified set of CLM operations, a second independent verification and confirmation is required before an operation becomes active in CLM.

2.4.3 Availability

The overall availability of the CLM 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 CLM component.

2.4.4 Monitoring

CLM operational monitoring provides tools to the T2 Operator for the detection of functional or operational problems in real-time. 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

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

2.6 Routing

Communication channels can be categorised as follows.

- store-n-forward
- 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 from 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 are provided in store-n-forward mode in 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 three items.

1. A technical receiver name which is represented by a distinguished name (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 CLM 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 in-bound communication.
Notifications being not as response to an instruction but belonging to a	store-n-forward message	The store-n-forward notification being not as response to an instruction is

RTGS business data exchanges	Communication channel	Deduction of technical address
business case triggered by an instruction, e.g. camt.054		sent to the technical address that is defined in the routing configuration.
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 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 messages are always returned to the technical sender of the message.

The CRDM routing configuration applies to notifications being not as 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 identifies the channel (message-based or file-based) depending on the size of the message to be sent.

There is the case of camt.053 serving two different purposes: statement of account and general ledger. These two purposes are reflected in CRDM as two different message types. This allows to define different technical addresses for these different uses of camt.053.

Connectivity requirements for actors due to routing

Store-n-forward mode:

Each external actor sending store-n-forward traffic to CLM/RTGS has to 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 has to support message and file channel.

Real-time mode:

Each external actors sending real-time traffic to CLM/RTGS has to 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.

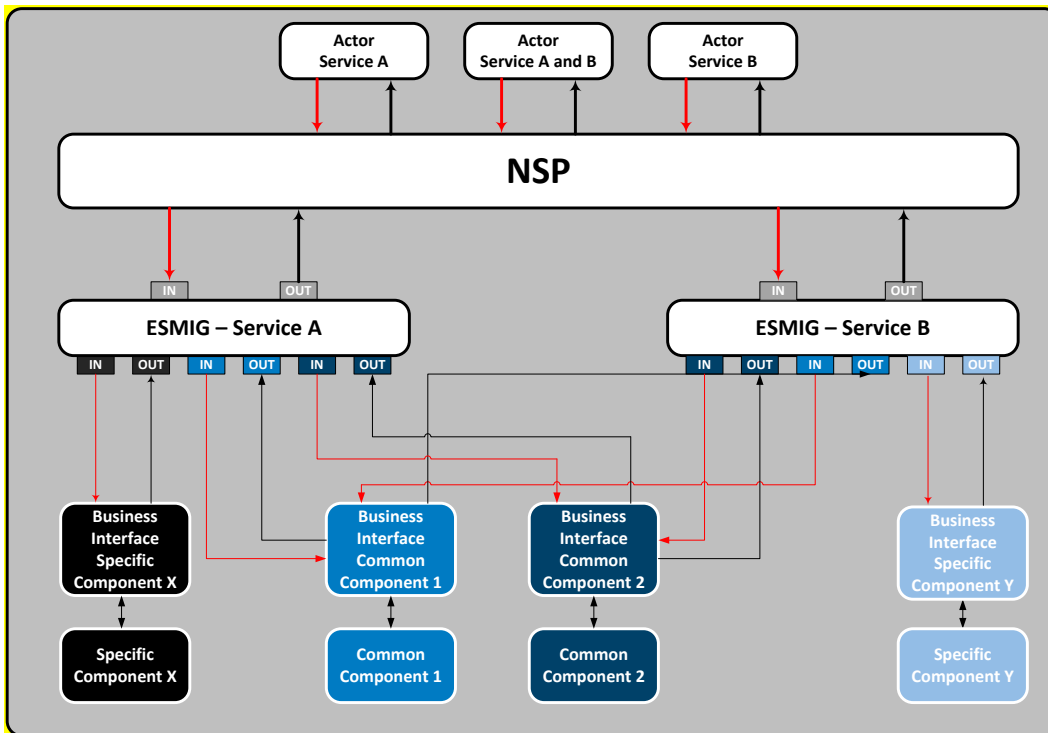


Figure 4 - Inbound routing

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, FILES NF, FILERT U2A
T2	RTGS	A2A: MSGSNF, MSGRT, FILES NF, FILERT U2A
T2	CRDM	A2A: MSGSNF, MSGRT, FILES NF, FILERT U2A

Business service	Component	Communication mode and protocol
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.

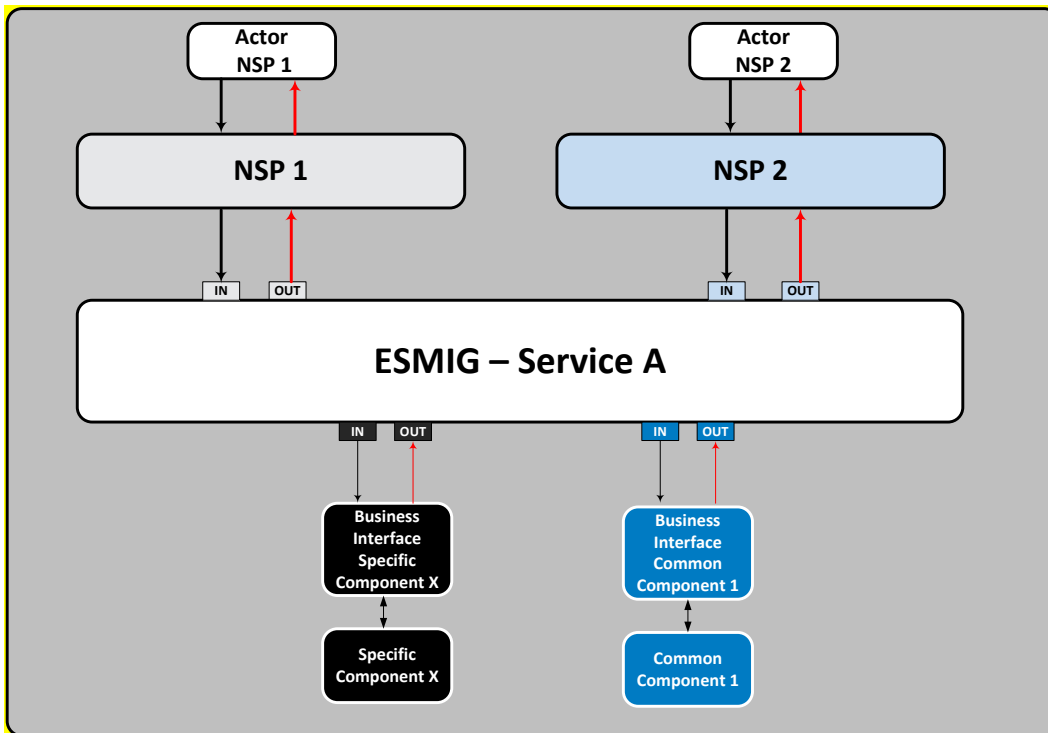


Figure 5 - Outbound routing

3 Parties and accounts

3.1 Parties

The CLM participation model envisions different types of CLM Actors, with different roles and responsibilities, as outlined in chapter [Concept of party in CLM](#) [▶ 48]. T2 Actors that interact with CLM are defined as different parties in CRDM.

This chapter provides a detailed description of all the objects and attributes CRDM stores and CLM uses for its participating T2 Actors. This chapter focuses in particular on the reference data in the context of parties used in CLM. In chapter [Overview of used common components in CLM component](#) [▶ 180] 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](#) [▶ 47] identifies the reference data related to the setup of CLM Actors and it provides detailed information as to who is responsible for the setup of these reference data. Chapter [Concept of party in CLM](#) [▶ 48] defines the concept of party in CRDM and the way this concept relates with the different types of legal entities that can interact with CLM. 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 CLM](#) [▶ 49] illustrates the reference data required by CLM for each T2 Actor, i.e. the way a party can be identified in CLM and which attributes have to be stored for each CLM Actor.

3.1.1 Setup of parties

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

The T2 Operator is responsible for setting up and maintaining party reference data for all CBs in CLM. 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 CLM and specifies the required communication mode.

Reference data object	Responsible actor	Mode
Party (CB)	T2 Operator	U2A
Party (CLM Participant)	CB	U2A

Table 2 - Setup of parties for CLM

CLM 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 CLM Account Holder and the CB. This implies the need to ensure that any given BIC can only be as-

signed to one CLM Account Holder and CB. Different CLM Account Holders or CBs must be assigned to a unique BIC.

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

3.1.2 Concept of party in CLM

The party model of CLM is based on a hierarchical three-level structure. The T2 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 CLM. Similarly, each party belonging to the second level is responsible for the setup of all parties of its community (i.e. CLM Account Holder), represented by parties of the third level.

This means that each CB is responsible for the reference data of its community. CLM MCA Holders are responsible for their own reference data. Further information about the hierarchical model can be found in chapter [Common reference data objects and the hierarchical party model](#) [▶ 193] and information about the data scope are included in chapter [Data scope](#) [▶ 194].

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

- | T2 Operator
- | CB
- | CLM Participant

The **T2 Operators** are the organisational entity that operate i.a. CLM. They are responsible for the initial setup and day-to-day operations of CLM and act 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 T2 Operators are also responsible for setting up and maintaining the reference data of the CBs in CRDM. Upon request of the respective CB T2 Operators may operate CLM functions on behalf of any CLM Actor. They have full access to all live and all archived reference data and transactional data in CLM.

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

In their CB role, they own CB accounts (see [Glossary](#) [▶ 587] for the definition of a CB account); all other account types need to be owned under its CLM Participant role.

CLM Participants represent CLM Actors that own MCAs. CLM Participants are responsible for their own liquidity management and have to make sure that sufficient liquidity is available in the different settlement services that they use. They are responsible for instructing liquidity transfers and monitoring the liquidity usage. However, the setup and maintenance of the MCAs is done by CBs upon request of the respective CLM Account Holder.

3.1.3 Reference data for parties used by CLM

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

Attribute	Description
Party BIC	It specifies the BIC11 to uniquely identify the party in CLM.
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.
Intraday credit indicator	It specifies the intraday credit indicator by either allowing or not allowing intraday credit.
Standing facilities indicator	It specifies the standing facilities indicator by either allowing or not allowing standing facilities.
Minimum reserve obligation	It specifies whether or not the party is subject to/exempted from minimum reserve requirement.
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.
MFI account for minimum reserves	It identifies the account used by the MFI leader for minimum reserves.
Marginal lending account	It specifies the account number of the marginal lending account managed within CLM and maintained by a CB to settle all marginal lending transactions.
Overnight deposit account	It specifies the account number of the overnight deposit account managed within CLM

Attribute	Description
	and maintained by a CB to settle all overnight deposit transactions.
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.
VAT1	It specifies the national rate of value added tax associated with a CB.
VAT 2	It specifies the additional national rate of value added tax associated with a CB.
Account to be credited (for CBs only)	It specifies the cash account to be credited within the billing process. Different accounts may be specified for each different service.
Direct invoicing flag (for CBs only)	This flag indicates whether invoices are sent directly to the CLM Account Holders or whether they are sent via the CB.

Table 3 - Party reference data attributes

3.1.4 Participation types

There are two possibilities to participate in CLM. This chapter gives an overview of the participation types: direct participation and the participation as co-manager.

Direct Participants

Direct Participants have direct access to CLM and hold their own MCAs. They are responsible for their own reference data and for their own liquidity management in CLM and for monitoring the settlement process. Furthermore, they are responsible for all liquidity transfers sent or received on their account(s).

Co-manager

The co-management functionality allows small banks to manage directly their minimum reserve requirements but to delegate cash flow management of the MCA to a different CLM MCA Holder, the so called co-manager. The co-manager is a CLM Account Holder that is allowed to

- | debit the managed MCA and credit its own MCA or DCA or
- | debit the managed MCA and credit a third party MCA or DCA.

3.1.5 Blocking/unblocking party

The blocking/unblocking of CLM Account Holders is possible.

It is up to CBs or any other authority in charge to declare actions to:

- | restrain the disposal of the assets and
- | withdraw the license.

As a consequence of this declaration or withdrawal the affected CLM Account Holder is blocked in CLM. The blocking is under the full responsibility of the respective CB. The CB initiates the blocking at party level (as 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 from which the party is blocked. If not stated, the next calendar date is 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 CLM Account Holder 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](#) [▶ 61].

3.2 Accounts structure and functionalities

Accounts are opened in CLM for the provision of liquidity and the settlement of CBOs and cash withdrawals.

This chapter provides a detailed description of the reference data CRDM stores and CLM uses for all its accounts.

The T2 Operator and/or CBs set-up and maintain the following categories of accounts ¹ in CRDM.

- | MCAs
- | CLM dedicated transit accounts
- | CLM CB accounts
- | overnight deposit accounts
- | marginal lending accounts
- | CB's European Central Bank (ECB) accounts
- | ECB mirror accounts

CLM Account Holders can set-up the following functionalities on their MCAs.

- | floor/ceiling
- | standing liquidity transfer orders

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

- | standing orders for reservation
- | current reservations(s)
- | message subscription
- | report configuration

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

Moreover, it is up to CBs to define the default RTGS DCAs in CLM.

The following chapters define the above mentioned reference data objects.

3.2.1 Account types

This chapter gives an overview of all account types used in CLM.

MCA

A MCA is an account used for the settlement of CBOs, cash withdrawals and liquidity transfers, as well as the management of the credit line (cash side).

A CLM Actor may own several MCAs. However, the credit line can only be assigned to one of them. As soon as a party wants to hold its minimum reserve directly, a MCA is to be opened.

Each MCA is linked to one and only one CLM Account Holder (i.e. the MCA Holder). Furthermore, each MCA Holder may be linked to one or many Liquidity Transfer Groups and to one or many account monitoring groups.

A MCA in CLM is identified by a BIC11 (that must be unique in CLM) and also by an account ID (that must be unique across all settlement services and components). In the case of settlement of credit transfers and direct debits, the CLM Participant's MCA is identified by a unique "BIC11" code. In the case of liquidity transfers, the CLM Account Holder's MCA is identified by the account ID.

It is up to CBs to set-up and maintain MCAs for their CLM MCA Holders.

CLM MCA Holders can establish a link between their MCA and a default RTGS DCA. This link 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, only one of the RTGS DCAs is the default one.

A default link between a MCA and DCAs of other services is not possible.

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

- | pulled from the linked default RTGS DCA (in the event the floor is breached) or
- | pushed to the default RTGS DCA (in the event the ceiling is breached).

It is up to CLM MCA Holders to decide which RTGS DCA should be the default one. CBs are in charge of the setup and maintenance.

CLM dedicated transit account

CLM dedicated transit accounts are accounts owned by CBs. They shall either have a zero or a 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 inter-service liquidity transfer process between CLM and the other services and cannot be involved in the settlement of CBOs. The CLM dedicated transit accounts could not be directly addressed by the CLM MCA Holder in a liquidity transfer.

Each CLM dedicated transit account related to one service is linked to one and only one CB (e.g. the ECB is linked to the CLM dedicated transit accounts for Euro).

There is only one CLM dedicated transit account per settlement service/settlement currency combination in CLM. The CLM dedicated transit accounts for Euro belong to the ECB.

However, the monitoring of CLM dedicated transit accounts is conducted by the T2 Operator.

It is up to the T2 Operator to set-up and maintain the CLM dedicated transit accounts.

CLM CB account

A CB account in CLM is a cash account that is owned by a CB and that is allowed to have a negative balance. It cannot be restricted or limited in its use.

The purpose of the account is to provide liquidity. It is especially used for CBOs and cash withdrawals.

A CLM CB account is identified by a BIC11 (that must be unique in CLM) and by an account ID (that must be unique across all settlement services).

It is up to the T2 Operator to set-up and maintain the CLM CB accounts.

Overnight deposit account

An overnight deposit account is owned by the relevant CB but is opened in the name of the CLM MCA Holder.

There is one overnight deposit account for each CLM MCA Holder subject to standing facilities.

An overnight deposit account in CLM is identified by a unique account ID (that must be unique across all settlement services).

It is up to CBs to set-up and maintain the overnight deposit accounts.

Marginal lending account

A marginal lending account is owned by the relevant CB but is opened in the name of the CLM MCA Holder.

There is one marginal lending account for each CLM MCA Holder subject to standing facilities.

A marginal lending account in CLM is identified by a unique account ID (that must be unique across all settlement services).

It is up to CBs to set-up and maintain the marginal lending accounts.

CB's ECB account

A CB's ECB account is an account that records the CB's asset/liability position towards the ECB in respect of cross-CB community transactions. If two CLM Actors from different countries have an account with the same CB, then a transaction between these two accounts (which is cross-border) is not reflected in the CB's ECB accounts. This account is owned by the relevant CB and is identified by a unique BIC11.

It is up to the T2 Operator to set-up and maintain the CB's ECB accounts.

ECB mirror account

A ECB mirror account is an account owned by the ECB for each CB on which the settlement postings done on the CB's ECB accounts are "mirrored". This account is owned by the ECB and is identified by a unique BIC11.

It is up to the T2 operator to set-up and maintain the ECB mirror accounts.

The following table shows an overview of the attributes of the account reference data objects used in CLM 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 CLM is as follows: <ul style="list-style-type: none"> MCA CLM dedicated transit account CLM CB account overnight deposit account

Attribute	Description
	<ul style="list-style-type: none"> marginal lending account CB's ECB account ECB mirror account
Currency code	It specifies the currency of the account.
Account owner	It specifies the BIC11 of the party owning the account (unique within CLM).
Status	Blocking status for the account. Exhaustive list of possible values: <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	If defined, it specifies the behaviour the system applies in case the floor on a MCA is breached. Exhaustive list of possible values: <ul style="list-style-type: none"> send notification initiate liquidity transfers
Ceiling	If defined, it specifies the behaviour the system applies in case the ceiling on a MCA is breached. Exhaustive list of possible values: <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.
Party to be billed	It specifies the party to whom the invoice is addressed.
Party to be charged	It specifies the party to whom the billable item is assigned, due to a contractual agreement.
MCA to be debited	It specifies the MCA to be debited within the billing process.

Attribute	Description
Management of minimum reserve	It specifies the method for the calculation of minimum reserve fulfilment. Possible values are: <ul style="list-style-type: none"> direct indirect indirect (pool) no
Linked RTGS DCA	It specifies the linked RTGS DCA.
Closing date	Closing date of the account.

Table 4 - Account reference data attributes

3.2.2 Functionalities

This chapter describes the functionalities available at MCAs level.

Direct debit mandate

The direct debit facility can be used in CLM by CBs in case of:

- | settlement of cash withdrawals
- | repayment of monetary policy operations
- | collections of fees.

For each MCA CRDM manages the information about the direct debit(s) the CLM MCA Holder has authorised and the related attributes (e.g. maximum amounts).

It is up to CBs to set-up and maintain the direct debit mandate(s) of a CLM MCA Holder in CRDM, while the definition is done by the CLM MCA Holder. All actions (setup, modify, delete) become effective as of the next business day or on the activation date of the MCA 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 MCA on which the direct debits are authorised.
Payee party identifier	It specifies the party whose payment requests are authorised under this 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.

Attribute	Description
Maximum amount (counter-part)	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 direct debit 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 MCA.
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 5 - Direct debit mandate reference data attributes

Floor/ceiling

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

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

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

Standing liquidity transfer order

A standing liquidity transfer order is an instruction of a CLM MCA Holder to transfer regularly (e.g. daily, weekly) an amount of liquidity from a MCA to another account (a MCA in CLM 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 CLM MCA.

This information is defined at the level of the MCA and it is up to the CLM MCA Holder to configure 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 CLM is as follows: <ul style="list-style-type: none"> Inter-service liquidity transfer from MCA to DCA Intra-service liquidity transfer to another MCA
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.
MCA to be debited	It specifies the MCA to be debited in CLM.
Account to be credited	It specifies the account (DCA or another MCA) to be credited.
Trigger event	It specifies the event type that triggers the transfer of liquidity.
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 6 - Standing liquidity transfer order reference data attributes

Standing order for reservation

A standing order for reservation is an instruction of a CLM MCA Holder to set-up an urgent reservation of a fixed amount for a business day on a MCA without a predefined end date. An existing standing order for reservation can be modified or deleted. All actions (setup, modify, delete) become effective as of the next business day or on the activation date of the MCA if this is later than the next business day.

The reservation remains valid until it is modified or deleted. It is dedicated to CBOs and seizure of funds.

It is up to the CLM MCA Holders to set-up and maintain its standing order for reservation information in CRDM.

Current reservations

For the execution of CBOs CLM MCA Holders can set-up a current reservation on liquidity in CLM. An existing reservation can be modified and/or deleted. All activities (setup, modify, delete) become effective immediately.

In case the amount changes to “0” the reservation is removed automatically. The reactivation is however possible during the business day.

This information is defined at the level of the MCA and it is up to CLM MCA Holders to set-up and maintain the current reservations in CLM.

3.2.3 Messaging

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

Message subscription

A message subscription is defined as a service that allows an authorised CLM Actor 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 MCA and it is up to the CLM Actor to set-up and maintain the message subscription in CRDM.

Changes to message subscription in CRDM 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 the list of parameter types that authorised CLM Actors can use for configuring their message subscription.

Parameter type	Description
Message type	It specifies the type of message to which the authorised CLM Actor wants to subscribe. The exhaustive list of possible message types is listed above.
Cash account	It specifies the cash account in CLM to which the subscription applies.
Business case	It specifies the business case for which the message is subscribed.

Parameter type	Description
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 7 - 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	Underlying message type
Camt.004	Yes	No	No	No
Camt.019	No	No	No	No
Camt.029	Yes	No	No	No
Camt.054	Yes	Yes	No	No
Pacs.002	Yes	No	Yes	Yes

Table 8 - 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 CLM Actor 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](#) [▶ 209].

Report configuration

The CLM Actor can configure one standard report (statement of accounts) that CLM shall create at a specific business day event (end of day (EoD)). CLM 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 CLM Actor to configure another CLM 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 CLM Actor to set-up and maintain the report configuration in CRDM.

For further information about the report generation please refer to chapter [CLM report generation](#) [▶ 174] .

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 CLM Actor to set-up and maintain the report configuration in CRDM.

3.2.4 Blocking/unblocking account

It is possible to block MCAs in CLM. Blocking of 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 is blocked. If not stated, the next calendar date applies 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 MCA as blocked for credit and debit, credits and debits are not allowed on the MCA. If the MCA is blocked for debit, credits are still allowed on this cash account. The reverse logic applies in case of blocking for credit (debits are allowed).

The procedure in CLM is the following for the blocking of MCAs:

- | The MCA of the CLM 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 can settle on the MCA.
- | Cash transfers involved in a running settlement process are not affected by the blocking. If the algorithm:
 - is successful, the involved cash transfers of the blocked MCA holder becomes final.

- fails, the cash transfers of the blocked MCA returns to the queue. They require confirmation by the CB before they can settle in one of the next running algorithms.
- I Payments (credit transfers and direct debits) sent by the blocked CLM Account Holder are stored for confirmation by the CB. If the CB:
 - confirms, the payments runs through the entry disposition. If they cannot settle in the entry disposition, they are queued and included in the process of dissolution of the payment queue.
 - disagrees, the payments are rejected.

Note:

- I The confirmation of payments is physically done by the CB of the blocked CLM Account Holder via the GUI. Nevertheless organisational rules outside T2 can be implemented to involve other bodies (e.g. the official receiver), depending on the legal requirements of each country.
- I Independent from the blocking of a CLM Account Holder it is possible to close the account of a CLM 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 Liquidity Transfer Group, a Banking Group, a monetary financial institution (MFI) group and an Account Monitoring Group in CLM.

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

Reference data object	Responsible actor	Mode
Banking Group	CB	U2A
MFI	CB	U2A
Liquidity Transfer Group	CB	U2A
Account Monitoring Group	CLM Participant	U2A

Table 9 - Setup of groups for CLM

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) become effective as of the next business day.

MFI

The role of **MFI** allows a pool for management of minimum reserves. MFI is not defined as a party, but as a code that CBs can define. One party can be part of exactly one MFI, but one MFI can consist of more than one party.

The following table gives an overview on the MFI reference data attributes.

Attribute	Description
MFI Code	It specifies the unique identifier of the MFI.
Current maintenance period from	It specifies the date range of the current maintenance period in case of MFI (pooling of reserves).
Current maintenance period to	It specifies the date range of the current maintenance period in case of MFI (pooling of reserves).
Minimum Reserves (EUR)	It specifies the minimum reserve requirement of the MFI (pooling of reserves).

Table 10 - The MFI reference data attributes

Liquidity Transfer Group

A Liquidity Transfer Group is an optional grouping of MCAs. CBs can set up Liquidity Transfer Groups for the purpose of arranging intra-CLM liquidity transfers between them. A cash account can be included in one or several Liquidity Transfer Group(s).

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

Account Monitoring Group

An Account Monitoring Group is an optional group of accounts (MCA(s) and DCA(s)) for pure liquidity monitoring purposes; they are not used for the context of payments or liquidity transfers. An account can be included in one or several Account Monitoring Groups. An Account Monitoring Group can include accounts owned by several parties, which have been opened in the books of different CBs.

It is up to CLM Account Holders to set up and maintain their Account Monitoring Groups and define the accounts linked to each Account Monitoring Group. All actions (setup, modify, delete) become effective as of the next business day.

3.4 Shared reference data

CLM calendar

The CLM calendar specifies the calendar days when CLM 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 T2 Operator to set-up and maintain the CLM calendars. All actions (setup, modify, delete) become effective as of the next business day.

CLM scheduled events

The CLM scheduled events specifies the scheduled events that will automatically trigger a specified process within CLM. Each trigger event might trigger one or several other events. The other way round each event might have one or several trigger events. It is up to the T2 Operator to set-up and maintain the CLM scheduled events. All actions (setup, modify, delete) become effective as of the next business day.

The following table shows the attributes of the CLM scheduled events.

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 11 - Attributes of the CLM scheduled events

CLM currency

The CLM currency specifies the available settlement currencies in CLM. It is up to the T2 Operator to set-up and maintain the settlement currencies. All actions (setup, modify, delete) become effective as of the next business day.

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

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 12 - Attributes of the CLM currency

CLM rates

The CLM rates specify the available rates in CLM (e.g. for minimum reserve calculation). It is up to the T2 Operator to set-up and maintain the CLM rates. All actions (setup, modify, delete) become effective as of the next business day.

The following table shows the attributes of the CLM rates.

Attribute	Description
Country code	It specifies the rate's country code.
Rate type	It specifies the type of rate amongst a list of values: <ul style="list-style-type: none"> Reserve management interest Reserve management penalties Overnight deposit Marginal lending
Validity period	It specifies the start and end date of the rate validity period.
Rate (%)	It specifies the rate value.
Status	It specifies the status of the related rate
Modification date	It specifies the date from which the displayed record has been or will be active.

Table 13 - Attributes of the CLM rates

Reserve management maintenance period

This reference data object provides general information on the reserve management maintenance periods. It is up to the T2 Operator to set-up and maintain the reserve management maintenance period. All actions (setup, modify, delete) become effective as of the next business day.

The following table shows the attributes of the reserve management maintenance periods in CLM.

Attribute	Description
Maintenance period ID	It specifies the ID of the reserve management maintenance period.
Validity start date	It specifies the start date of the maintenance period.
Validity end date	It specifies the end date of the maintenance period.

Table 14 - Attributes of the reserve management maintenance periods

Duplicate check

There are duplicate checks on:

- | files and individual messages received (for A2A communication only); and
- | payment and liquidity 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.

No specific configuration by the T2 Actor is required. It is up to the T2 Operator to set-up and maintain the duplicate check parameter. All actions (setup, modify, delete) 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	T2 Operator	T2 Operator	M	1 day (same business day)
Business Validation/Duplicate Check	Number of business days in the past for duplicate check on liquidity/payment orders	T2 Operator	T2 Operator	M	1 day (same business day)

Table 15 - Attributes of the duplicate check

Warehoused payment period

It is possible to send warehoused payments a few calendar days in advance to CLM. The payment message shall pass technical and business validation and shall be warehoused until CLM opens for that date. The system parameter regarding the warehoused payment period is defined in the table below. It is up to the T2 Operator to set-up and maintain the warehouse payment period parameter. All actions (setup, modify, delete) 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 calendar days in the future for warehoused payments	T2 Operator	T2 Operator	M	10 calendar days

Table 16 - Attributes of the warehoused payment period

3.5 Interaction between CLM 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 timely and consistent. Further detailed information can be found in chapter [CRDM features](#) [▶ 180] .

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

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

- | modifying of limits
- | deleting of limits, see chapter [Maintain local reference data object - maintain reservation](#) [▶ 297]

The reference data stored in CRDM are propagated from the CRDM to CLM 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 CLM.

Every CRDM opening day, an ad-hoc event triggers the propagation of all CLM reference data from CRDM to CLM. The event takes place at the EoD phase of CRDM business day, to ensure a smooth and complete reference data propagation before CLM receives the notification that a new business day is starting. The propagated reference data is then loaded into CLM during the SoD phase.

4 Business day

4.1 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.

Example: If the 1st of January is a Tuesday and CLM, RTGS and T2S is closed, the business day 2nd of January already starts on calendar day the 31th of December at 18:45 Central European Time (CET) until the maintenance window start on Monday the 1st of January 0:30. The end of the maintenance window is on Tuesday the 2nd 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 end-of-day 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.

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 different infrastructure services and if necessary to initiate and coordinate overarching processes between different services and components e.g. RTGS, CLM, CRDM, 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 monitoring 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 waits for a feedback from the triggered process (e.g. processes on the critical path), for other events the scheduler 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 scheduler are 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 different from the scheduled time.

The business day management also defines 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 in CLM.

	CLM
Change of business day	18:45
SoD procedures	18:45-19:00
Start of settlement window for CBOs	19:00
Settlement window for liquidity transfer orders	19:30-18:00 (interrupted by maintenance window)
Maintenance window	00:30-02:30

	CLM
Cut-off for liquidity transfer orders and CBOs (except standing facilities)	18:00
EoD procedures	18:00 - 18:45
Cut-off for standing facilities	18:15 + 15 min on the last business day of the reserve maintenance period) 18:40 cut-off marginal lending on request for CB operators

Table 17 - Main windows during the business day

4.3 Detailed description of CLM business day phases

4.3.1 Start of day

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

As from begin of availability for users CBOs and liquidity transfer orders are possible.

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

Actions

Transfer of liquidity and interest from overnight deposit.

CLM calculates the interest to be paid on the overnight deposit. At the start of the following business day, the overnight deposits are reimbursed and the interest is calculated and settled.

Debiting of reimbursement and interest from standing facilities (marginal lending)

CLM calculates the interest to be paid by the credit institution on the marginal lending and, at the start of the following business day, reimburses the marginal lending and settles the interests.

Possibility to update credit lines at the MCAs (It is up to each CB to decide whether it update credit lines in the evening or during the next T2 working day.)

Start of execution of liquidity transfer orders from CLM to other services and components

Events

18:45 SoD procedures

19:00 Availability for users (settlement window for CBOs)

19:30 Start of settlement window liquidity transfers

4.3.2 Processing times/windows

This section presents the two settlement windows during the CLM availability times.

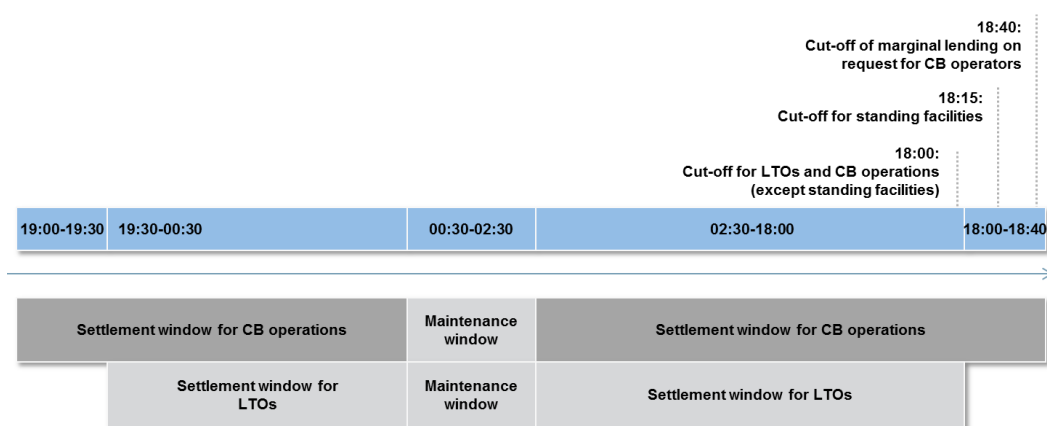


Figure 6 - Processing times windows

Settlement window for CBOs

The settlement window for CBOs starts after the successful completion of SoD processes (19:00) and ends with the cut off for standing facilities for CBs (18:40). It is interrupted by the maintenance window.

During this settlement window CLM processes all kind of CBOs e.g.:

- | cash withdrawals
- | monetary policy operations
- | overnight deposits
- | marginal lending
- | connected payments
- | collections of fees
- | modify credit line

The cut-off for all CBOs except standing facilities takes place at 18:00 with the start of EoD processing.

Settlement window for liquidity transfer orders

The settlement window for liquidity transfer orders starts at 19:30 and ends with the start of EoD processing. It is interrupted by the maintenance window.

During this settlement window CLM processes liquidity transfers to and from all other services and components.

Liquidity transfers sent during the maintenance window are technically stored and processed after the end of the maintenance window.

4.3.3 Maintenance window

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 services². 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 available³.

4.3.4 End of day

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

Actions

Closure for liquidity transfers (no new liquidity transfers are accepted and new incoming liquidity transfers are therefore be rejected).

Inform business day management about the closure of CLM

Rejection of pending payments, liquidity transfers and credit line changes not executed before the start of the EoD process of the current business day.

Stop processing of pending reservation order and release the remaining reserved amounts.

Rejection of pending verifications for creations, amendments or deletions in four-eyes-principle related to payments

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

3 For the sake of efficiency, the Eurosystem aligns 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.

Collection of EoD balances from each settlement service/components by receiving general ledger (camt.053) from other services/components.

EoD reporting for MCA Holders depending on the report configuration.

Inform business day management about the EoD processing of CLM.

Change of business day when EoD processing is finished

Events

18:00 cut off for liquidity transfers and CBOs (except standing facilities)

18:15 cut off for use of standing facilities ⁴

18:40 cut-off marginal lending on request for CB operators ⁵

18:45 change of business day ⁶

Until 18:15 (18:30 on the last day of the minimum reserve maintenance period) parties can use the standing facilities to either obtain liquidity or deposit liquidity with the CB.

The automatic marginal lending takes place during the EoD processing after the cut off for standing facilities for CB operators (18:40). The successful processing of an automatic marginal lending leads to the credit on the CLM Account Holder's MCA, the debit on the marginal lending account and the concurrent update of the CLM Account Holder's credit line.

During the EoD process, CLM receives a general ledger file from each service and component. After a successful consistency check on the received general ledger file CLM produces one general ledger file per CB and service and forwards them to the respective CBs. In case the consistency check fails TARGET Service desk receives an alert.

Once deadline for the usage of standing liquidity is over and CLM has gathered all cash balances from RTGS, TIPS and T2S, it calculates the final EoD balance of each party. In case the balance is negative, the system converts any outstanding amount into an automatic marginal lending ^{7 8}.

After all standing facilities are processed and the CB's ECB account balances are calculated and the general ledger files of the other services are received the CLM general ledger files for CBs are produced during the EoD process.

⁴ 18:30 on the last day of the minimum reserve maintenance period

⁵ 18:55 on the last day of the minimum reserve maintenance period

⁶ 19:00 on the last day of the minimum reserve maintenance period

⁷ It is possible for CBs to insert requests into the system until 40 minutes after the start of the EoD process (+15 minutes on last business day of minimum reserve maintenance period). The cut-offs are configurable with parameters. The mentioned points in time are only indicative values to define the order of the different cut offs and the timing regarding the phase of business day they have to take place.

⁸ T2S currently sends all DCA balances with an automated cash sweep back to the RTGS accounts. For the sake of alignment of the Eurosystem functionalities the proposal is to change the current mandatory cash sweep in an optional one. This shall be agreed among all T2S communities.

In case of an error, the EoD processing and the transmittance of the general ledger files is stopped in CLM.

At 18:45 the change of business day takes place. When all tasks EoD have been completed, the current business day is closed by sending a notification to the Business Day Management and the next business day is opened upon the reception of the trigger of 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.

Receive general ledger files von RTGS, TIPS, T2S⁹

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 in a general ledger file. A delay of any of the other services/components can lead to a delay of the CLM business day schedule.

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.

Automated liquidity transfers for pending CBO

In case there is not enough liquidity on a MCA to fully execute a CBO (e.g. overnight deposit, open market operation), CLM pulls 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.

⁹ T2S remains subject to decision about mandatory cash sweep.

Update of credit lines from collateral management

Active credit lines in CLM need to be updated on a daily basis by collateral management according to the latest available prices for the collateral placed.

4.5 Calculating the positions of CBs vis-à-vis other CBs

National central bank (NCB)/ECB accounts

In CLM each national CB has an “CB’s ECB account” and at least one account as CLM Account Holder. Also the ECB has at least one account as CLM Account Holder.

The CB’s ECB account is necessary to record the national CB’s asset/liability position vis-à-vis the ECB in respect of cross border transactions. Therefore, the national CB’s ECB account held by each CB in CLM is debited/credited at the end of the day as part of an automatic procedure. The postings on this account reflect the daily activities of each CB and its “local” account holders as a result of the cross border transactions the CB and its account holders processed during the day.

The ECB has a mirror account for each CB in CLM on which the postings done on the CBs’ ECB accounts are “mirrored”.

In order to enable CBs to settle their payments in CLM, each CB hold at least one MCA in its own name.

The following table lists some examples which transactions can be settled on this account:

- | monetary policy operations
- | transactions stemming from CB’s own business
- | balances from other services and components

During the EoD procedure the sum of all bilateral debits as well as bilateral credits vis-à-vis each other CB is calculated and booked on the CBs’ ECB account. This is done for all bilateral relationships between CBs. On an optional basis a CB can receive individual confirmations of credit or debit for each posting on this specific account.

During the EoD procedure each CB receives all information necessary to update the accounting system in order to be able to generate a daily balance sheet. Therefore, the general ledger file sent contain the following information

NCB’s ECB account

- | start balance
- | the sums of debits and the sums of credits towards each CB from CLM and other components and services(except TIPS) and

delta of the sum of closing balances of all TIPS DCAs in the books of respective CB on day D compared to day D-1 towards Eurosystem/ECB (shown in a "second EU line". Only available, if there has been cross border traffic within TIPS)

the EoD balance of each CB on its CB's ECB account

CB account(s) as MCA Holder

start balance

the sums of debits and the sums of credits and

the EoD balance on its CBs'account(s) as MCA Holder.

MCA Holder

start balance

the sum of debits and sum of credits for each MCA and

the EoD balance of each MCA

Note: In addition, the general ledger file sent to the ECB contains information on the ECB's mirror accounts.

Examples:

The following table assumes that only five credit institutions (assigned to three different CBs) hold a MCA or DCA in CLM or another components or service:

Bank A and Bank B are assigned to CB 1

Bank C and Bank D are assigned to CB 2

Bank E is assigned to CB 3

The figures in the columns show the settlement items resulting from credit transfers (debtor: negative amount (-), creditor: positive amount (+)).

bank a	bank b	bank c	bank d	bank e
CB 1	CB 1	CB 2	CB 2	CB 3
-50			+50	
	+30	-30		
+20	-20			
		-40	+40	
		+50		-50
			-30	+30

bank a	bank b	bank c	bank d	bank e
-10				+10
	+80		-80	
-40			+40	

Table 18 - Settlement items resulting from credit transfers

At the end of the day, CLM calculates the sum of all bilateral debits as well as the sum of all bilateral credits vis-à-vis each other CB. These bilateral sums of CLM and all other services and components are settled on the CBs' ECB accounts. These postings are "mirrored" on the ECB mirror accounts.

The result of these postings is the multilateral position of each CB vis-à-vis the ECB as the result of the daily activities ¹⁰.

country 1	country 2	country 3
⇒ Country 2 C 110 D 90	⇒ Country 1 C 90 D 110	⇒ Country 1 C 10 D 0
⇒ Country 3 C 0 D 10	⇒ Country 3 C 50 D 30	⇒ Country 2 C 30 D 50
multilateral position vis-à-vis the ECB: C 10	multilateral position vis-à-vis the ECB: C/D 0	multilateral position vis-à-vis the ECB: D 10

Table 19 - Postings

The postings take place as follows:

- | CB 1's ECB account is credited with 90 and CB 2's ECB account is debited with 90.
- | CB 1's ECB account is debited with 110 and CB 2's ECB account is credited with 110.
- | CB 1's ECB account is credited with 10 and CB 3's ECB account is debited with 10.
- | CB 2's ECB account is credited with 30 and CB 3's ECB account is debited with 30.
- | CB 2's ECB account is debited with 50 and CB 3's ECB account is credited with 50.

The postings on the ECB mirror accounts take place as follows:

- | The ECB mirror account for CB 1 is debited with 90 and the ECB mirror account for CB 2 is credited with 90.

¹⁰ Whether the overall (accumulated) position of each CB vis-à-vis the ECB is reflected in the CB's ECB account on the following day depends on the decision of each CB to either maintain the balances at the EoD in CLM or to transfer it back.

The ECB mirror account for CB 1 is credited with 110 and the ECB mirror account for CB 2 is debited with 110.

The ECB mirror account for CB 1 is debited with 10 and the ECB mirror account for CB 3 is credited with 10.

The ECB mirror account for CB 2 is debited with 30 and the ECB mirror account for CB 3 is credited with 30.

The ECB mirror account for CB 2 is credited with 50 and the ECB mirror account for CB 3 is debited with 50.

The above mentioned positions mean:

CB of country 1 has a claim of 10 towards the ECB.

CB of country 3 has a liability of 10 towards the ECB.

The sum of all bilateral debits and credits vis-à-vis each other CB posted to the respective CBs' ECB accounts. The multilateral position of each CB vis-à-vis the ECB is the result of all these postings. Each CB has, at the EoD, only one position.

The CBs are informed about these transactions on their accounts via a corresponding credit/debit notification.

The above mentioned postings are also part of the general ledger file.

Example for calculating TIPS DCAs positions of a CB vis-à-vis Eurosystem/ECB:

The following table assumes that only five credit institutions (assigned to three different CBs) hold a DCA. The figures in the columns show the settlement items resulting from credit transfers (debitor: -, creditor: +).

Transactions on DCA side						
	Bank A	Bank B	Bank C	Bank D	Bank E	
Previous business day (d-1)						
Responsible CB	CB1	CB1	CB2	CB2	CB3	
Closing balance	55	200	20	60	35	
Sum of all closing balances per CB	255		80		35	
Current business day (D)	+20			-20		
		-80	+80			
	-10					+10
		+25	-25			
				-15		+15

Transactions on DCA side						
Closing balance	65	145	75	25	60	
Sum of all closing balances per CB	210		100		60	

Table 20 - Transactions on DCA side

At the end of the day, the TIPS sends its general ledger file to CLM.

CLM calculates the delta of the sum of the closing balances of all the TIPS DCAs in the books of the respective CB on day D compared to day D-1 for each CB towards the Eurosystem/ECB. This delta is booked on the CBs' ECB accounts. This posting is "mirrored" on the ECB mirror accounts.

The delta is calculated as follows:

- I Sum of closing balances of all the TIPS accounts in the books of CB 1 on current business day compared to previous business day decreased around 45.
- I Sum of closing balances of all the TIPS accounts in the books of CB 2 on current business day compared to previous business day decreased around 20
- I Sum of closing balances of all the TIPS accounts in the books of CB 3 on current business day compared to previous business day increased around 25

The postings on CB's ECB accounts takes place as follows:

- I CB 1's ECB account is credited with 45 and ECB's ECB account is debited with 45.
- I CB 2's ECB account is credited with 20 and ECB's ECB account is debited with 20.
- I CB 3's ECB account is debited with 25 and ECB's ECB account is credited with 25.

The postings on ECB mirror accounts takes place as follows:

- I The ECB mirror account for CB 1 is debited with 45 and the ECB mirror account for ECB is credited with 45.
- I The ECB mirror account for CB 2 is debited with 20 and the ECB mirror account for ECB is credited with 20.
- I The ECB mirror account for CB 3 is credited with 25 and the ECB mirror account for ECB is debited with 25.

5 Business and features description

5.1 Settlement of payments linked to CBOs

5.1.1 Overview

In CLM the following CBOs ¹¹ are processed and settled on the MCAs of the CLM Account Holder:

- | update of credit line (cash side)
- | marginal lending and overnight deposits (summarised as standing facilities)
- | cash withdrawals
- | tender operations (e.g. open market operations like the main refinancing operation or the longer-term refinancing operations)
- | debit of the invoiced amount
- | interest payment orders linked to marginal lending, overnight deposits, minimum reserves and excess of reserves
- | any other activity carried out by CBs in their capacity as CB of issue

All CBOs are settled with a priority before the liquidity transfers and are either fully executed or queued, i.e. payments linked to CBOs are never settled partially.

CBOs can be initiated by the CB in A2A or in U2A mode. The following payment types can be submitted:

- | credit transfers or
- | direct debits (e.g. used for the execution of cash withdrawals, repayment of monetary policy operations and collection of fees)
- | connected payments
- | warehoused payments

A payment order linked to CBOs lead to a debit (or credit) of the MCA with the simultaneous credit (debit) of the CB account/marginal lending account/overnight deposit account.

With the exception of overnight deposits, which are initiated by a liquidity transfer (camt.050), a CB can send the above mentioned CBOs (depending on the underlying business case) as:

- | a credit transfer (pacs.009) or
- | a direct debit (pacs.010) to CLM (for further details please refer to chapter [Flow of payments](#) [► 84]).

¹¹ Generally within this CLM UDFS the term CBO covers one of the here mentioned operations initiated by CBs.

Note: In case the CB sends a direct debit in general no direct debit mandate is needed. Only in case the CB wants to have a direct debit booked on a MCA of a CLM Account Holder not belonging to “its” own banking community a direct debit mandate is needed.

Beside that the CB can send credit transfers and/or direct debits as **connected payments**. They are called “connected payments”, due to the link between payment (an immediate debit/credit of its MCA) and a corresponding change of credit line. Connected payments are also used for tender operations. For further details please refer to chapter [Connected payment](#) [▶ 242].

Within the payment, CBs have the possibility to define the execution time ([Definition of execution time](#) [▶ 82])
It is possible to set

- | an “earliest debit time indicator” (FromTime) and
- | a “latest debit time indicator” (RejectTime)

Furthermore, payments can be submitted as “**warehoused payments**” which means that the CBO is sent up to ten calendar days in advance. In this case, the payment is warehoused until CLM opens for the settlement on the intended settlement day.

5.1.2 Definition of execution time

The above mentioned CBOs can be processed throughout the whole business day with the exception of the EoD processing and the maintenance window. Connected payments are processed up until the CB general cut-off for the use of standing facilities (i.e. 18:40).

In addition, T2 Actors in CLM have the possibility to determine the settlement time of their payments. The following options are available:

- | an “earliest debit time indicator”
- | 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)	Payments which should be executed up to a certain time and is rejected, if that is not the case (message element: RejectTime)
Effect	Payment is stored until the indicated time (with status earmarked). At the earliest debit time, the payment runs through the entry disposition.	RejectTime the payment is rejected, if it could not be executed until the latest debit time.
Processing	If the payment cannot be settled at the earliest debit time, it is queued until the cut-off time for payment type. The payment can be revoked.	If the payment with the RejectTime indicator cannot be settled until the indicated debit time, the payment is rejected.

Table 21 - Payments with 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 via U2A only is triggered. The notification is directly displayed on top of all screens of the CLM Account Holder whose MCA is debited.

It is possible to combine the “earliest debit time indicator” with the “latest debit time indicator”. 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. For the effect of changing settlement time see chapter [Amendment of payments](#) [▶ 94] as well as chapter [Comprehensive queue management](#) [▶ 111].

Note: It is not possible to change the “earliest debit time indicator” of a payment which is queued due to the fact that the original “earliest debit time indicator” has been reached and it was already tried to settle this payment.

5.1.3 Revalidate warehoused payments at SoD

Basics

Warehoused payments are stored in RTGS with the certain payment order 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.

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](#) [▶ 82]
- | further checks as described in chapter [Entry disposition](#) [▶ 106]

Technical validations like schema validations are only carried out on message level on the submission day. The same is valid for the duplicate payment order check. They are not repeated at SoD.

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. Exception: Warehoused payments with a set execution from time indicator which has not been reached are set to status “earmarked”.

5.1.4 Flow of payments

5.1.4.1 Payments initiated by CB - credit transfer

Only CBs can send a credit transfer linked to a CBO to a CLM MCA Holder.

The credit transfer is used inter-alia in case of:

- | credits in the context of tender operations or
- | payment of interests

Positive case of CBO credit transfer initiated by the CB

In case the technical and business validation is passed successfully, the CBO is settled.

Message flow

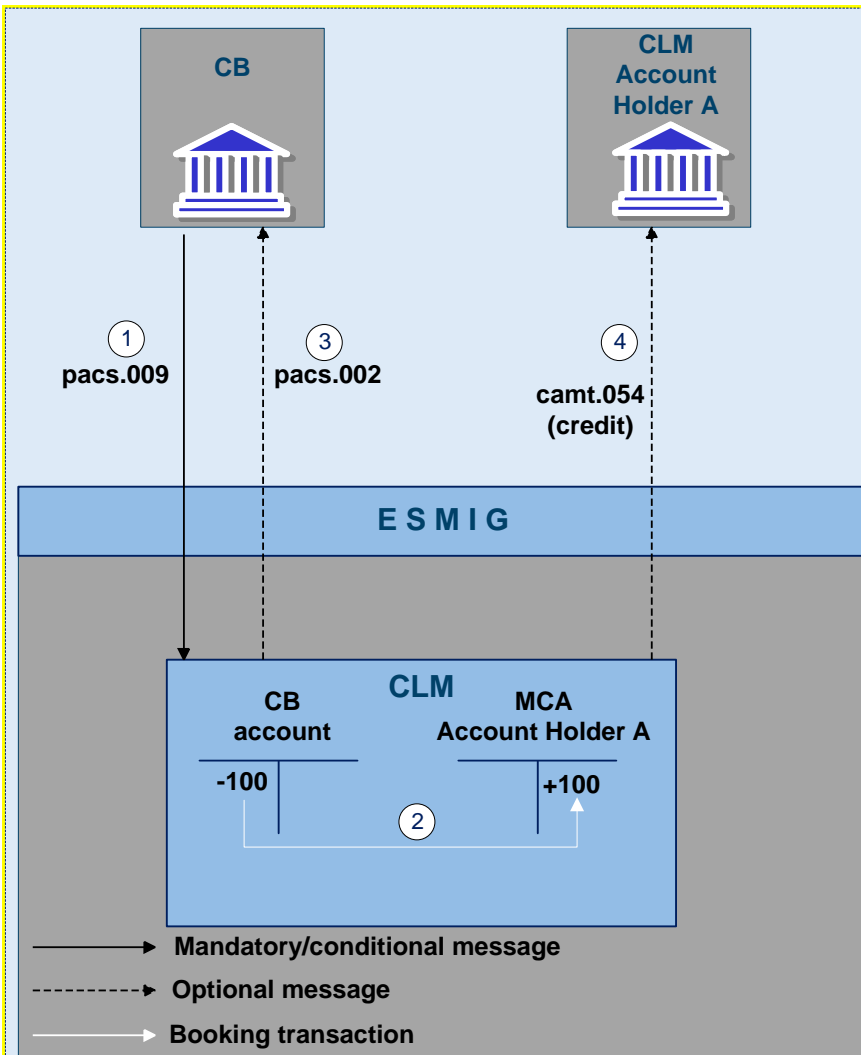


Figure 7 - pacs.009 CBOs

Process description

Step	Processing in/between	Description
1	CB via ESMIG to CLM	The CB sends a pacs.009 via ESMIG to CLM.
2	CLM	Booking takes place in CLM if CLM message check and validations are positive.
3	CLM via ESMIG to CB	CLM creates and forwards pacs.002 via ESMIG to CB (optional).
4	CLM via ESMIG to CLM Account Holder	CLM creates and forwards a camt.054 (credit) via ESMIG to CLM MCA Holder A (optional).

Table 22 - CB credit transfer (technical and business validations passed)**Used messages**

| [FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491]

| [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]

| [PaymentStatusReport \(pacs.002\)](#) [▶ 487]

Technical validation failure

CLM performs the technical validations. For further details please refer to [Rejection of payments](#) [▶ 93].

CLM continues the technical validation even if a first error has been detected.

If the technical validation fails CLM rejects the CBO and provides all negative results in form of error codes in a single message (please refer to message flow described below).

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.

Message flow

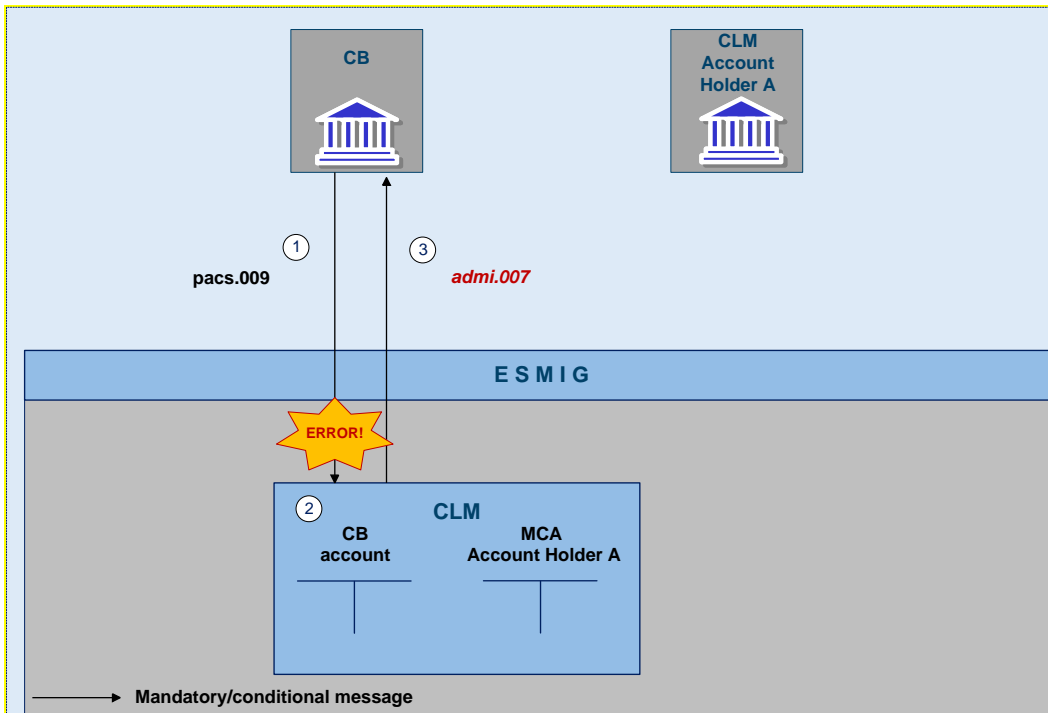


Figure 8 - pacs.009 CBOs technical validation failed

Process description

Step	Processing in/between	Description
1	CB via ESMIG to CLM	The CB sends a pacs.009 via ESMIG to CLM.
2	CLM	CLM technical validation failed.
3	CLM via ESMIG to CB	CLM creates and forwards an admi.007 via ESMIG to the CB.

Table 23 - CB credit transfer (technical validation failure)

Used messages

- FinancialInstitutionCreditTransfer (COR) (pacs.009) [491]
- ReceiptAcknowledgement (admi.007) [347]

Business validation failure

CLM performs the business validations. For further details please refer to [Rejection of payments](#) [93].

If the business validation fails CLM rejects the CBO and provides the rejection notification to the CB which submitted the CBO (please refer to message flow described below).

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.

Message flow

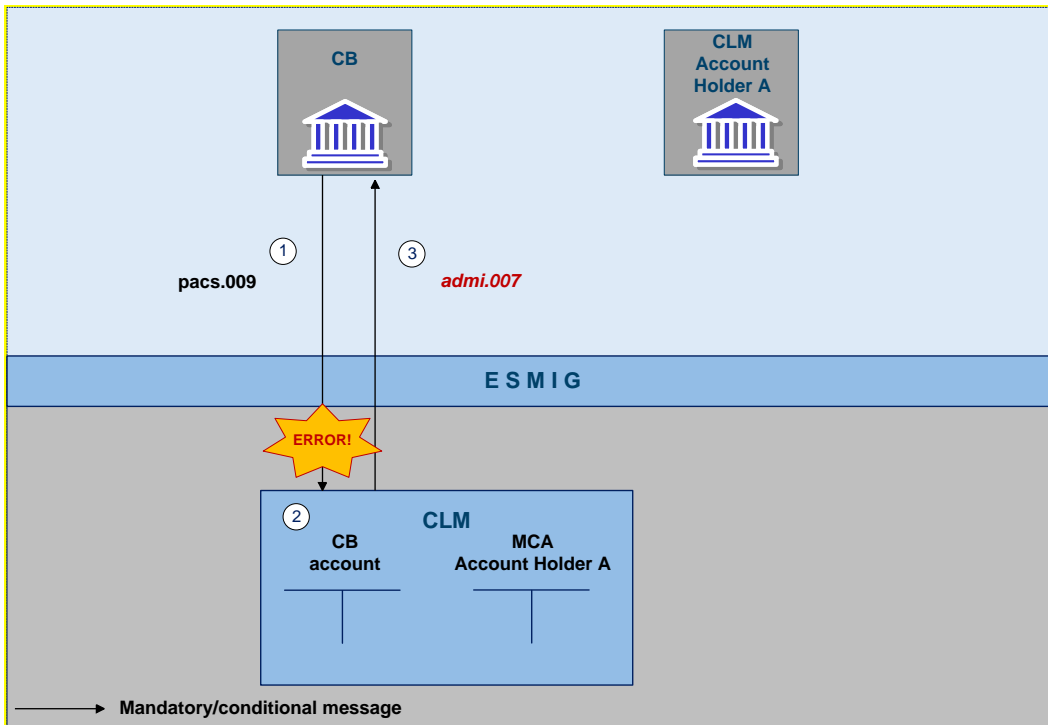


Figure 9 - pacs.009 CBOs business validation failed

Process description

Step	Processing in/between	Description
1	CB via ESMIG to CLM	The CB sends a pacs.009 via ESMIG to CLM.
2	CLM	CLM business validation failed.
3	CLM via ESMIG to CB	CLM creates and forwards a negative pacs.002 via ESMIG to CB.

Table 24 - CB credit transfer (business validation failure)

Used messages

- FinancialInstitutionCreditTransfer (COR) (pacs.009) [491]
- PaymentStatusReport (pacs.002) [487]

5.1.4.2 Payments initiated by CB - direct debit

Only a CB can send a direct debit linked to a CBO to a CLM MCA Holder. CBs are allowed to send direct debits within its market by default. No direct debit mandate is required in CRDM for CBOs.

The direct debit is used inter alia in case of:

- | debits in the context of open market operations
- | debit of invoiced amounts
- | cash withdrawals
- | debit of interest

Positive case of CBO direct debit initiated by the CB

In case the technical and business validation is passed successfully, the CBO is settled.

Message flow

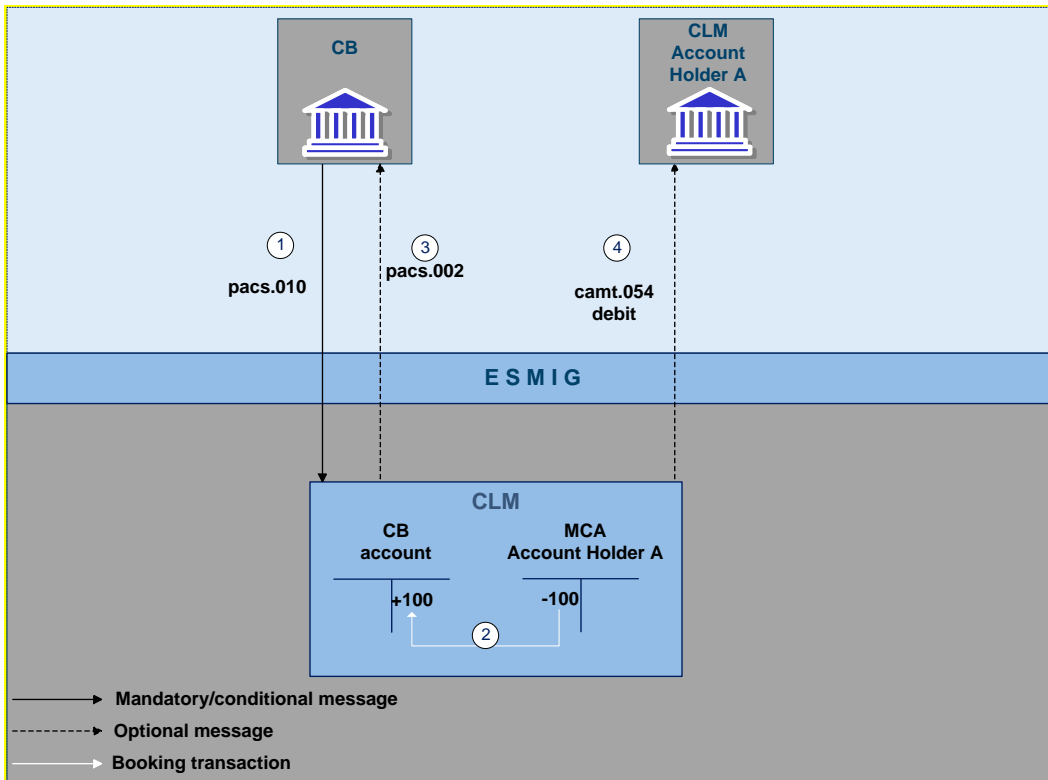


Figure 10 - pacs.010 CBOs

Process description

Step	Processing in/between	Description
1	CB via ESMIG to CLM	The CB sends a pacs.010 via ESMIG to CLM.
2	CLM	Booking takes place in CLM if CLM message check and validations are positive.
3	CLM via ESMIG to CB	CLM creates and forwards a pacs.002 via ESMIG to CB (optional).
4	CLM via ESMIG to CLM Account Holder	CLM creates and forwards a camt.054 (debit) via ESMIG to CLM MCA Holder A (optional).

Table 25 - CB direct debit (technical and business validations passed)

Used messages

- FinancialInstitutionDirectDebit (pacs.010) [496]
- BankToCustomerDebitCreditNotification (camt.054) [442]
- PaymentStatusReport (pacs.002) [487]

CLM performs the technical validations. For further details please refer to [Rejection of payments](#) [93].

CLM continues the technical validation even if a first error has been detected.

If the technical validation fails CLM rejects the CBO and provides all negative results in form of error codes in a single message.

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.

Message flow

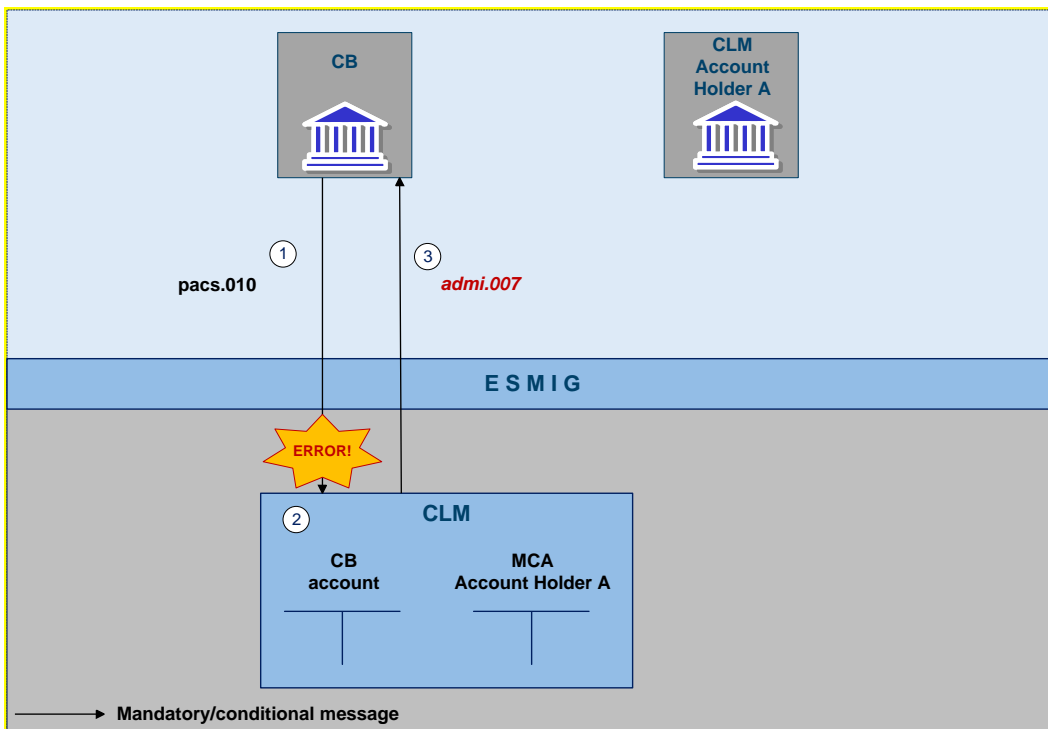


Figure 11 - pacs.010 CBOs - technical validation failed

Process description

Step	Processing in/between	Description
1	CB via ESMIG to CLM	The CB sends a pacs.010 via ESMIG to CLM.
2	CLM	CLM technical validation failed.
3	CLM via ESMIG to CB	CLM creates and forwards an admi.007 via ESMIG to CB.

Table 26 - CB direct debit (technical validation failure)

Used messages

- [FinancialInstitutionDirectDebit \(pacs.010\) \[496\]](#)
- [ReceiptAcknowledgement \(admi.007\) \[347\]](#)

Business validation failure

CLM performs the business validations. For further details please refer to [Rejection of payments](#) [93].

If the business validation fails, CLM rejects the CBO and provides the rejection notification to the CB which submitted the CBO.

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.

Message flow

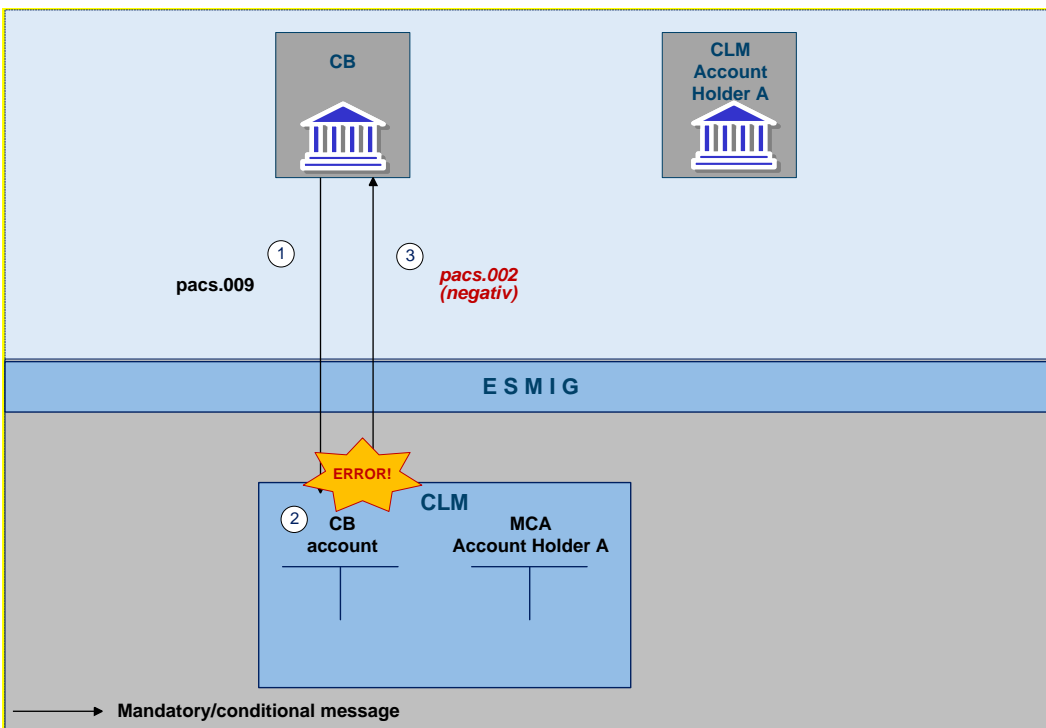


Figure 12 - pacs.010 CBOs - validation failed

Process description

Step	Processing in/between	Description
1	CB via ESMIG to CLM	The CB sends a pacs.010 via ESMIG to CLM.
2	CLM	CLM business validation failed.
3	CLM via ESMIG to CB	CLM Creates and forwards a negative pacs.002 via ESMIG to CB.

Table 27 - CB direct debit (business validation failure)

Used messages

- FinancialInstitutionDirectDebit (pacs.010) [496]

[PaymentStatusReport \(pacs.002\) \[487\]](#)

5.1.5 Rejection of payments

Payment orders sent to CLM have to pass several validations before the payment is effectively settled. Validations include technical checks and format checks (both technical validations) as well as checks for the correct content (business validations). A payment is rejected by CLM if either the technical validation or the business validation fails.

Technical validation

The following technical validation is performed in CLM interface:

- I schema validation - syntax, format and structure of the message are compliant (e.g. check that all mandatory fields 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 admi.007 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.

Business validation

The validations described below are performed in one step in order to capture all the possible breaches; the checks therefore do not stop after the first breach occurring, as if there could be further breaches in the subsequent checks. If the business validation fails, a rejection notification with appropriate reason codes for all breaches occurred is sent to the instructing party.

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

- I check for duplicate submission for incoming CBOs including the fields:
 - sender of the message
 - message type
 - receiver
 - transaction reference number
 - related reference
 - value date
 - amount
- I process specific access rights/authorisation checks:

- Is the sender of the payment order the owner of the account to be debited or another actor operating on its behalf?
- In case of direct debit: is the sender of the payment order the owner of the account to be credited?
- Is the CB allowed to send CBOs for the provided MCAs?
- In case a CB acts on behalf of a credit institution: does the credit institution belong to the acting CB?
- | check on value date
 - If the value date is in the future (up to ten calendar days), it is treated as warehoused payment.
 - If the value date is the current business day, it is treated as like any other payment.
- | payment type specific checks
- | field and reference data checks:
 - field value validation - codes are valid, domain values are within allowed range.
(e.g. the MCA and the CB account mentioned in the CBO exist and are active for settlement in the relevant currency or the MCA owner is not blocked at account or party level.)
 - Cross-field validation - all provided values are valid according to predefined values or cross-field validations.
 - database checks, e.g. existence of parties and accounts
- | direct debit check
- | check of back-up payments
- | account checks

Error codes for possible rejections are listed in chapter [Index of business rules and error codes](#) [▶ 537].

If business validation fails CLM creates and forwards a pacs.002 (negative – payment status report) to the instructing party (meaning the CB). The pacs.002 refers to the original transaction reference number and a set of elements from the original instruction. The pacs.002 message is a conditional message, i.e. it is mandatory in case of failed business validation.

5.1.6 Amendment of payments

As long as a CBO initiated in CLM is not settled (including warehoused payments), the CB has the ability to change certain parameters of this payment.

The amendment of CBOs is possible throughout the whole business day with the exception of the EoD processing and the maintenance window. CBs can initiate an amendment in U2A mode only.

If the message content is valid (see chapter [Rejection of payments](#) [▶ 93]) CLM checks the status of the original CBO the amendment is referring to. The CBO to be amended has to be in an intermediate (i.e. not final) status to be eligible for amendment.

If the amendment operation succeeds, CLM modifies the original CBO according to the amendment request and send a success notification to the submitting CB.

If the amendment operation fails, a reject notification with appropriate reason code is sent to the CB.

Two following different types of amendment are possible in CLM.

Parameter/action	Actor
Re-ordering within the respective queue (increase/decrease position)	CB
Change of set execution time (if defined before sending to CLM)	CB

Table 28 - Possible amendment types in CLM

These features enable a CB to react on changed conditions during the day.

In principle, amendments can be provided to CLM in U2A. A description of the respective screen can be found in the CLM user handbook.

As a consequence of the amendment of CBOs the dissolution of the payment queue process might be started. For further details please refer to chapter [Dissolution of the payment queue](#) [▶ 113].

Case: re-ordering the queued transactions

A CB can change the queue position of CBOs. The selected CBO can be placed:

- | to the top of the queue
- | to the end of the queue

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](#) [▶ 111].

Case: changing the execution time

CBOs can include a time that indicates when they should be settled, i.e. when the first settlement attempt is started (transactions with an “earliest debit time indicator”) and/or a time that indicates until when they should have been settled, i.e. after which no further settlement attempt takes place (transactions with a “latest debit time indicator”).

The execution time can be changed in CLM via U2A (i.e. the time may be advanced or postponed). The change has no impact on the payment processing, but on the queue management.

The change of the execution time can be done at any time during the business day. A detailed description of the process and the effect of the changed execution time can be found in chapter [Comprehensive queue management](#) [▶ 111].

Successful amendment

Message flow

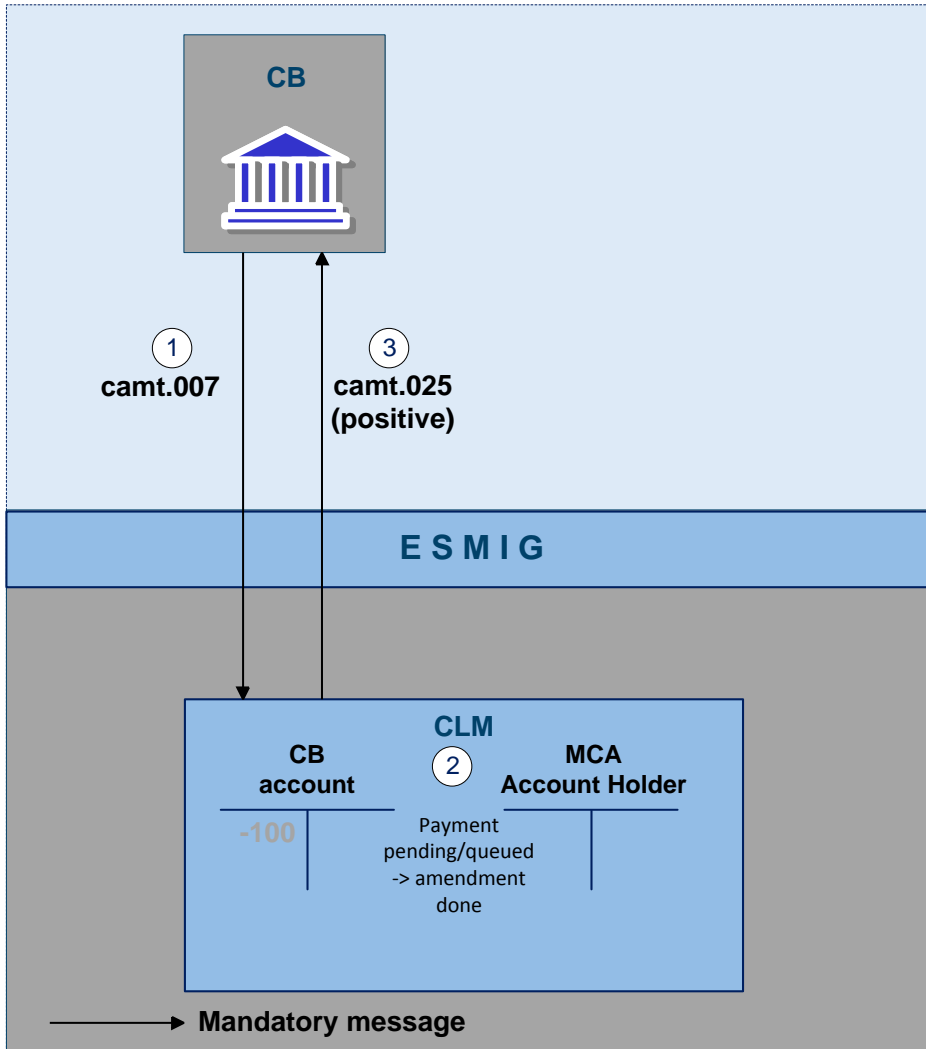


Figure 13 - Amend payment succeeded

Process description

Step	Processing in/between	Description
1	CB to CLM via ESMIG	The CB sends a camt.007 via ESMIG to CLM.
2	CLM	CLM business validation passed. CLM processes the requested amendment.
3	CLM to CB via ESMIG	CLM creates and forwards a positive camt.025 via ESMIG to the CB.

Table 29 - Successful amendment of payment

Used messages

- | [ModifyTransaction \(camt.007\)](#) [▶ 384]
- | [Receipt \(camt.025\)](#) [▶ 397]

Failed amendment

Message flow

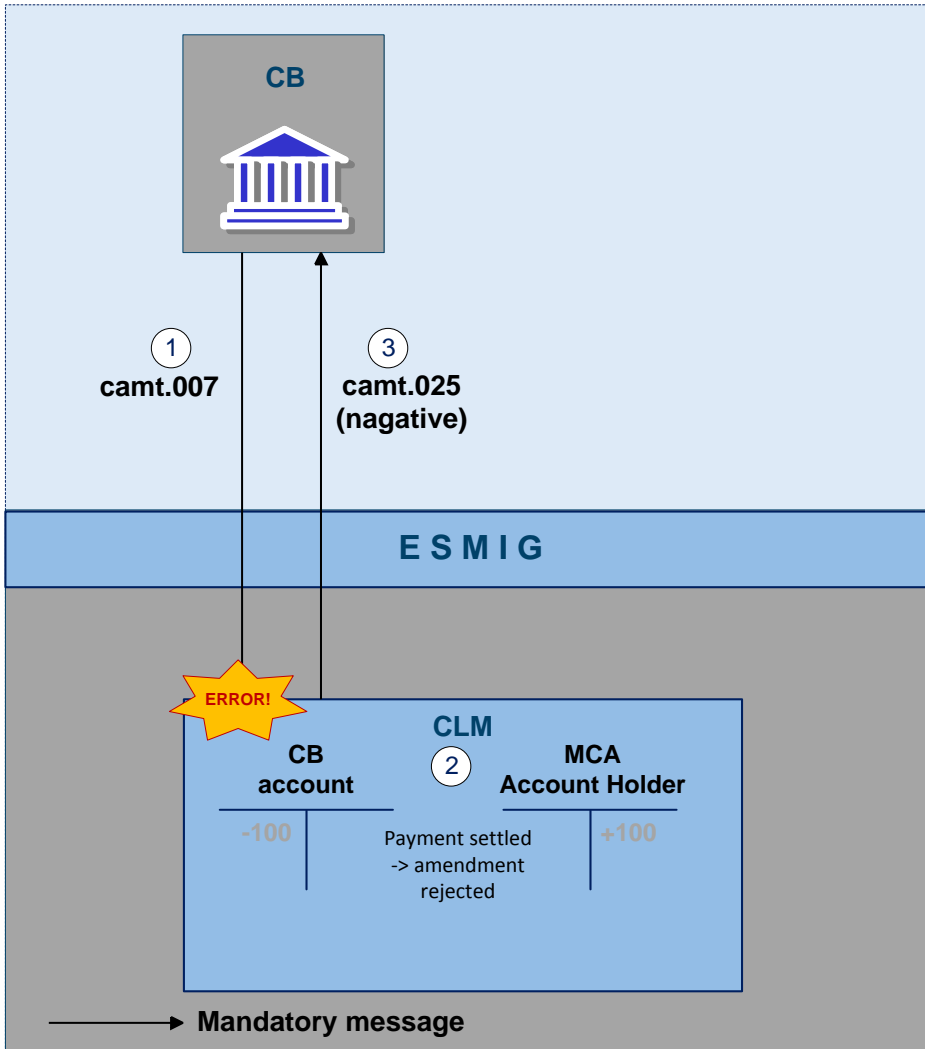


Figure 14 - Amend payment failed

Process description

Step	Processing in/between	Description
1	CB to CLM via ESMIG	The CB sends a camt.007 via ESMIG to CLM.
2	CLM	CLM business validation failed.
3	CLM to CB via ESMIG	CLM creates and forwards a negative camt.025 via ESMIG to the CB.

Table 30 - Failed amendment of payment**Used messages**

- | [ModifyTransaction \(camt.007\)](#) [▶ 384]
- | [Receipt \(camt.025\)](#) [▶ 397]

5.1.7 Revocation of payments

As long as a CBO is not settled (including warehoused payments), a CB has the ability to revoke this payment.

To revoke a payment the following pre-conditions apply.

- | a CBO has been initiated in CLM and
- | the status of the payment is not final, i.e. the payment is in the CLM queue or it is warehoused.

The revocation of CBOs is possible throughout the whole business day with the exception of the EoD processing and the maintenance window. CBs can initiate a revocation in A2A as well as in U2A mode. A description of the individual U2A process can be found in the CLM user handbook.

A cancellation request can be sent to revoke CBOs which were sent via pacs.009 or pacs.010. For each CBO submitted the CB needs to send a dedicated cancellation request ([FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 452]).

If the message content is valid ([Rejection of payments](#) [▶ 93]) CLM checks the status of the original CBO the revocation is referring to. The CBO to be revoked has to be in an intermediate (i.e. not final) status to be eligible for revocation. If the revocation operation succeeds, CLM cancels the original CBO and sends a revoke success notification to the CB as initiator. Where the revocation operation fails, a revocation reject notification with appropriate reason code is sent to the CB ([Index of business rules and error codes](#) [▶ 537]).

Successful revocation

Message flow

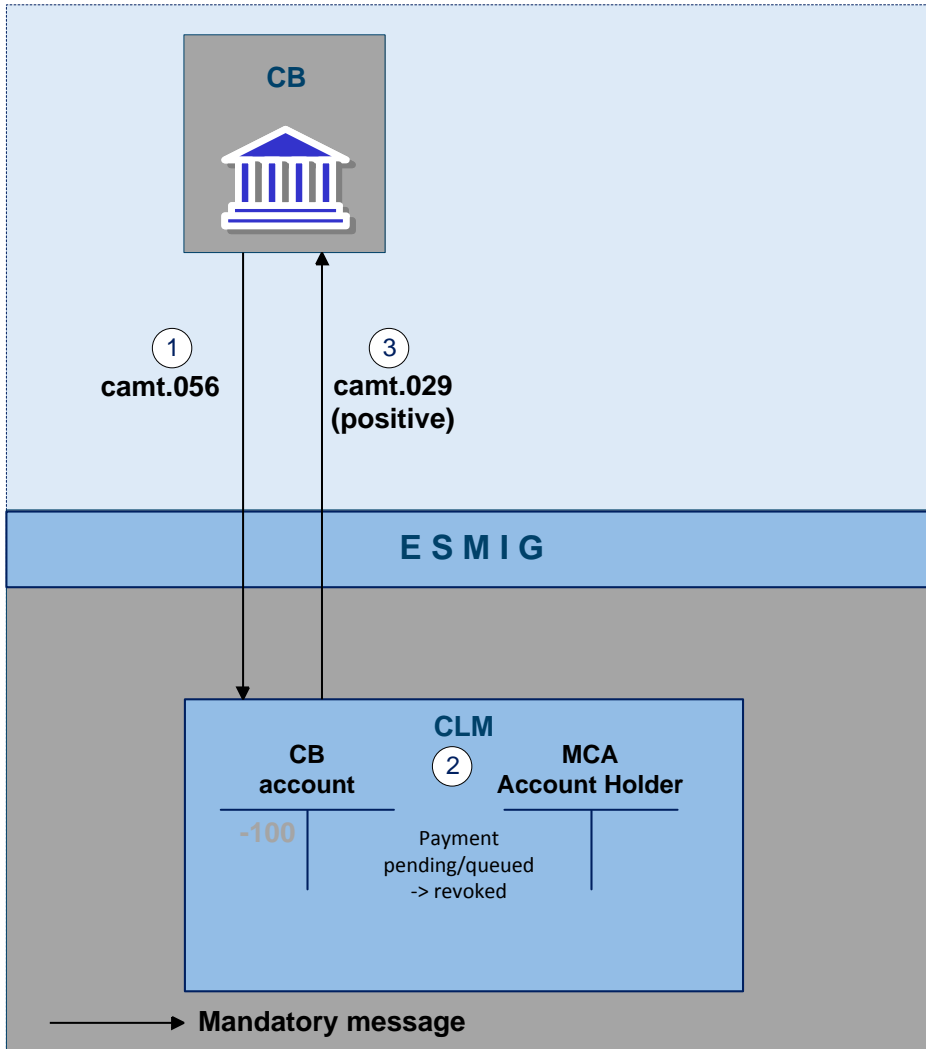


Figure 15 - Revoke payment via camt.056 – positive

Process description

Step	Processing in/between	Description
1	CB to CLM via ESMIG	CB sends a camt.056 (via ESMIG) to request revocation of payment to CLM.
2	CLM	CLM checks status of requested payment (not final); the payment is revoked and deleted from payment queue.
3	CLM to CB via ESMIG	CLM sends a positive camt.029 to confirm the revocation.

Table 31 - Successful revocation of pending payment

Used messages

- | [ResolutionOfInvestigation \(camt.029\)](#) [▶ 406]
- | [FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 452]

Failed revocation

Message flow

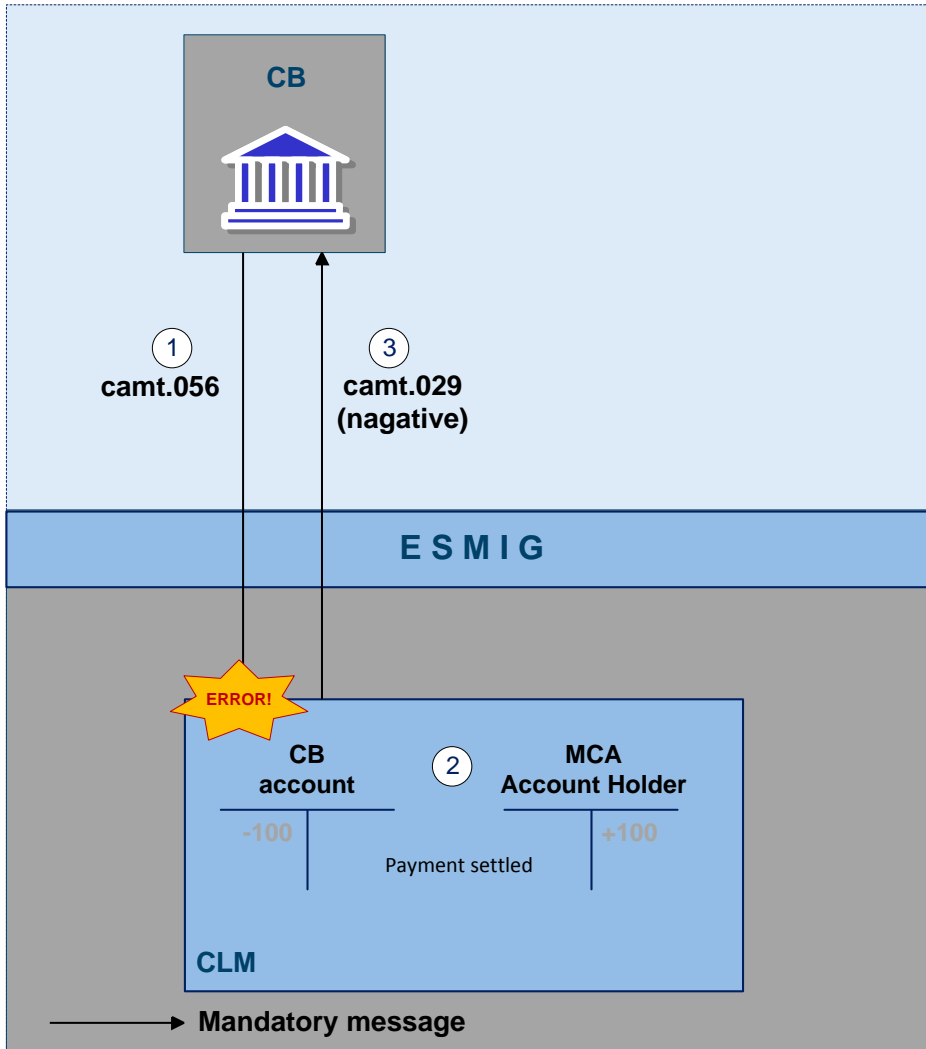


Figure 16 - Revoke payment via camt.056 - negative

Process description

Step	Processing in/between	Description
1	CBs to CLM via ESMIG	CB sends a camt.056 (via ESMIG) to request revocation of payment to CLM.
2	CLM	CLM checks the status of requested payment (settled); revocation request not executed.
3	CLM to CBs via ESMIG	CLM sends a negative camt.029 to notify a failed revocation.

Table 32 - Failed revocation of payment**Used messages**

- | [ResolutionOfInvestigation \(camt.029\)](#) [▶ 406]
- | [FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 452]

Technical validation

CLM performs the technical validations. For further details please refer to chapter [Rejection of payments](#) [▶ 93].

If the validation fails, a rejection notification with appropriate reason code is sent to the initiator of the revocation request (depending on the submission channel, an admi.007 in A2A mode or a rejection notification is displayed directly on the screen in U2A mode).

Business validation

CLM performs the business validations. For further details please refer to chapter [Rejection of payments](#) [▶ 93].

5.1.8 Processing of payment orders

Basics

The efficient management of liquidity and the settlement of CBOs in an optimised manner are of key importance. Therefore, offering a broad set of liquidity management features helps fulfilling the objectives of the CLM component.

These features may, i.a.:

- | result in faster settlement, with a reduced amount of liquidity
- | increase transparency for CLM Account Holders
- | contribute achieving a higher degree of efficiency
- | allow achieving a flexible and need-based control of payment flows

Objective for settlement

The aim of the processing in CLM is a fast settlement of CBOs with the following characteristics:

- | settlement in CB money
- | immediate, irrevocable booking of settled CBOs

Moreover, it is the aim of the CLM processing to enable an efficient allocation of liquidity among the various services or components and its fast, immediate irrevocable settlement.

Influencing factors

The payment processing in CLM is influenced by the following factors:

- | balance on the MCA
- | credit line connected to the MCA
- | used reservations for CBOs
- | seizure of funds
- | order of payment orders (including CBOs and liquidity transfers) submitted
- | set execution time

Effective settlement order

All cash transfer and credit line orders processed by CLM are characterised by urgent priority by default. Nevertheless in CLM it is acknowledged that the cash transfer and credit line orders serve different business what requires a further categorisation within the settlement priority, e.g. credit line decrease vs. liquidity transfer. The following table illustrates the effective settlement order for debits on MCAs depending on the triggering business in CLM.

effective settlement order	business case
1	Credit line decrease
2	CBOs (including cash withdrawal)
3	Liquidity transfer

Table 33 - Effective settlement order

Basic principles

The following principles apply to the processing of CBOs in CLM.

- | All CBOs have the same priority. They are not distinguished between urgent and normal payments.
- | Attempt to settle immediately after their submission - with the exception of CBOs with a set earliest debit time indicator "FROTIME".
- | In case a "FROTIME" is defined, these CBOs are included in the settlement process only from that time indicated as earliest debit time.
- | The CBO can include the latest debit indicator "REJTIME" and "TILLTIME".
- | In case a "REJTIME" is defined, the CBOs are excluded from the settlement process and are rejected at that time indicated as latest debit time (if not settled before).

Warehoused payments can be initiated by default ten calendar days in advance¹². The payment message passes the schema validation and the business validation of CLM and is warehoused until the SoD of CLM of that date.

Offsetting mechanisms are not used.

For CBOs a defined amount of liquidity can be reserved in advance to separate it from the “non-reserved” part of the MCA used inter alia for liquidity transfers.

CBOs that are not yet executed can be revoked.

CBOs that cannot settle immediately are queued. The orders within the queue are then processed following the FIFO-principle. CBs can intervene on queued CBOs by

- changing the set execution time

Note: This is only possible in case an execution time has been set in the original payment order.

- re-ordering of queued CBOs

- revoking the queued CBOs

CLM continuously attempts to settle the CBOs in the queue.

5.1.8.1 Entry disposition

General remarks

In CLM, the available liquidity of the MCA can be divided into a non-reserved part and a part reserved for CBOs (see chapter [Available liquidity](#) [▶ 114]).

CBOs use the available liquidity in the dedicated reserved part of the MCA first. Only in case this reserved part does not include any (or not enough) liquidity, the liquidity on the non-reserved part of the available liquidity on the MCA is used in a second step. Moreover, the FIFO-principle applies among all CBOs.

Liquidity transfers use only the liquidity in the non-reserved part of the MCA. Liquidity transfers are only settled immediately. Therefore, no FIFO-principle is needed. Standing liquidity transfer orders are treated like immediate liquidity transfer as soon as the triggering event occurs. The only difference is that standing liquidity transfer orders could also settle partially in case of insufficient liquidity in the non-reserved part of the MCA.

Offsetting mechanisms are not required in CLM. They are neither used for CBOs nor for liquidity transfers.

Unsuccessful entry disposition

If the submitted CBO cannot settle in the entry disposition, it is placed into the queue of CBOs according to the FIFO-principle.

¹² The numbers of days are defined as a parameter that indicates the number of days payments can be submitted to CLM in advance.

Note: In contrast to CBOs, liquidity transfers other than SLTs are not placed into a queue and are rejected with appropriate error code in case of insufficient liquidity.

Settlement of payments in the entry disposition

CBOs have the highest priority and are settled prior to liquidity transfers, regardless of the priority assigned to liquidity transfers. Therefore, CLM checks first which kind of payment order the CLM Account Holder has submitted, i.e. whether it is a payment (meaning a CBO) or a liquidity transfer.

CBOs

- | First, the liquidity on the reserved part for CBOs of the available liquidity on the MCA is checked.
- | In case of sufficient liquidity on the reserved part, the CBO is settled.
- | In case of insufficient liquidity, the liquidity on the non-reserved part of the available liquidity on the MCA is checked.
 - If there is overall sufficient liquidity, the CBO is settled.
 - If there is not sufficient liquidity, the CBO is queued. In case of queued CBOs, CLM creates and sends an automated inter-service liquidity transfer to pull the missing liquidity from the linked RTGS DCA.
- | When the payment is submitted to CLM and in case there are already other CBOs queued due to insufficient available liquidity on the MCA, the submitted payment is queued as well and it is put at the end of the queue following the FIFO-principle. When putting a new payment at the end of the queue CLM again creates and sends a new automated inter-service liquidity transfer to the RTGS component to pull liquidity from the linked RTGS DCA. The amount within this new automated inter-service liquidity transfer is the sum of all pending CBOs that are currently in the queue minus the available liquidity (that is still not sufficient to settle the first CBO in the queue).

Note: As soon as a new automated inter-service liquidity transfer arrives in the RTGS, the RTGS component deletes the previous automated inter-service liquidity transfer and considers only the current one with the sum of all queued CBOs.

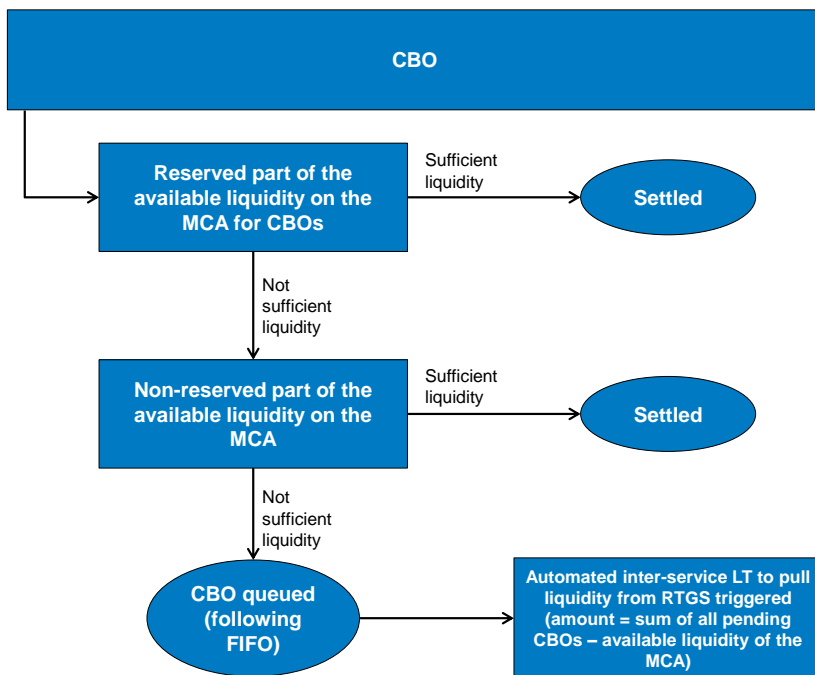


Figure 17 - Entry disposition of CBOs

Liquidity transfers

For liquidity transfers, only the non-reserved part of the available liquidity on the MCA can be used for the settlement. In case the liquidity is sufficient and there are no pending CBOs queued, the liquidity transfer is immediately settled. In case the liquidity on the non-reserved part of the available liquidity on the MCA is not sufficient, the behavior of CLM depends on the way of initiation of the liquidity transfer:

- Immediate liquidity transfers:** In case the liquidity on the non-reserved part of the MCA is not sufficient and in case there are no pending CBOs in the queue, the immediate liquidity transfer is rejected and a camt.025 receipt is sent to the CLM Account Holder who submitted the original liquidity transfer.
- Standing liquidity transfer order:** In case the liquidity on the non-reserved part of the MCA is not sufficient and in case there are no pending CBOs in the queue, the standing liquidity transfer order is partially settled up to the amount that is available. In case that more than one standing liquidity transfer order is in place, the available liquidity is used “pro rata” for all existing standing liquidity transfer orders. For the remaining amount(s) that could not settle in the first settlement attempt no further attempt(s) take(s) place.

Note: 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 CLM Account Holder is informed accordingly via a camt.054 BankToCustomerDebitCreditNotification.

Event-based liquidity transfer orders, e.g. stemming from floor/ceiling functionality: Analogue of the standing liquidity transfer orders, that means, in case the liquidity on the non-reserved part of the MCA is not sufficient and there are no pending CBOs in the queue, the event-based liquidity transfer orders, e.g. stemming from floor/ceiling functionality are partially settled up to the amount that is available. For the remaining amount that could not settle in the first settlement attempt no further attempt takes place.

Note: 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 CLM Account Holder is informed accordingly via a camt.054 BankToCustomerDebitCreditNotification.

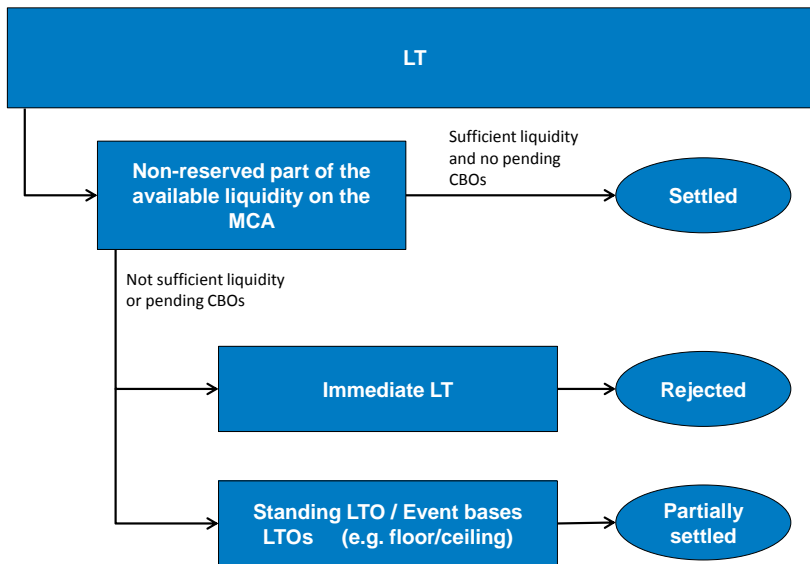


Figure 18 - Entry disposition of liquidity transfers

Action	Reserved part of the MCA for CBOs	Non-reserved part of the MCA	Queued CBOs	Automated inter-service liquidity transfer pending in RTGS	Remarks
Starting situation	100	50	0		
First CBO - amount: debiting 50	50 ↓	50	0		
Second CBO – amount: debiting 500	50	50	500 ↑	400 ↑	
Inter-service liquidity transfer from T2S – amount: crediting 10	50	60 ↑	500	390	
Third CBO – amount: debit-	50	60	650		

Action	Reserved part of the MCA for CBOs	Non-reserved part of the MCA	Queued CBOs	Automated inter-service liquidity transfer pending in RTGS	Remarks
ing 150			↑		
Intra-service liquidity transfer – amount: debiting 30	50	60	650		Rejected due to queued CBOs
Automated inter-service liquidity transfer from RTGS – amount: crediting 300	50	360 ↑	650	whereas only 240 pending	
Automated inter-service liquidity transfer from RTGS – amount: crediting 240	0 ↓	0 ↓	0 ↓	0 ↓	

Table 34 - Entry disposition of liquidity transfers - example

Rejection during EoD processing

If queued CBOs cannot be settled until the EoD and are still queued due to lack of liquidity, these payments are rejected during EoD processing.

5.1.8.2 Comprehensive queue management

If a submitted CBO cannot be settled in the entry disposition, it is placed in the queue.

As long as CBOs are not settled, the CB of the CLM Account Holder has the ability to change the parameters of the payment.

Three different control options are offered:

- changing the set execution time (if already defined in the CBO before sending it to CLM)
- re-ordering the queued payments
- revocation of a queued payment

These control options enable the CB to react on changed conditions during the business day. It is possible to modify a single CBO or several CBOs at the same time. In case it is not possible to execute a modification request, the CB is informed accordingly. Further details on the interventions done in U2A can be found in the CLM user handbook.

In case of successful interventions, the process to resolve the queue in CLM is started.

Changing the set execution time

In principle, CBOs can be submitted 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](#) [82]).

In case a submitted CBO 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 U2A as long as the time is not reached. Such a change has no impact on the processing of the CBO, but on the queue management as the time indication only supports the queue management.

Action	Effect
Deleting the earliest debit time indicator of a CBO (FROTIME)	This CBO is not in the queue yet, as the earliest debit time indicator is not reached so far. With the deletion, the entry disposition is done by CLM and a first settlement attempt takes place. As a result the CBO is either settled or put at the end of the queue.
Changing the earliest debit time indicator of a CBO (FROTIME)	The CBO is included in the settlement process from the new indicated time.

Table 35 - Effect of changing the execution time

Note: Since the deletion or modification of the latest debit time indicator has no direct effect on the queue management, it has not been considered in the table.

Re-ordering the queued payments

The CB can change the queue position for a single or a sequence of CBOs via U2A. The CBO selected can be placed on:

- I to the top of the queue of CBOs
- I to the end of the queue of CBOs

Action	Effect
Moving a CBO to the top of the queue	Immediate check whether the new CBO on top (and possibly any further in the queue) can be executed
Moving a CBO from the top to the end of the queue	
Moving a CBO that is not on top to the end of the queue	The action is taken into account during the next settlement process – no immediate attempt to settle.

Table 36 - Effect of changing the order of queued CBOs

In case of such a change, the CBO:

- keeps its original submission time or
- is placed in the queue according to the change

Revocation of a queued payment

A CB can revoke CBOs that are queued and not yet successfully settled. The revocation can be done via U2A and A2A at any time during the day. The queue of CBOs is reduced by the revoked payment.

For further details, please refer to chapter [Revocation of payments](#) [▶ 100].

5.1.8.3 Dissolution of the payment queue

The queue is resolved in an event-oriented way starting with the CBO on top.

Events	By ...
Liquidity increase	<ul style="list-style-type: none"> incoming settled CBOs (i.e. credits) incoming settled intra-service liquidity transfers incoming inter-service liquidity transfers from other services/components
Intervention on queue level	<ul style="list-style-type: none"> If the CBO on top of the queue is changed <ul style="list-style-type: none"> – change of order – revocation – rejection of the CBO due to the fact that the latest debit time is reached

Table 37 - Origin of possible events

As soon as one of the above mentioned events occurs, further settlement attempts take place to settle the CBOs starting with the one on top of the queue. There are no additional algorithms as it is the case for the RTGS component.

The resolving queue process and the entry disposition are handled in the same way. If a single CBO cannot be settled, it remains in the queue (at maximum until the end of the business day).

5.2 Liquidity management

This chapter describes the tools and processes for an efficient management and usage of liquidity across the TARGET Services in a harmonised and generic way. It covers the different kinds of liquidity transfers, liquidity reservations, floor/ceiling management as well as the standing facilities.

5.2.1 Available liquidity

The MCA is used for the settlement of:

- liquidity transfers - to ensure an efficient liquidity provisioning for the settlement in T2S, RTGS and TIPS
- payments in the context of CBOs

In principle the available liquidity of a MCA consists of:

- the balance on the MCA
- the credit line linked to the MCA

Note: In case a CLM Account Holder has more than one MCA, the credit line can only be linked to one MCA. Only this MCA (with the linked credit line) can be used for the CBOs of that CLM Account Holder.

It is up to the CLM Account Holder to decide whether the available liquidity should be divided into:

- the reserved part for CBOs and
- the non-reserved part

This would be done by using the reservation function.

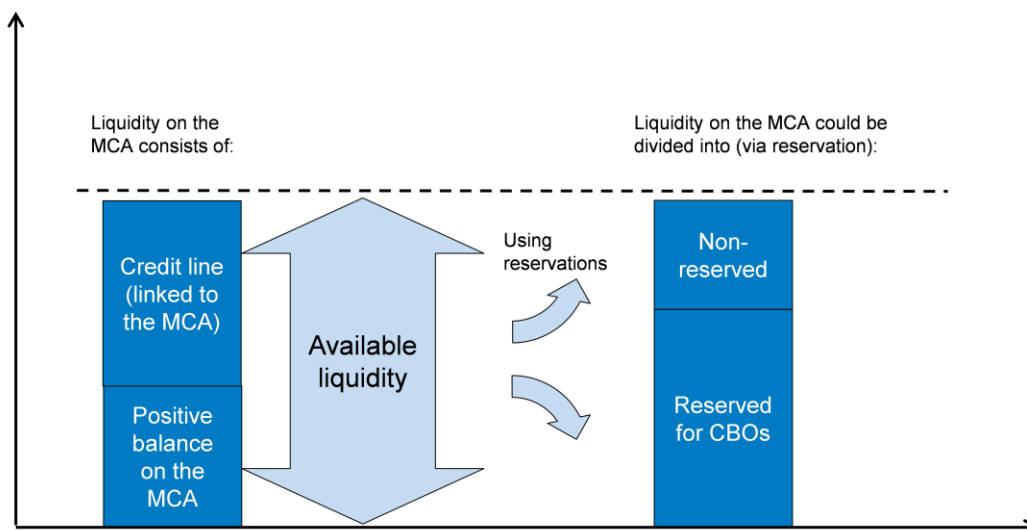


Figure 19 - Available liquidity

Without using the reservation function (please also refer to chapter [Liquidity reservation](#) [131]) the MCA just consists of the available liquidity that is used for CBOs and liquidity transfers.

With using the reservation function, the reserved part for CBOs cannot be used for liquidity transfers. The reserved liquidity is only available for the settlement of CBOs.

5.2.2 Liquidity transfer

5.2.2.1 Overview

The MCA is the central source of liquidity for the different services/components the CLM Account Holder joined in. Therefore CLM has to ensure an efficient liquidity provision by liquidity transfers within CLM, to DCAs of other services or components. Furthermore CLM optimises the efficient usage of liquidity for the different services/components and transfers liquidity between them.

Liquidity transfers are not classified as payments (i.e. pacs); they are cash management instructions using camt messages. The liquidity transfer order message (camt.050) is exchanged between users and the system in order to instruct the transfer of cash from one cash account to another cash account.

Liquidity can be transferred:

- | between different MCAs within the CLM (under certain preconditions – for further details see chapter [Immediate liquidity transfer between two MCAs](#) [▶ 122]).
- | between the MCAs and the DCAs of the different services/components
- | between DCAs within the same service/component (out of scope of this UDFS and described in the relevant service/component UDFS)
- | between DCAs of different services/components (via CLM transit accounts)

The following types of liquidity transfers exist:

- | immediate liquidity transfer
- | automated liquidity transfer
- | rule-based liquidity transfer
- | standing order liquidity transfer

In general, liquidity transfers are never queued. They are either immediately settled (full or partially) or rejected. Only under following conditions automatically generated liquidity transfers can become pending in RTGS.

- | The MCA has insufficient liquidity for a CBO and there is not sufficient liquidity on the RTGS DCA for an automatically triggered liquidity transfer to the MCA.
- | Any incoming liquidity (up to the required amount) on the RTGS DCA is then transferred stepwise (partially) to the MCA until the pending CBO can be settled.
- | The pending automated inter-service liquidity transfer from CLM is set on the top of the payment queue in RTGS.

Within CLM, liquidity can be transferred between MCAs belonging to the same party or Liquidity Transfer Group. Liquidity Transfer Groups are configured by the respective CB. For further details please refer to chapter [Types of groups](#) [▶ 62].

The rules for Liquidity Transfer Groups do not apply for CBs. That means a liquidity transfer within CLM is always possible as soon as a CB account is involved.

5.2.2.2 Initiation of liquidity transfers

A liquidity transfer can be submitted via U2A or A2A (camt.050) to the CLM by:

- | a CLM Account Holder
- | another actor on behalf of the CLM Account Holder or
- | a CB

A liquidity transfer can be initiated as

- | immediate liquidity transfer order - the amount is immediately transferred after initiation
- | automaticated liquidity transfer order - if there is not sufficient available liquidity for the CBO settlement
- | rule-based liquidity transfer orders triggered by floor or ceiling amount
- | standing liquidity transfer orders – the fixed amount is transferred regularly at predefined event

A partial execution of a liquidity transfer takes place for standing orders and automated and rule-based liquidity transfer orders. For several standing orders, where the sum of all standing orders of the CLM Account Holder to be settled at the same event is larger than the available liquidity, CLM reduces all respective standing orders in a pro-rata mode.

The characteristics of different kinds of liquidity transfers are summarised in the following table.

	Trigger	Execution	Partial execution	Frequency of execution
Immediate liquidity transfer	External party, i.e. account holder, Co-manager, CB acting on behalf	Immediately after the submission during the operating hours	Not enabled but rejection in case of lack of liquidity	Once
Automated liquidity transfer order	CLM	Automatically triggered by event, then immediate	In case liquidity available in RTGS is not sufficient to settle the full amount of the order	In case of pending CBOs
Rule-based liquidity transfer	Triggered by CLM, if a target amount due to floor/ceiling or pending U-H-payment configuration rules	Automatically triggered by event, then immediate	In case liquidity available in RTGS is not sufficient to settle the full amount of the order; for liquidity transfers triggered by floor amount	Whenever a floor/ceiling amount is breached
Standing order	Based on data propagated by CRDM	Triggered by event, then immediate changes become effective on the following business day	In case liquidity on the debited account is not sufficient to settle the full amount of the order	On a regular basis

Table 38 - Underlying liquidity transfer characteristics

The liquidity provisioning for the settlement of all payment types in the MCA shall be processed following the FIFO principle. For further details, refer to chapter [Processing of payment orders](#) [▶ 104].

Detailed information regarding the initiation of liquidity transfers in U2A mode can be found in the CLM user handbook.

5.2.2.3 Liquidity transfer process

In the following process descriptions successful liquidity transfers are described. The unsuccessful processes are described in chapter [Rejection of liquidity transfer orders](#) [▶ 129].

The processing of liquidity transfers is dependent on how the order is triggered. There is a need to distinguish between immediate liquidity transfers submitted by an external party (via camt.050 LiquidityCreditTransfer) and system generated liquidity transfer orders (i.e. standing orders, rule-based and automated liquidity transfers).

5.2.2.3.1 Immediate liquidity transfer from MCA to DCA

A CLM Account Holder can transfer liquidity from its MCA to any DCA within another settlement service/component (T2S, RTGS or TIPS).

Message flow

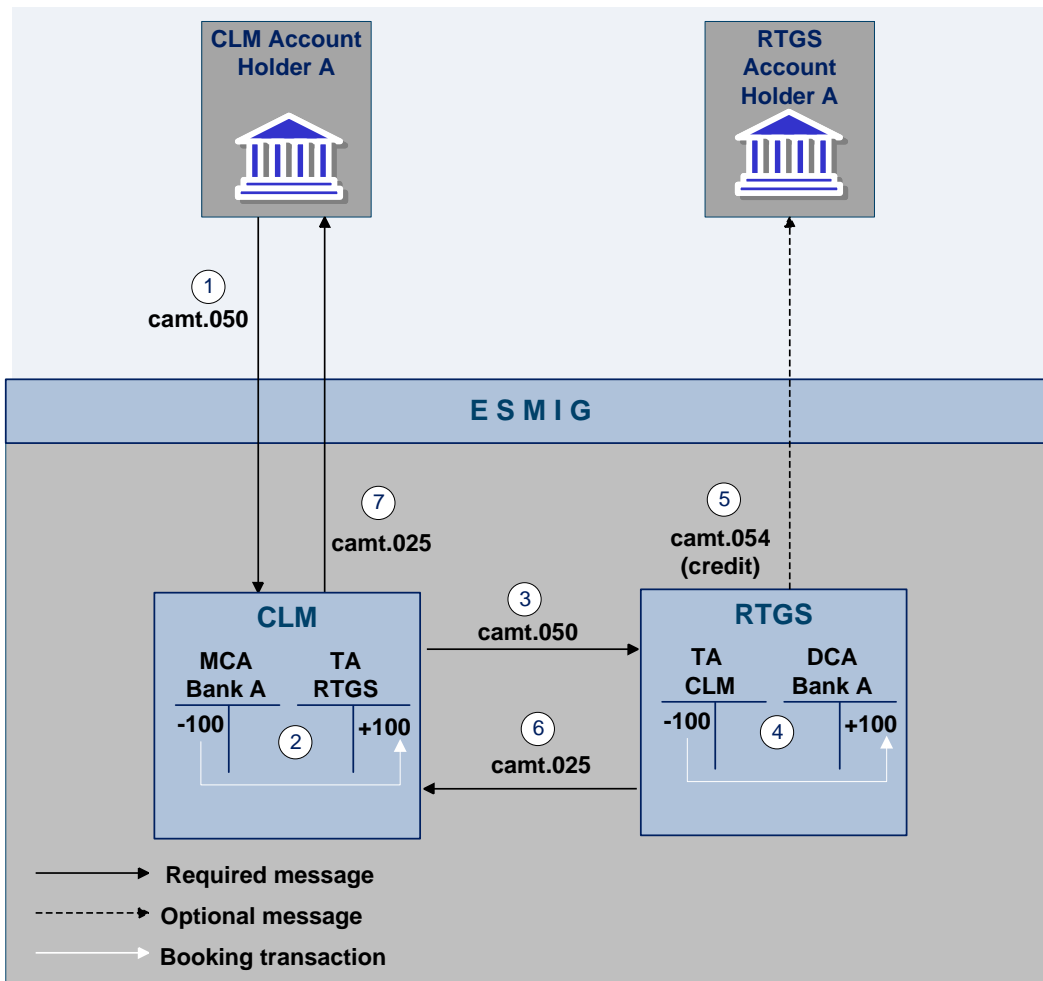


Figure 20 - Example camt.050 liquidity transfer from MCA to RTGS DCA

Process description

Step	Processing in/between	Description
1	CLM Account Holder via ESMIG to CLM	A camt.050 is sent from a CLM Account Holder via ESMIG to CLM.
2	CLM	Booking on MCAs (MCA to transit account RTGS)
3	CLM to RTGS	A camt.050 is forwarded to RTGS.
4	RTGS	Booking on RTGS DCAs (transit account-CLM to RTGS DCA)
5	RTGS via ESMIG to RTGS account holder	A camt.054 (credit) is sent by RTGS via ESMIG to the RTGS account holder (optional).
6	RTGS to CLM	For execution a camt.025 generated in RTGS is sent to CLM.
7	CLM via ESMIG to CLM Account Holder	A camt.025 is sent by CLM via ESMIG to the CLM Account Holder.

Table 39 - Liquidity transfer from MCA to RTGS DCA**Used messages**

- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]
- | [Receipt \(camt.025\)](#) [▶ 397]

5.2.2.3.2 Immediate liquidity transfer from DCA to MCA

A settlement service/component account holder can transfer liquidity from its DCA within a settlement service/component (T2S, RTGS or TIPS) to any MCA.

Message flow

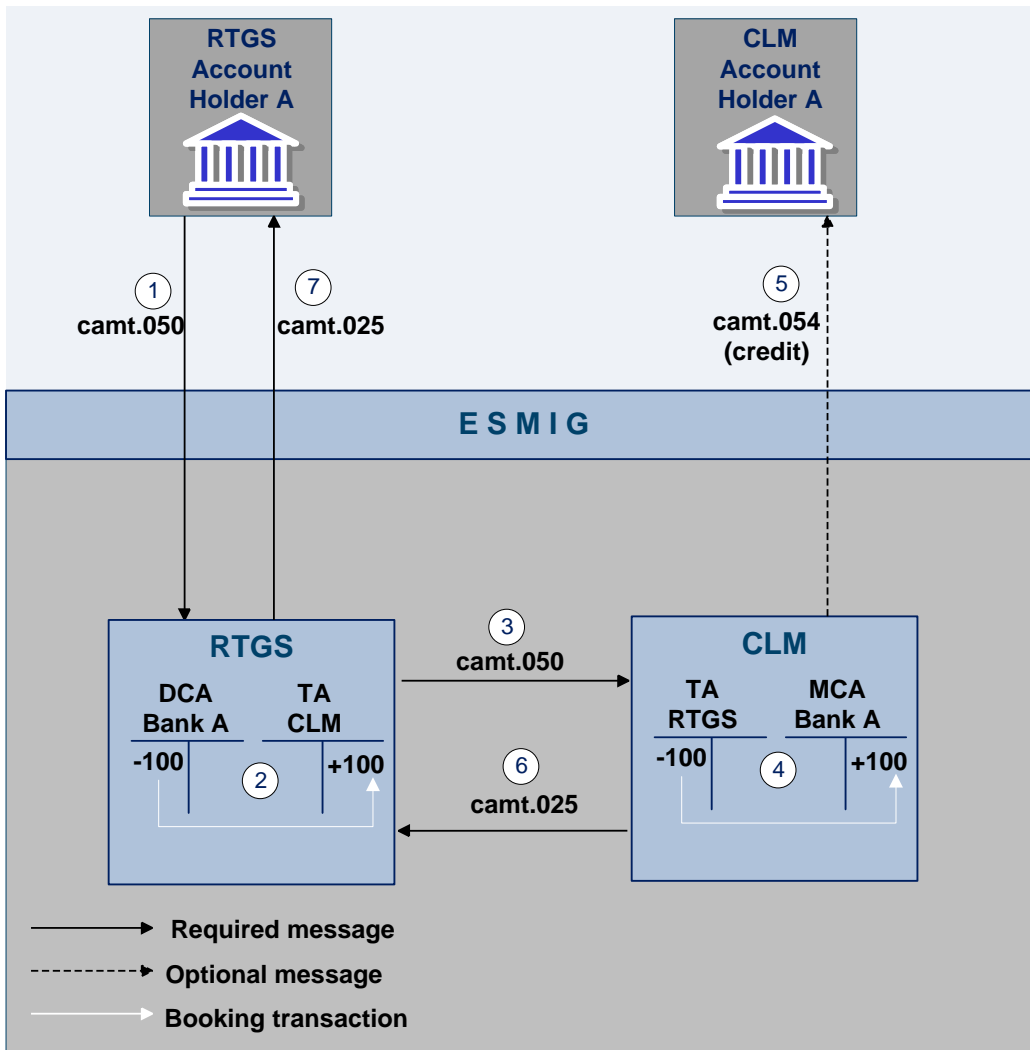


Figure 21 - Example camt.050 liquidity transfer from RTGS DCA to MCA

Process description

Step	Processing in/between	Description
1	RTGS account holder via ESMIG to RTGS	A camt.050 is sent from a RTGS account holder via ESMIG to RTGS.
2	RTGS	Booking on RTGS DCAs (RTGS DCA to transit account-CLM)
3	RTGS to CLM	A camt.050 is forwarded to CLM.
4	CLM	Booking on MCA (transit account-RTGS to MCA)

Step	Processing in/between	Description
5	CLM via ESMIG to CLM Account Holder	A camt.054 (credit) is sent by CLM via ESMIG to the CLM Account Holder (optional).
6	CLM to RTGS	For execution a camt.025 generated in CLM is sent to RTGS.
7	RTGS via ESMIG to the RTGS account holder	For execution, a camt.025 is sent by RTGS via ESMIG to the RTGS account holder.

Table 40 - Liquidity transfer from RTGS DCA to MCA

Used messages

- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]
- | [Receipt \(camt.025\)](#) [▶ 397]

5.2.2.3.3 Immediate liquidity transfer between two MCAs

A CLM Account Holder can transfer liquidity from one MCA to another MCA. The owners of the MCAs have to be in the same Liquidity Transfer Group to work with the MCA to be credited.

Message flow

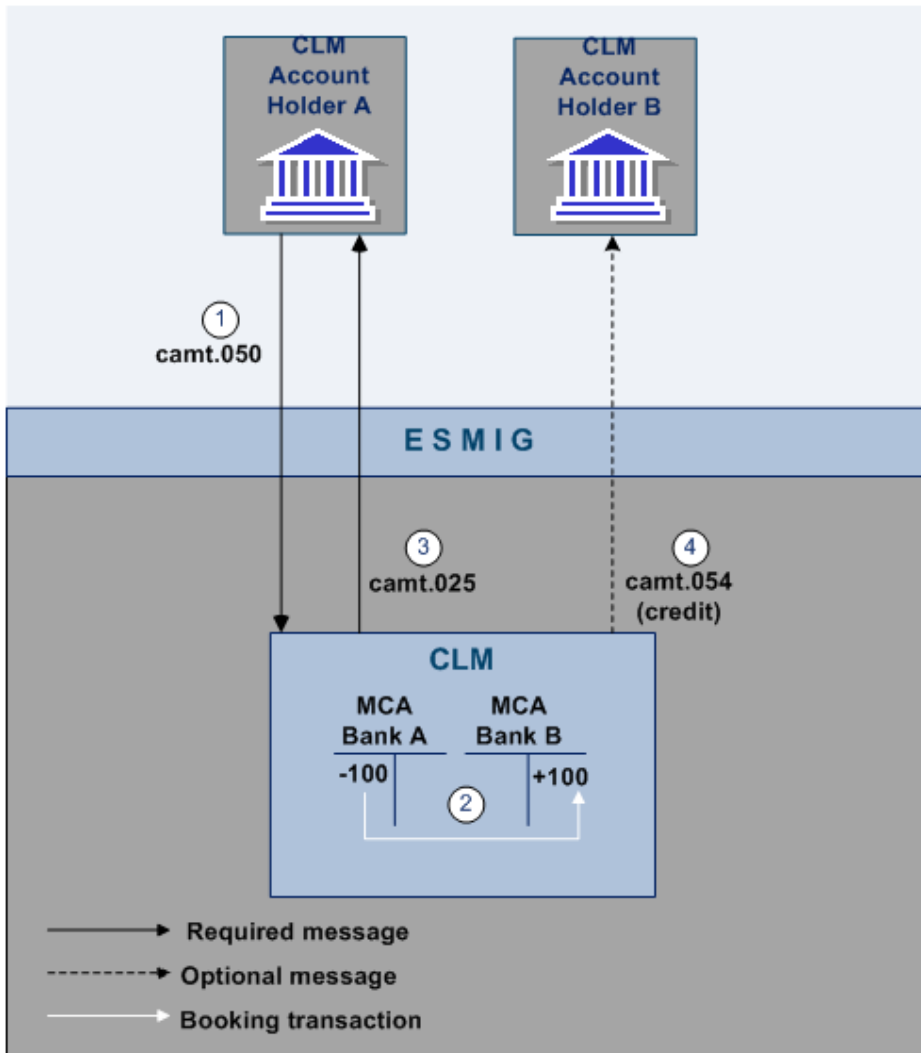


Figure 22 - camt.050 liquidity transfer intra-CLM

Process description

Step	Processing in/between	Description
1	CLM Account Holder A via ESMIG to CLM	A camt.050 is sent from a CLM Account Holder A via ESMIG to CLM.
2	CLM	Booking on MCAs
3	CLM via ESMIG to CLM Account Holder A	A camt.025 is sent by CLM via ESMIG to CLM Account Holder A.
4	CLM via ESMIG to CLM Account Holder B	A camt.054 is sent by CLM via ESMIG to CLM Account Holder B (optional).

Table 41 - Liquidity transfer intra-CLM**Used messages**

- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]
- | [Receipt \(camt.025\)](#) [▶ 397]

5.2.2.3.4 Immediate liquidity transfer between two DCAs in different settlement services/components

A settlement service/component account holder can transfer liquidity from a DCA in one settlement service/component to a DCA within another settlement service/component.

Message flow

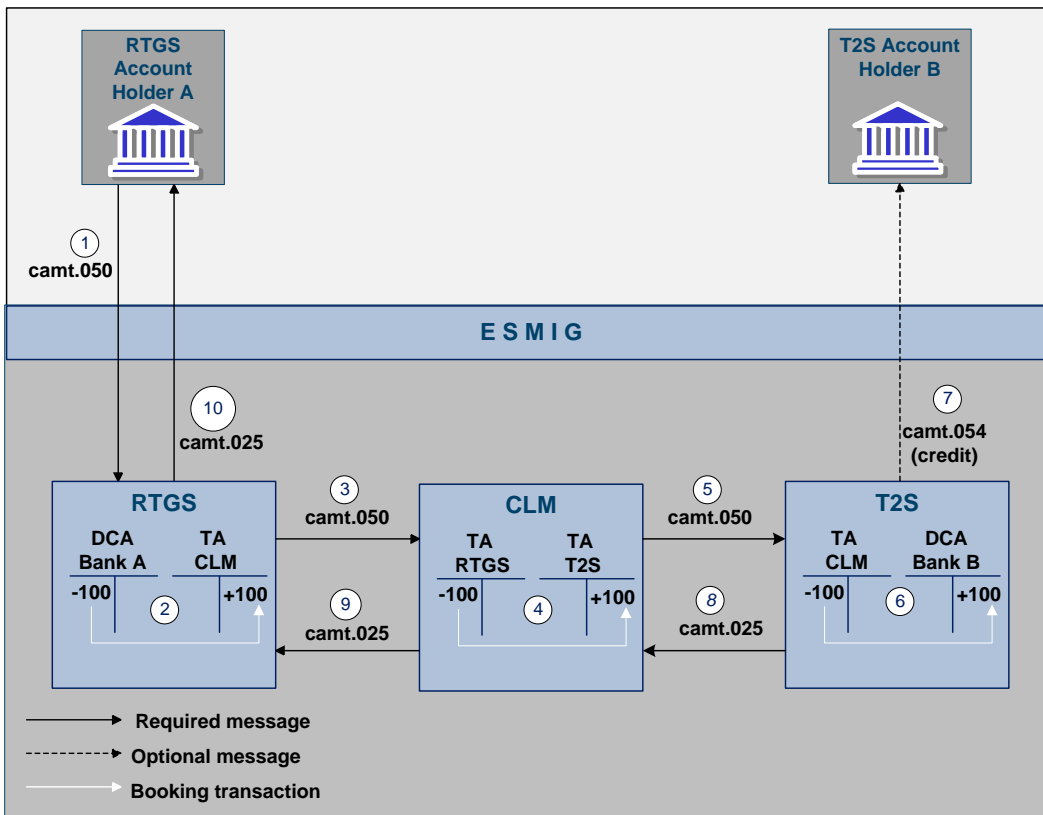


Figure 23 - Example camt.050 liquidity transfer inter-service/component

Process description

Step	Processing in/between	Description
1	RTGS account holder via ESMIG to RTGS	A camt.050 is sent from a RTGS account holder via ESMIG to RTGS.
2	RTGS	Booking on RTGS DCAs (RTGS DCA to transit account-CLM)
3	RTGS to CLM	A camt.050 is forwarded to CLM.
4	CLM	Booking on technical accounts in CLM (transit account -RTGS to transit account-T2S)
5	CLM to T2S	A camt.050 is forwarded to T2S.
6	T2S	Booking on T2S accounts (transit account -CLM to T2S DCA)
7	T2S via ESMIG to T2S account holder	A camt.054 (credit) is sent by T2S via ESMIG to the T2S account holder (optional).

Step	Processing in/between	Description
8	T2S to CLM	A camt.025 generated in T2S is sent to CLM.
9	CLM to RTGS	CLM forwards the camt.025 to RTGS.
10	RTGS via ESMIG to direct RTGS account holder	A camt.025 is sent by RTGS via ESMIG to the RTGS account holder.

Table 42 - Liquidity transfer inter-service/component

Used messages

- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]
- | [Receipt \(camt.025\)](#) [▶ 397]

5.2.2.3.5 System-generated liquidity transfer orders

In addition to immediate liquidity transfers submitted by external parties, the following system-generated liquidity transfers are available.

- | standing liquidity transfers
- | rule-based liquidity transfers
- | automated liquidity transfers

The aim of this chapter is to illustrate the booking process as well as the related communication to the CLM MCA Holders.

Rule-based and automated liquidity transfers are only possible between CLM and RTGS.

Standing liquidity transfers are possible between CLM and the various settlement services/components.

The following table provides an overview about the different types of liquidity transfers between CLM and the other settlement services/components:

Liquidity transfer type	RTGS	T2S	TIPS
Standing Order	Provided	Provided	Provided
Rule-based liquidity transfer	<ul style="list-style-type: none"> ▮ Provided ▮ Pull liquidity in case of breached floor amount in CLM ▮ Push liquidity in case of breached ceiling amount in CLM 	n. a.	n. a.
Automated liquidity transfer because of pending CBOs	<ul style="list-style-type: none"> ▮ Provided ▮ Pull liquidity from RTGS 	n. a.	n. a.

Note: In contrast to immediate liquidity transfers there is a partial execution of system-triggered liquidity transfers. Depending on the trigger type the processing of the liquidity transfer differs. The following table provides an overview of the possible processing in CLM.

Rule-based liquidity transfer		Standing order	Pending CBOs
Breach of floor/ceiling threshold in CLM			
In case of floor amount breach in CLM: After partial execution no further processing, i.e. order is set to final status. New order is generated if floor amount is still breached.		Execution in pro-rata mode (distributed to all targeted accounts) is liquidity is not sufficient to serve all standing orders. After partial execution no further processing, i.e. order is set to final status.	If the available liquidity in CLM is not sufficient to serve the execution of CBOs: Remaining amount of order is queued on top of RTGS queue (queueing is repeated till complete amount of origin order is executed) Every execution booking is notified (optional) to account holder.

Table particular processing of liquidity transfers from CLM to other components in case of partial execution

Message flow

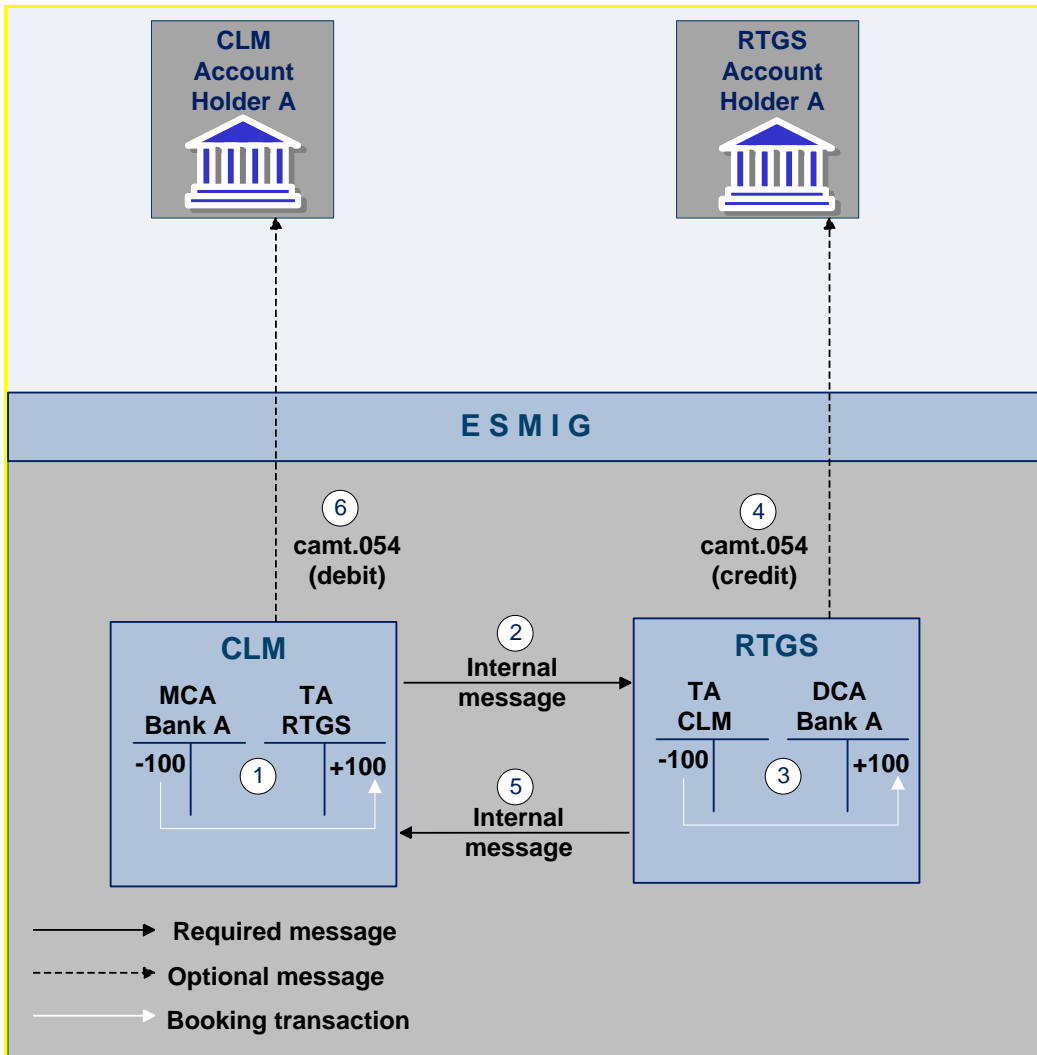


Figure 24 - Liquidity transfer from CLM triggered by automatic processes

Process description

Step	Processing in/between	Description
1	CLM	Booking on CLM MCA and RTGS transit account-CLM.
2	CLM to RTGS	CLM forwards liquidity transfer order to RTGS.
3	RTGS	Debit of transit account CLM and credit on RTGS DCA in RTGS.

Step	Processing in/between	Description
4	RTGS to CLM	A camt.054 credit notification (optional) generated in RTGS is sent to RTGS account holder.
5	RTGS to CLM	A settlement confirmation is sent to CLM.
6	CLM via ESMIG to direct RTGS account holder	A camt.054 debit notification (optional) is sent by RTGS via ESMIG to the CLM Account Holder.

Table 43 - Liquidity transfer inter-service/component

Used message

[BankToCustomerDebitCreditNotification \(camt.054\) \[442\]](#)

5.2.2.3.6 Rejection of liquidity transfer orders

Liquidity transfer orders sent to the CLM have to pass several validations before the liquidity is effectively transferred. Validations performed include technical checks, format checks as well as checks for the correct content.

For different reasons a liquidity transfer can be rejected and a notification providing the reason codes for rejection is returned to the sending actor (see chapter [Index of business rules and error codes \[537\]](#)). The validations are distinguished in two types:

Technical validations

CLM shall parse the message and perform a field level validation, e.g. on correct data types. CLM shall check whether all mandatory fields are populated. If the validation fails, a rejection notification with appropriate reason code is sent to the sender of the message (depending on the submission channel, a message in A2A mode or an error message on the screen in U2A mode).

Business validations

In case a liquidity transfer does not pass the business validation check, a receipt message (camt.025) is sent to the sending actor including the respective error code(s). For further details, please refer to [Index of business rules and error codes \[537\]](#).

Check for duplicate liquidity transfer

CLM carries out a duplicate submission control for incoming liquidity transfers. This control shall include the following fields: Sender of the message, Message Type, Receiver, Transaction Reference Number, Related Reference, Value Date and Amount.

Process specific authorisation checks

CLM performs specific authorisation checks. The liquidity transfer order can also be triggered by the scheduler in the case of standing orders.

Liquidity Transfer Group

For intra-service liquidity transfers, CLM checks whether both accounts belong to the same Liquidity Transfer Group or not. If not, the order is rejected. This check is not performed for CB accounts.

Field and reference data checks

The service performs the following field and reference data checks.

- | **field value validation** - codes are valid, domain values are within allowed range
- | **cross-field validation** – e.g. currency of the accounts involved equals the amount currency
- | **common reference data checks** – e.g. existence of active parties and accounts

The validations described above are performed in one step in order to capture all the possible breaches. The checks therefore have not to stop after the first breach occurring, as there could be further breaches in the subsequent checks. If the validation fails overall, a rejection notification with appropriate reason codes for all breaches which occurred is sent to the sender. This principle applies to technical and business validations separately.

Subsequent processes and checks

- | **Check available liquidity vs. amount to be transferred** - CLM checks whether enough liquidity is available. In case the liquidity is not sufficient the liquidity transfer order is rejected. The sender is notified with a negative receipt providing the related error code.
Note: In case reservations are used, only the non-reserved part of the available liquidity can be used for liquidity transfers.
- | **Update cash balances** - CLM books the liquidity transfer finally and irrevocably on both accounts and updates the defined value. CLM sends a settlement notification to the sending party and to the owner of the credited account.

5.2.3 Liquidity management features

5.2.3.1 Liquidity reservation

5.2.3.1.1 Overview

CLM offers the possibility to reserve cash of the MCA, so that therefore MCA has two types of liquidity sources:

- | reserved for CBOs
- | reserved for dedicated for seizure based on court decision(s)
- | non-reserved

The available liquidity in the reserved part of the MCA is used for CBs (e.g. reimbursement of liquidity providing tender operations) or for credit line decreases ¹³.

Reservations can be created, modified and deleted by the owner of the MCA (or another actor acting on behalf of the MCA owner) using U2A or A2A. Further details on the U2A functionality can be found in the CLM user handbook.

The owner of the MCA (or another actor acting on behalf of the MCA owner) has the following possibilities.

- | Create and/or modify reservations with immediate effect during the current business day as a one-time reservation in CLM, including:
 - establishing a specific amount during the current day with immediate effect as a one-time reservation (e.g. setting a new reservation of 300)
 - “resetting” to zero the liquidity reserved for the current business day only with immediate effect.
 - changing the amount on demand during the day with immediate effect (e.g. from 300 to 200 or from 300 to 400).
- | Create, modify or delete a standing order for reservation in CRDM valid as of the following day (i.e. valid as of the next business day until next change or the deletion of the standing order reservation amount)

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.

¹³ The latter one uses the reserved part of the MCA only in case there is not enough liquidity on the non-reserved part of the MCA.

Standing order reservation

Standing order reservations are created and managed in CRDM. The amount defined in the standing order for reservation is valid at the SoD, even if the amount of the reservation is changed during the preceding business day with immediate effect (such a change is only valid for the respective business day).

At the SoD, reservations are set according to the standing orders and up to the available liquidity on the MCA. The amount defined in the standing order for reservation is valid at the SoD even if the amount of the reservation is changed during the preceding business day with immediate effect (such a change is only valid for the respective business day).

In case that the amount of non-reserved available liquidity is not sufficient to fulfil the liquidity reservation set-up via standing order, the reservation is partially executed. CLM continues attempting to reserve the remaining amount until the reservation amount is reached whenever there is an increase of non-reserved liquidity on the MCA.

Note: All CBOs are settled with priority prior to liquidity transfers and are either fully executed or queued, before the reservation is fully executed.

One-time reservation with immediate effect

One-time reservations are created and managed directly in CLM. 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 CLM incoming liquidity might be blocked and used by a parallel booking process before the attempt to increase the reservation is performed.

Upon receipt EoD notification, reservation revocation or a new reservation order, CLM stops processing the original (queued) reservation order.

In case that the amount of non-reserved available liquidity is not sufficient to fulfil the liquidity reservation order, the reservation is partially executed. CLM attempts to reserve the remaining amount until the reservation amount is reached whenever there is an increase of non-reserved liquidity on the MCA.

5.2.3.1.2 Liquidity reservation process

Reservation process – one-time reservation with immediate effect

The following message flows illustrate the reservation creation, the amendment ([ModifyReservation \(camt.048\)](#) [▶ 414]) and the “reset to zero” ([DeleteReservation \(camt.049\)](#) [▶ 416]) in CLM.

Note: The creation and the management of standing order reservations are done in CRDM.

Message flow

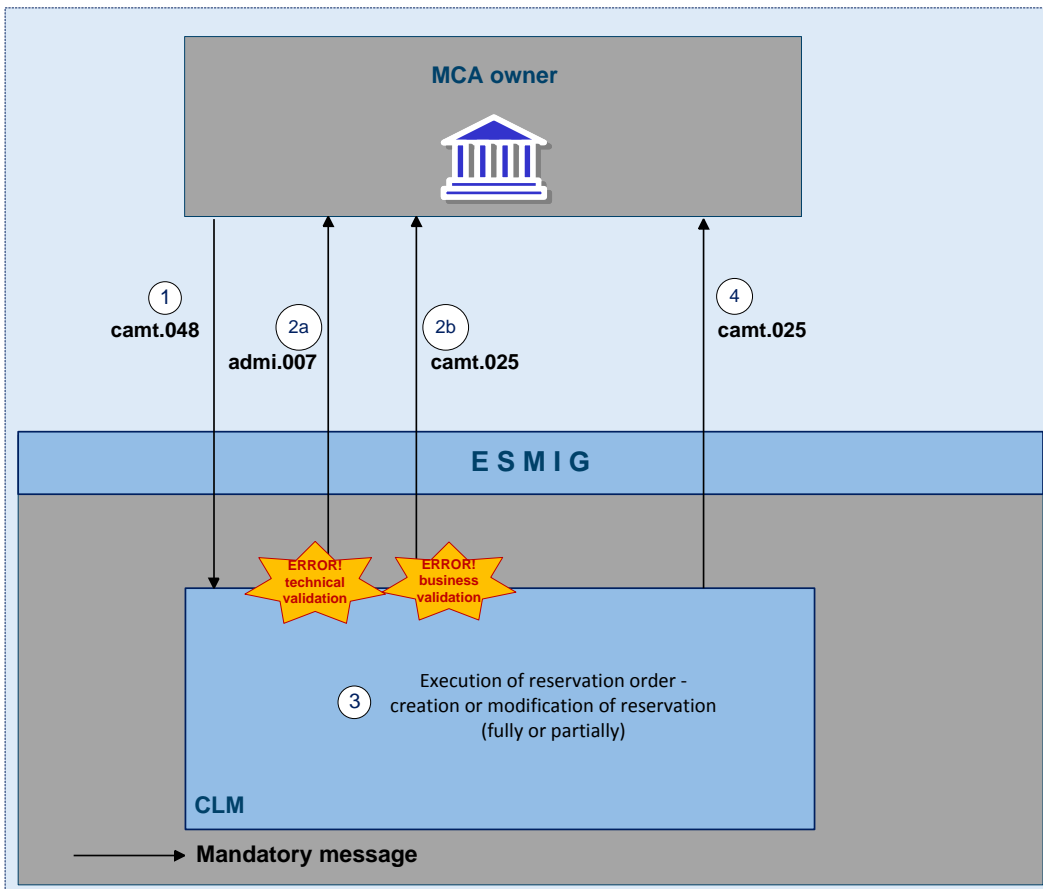


Figure 25 - One time reservation with immediate effect

Process description

Step	Processing in/between	Description
1	Owner of the MCA via ESMIG to CLM	The owner of the MCA sends a camt.048 via ESMIG to CLM.
2a	CLM	CLM performs a message check and technical validation. In case of a negative technical validation, an error message (admi.007) is sent. In case of a successful technical validation, CLM performs the business validation checks.
2b	CLM	CLM business validation. In case of a negative business validation, an error message (camt.025) is sent.

Step	Processing in/between	Description
3	CLM	CLM executes one-time reservation with immediate effect.
4	CLM via ESMIG to the owner of the MCA	Upon (partial) execution, CLM sends a camt.025 via ESMIG to owner of the MCA.
5	CLM	<p>The remaining reservation request is queued and processed in an event-oriented way.</p> <p>In case of an increase of the available liquidity an asynchronous resolving process attempts to process the pending reservation order.</p> <p>Note: Even if the increase of available liquidity is not sufficient for the complete processing, the pending reservation is processed partly (the pending reservation is decreased and the existing reservation is increased).</p>

Table 44 - Creation of a one-time liquidity reservation with immediate effect

Used messages

- | [ModifyReservation \(camt.048\)](#) [▶ 414]
- | [Receipt \(camt.025\)](#) [▶ 397]

Error message (admi.007) Modification of a reservation with immediate effect

Message flow

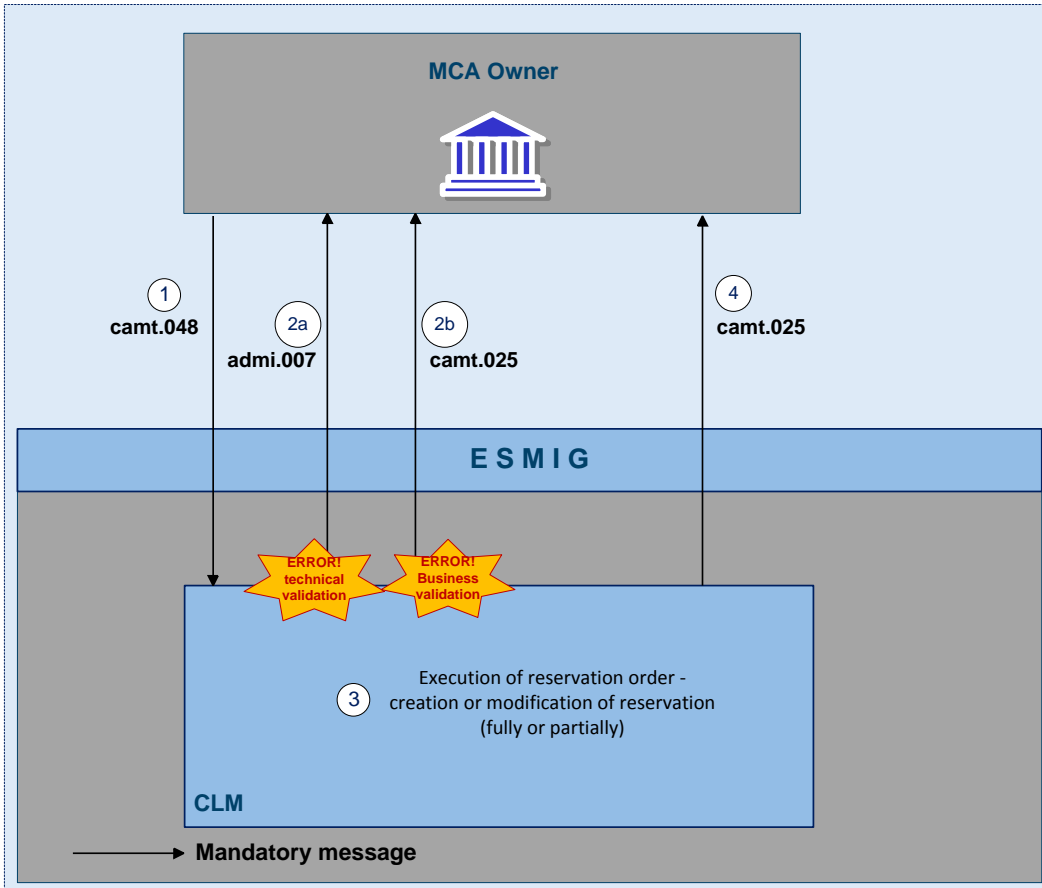


Figure 26 - One time reservation with immediate effect

Since the same messages are used for creating a reservation as well as modifying a reservation, the message flow for creating a one-time reservation applies here, too.

Process description

Step	Processing in/between	Description
1	Owner of the MCA via ESMIG to CLM	The owner of the MCA sends a camt.048 via ESMIG to CLM in order to modify the reservation with immediate effect.
2a	CLM	CLM performs a message check and technical validation. In case of a negative technical validation, an error message (admi.007) is sent. In case of a successful technical validation, CLM performs the business validation checks.
2b	CLM	CLM business validation: in case of a negative business valida-

Step	Processing in/between	Description
		<p>tion, an error message (camt.025) is sent.</p> <p>In case of successful business validation, CLM starts executing the reservation.</p>
3	CLM	CLM executes the modification of reservation.
4	CLM via ESMIG to the owner of the MCA	Upon (partial) execution, CLM sends a camt.025 via ESMIG to owner of the MCA.
5	CLM	<p>The remaining reservation request is queued and processed in an event-oriented way.</p> <p>In case of an increase of the available liquidity an asynchronous resolving process attempts to process the pending reservation order.</p> <p>Note: Even if the increase of available liquidity is not sufficient for the complete processing, the pending reservation is processed partly (the pending reservation is decreased and the existing reservation is increased).</p>

Table 45 - Modification of a one-time liquidity reservation with immediate effect

Used messages

- | [ModifyReservation \(camt.048\)](#) [▶ 414]
- | [Receipt \(camt.025\)](#) [▶ 397]

Error message (admi.007)

“Resetting to zero” reservation

Message flow

Aim of this chapter is to illustrate the deletion of a liquidity reservation in CLM with immediate effect.

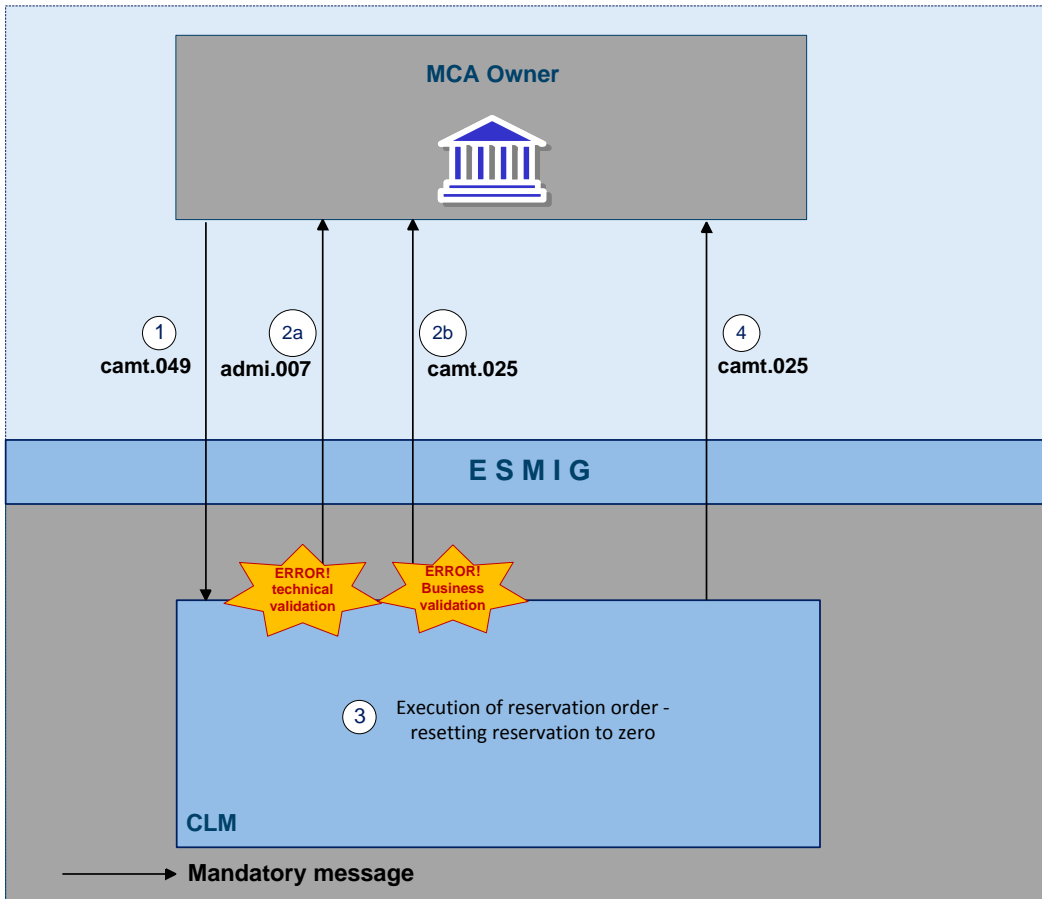


Figure 27 - Deletion reservation with immediate effect

Process description

Step	Processing in/between	Description
1	Owner of the MCA via ESMIG to CLM	Owner of the MCA (or another actor acting on behalf of the MCA owner) sends a camt.049 via ESMIG to CLM in order to reset to zero.
2a	CLM	CLM performs a message check and technical validation. In case of a negative technical validation, an error message (admi.007) is sent. In case of a successful technical validation, CLM performs the

Step	Processing in/between	Description
		business validation checks.
2b	CLM	CLM business validation. In case of a negative business validation, an error message (camt.025) is sent. In case of successful business validation, CLM starts executing the reservation.
3	CLM	CLM executes "resetting to zero".
4	CLM via ESMIG to the owner of the MCA	CLM sends a camt.025 via ESMIG to owner of the MCA. Note: Only in case of a successful "reset to zero", a notification (camt.025) is sent to the owner of the MCA (or another actor acting on behalf).

Table 46 - "Resetting to zero" reservation

Used messages

- | [ModifyReservation \(camt.048\)](#) [▶ 414]
- | [Receipt \(camt.025\)](#) [▶ 397]

5.2.3.1.3 Effect and tapping of liquidity reservation

Error message (admi.007)

Basic principles of liquidity tapping

The CLM provides the opportunity to define a dedicated liquidity pool for CBOs. The definition of reservation finally determines the sequence of liquidity tapping from these pools. The tapping of liquidity also includes consideration of RTGS liquidity pools, which can be pulled by CLM via automated liquidity transfer orders.

The generic sequence of liquidity tapping in CLM for debits on MCA can be illustrated as follows.

Business case	Tapping of liquidity reservation				
	MCA		RTGS DCA		
	CBOs	Non-reserved	Urgent (U)	High (H)	Non-reserved

Business case	Tapping of liquidity reservation				
	MCA		RTGS DCA		
	CBOs	Non-reserved	Urgent (U)	High (H)	Non-reserved
8Credit line decrease	2	1	5 ¹⁴	4 ¹⁵	3 ¹⁶
CBOs (including Cash Withdrawal)	1	2	5 ¹⁷	4 ¹⁸	3 ¹⁹
Liquidity transfer		1			

Table 47 - Liquidity tapping in CLM

Numeric example of reservation usage

The following table explains the effect of the reservation functionality for the processing of cash transfers in CLM. It illustrates the changes of the different liquidity type sources and provides numeric examples.

Activity	Balance on MCA of Bank A	Liquidity reserved for CBOs	Non-reserved liquidity
Start	1,000	300	700
Settlement liquidity transfer = 50 (debit)	950 ↓	300 ↔	650 ↓
Reimbursed marginal lending to CB = 200	750 ↓	100 ↓	650 ↔
Receiving liquidity transfer from Bank C = 20 (credit)	770 ↑	100 ↔	670 ↑
Set-up of overnight deposit = 100	670 ↓	0 ↓	670 ↔
Incoming liquidity from RTGS DCA = 80	750	0	750

- 14 Related to automated liquidity transfer due to pending CBO or credit line decrease
- 15 Related to automated liquidity transfer due to pending CBO or credit line decrease
- 16 Related to automated liquidity transfer due to pending CBO or credit line decrease
- 17 Related to automated liquidity transfer due to pending CBO or credit line decrease
- 18 Related to automated liquidity transfer due to pending CBO or credit line decrease
- 19 Related to automated liquidity transfer due to pending CBO or credit line decrease

Activity	Balance on MCA of Bank A	Liquidity reserved for CBOs	Non-reserved liquidity
	↑	↔	↑
Creation of one-time reservation = 200	750 ↔	200 ↑	550 ↓
Set-up of overnight deposit = 150	600 ↓	50 ↓	550 ↔
Submitting a "resetting to zero" reservation = 50	600 ↔	0 ↓	600 ↑

Table 48 - Usage of reserve for CBOs – numeric example

5.2.3.2 Floor/ceiling

5.2.3.2.1 Definition of floor/ceiling threshold

A floor is defined as a lower threshold of an amount of available liquidity (balance plus credit line) defined by the CLM Account Holder.

A ceiling is defined as an upper threshold of an amount of available liquidity defined by the CLM Account Holder.

The target amount is the amount up to which the balance (available liquidity) of an MCA

- | 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.

The floor/ceiling threshold manages the behaviour of CLM after the successful settlement of a payment (CBO) whenever the balance of the account undercuts the floor amount or exceeds the ceiling amount. Since this functionality is optional, it is up to the owner of the MCA (or another actor acting on behalf of the MCA owner) to define the floor/ceiling threshold in CRDM.

The owner of the MCA (or another actor acting on behalf of the MCA owner) can define a minimum ("floor") and maximum ("ceiling") liquidity amount for its MCA(s). The CLM Account Holder has the option to choose the behaviour of CLM once the balance is below the defined floor or above the defined ceiling amount. Two options are available.

1. CLM generates a notification to be sent to the owner of the MCA informing about the floor/ceiling breach (upon which the CLM Account Holder can take action).
2. CLM automatically generates an inter-service liquidity transfer to pull cash from the CLM Account Holder's RTGS DCA used for payments (where the floor is breached) or push cash to the CLM Account Holder's RTGS DCA used for payments (where the ceiling is breached).

5.2.3.2.2 Breach of floor/ceiling threshold - notification

If the CLM Account Holder chooses the first option, CLM generates and sends out a notification with the information that the available liquidity is below the floor or that the available liquidity is above the ceiling respectively:

- I in U2A (see CLM user handbook) or
- I in A2A mode (via [ReturnAccount \(camt.004\)](#) [▶ 355]; [Process floor and ceiling](#) [▶ 270]).

The notification is sent every time the threshold is undercut (floor) or exceeded (ceiling). However, CLM does not send the notification if -after passing the threshold- the MCA available balance remains consistently below the floor or above the ceiling.

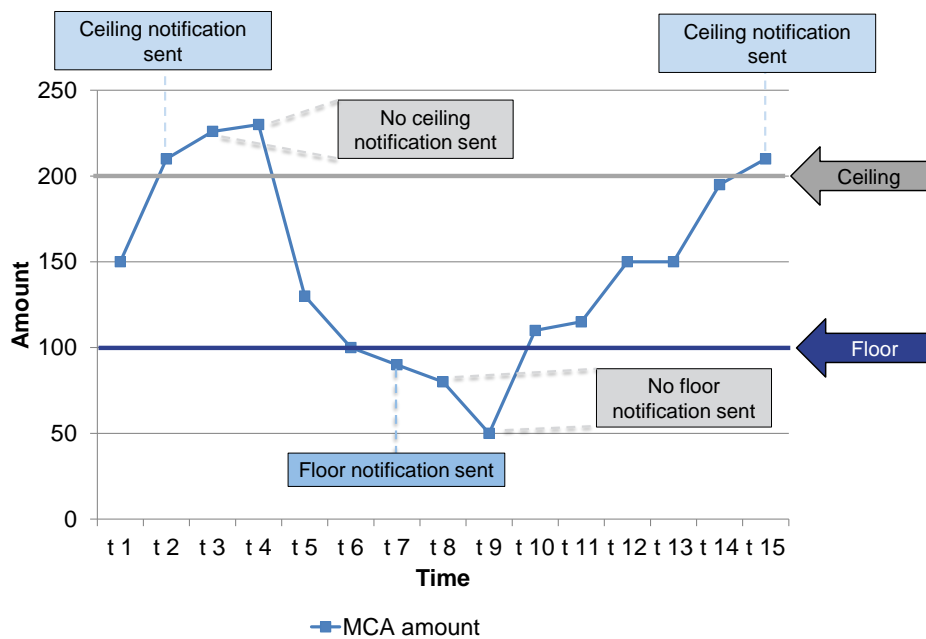


Figure 28 - Breach of floor/ceiling threshold – notification

5.2.3.2.3 Breach of floor/ceiling threshold - automatic liquidity transfer

If the CLM Account Holder chooses the second option, CLM creates and releases a rule-based inter-service liquidity transfer.

- | In case of a breach of the floor threshold a certain amount is pulled from the RTGS DCA and credited on the MCA.
 - The used RTGS DCA is linked to the MCA as defined in CRDM.
 - The amount to be transferred is the difference between the current available liquidity on the MCA and the predefined target amount, whereas the target amount can be different but equal or above the floor amount. If the available liquidity on the RTGS account is not sufficient, the liquidity transfer is partially settled in RTGS.
- | In case of a breach of the ceiling threshold a certain amount is pushed to the RTGS DCA and debited on the MCA.
 - The used RTGS DCA is the same as for the floor threshold, meaning it is linked to the MCA as defined in CRDM.

- The amount to be transferred to the RTGS DCA is the difference between the current available liquidity on the MCA and the predefined target amount.
- The target amount could be different but is below the ceiling amount.
- The target amount for ceiling is a different one compared to the target amount of the floor threshold.

After the successful execution of the inter-service liquidity transfer the available liquidity on the MCA is within the boundaries of the floor or ceiling amount again.

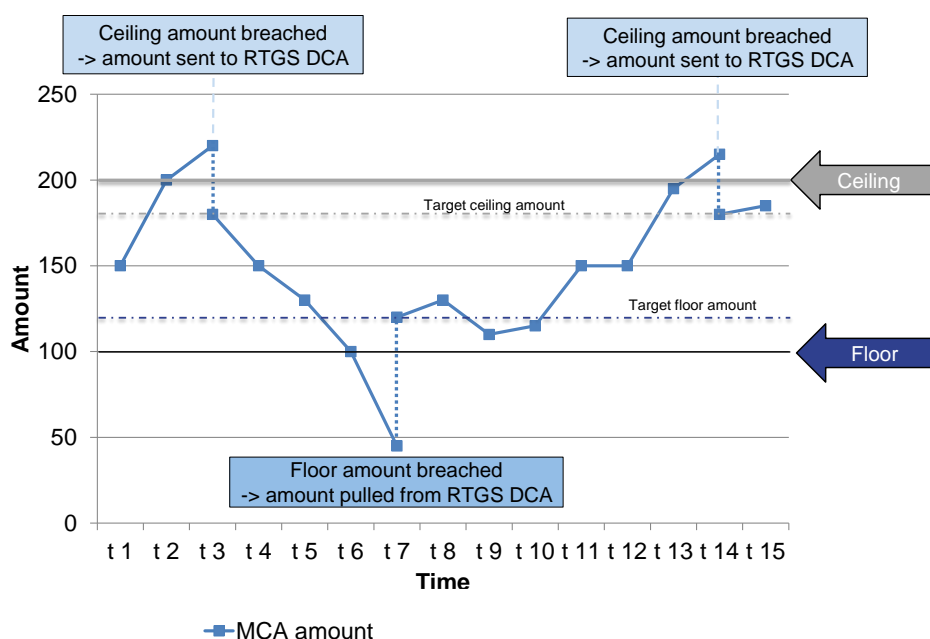


Figure 29 - Breach of floor/ceiling threshold - automatic liquidity transfer

5.3 Reserve management

5.3.1 Overview

CLM receives - automatically at the EoD – general ledger files containing account balances from the various settlement services and components (RTGS, T2S and TIPS) in order to manage minimum reserves and excess of reserve. Furthermore CBs can send information about the EoD balances of accounts held with them outside the TARGET Services to be included in the minimum reserve process (see chapter [Standing facilities - specific functions for CBs](#) [> 252]).

CLM will:

- verify the minimum reserve fulfilment

calculate the interest to be paid to MFI for minimum reserves

calculate the penalties related to the reserve requirements infringement to be submitted to the relevant CBs validation process

calculate negative interest on excess of reserve

notify the CBs on the minimum reserve fulfilment, due interest and possible penalties for the respective credit institutions

create automatically the related credit and debit instructions for minimum reserve fulfilment (the latter only after the CB validation process) and process them (at the end of the maintenance period)

automatically create the related credit and debit instructions for excess of minimum reserve and process them (at the end of the maintenance period)

The following set-ups are possible for holding minimum reserves.

1. Direct holding of minimum reserves

– account holders which manage the minimum reserve directly

2. Indirect holding of minimum reserves through an intermediary

– please see below for more information

3. Pool of reserve accounts of a MFI

– please see below for more information

Indirect holding of minimum reserves through an intermediary

CLM offers also the possibility of managing indirectly the reserve requirements and the excess reserve according to the “General documentation on Eurosystem monetary policy instruments and procedures“ and the Council Regulation (EC) No 1745/2003 of 12 September 2003.

On the basis of the list of MFIs that decide to fulfil minimum reserves indirectly and of the intermediaries selected for its management, CLM is able to verify the fulfilment of minimum reserves and to calculate the excess reserve. Whether a MFI is holding its minimum reserve directly or indirectly is stored in CRDM.

In case of indirect reserve management the total amount of the minimum reserves (intermediary + institutions managing their minimum reserves through the intermediary) is taken into account. However, only the balance of the intermediary is considered for the fulfilment of the minimum reserve requirement and for possible infringements at the end of the maintenance period. The intermediary is only debited in case of penalty.

Pool of reserve accounts of a MFI

Within CLM the so-called “pool of reserve accounts of a MFI“ can be used, which enables the fulfilment of reserve requirements for a group of CLM MCA holders (which are part of the same MFI).

In this case, the fulfilment of reserve requirements by the MFI is evaluated on the basis of the sum of balances of all the accounts (in CLM, RTGS, T2S and in TIPS) belonging to the pool, even if from a technical

point of view the minimum reserve of the MFI is linked only to a single predefined MCA indicated by the MFI leader (i.e. the account holder which is holding the reserve of the pool).

No consolidation is possible on a cross-border basis. At the end of the maintenance period the accrued interest is credited on the MCA associated to the minimum reserve indicated by the MFI leader.

The same MCA would be debited interest in case of a negative interest rate as well as potential infringement penalty, once validated by the relevant CB.

The balances of all participant accounts belonging to a pool are considered for the calculation of the excess of reserve, but only the leader MCA is debited in case of negative interest.

It is not possible for the single participants to have access to both functions “pool of reserve accounts of a MFI” and indirect reserve management. As a consequence participants belonging to the same MFI and availing themselves of the minimum reserve “pooling” functionality cannot make use of the indirect reserve management.

5.3.2 Collection of EoD balances

To be able to calculate the minimum reserve requirements CLM needs to collect the EoD balances of the accounts of all TARGET Services, i.e. DCAs of RTGS, T2S²⁰ and TIPS as well as the MCAs from CLM. The EoD balances are provided by the other services/components within the CLM EoD process. Only balances from accounts which are labelled in CRDM (see chapter [Reference data for parties used by CLM](#) [▶ 49]) as being relevant for minimum reserve are processed.

In addition to the balances provided by the EoD process, CLM has to include the additional balances which can be sent by CBs in a separate process (see chapter [Standing facilities - specific functions for CBs](#) [▶ 252]).

5.3.3 Daily calculations

Having received all EoD balances of the TARGET Services, CLM executes the following calculations for the current maintenance period²¹.

Accumulation of balances: Building the sum of all relevant EoD balances of the accounts to be included in the minimum reserve requirement.

20 Only applicable in case the mandatory cash sweep is not mandatory any longer. Currently, the cash sweep at EoD from T2S towards T2 is mandatory. A corresponding change request would be needed.

21 In case a balance or balances are missing, the “crisis management” decides about the way forward.

Moving average: Calculated as the arithmetic mean of the accumulated balances from the first business day of the current maintenance period until the day before the next business day (i.e. on Friday the moving average is calculated including Saturday and Sunday with Fridays balance).

Adjustment balance: In order to verify the fulfilment of the minimum reserve requirement, CLM compares the moving average with the minimum reserve requirement. In case the moving average is below the minimum reserve requirement, the adjustment balance is the difference between the moving average and the balance needed to fulfil the minimum reserve requirement at the end of the maintenance period.

The calculations above are done separately for each MFI.

Preconditions

To be able to manage the minimum reserve requirements, i.e. verifying the fulfilment and calculating the adjustment balance, the individual minimum reserve requirement numbers per MFI have to be submitted by the CB (see Chapter [Standing facilities - specific functions for CBs](#) [> 252])

Triggers and cut-off times

The daily calculations are started after the settlement of standing facilities and before the start of the new business day. They are triggered upon reception of all EoD balances.

5.3.4 Periodic calculations

At the end of the maintenance period, CLM calculates

- the interest to be paid to MFIs for the amounts up to the minimum reserve requirement according to the relevant interest rate.

- the penalties related to the reserve requirements infringement in case the running average during the maintenance period is lower than the minimum reserve requirement for an MFI.

- the excess of minimum reserve and the interest on excess of reserve according to the relevant interest rate.

For MFIs subject to minimum reserve requirements, the excess of reserve is the difference between the aggregated EoD balance running average and the minimum reserve requirements.

For MFIs not subject to minimum reserve requirements (with the exception of CBs), the excess of reserve is the aggregated EoD balance running average.

After verifying the minimum reserve fulfilment and the interest and penalty calculations, CLM sends a notification (camt.998) to the CBs on the minimum reserve fulfilment of the related MFIs.

Further CLM informs the CBs about the due interest and possible penalties for the pertaining parties by sending a camt.998 message.

With regards to penalties, a feedback from the CBs is required before the creation of payment orders by CLM.

5.3.5 Generate payment orders

At the end of the maintenance period CLM creates the following related credit and debit instructions.

Payment orders for minimum reserve fulfilment:

Based on the interest and penalties calculation, CLM creates the related payment orders for minimum reserve fulfilment, whereby the payment order with regards to penalties is only created after the CB validation process.

Payment orders for excess of minimum reserve:

Based on the interest calculation, CLM creates the related payment orders for excess of minimum reserve. In case of interest rate of 0,00 % no payment order is created.

In case of a positive overall interest calculation result, the CLM Account Holder's MCA is credited and the CB account is debited by creating a credit transfer pacs.009 FinancialInstitutionCreditTransfer (see chapter [Payments initiated by CB - credit transfer](#) [▶ 84]).

In case of a negative overall interest calculation result, the CLM Account Holder's MCA is debited and the CB account is credited by creating a direct debit pacs.010 FinancialInstitutionDirectDebit (see chapter [Payments initiated by CB - direct debit](#) [▶ 89]).

The value date of interest related payments is two business days after the end of the maintenance period.

5.4 Standing facilities management

5.4.1 Overnight deposit

5.4.1.1 Overview

The overnight deposit process is an element of the CLM standing facilities and breaks down into three parts:

- | setup of an overnight deposit
- | overnight deposit reverse transaction
- | overnight deposit reimbursement and interest calculation

CLM Account Holders with access to monetary policy operations can use the deposit facility to make overnight deposits with their national CBs.

To setup an overnight deposit, CLM Account Holders are able to transfer liquidity from their MCA to the relevant overnight deposit account.

Note: The owner of overnight deposit account to be set-up is the CB. A CB has to open a separate overnight deposit account per monetary counterparty using the overnight deposit functionality.

It is also possible to activate a reverse transaction in order to reduce the amount deposited in the overnight deposit account. This has to be initiated before the deadline for the usage of standing facilities. CLM calculates the interest to be paid on the overnight deposit and, at the start of the next business day, returns automatically the capital amount and credits the interest on the CLM Participant's MCA. In case of a negative interest rate, CLM calculates the interest to be paid by the CLM Account Holders on the overnight deposit and, at the start of the next business day, returns automatically the capital amount to CLM and debits the interest to be charged from the CLM Participant's MCA.

Note: For CBs outside the Eurosystem interests are always accumulated on a monthly basis. CLM calculates the accumulated interest at the end of a calendar month and posts it ten business days after the first business day of the following month (warehoused payment). The respective connected CB has the possibility to check the calculated interest and to cancel the warehoused payment if the calculation is not correct.

Preconditions

A participant wishing to initiate an overnight deposit needs to:

- | be a monetary counterparty
- | be eligible to the overnight deposit facility
- | have an active MCA
- | have dedicated overnight deposit account(s) set-up in CLM by the respective CB
- | for reverse transactions only: an overnight deposit for that business day has been set-up previously

Furthermore, a control mechanism is in place to verify that the total amount envisaged for each CB outside the Eurosystem is not exceeded.

Triggers and cut-off times

The setup and reversal of an overnight deposit can be initiated by

- | an overnight deposit or reverse transaction request sent by the CLM Account Holder in A2A or
- | manual input via U2A screen by the CLM Account Holder (or CB operator acting on behalf of the CLM Account Holder).

The reimbursement of deposited capital and calculation of interest is triggered by the start of the next business day. CLM triggers automatically the liquidity transfer for the repayment of the capital amount and the interest payment. Interest for non-Eurosystem CBs is processed differently.

It is possible for CLM Account Holders to set-up and/or to reverse an overnight deposit from the opening time of CLM (i.e. 19:00 and after overnight deposit, marginal lending reimbursement and interest calculation) until the general cut-off for the use of standing facilities (i.e. 18:15 with additional fifteen minutes on the last day of the reserve maintenance period) with the exception of the maintenance window.

Settlement principles

The following principles apply to the processing of liquidity transfer orders linked to overnight deposits.

- | Attempt to settle liquidity transfer immediately after its submission.
- | Liquidity transfer orders: are either settled completely or cancelled (no partial settlement).
- | Liquidity transfer orders are not queued.
- | Liquidity from RTGS-DCA(s) is used to supplement insufficient liquidity on the MCA.

5.4.1.2 Overnight deposit process

5.4.1.2.1 Setup overnight deposit

Message flow

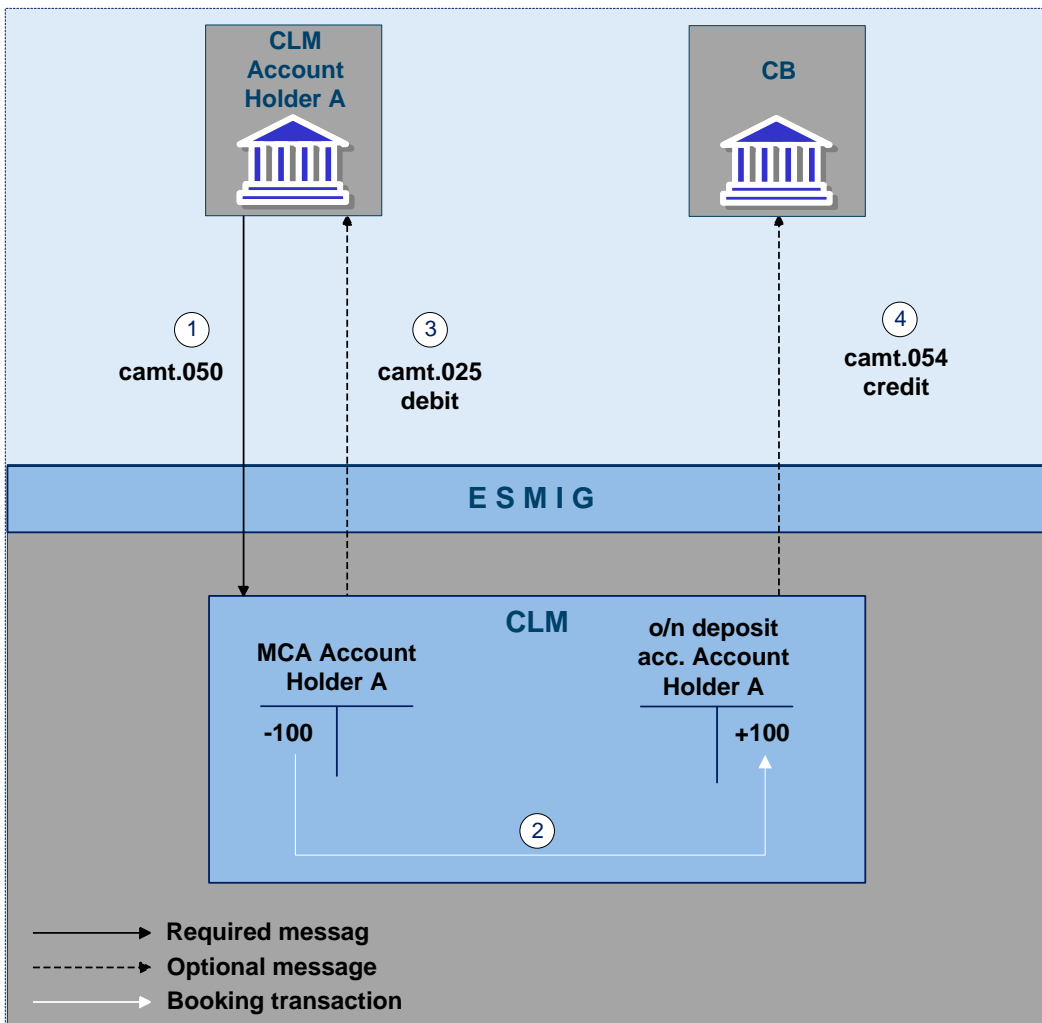


Figure 30 - camt.050 - setup overnight deposit

Process description

The process of setting up an overnight deposit in CLM consists of the following steps.

Step	Processing/between	Description
1	CLM Account Holder via ESMIG to CLM	The CLM Account Holder sends a camt.050 to CLM.
2	CLM	CLM credits the overnight deposit account at the CB and debits the MCA of the participant, if validations are positive.
3	CLM via ESMIG to CLM Account Holder	CLM sends a receipt (camt.025) to the CLM Account Holder.
4	CLM via ESMIG to CB	CLM sends an optional notification (camt.054 debit) to the CB.

Table 49 - Setup overnight deposit

Used messages

- | [Receipt \(camt.025\)](#) [▶ 397]
- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]

Participants are allowed to send multiple camt.050 to set-up overnight deposit. Each new instruction increases the deposited amount.

Technical validations

At the reception of an overnight deposit request, the CLM interface performs technical validations. For further details please refer to chapter [Rejection of liquidity transfer orders](#) [▶ 129]s.

After encountering the first negative validation result, the CLM continues to validate as far as possible and reports all negative results combined in a single reply message. CLM rejects the order not until performing all possible technical validations. In case of a negative result of the technical validation the request is rejected and a negative notification (admi.007) is sent to the instructing CLM Account Holder.

If all technical validations are passed without any error, the request is further processed, i.e. business validations.

Business validations

Once the technical validations are positively completed, the overnight deposit request proceeds the business validations. For further details please refer to chapter [Rejection of liquidity transfer orders](#) [▶ 129]s.

If any of the business validations fails, the overnight deposit request is rejected and a negative receipt (camt.025) is sent to the instructing CLM Account Holder.

5.4.1.2.2 Overnight deposit reverse transaction

Once CLM Account Holders have sent a set-up overnight deposit order, it is possible for the CLM Account Holder (before the deadline for the usage of standing facilities) to activate a reverse transaction in order to reduce the amount deposited in the overnight deposit account.

Message flow

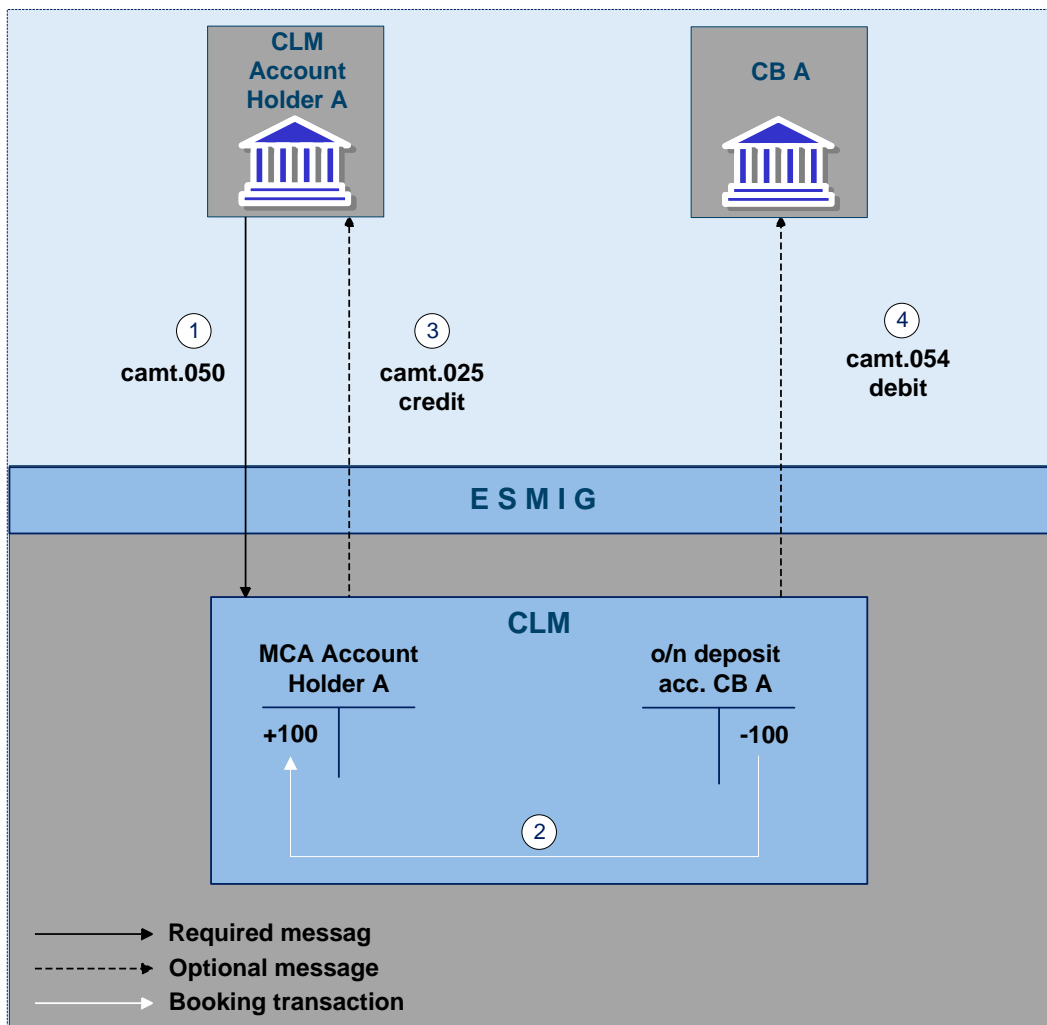


Figure 31 - camt.050 - reverse overnight deposit

Process description

The process of reversing an overnight deposit in CLM consists of the following steps.

Step	Processing in/between	Description
1	CLM Account Holder via ESMIG to CLM	The CLM Account Holder sends a camt.050 to CLM.
2	CLM	CLM debits the overnight deposit account of the CB and credits the MCA of the participant if business validations are positive.
3	CLM via ESMIG to CLM Account Holder	CLM sends an optional receipt (camt.025) to the CLM Account Holder.
4	CLM via ESMIG to CB	CLM sends an optional notification (camt.054 debit) to the CB.

Table 50 - Reverse overnight deposit

Used messages

- | [Receipt \(camt.025\)](#) [▶ 397]
- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]

Technical and business validations

Technical and business validations check that a corresponding overnight deposit is set-up previously. Apart from this additional check the same technical and business validations apply as described in chapter [Rejection of payments](#) [▶ 93].

5.4.1.2.3 Overnight deposit reimbursement and interest calculation

At start of the next business day CLM calculates the interest on the overnight deposit and automatically books the capital amount and the interest amount to the participant’s MCA ²².

22 Interest calculation and payment for non-Eurosystem CBs is done at the end of the calendar month.

Message flow

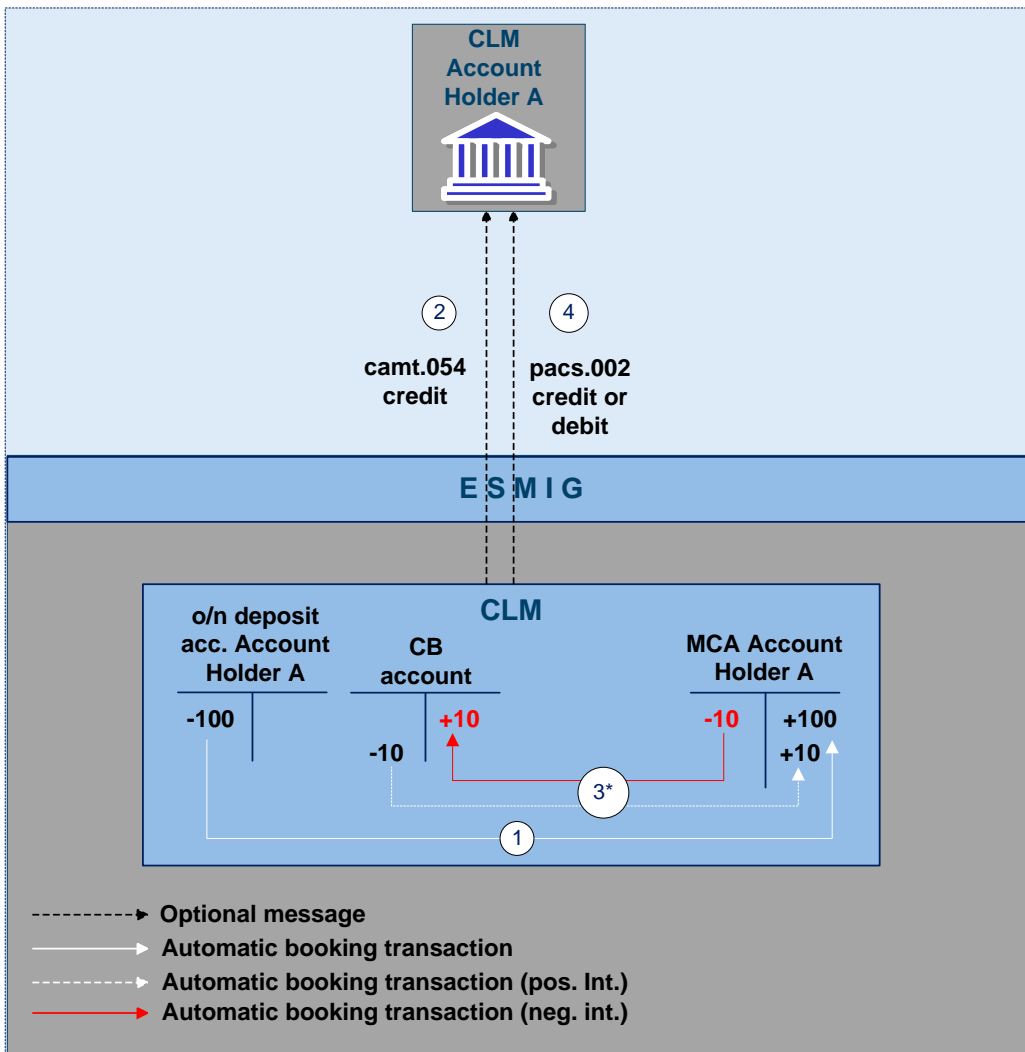


Figure 32 - Reimburse overnight deposit and booking of interest

Process description

The process of overnight deposit reimbursement and interest calculation in CLM consists of the following steps.

Step	Processing in/between	Description
1	CLM	CLM automatically debits the overnight deposit account of the CB with the deposited amount and credits the MCA of the CLM Account Holder.
2	CLM via ESMIG to CLM Account Holder	CLM sends an optional notification (camt.054 credit) to the CLM Account Holder.
3	CLM	CLM automatically debits the CB account and credit the MCA of the CLM Account Holder, if the overnight deposit rate is positive ²³ . CLM automatically credits the MCA of the CB and debit the MCA of the CLM Account Holder, if the overnight deposit rate is negative ²⁴ .
4	CLM via ESMIG to CLM Account Holder	CLM sends an optional notification (camt.054 credit or debit) to the CLM Account Holder.

Table 51 - Reimburse overnight deposit

Used messages

- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]

Technical and business validations

The same validation processes as for setup of overnight deposits apply (see “Technical validation” in chapter [Setup overnight deposit](#) [▶ 150]).

5.4.2 Marginal lending "on request"

5.4.2.1 Overview

Process context

Eligible monetary policy counterparties can use the marginal lending on request facility to obtain overnight liquidity from CBs at a pre-specified interest rate against eligible assets.

23 CLM generates an interest payment even if the overnight deposit rate is zero.

24 CLM generates an interest payment even if the overnight deposit rate is zero.

Settlement principles

The following principles apply to cash transfers linked to marginal lending on request, to its reimbursement and to its interest payments.

- | cash transfers all have the same priority
- | attempt to settle a cash transfer immediately after its submission
- | offsetting mechanisms to save liquidity are not required
- | cash transfers may be cancelled as long as they are not executed
- | cash transfers, which cannot be settled immediately, are queued (as CBOs they are placed on top of the queue)
- | among CBOs, cash transfers in the queue are processed according to the FIFO-principle

5.4.2.2 Setup marginal lending "on request"

Preconditions

To set-up a marginal lending on request, a party needs to:

- | be a participant in CLM,
- | be eligible to the marginal lending facility,
- | have an MCA in CLM and
- | the corresponding CB has to open a dedicated marginal lending account in CLM – one for each CLM Account Holder eligible for marginal lending.

In addition all parties'EoD balances need to be available to CLM.

Triggers and cut-off times

The marginal lending on request is requested by the participant at its CB. The CBs collateral management system either sends an A2A message to CLM or a CB operator enters it manually via U2A screen.

It is possible for CLM Account Holder to request a marginal lending from the opening time of CLM (i.e. 19:00 and after overnight deposit and marginal lending reimbursement and interest calculation) until the general cut-off for the use of standing facilities (i.e. 18:15 with additional 15 minutes on the last day of the reserve maintenance period) with the exception of the maintenance window. CBs can set-up a marginal lending on request until 18:40 (with additional 15 minutes on the last day of the reserve maintenance period).

Process flow

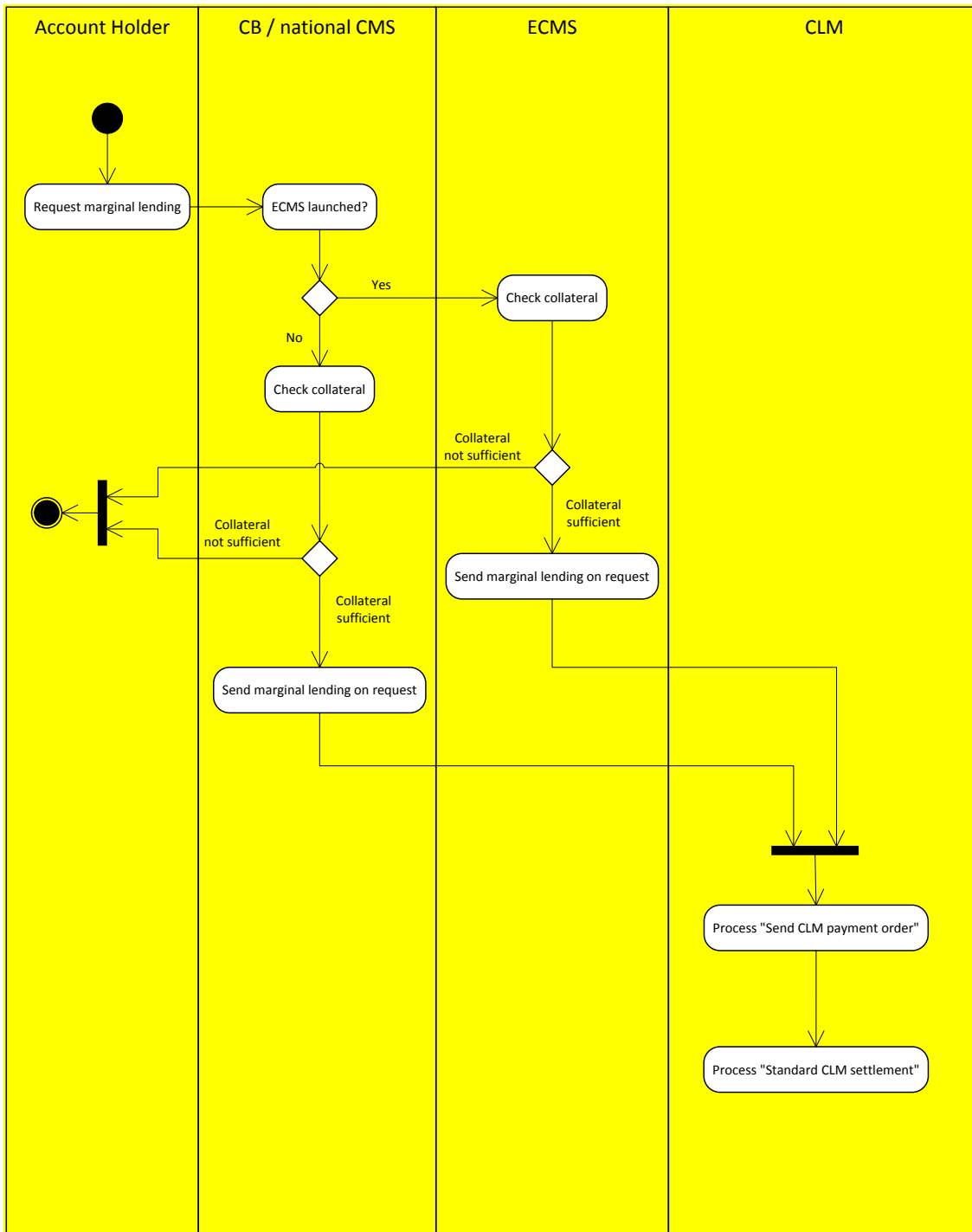


Figure 33 - Marginal lending “on request”

Process description

Step	Actors	Description
1	CLM Account Holder/CB	CLM Account Holder requests a marginal lending at its CB.

Step	Actors	Description
2	CB/local collateral management	CB checks available collateral with their local collateral management (system).
3	Local collateral management/CLM	The local collateral management send an marginal lending “on request” order via A2A (camt.050) or U2A (screen entry) to CLM
4	CLM	CLM starts the process “Send payment order” and “Standard settlement process”

Table 52 - Marginal lending “on request”

Note: The collateral management function is managed outside of T2 under the responsibility of the respective CB. CLM only checks the formal correctness of the message (A2A or U2A) sent by the collateral management (system).

Used messages

- ▶ [PaymentStatusReport \(pacs.002\)](#) [487]
- ▶ [FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [491]
- ▶ [BankToCustomerDebitCreditNotification \(camt.054\)](#) [442]
- ▶ [ReceiptAcknowledgement \(admi.007\)](#) [347]

5.4.2.2.1 Before launch of ECMS

The marginal lending on request is initiated by CLM Account Holder (or a user authorised by the CLM Account Holder) via their CBs collateral management system. Before the Eurosystem Collateral Management System (ECMS) is launched, every CB is using its own national collateral management, which can be a system managed by the CB or a service done by a collateral manager on behalf of the respective CB. Cash transfers for reimbursing the marginal lending amount are sent from the national collateral management to CLM.

5.4.2.2.2 After launch of ECMS

Note: Details about ECMS and the roles in different processes are still subject to discussions.

ECMS replaces the collateral management of the national CBs. Consequently a marginal lending “on request” submitted by a CLM Account Holder to its CB is handled by ECMS and marginal lending “on request” messages are sent to CLM by ECMS. The collateral manager can send the message to CLM via A2A or

enter it via a screen in U2A mode ²⁵. From the point on when the request is submitted to CLM, the process is carried on as described in chapter [Setup marginal lending "on request"](#) [▶ 156].

5.4.2.3 Marginal lending reimbursement and interest calculation

For marginal lending “on request” reimbursement and interest calculation are done automatically by CLM on the following business day. For the interest calculation CLM uses the relevant marginal lending interest rate. In case of multiple marginal lending operations for the same participant, the interest is calculated on the basis of the aggregated marginal lending amount.

The CB operator can decide to exclude marginal lending on request from the interest calculation, i.e. CLM in this case does not calculate interest for marginal lending on request.

Preconditions

The reimbursement and interest payment requires an existing marginal lending on request transaction from the previous business day.

Triggers and cut-off times

As of the start of provisioning of liquidity phase on the following business day (i.e. 19.00) CLM calculates the interest to be paid on marginal lending on request and automatically posts the capital amount and interest.

Process Flow

See chapter 10.5.1 “Reimburse marginal lending”

Used Messages

- | [Receipt \(camt.025\)](#) [▶ 397]
- | [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]
- | [PaymentStatusReport \(pacs.002\)](#) [▶ 487]
- | [FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491]

Expected results

The liquidity transfer linked to the reimbursement of a marginal lending on request leads to the debit on the CLM Participant’s MCA and the credit on the marginal lending account.

²⁵ U2A is mainly envisaged for contingency situations.

The interest payment order debits the CLM Account Holder’s MCA and credits the CB account.

Note: CLM generates an interest payment even in case the marginal lending rate is zero.

5.4.2.3.1 Before launch of ECMS

Before ECMS is launched, every CB is using its own national collateral management, which can be a system managed by the CB or a service done by a collateral manager on behalf of the respective CB. Cash transfers for reimbursing the marginal lending amount are created by CLM.

Process description

Step	Actor	Description
1	CLM	Upon the start of the EoD phase, CLM triggers the reimbursement and the interest calculation for marginal lending on request.
2	CLM	Create payment messages to reimburse and to post interest for marginal lending on request
3	CLM	Debit CLM Account Holder’s MCA and credit marginal lending account with reimbursement amount
4	CLM	Debit CLM Account Holder’s MCA and credit CB account with interest for marginal lending on request
5	CLM	Send out (optional) DebitCreditNotifications to CB and to the CLM Account Holder

Table 53 - Process steps of marginal lending reimbursement and interest calculation

Note: The collateral management function is managed outside of T2 under the responsibility of the respective CB. CLM only checks the formal correctness of the message (A2A or U2A) sent by the collateral management.

5.4.2.3.2 After launch of ECMS

Note: Details about ECMS and the roles in different processes are still subject to discussions.

ECMS replaces the collateral management of the national CBs. Consequently the reimbursement of a marginal lending “on request” is handled by ECMS and payment orders to reimburse the marginal lending amount and to post the interest are sent to CLM by ECMS. The collateral manager can send the message via A2A or enter it via a screen in U2A mode ²⁶. From the point on when the request is submitted to CLM, the process is carried on as described in chapter [Setup marginal lending "on request" \[p 156\]](#).

²⁶ U2A is mainly envisaged for contingency situations.

Process description

Step	Actor	Description
1	ECMS	Upon the start of the EoD phase, ECMS triggers the reimbursement and the interest calculation for marginal lending on request.
2	ECMS	Create payment message to reimburse marginal lending on request and forward it to CLM for further processing
3	ECMS	Create payment to post interest for marginal lending on request and forward it to CLM for further processing
4	CLM	Debit CLM Account Holder's MCA and credit marginal lending account with reimbursement amount
5	CLM	Debit CLM Account Holder's MCA and credit CB account with interest for marginal lending on request
6	CLM	Send out (optional) DebitCreditNotifications to CB and to the participant

Table 54 - Process steps of marginal lending reimbursement and interest calculation

Note: The collateral management function is managed outside of T2 under the responsibility of the respective CB. CLM only checks the formal correctness of the message (A2A or U2A) sent by the collateral management.

5.4.3 Automated marginal lending

5.4.3.1 Overview

The automated marginal lending facility is an instrument to transform an intraday credit into an overnight credit at the end of the business day in case the aggregated balance of the CLM Account Holder is negative. The aggregated balance is the sum of all balances of the participant's DCA's and MCA's. The process is initiated by CLM as part of the EoD process.

Preconditions

For the set-up of an automated marginal lending, a party needs to

- | be a CLM Account Holder,
- | have an MCA in CLM and
- | be eligible to the marginal lending facility.

Moreover, the corresponding CB has to open a marginal lending account in CLM – one for each CLM Account Holder eligible for marginal lending.

Before initiating the process, CLM needs to make sure that all EoD balances of the other services and components are available.

Triggers and cut-off times

The automated marginal lending process is triggered by CLM if the aggregated balance of the party is negative. It takes place during the EoD processing after the cut-off for CBs for standing facilities (i.e. 18:40).

5.4.3.2 Automated marginal lending process

5.4.3.2.1 Process automated marginal lending

The process starts after CLM has received the general ledger files from all other services and components and the aggregated EoD balance is calculated for the eligible monetary policy counterparty, if he is allowed to access the marginal lending facility. In case the aggregated balance of the eligible monetary policy counterparty is negative, CLM initiates the automated marginal lending to cover the negative balance.

Process flow

Please refer to chapter [Process automated marginal lending](#) [▶ 287].

Used messages

- | [PaymentStatusReport \(pacs.002\)](#) [▶ 487]
- | [FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491]
- | [ReturnAccount \(camt.004\)](#) [▶ 355]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]

5.4.3.2.2 Marginal lending reimbursement and interest calculation

For automated marginal lending the reimbursement and interest calculation are done automatically by CLM on the following business day. For the interest calculation CLM uses the relevant marginal lending interest rate. In case of multiple marginal lending operations for the same participant, the interest is calculated on the basis of the aggregated marginal lending amount.

Preconditions

The reimbursement and interest payment requires an existing automated marginal lending transaction from the previous business day.

Triggers and cut-off times

As of the start of provisioning of liquidity phase on the following business day (i.e. 19.00) CLM calculates the interest to be paid on marginal lending and automatically posts the capital amount and interest to the respective accounts.

Process Flow

Please refer to Chapter [Reimburse marginal lending](#) [▶ 292].

Used messages

- | [PaymentStatusReport \(pacs.002\)](#) [▶ 487]
- | [FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491]
- | [ReturnAccount \(camt.004\) - specific for CBs](#) [▶ 518]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]

Expected results

The connected payment linked to the reimbursement of an automated marginal lending leads to the debit on the CLM Account Holder's MCA and the credit on the marginal lending account. The connected payment leads to a concurrent update (increase) of the CLM Account Holder's credit line.

The interest payment debits the CLM Account Holder's MCA and credits the CB account.

Note: CLM generates an interest payment even in case the marginal lending rate is zero.

5.5 Information management for CLM

5.5.1 CLM status management

5.5.1.1 Concept

CLM informs its CLM Actors of the processing results. This information is provided to the CLM Actors via a status reporting which is managed by the status management. The communication of status to CLM Actors is complemented by the communication of reason codes in case of negative result of a CLM process (e.g. validation failure notifications).

5.5.1.2 Overview

The status management process manages the status updates of the different instructions existing in CLM in order to communicate these status updates through status advice messages to the CLM Actors throughout the lifecycle of the instruction. Status information on push basis is only available in A2A mode. Respective status advice messages are pushed via store-n-forward network service.

The status management handling also provides the reason codes to be sent to CLM Actors in case of negative result of a CLM 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 CLM Actors with information about the situation of this instruction with respect to a given CLM process at a certain point in time.

Since each instruction in CLM can be submitted to several processes, each instruction in CLM may have several statuses. However, each of these statuses 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 CLM. Consequently, the status featuring each instruction depends on the considered instruction type.

The following sections provide:

- I the generic principles for the communication of statuses and reason codes to CLM Actors
- I the list of statuses 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 CLM Actors

CLM Actors can query the status values and reason codes of their instructions (e.g. cash transfers, tasks, reference data updates) during the day.

The status can be classified in the following two types, common to all types of instructions.

- I "Intermediate status" - in general an instruction has 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 CLM. With each step in the process of the instruction the status changes until a final status is reached. Further status updates are communicated to the CLM Actors if reached.
- I "Final status" - this is the last status of an instruction (i.e. the status of an instruction when processing ends). At a point in time, any instruction in CLM reaches a final status and all respective processes are completed.

For some status updates mandatory information is provided. For other status updates, the status management process informs the CLM Actor of the status change by means of the sending of status advice messages (according to their message subscription configuration – please refer [Messaging](#) [▶ 59]).

Statuses and status values in CLM

As previously mentioned, the statuses of an instruction depend on the considered instruction type. The following paragraphs provide the list of statuses and status values. None of the statuses are stored for queries.

CLM component statuses are:

- | CLM file statuses
- | CLM message statuses
- | cash transfer statuses
- | task queue statuses.

CLM file statuses

CLM file statuses indicate the status of the file in CLM. There are the following statuses.

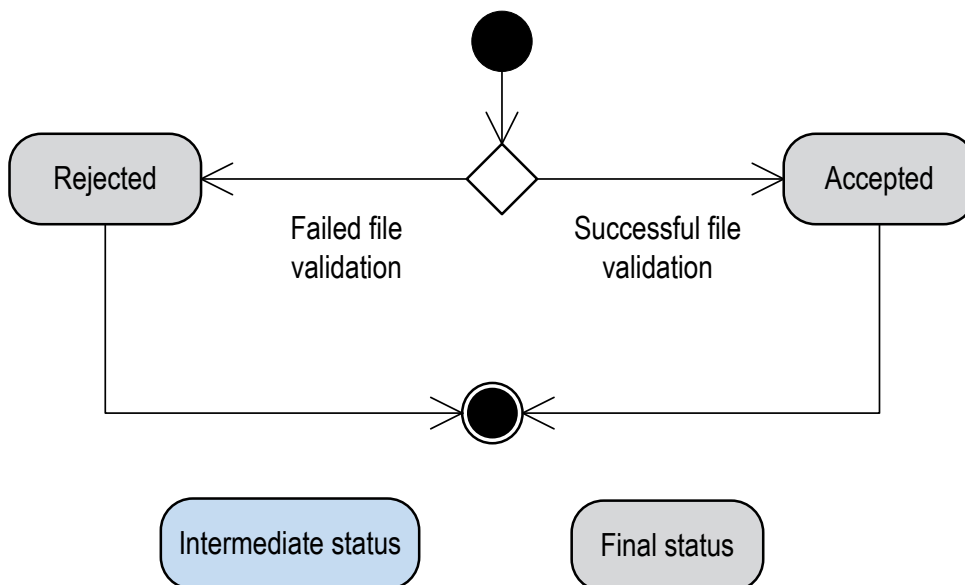


Figure 34 - File state diagram

Status value	Definition	Direction	Transition possible to status	Intermedi-ate/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 55 - CLM file statuses

CLM message statuses

CLM message statuses indicate the status of the message in CLM. They are the following statuses.

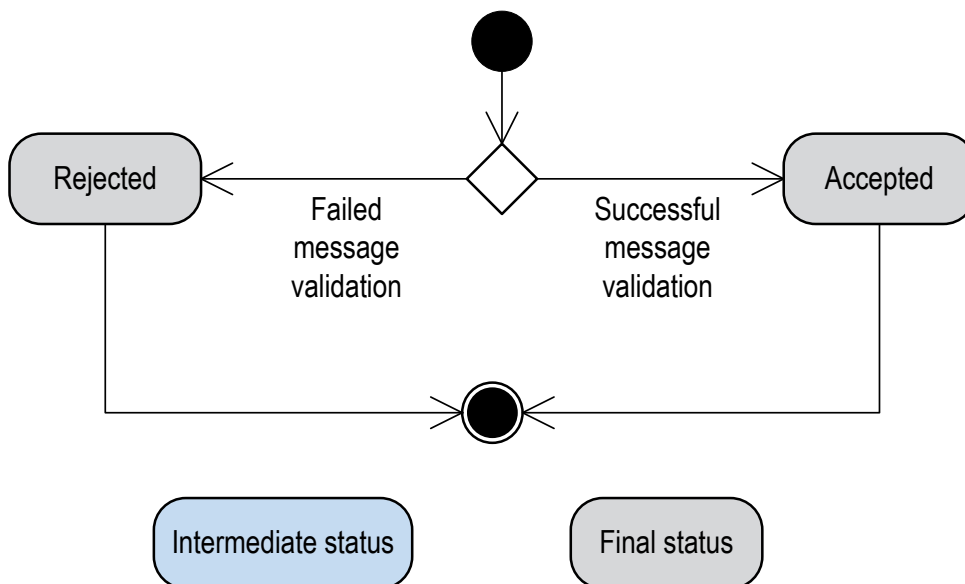


Figure 35 - Message state diagram

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 56 - CLM message statuses

Cash transfer statuses

Cash transfer statuses indicates the status of the cash transfer in CLM. They are the following statuses.

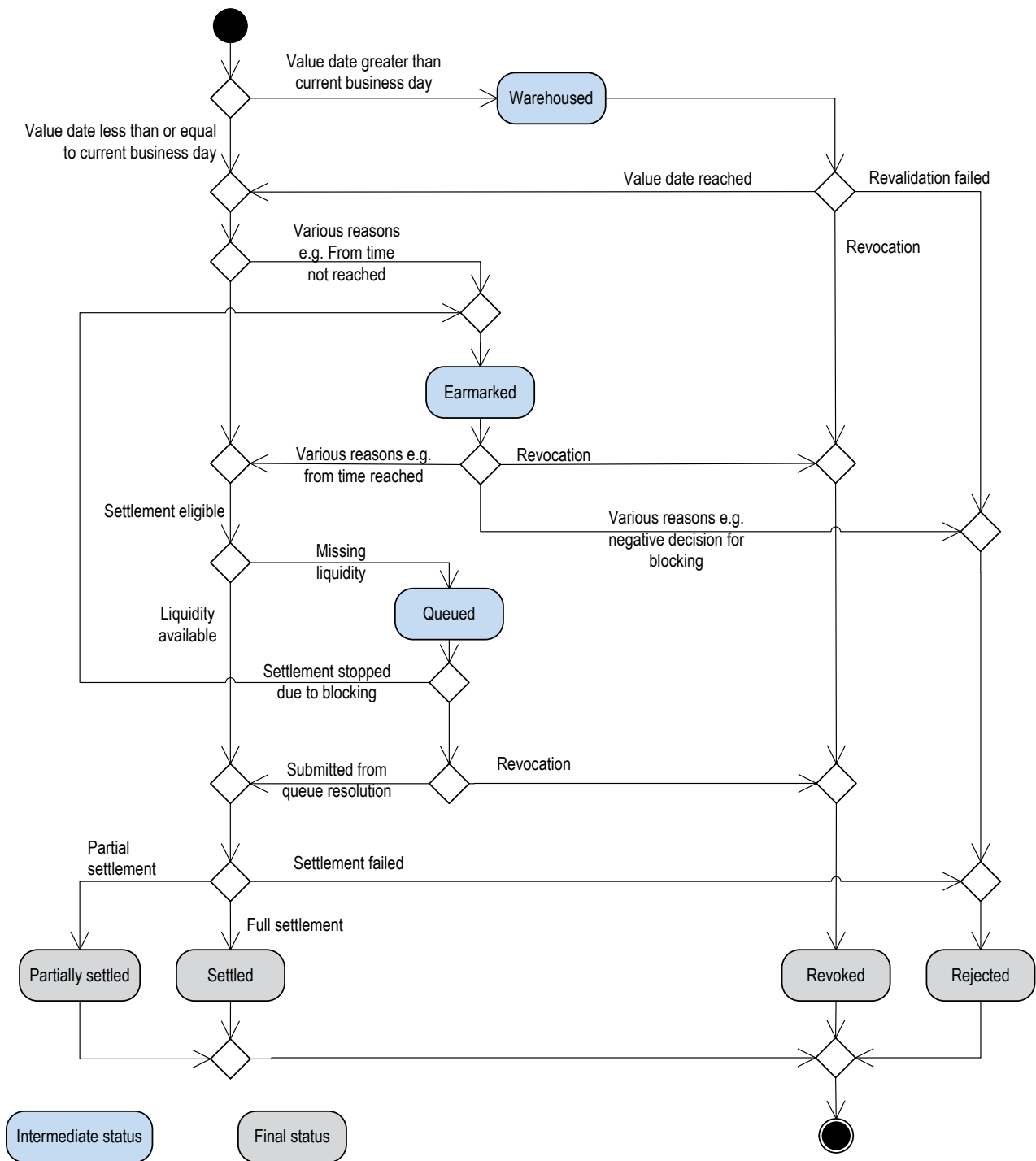


Figure 36 - Inbound CLM message state diagram

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 is forwarded to the processing at the start of the business day - from then on they are 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 accounting stopped due to blocking pending decision on blocking waiting for end of cycle 	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	-

Status value	Definition	Transition possible to status	Intermediate/final status	Reported via status notification to the sender
Partially settled	Status of cash transfer after settlement with an amount lower than ordered. For business cases where the remaining (unsettled) amount should be settled the service creates a new cash transfer.	-	Final	Mandatory
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 57 - Cash transfer statuses

Task queue order statuses

Task queue order statuses indicates the status of the task queue order in CLM. There are the following statuses.

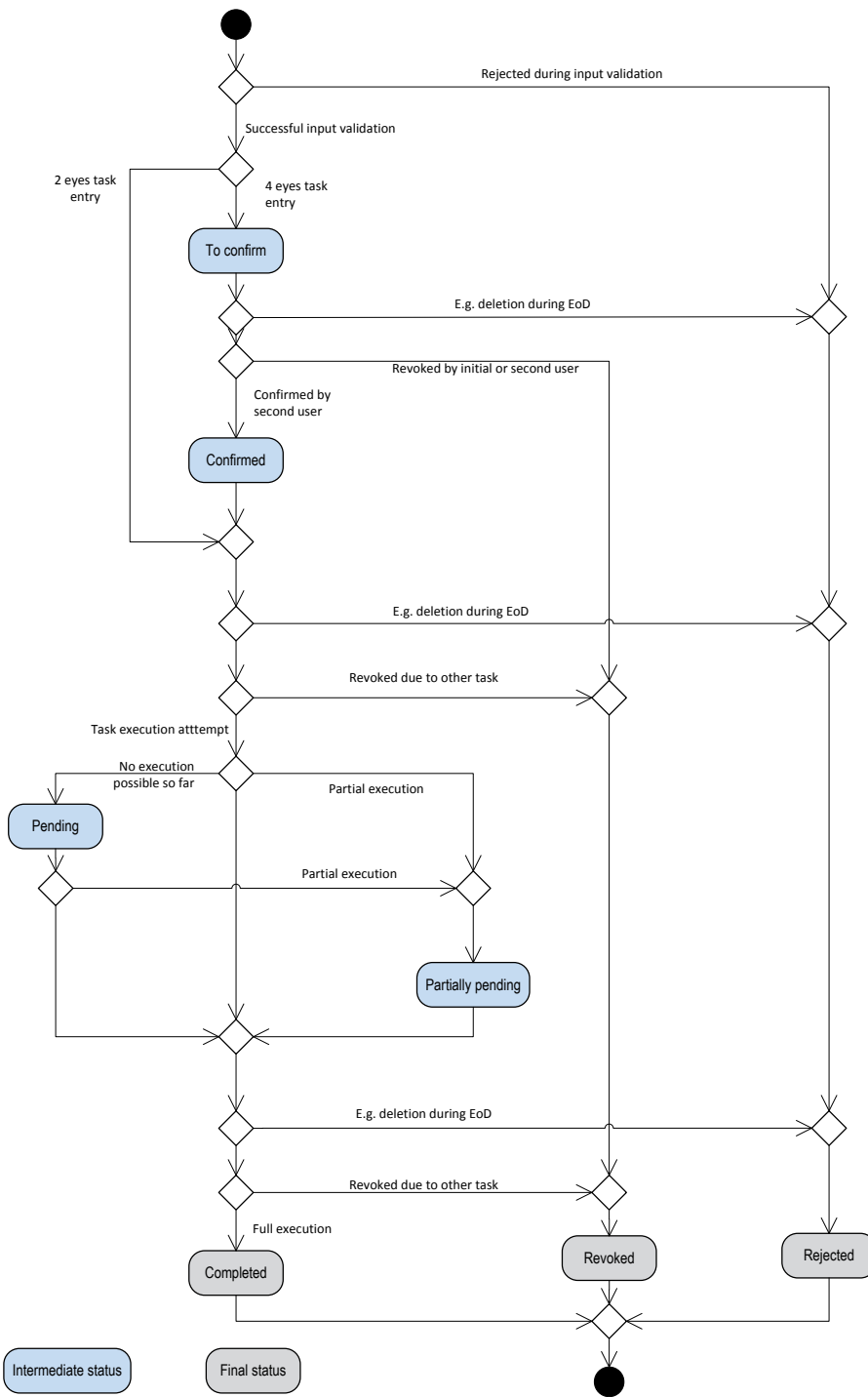


Figure 37 - Task queue order state diagram

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 is 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.	Confirmed, 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 is be updated and further processed, if the preconditions for the pending status (eg liquidity increase) are changed. Tasks with status "pending" can	Partially pending, completed, revoked, rejected	Intermediate	-

Status value	Definition	Transition possible to status	Intermediate/final status	Reported via status notification to the sender
	only be revoked via a new task.			
Partially pending	A task should be stored with status “partially pending” if the user's order cannot be processed completely (e.g. an increase of reservation cannot be executed completely because of lack of liquidity). The order is processed as far as possible. The task is updated and further processed, if the pre-conditions for the “partially pending” status (e.g. liquidity increase) are changed.	Completed, revoked, rejected	Intermediate	-

Status value	Definition	Transition possible to status	Intermediate/final status	Reported via status notification to the sender
Revoked	Status based on an action by the user to prevent the processing due to four-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 cash transfer) does not have to be final. To this task queue status a time stamp is added.	-	Final	Mandatory

Table 58 - Task queue order statuses

Tasks with status “pending” can only be revoked via a new task.

5.5.2 CLM report generation

5.5.2.1 Concept

CLM provides the possibility to create the predefined report “statement of account” periodically. The CLM component triggers the generation of the “statement of account” report based on the reference data configu-

ration. It is only foreseen at the business event EoD. The report is not created intraday. Depending on the CLM 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) [425]

The respective business process is described in chapter [Receive report](#) [▶ 303].

5.5.2.2 Overview

The report "statement of account" includes information on one single CLM cash account of a CLM Actor. It is not possible to receive one combined "statement of account" for more than one CLM 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 are booked on the CLM cash account and balance information of the current business day.

It is provided as a complete report, i.e. no delta version is offered.

Reports 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 CLM does not create reports automatically. To initiate the creation of a report, the report receiver has to configure the report in advance. The configuration is done via the GUI for the reference data, which is described in the CLM 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 EoD of the CLM component after finalisation of booking processes [business event "EoD"] – please see [End of day](#) [▶ 73]. A new report configuration can be set-up for the next business day at the earliest. 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 CLM

A generated report is available for download until it is replaced by a new 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 CLM. In A2A mode CLM 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 CLM 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 65 - [Report configuration](#) [▶ 188]) for the set-up of a report.

Parameter	Mandatory/optional	Possible values	Further information
Report type	Mandatory	Statement of accounts	
Concerned account	Mandatory	CLM Cash Account	
Possible recipient of a report	Mandatory	CLM Actor	
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 59 - Parameters for the set-up of a report

Concerned account

Each report provides information on a certain scope of data. The data scope is indicated by the CLM cash account for which it is configured. The feature is available for all CLM cash account types.

The concerned account has to be specified, when the report is configured for the first time. It is necessary to store one configuration per CLM 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 CLM 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 CLM cash account or by the technical address defined for the other authorised party (see chapter [Routing](#) [▶ 40]).

If a recipient does not wish to receive a report directly after its creation but to request it afterwards, this CLM 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 CLM user handbook chapters related to report configuration setup.

5.5.3 Query management for CLM, CRDM, scheduler and billing

5.5.3.1 Concept for CLM

Queries are provided by CLM 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 CLM. These are answered on the basis of the latest data available.

For requests on CLM 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 [Information services](#) [▶ 299].

5.5.3.2 Overview for CLM

CLM 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 CLM Actors.

They can send query requests to CLM 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 maintenance window is handled by using timeout management. In case the network interface is closed the NSP informs the authorised submitting actor about the closure of the real-time channel.

5.5.3.3 Query management process for CLM

Initiating queries for CLM

In order to obtain the desired information the submitting actor needs to submit a query request to CLM. For the communication with CLM in A2A mode all query and response messages are set-up as XML messages compliant with the ISO20022 standard. For the communication with CLM 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 CLM user handbook for U2A mode and in chapter [Dialogues and processes](#) [▶ 306] with link to MyStandards for A2A mode.

Query type	Initiation via GUI (U2A mode)	Initiation via XML message (A2A mode)
Account statement query	X	X
Audit trail for CLM query	X	X
Available liquidity CLM query	X	X
Available liquidity overall query	X	-
Broadcast query	X	-
Business case query	X	-
Cash transfer query	X	X
Current reservations query	X	X
Event query	X	X
File query	X	-
Message query	X	-
Minimum reserve query	X	X
System time query	X	X
Task queue query	X	-

Table 60 - Initiating queries for CLM

The different types of queries in CLM are static regarding the set of selection parameters, which can be mandatory, optional or conditional.

Preconditions for successful processing of queries

CLM validates the plausibility of search criteria that were specified by the submitting actor. In addition, CLM ensures that the submitting actor of the query request is allowed to initiate the query and to retrieve the requested data 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 CLM 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 CLM 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, CLM 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 sends the query via U2A mode, the response is given to the submitting actor in U2A mode. The U2A dialogue is described more in detail in the CLM user handbook.

If the submitting actor sends the query via A2A mode, the response is given to the same submitting actor in A2A mode. The CLM does not allow the routing of the query response to a dedicated technical address.

Parameter synthesis

No specific query configuration from the submitting actor is needed.

6 Overview of used common components in CLM 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](#) [▶ 214]). 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](#) [▶ 181]).

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](#) [▶ 181] presents some basic concepts (e.g. user, privilege, role and data scope) related to access rights management. On this basis, chapter [Access rights configuration](#) [▶ 196] illustrates all the available options for the configuration of access rights. Finally, chapter [Access rights configuration process](#) [▶ 205] 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](#) [▶ 193], as well as the concept of system entity, which is introduced in chapter [Data scope](#) [▶ 194].

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 ²⁷	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

²⁷ 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 61 - 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 62 - 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 63 - 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 64 - 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 65 - 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 66 - 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 67 - TIPS functions

Privilege	User function	Data scope
DMT access	n/a	n/a

Table 68 - Other

See chapter [Configuration of privileges](#) [▶ 196] 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](#) [▶ 203] 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](#) [▶ 196] 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 ²⁸. All the other reference data objects are linked to a party. For example:

²⁸ 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.

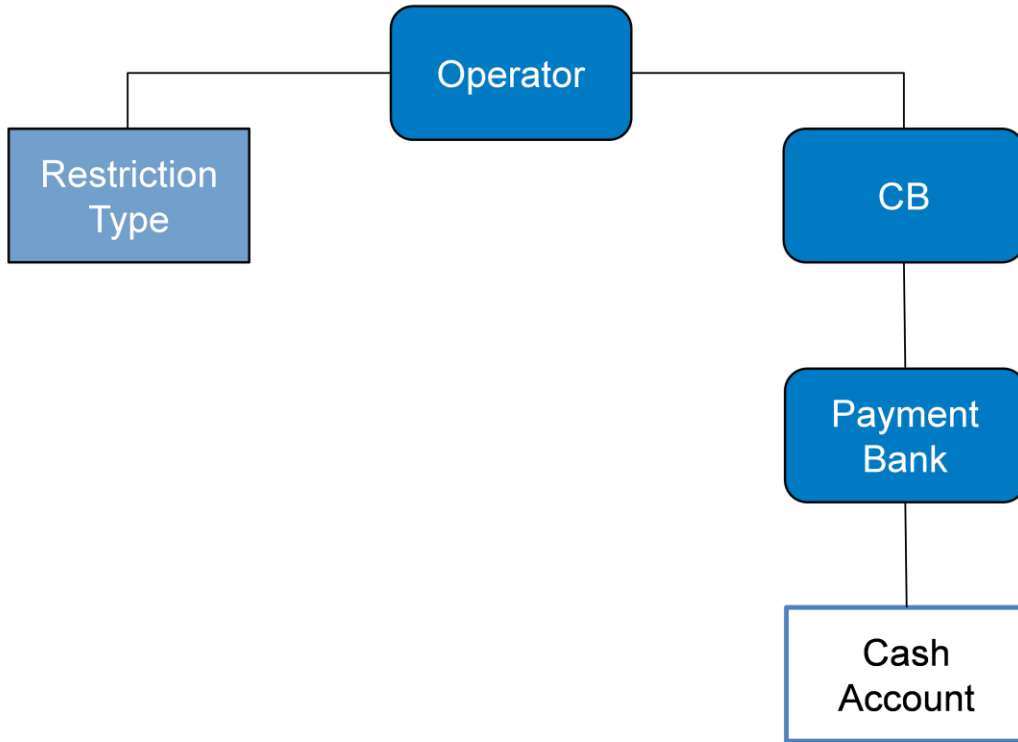


Figure 38 - Common reference data objects and the hierarchical party model

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 entity²⁹
- | 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 scope³⁰.

29 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.

30 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 69 - User privileges (data scope)

The following diagram shows the data scopes stemming from this access rights configuration for the three users.

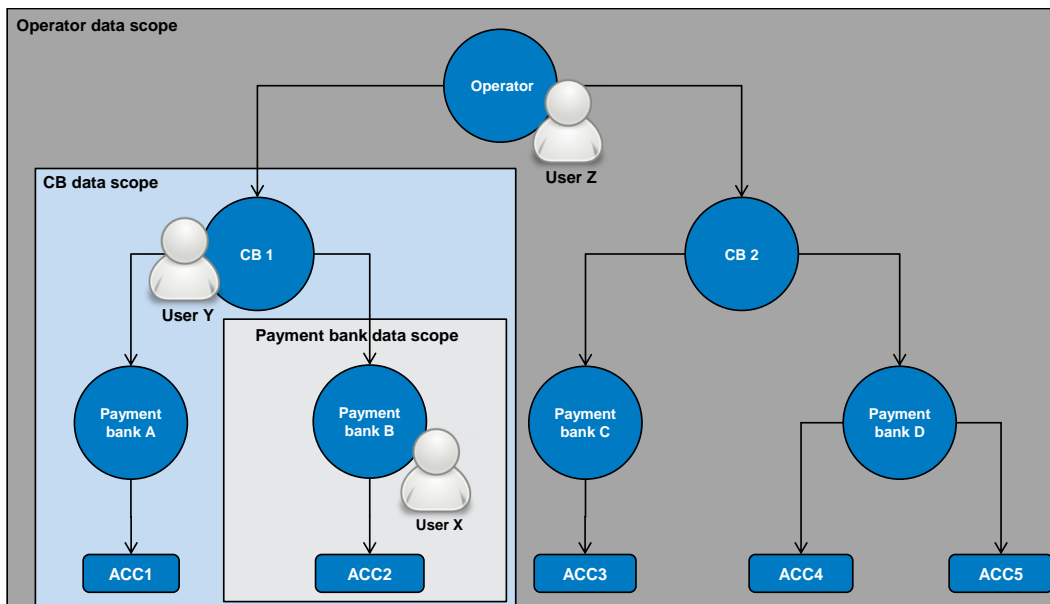


Figure 39 - Data scopes

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](#) [► 194]). 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.³¹

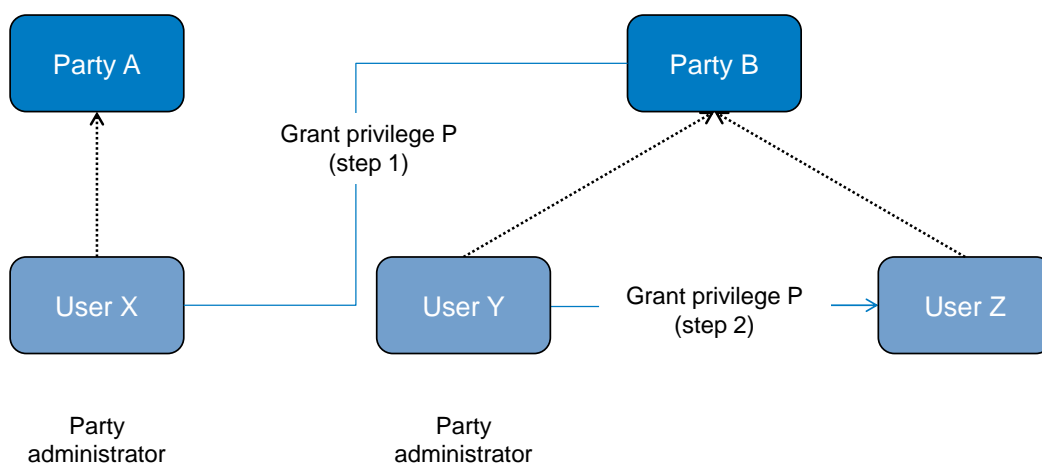


Figure 40 - Access rights configuration steps

The two configuration steps are as follows.

- 1 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.
- 2 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](#) [▶ 203].

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.

³¹ 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 70 - 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 71 - 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 ³²;
- | 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

32 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 72 - 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 73 - 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 74 - 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 75 - 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 74 - Cascade process when revoking privileges [202]).</p>

Table 76 - 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 77 - Assignment of privileges to roles

Row	Role	Privilege	Deny	Admin	Four-eyes
1	Party management	User A1	False	True	True

Table 78 - Assignment of roles to users

Row	Role	Privilege	Deny	Admin	Four-eyes
1	Update party	User A2	False	True	True

Table 79 - 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](#) [196], 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).

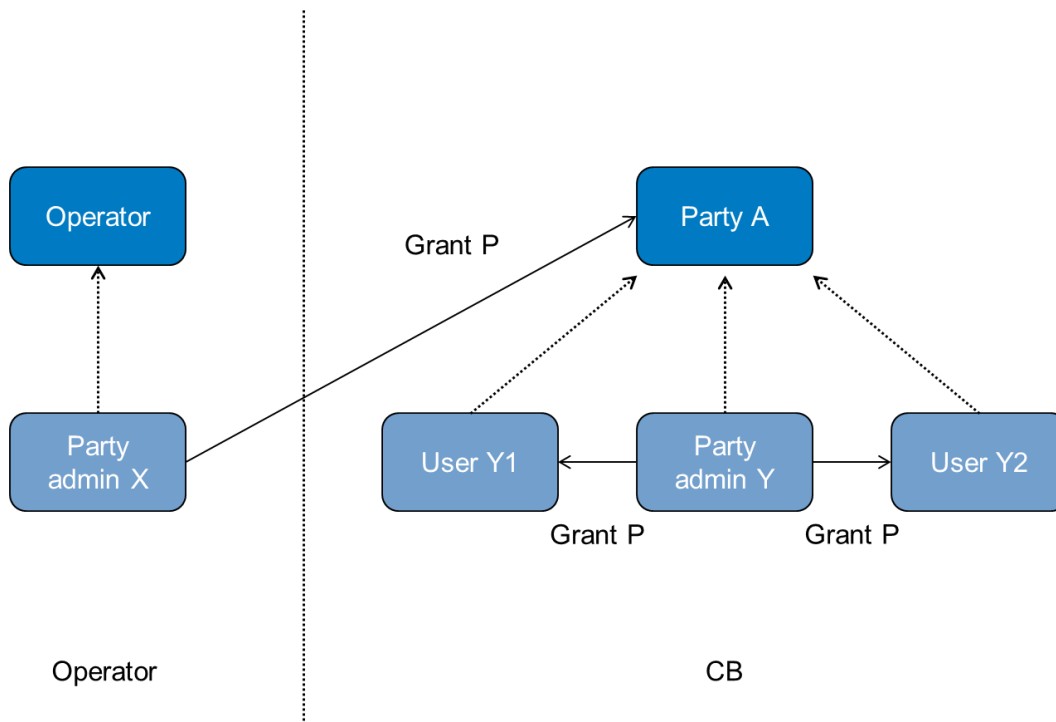


Figure 41 - Access rights configuration process (A)

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).

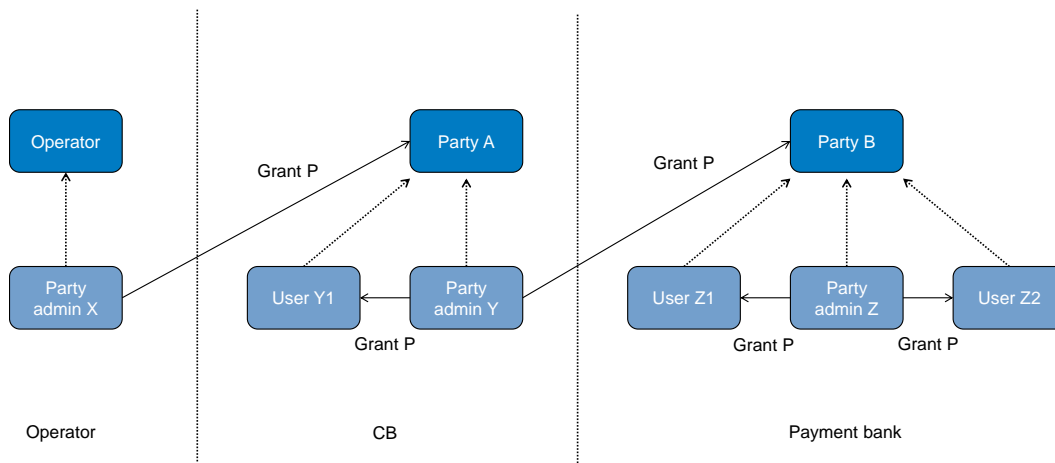


Figure 42 - Access rights configuration process (B)

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](#) [▶ 196], 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](#) [▶ 203].

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.

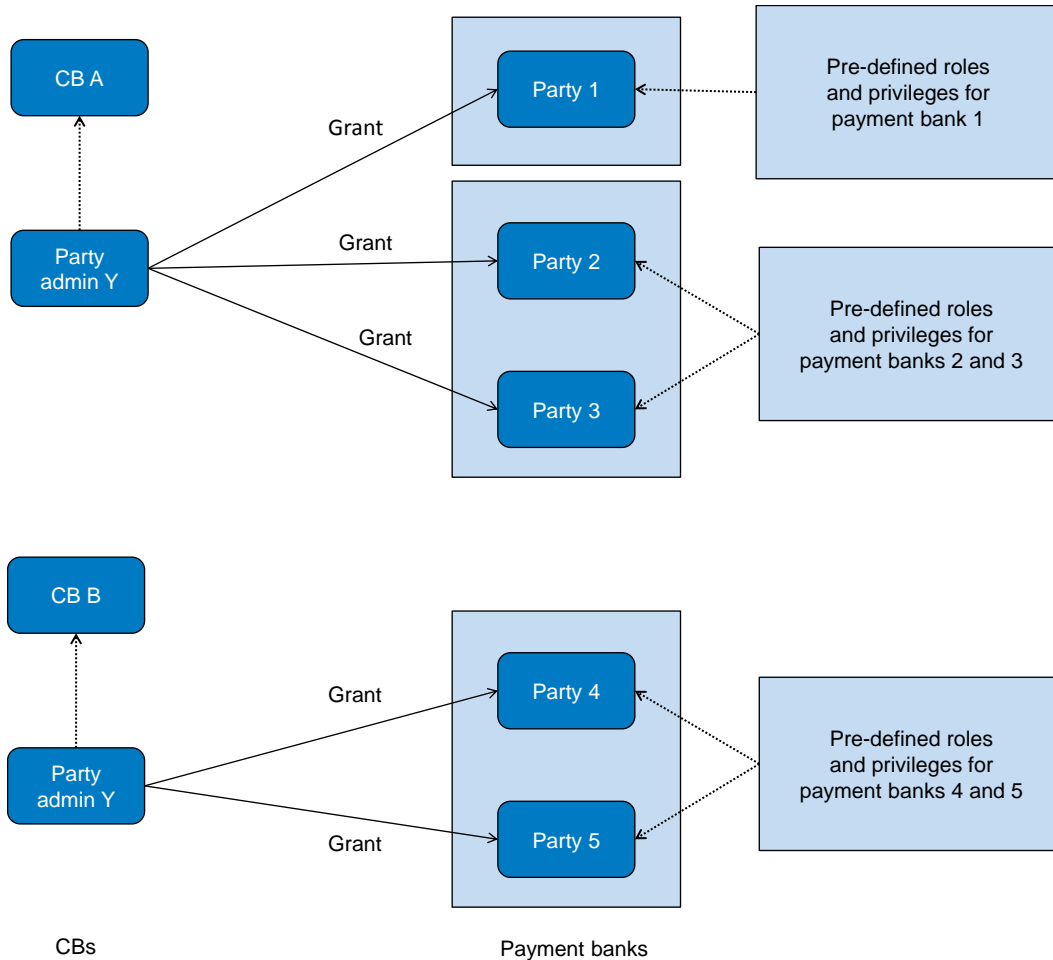


Figure 43 - Example - configuration of access rights at party level by the operator

A party administrator of each CB assigns the relevant set of roles ³³ 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.

³³ 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.

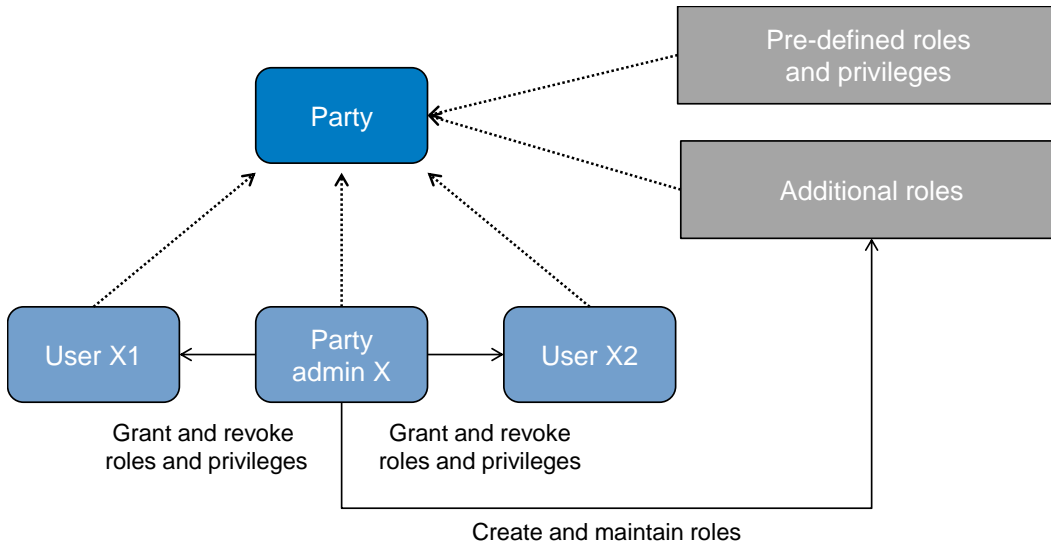


Figure 44 - Configuration of access rights at user level

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 ³⁴ additional roles, besides the ones previously granted at party level ³⁵.
- | 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:

³⁴ 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.

³⁵ 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 \[181\]](#)).

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 80 - 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 81 - 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 82 - 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 ^{36 37}
Party	Party	Operator, CB
	Party service link	Operator, CB
	Banking Group	CB
	MFI	CB
Cash account	Cash account	All ³⁸
	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	All
	Role	Operator, CB
	Privilege	Operator
	Certificate DN	All
	User-certificate DN link	All
	Role user ³⁹	All
	Role party ⁴⁰	Operator, CB
	Grantee privilege ⁴¹	Operator, CB, payment bank
Message sub- scription configu-	Message subscription rule	CB, payment bank
	Message subscription rule set	CB, payment bank

36 "All" indicates that all types of CRDM actors (operator, CBs, payment banks) have the ability to manage the object type.

37 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.

38 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.

39 This object is related to the granting/revoking of roles to/from users.

40 This object is related to the granting/revoking of roles to/from parties.

41 This object is related to the granting/revoking of privileges to/from roles, parties and users.

Area	Object	Responsible CRDM Actors ^{36 37}
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 83 - 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](#) [▶ 214]). 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.

- | **create:** creates a new common reference data object.
- | **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.
- | **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](#) [▶ 219]).
- | **restore**⁴²: 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](#) [▶ 181] 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.

⁴² 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 84 - 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 85 - 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 86 - 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 ⁴³	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

⁴³ 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 87 - Common reference data objects with limited validity period ⁴⁴

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.

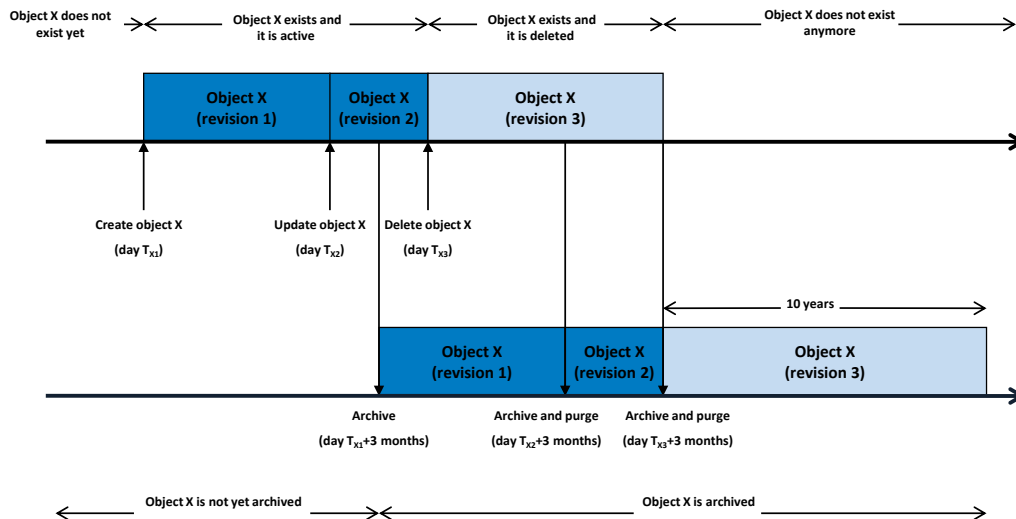


Figure 45 - Example - archiving and purging after deletion of a common reference data object

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.

⁴⁴ 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.

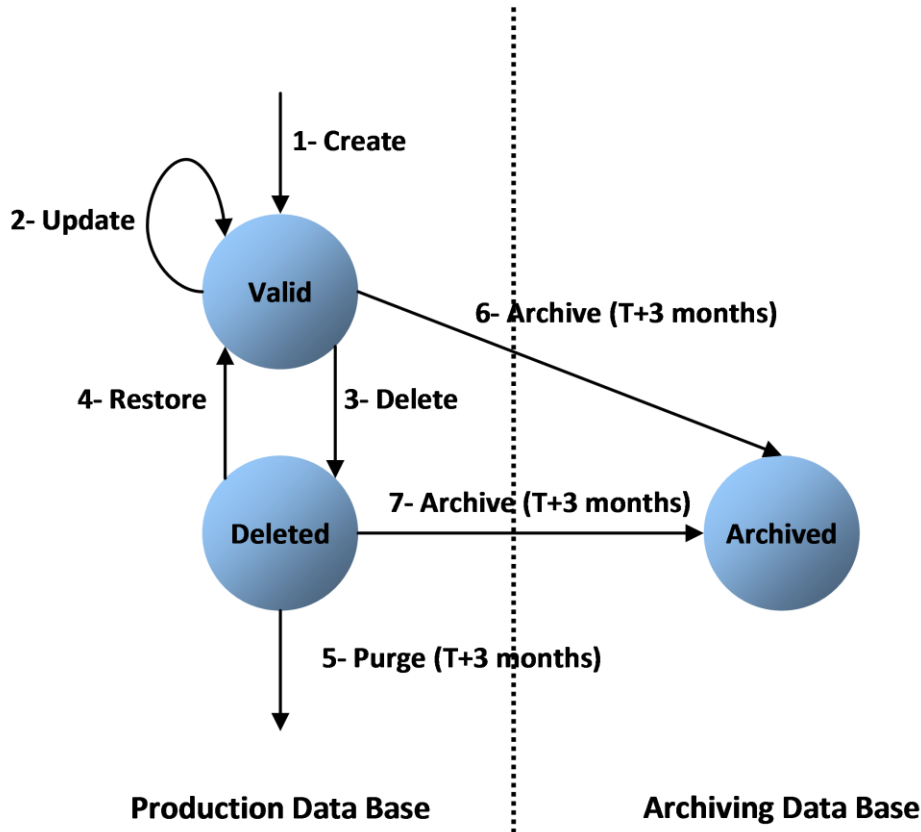


Figure 46 - Lifecycle of common reference data objects with unlimited validity period

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.

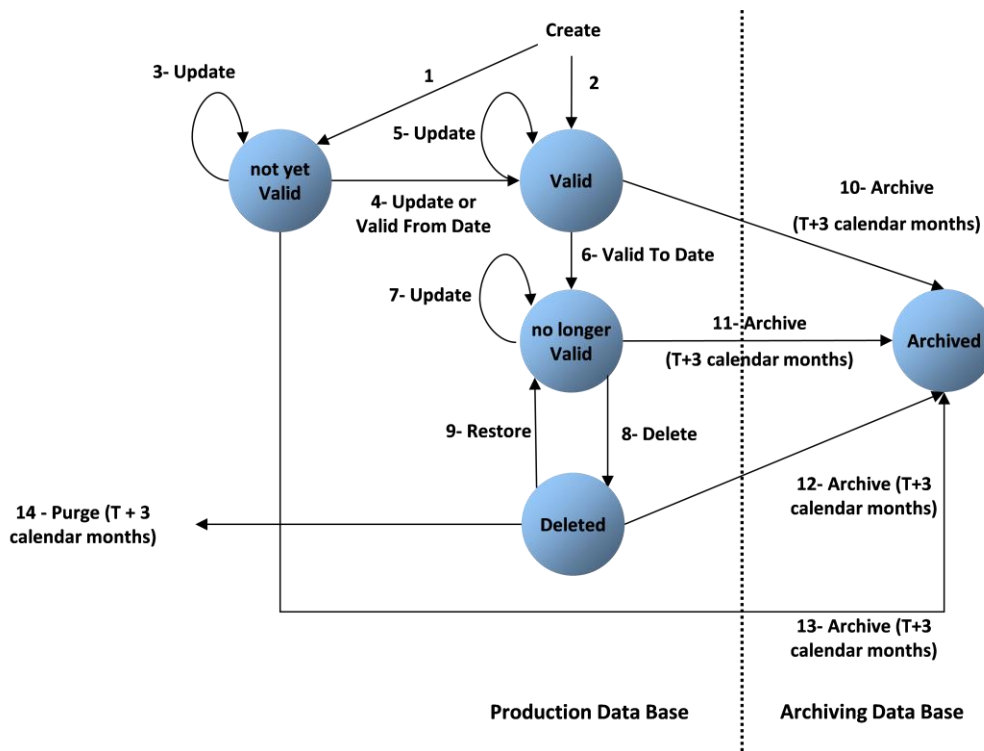


Figure 47 - Lifecycle of common reference data objects with limited validity period

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.

- I 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).
- I 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 88 - CRDM data segregation per service/component

6.2 Data warehouse

Will be completed in v2.0.

6.2.1 Introduction

6.2.2 Scope of the data warehouse

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

Will be completed in v2.0.

6.5.2.2 Network agnostic communication

Will be completed in v2.0.

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:

- | **choose the application** among the authorised applications accessible by at least one user-ID linked to the DN of the user
- | **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

Will be completed in v2.0.

8.5.1 Technical monitoring

Will be completed in v2.0.

8.6 Archiving management

8.7 Trouble management

9 Additional information for CBs

9.1 Role of CBs in CLM

General

The processes with CLM – as a component of the 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 CLM 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 CLM, the CBs have the following responsibilities.

Administrative tasks	Operational tasks
All contacts and provision of any kind of support to their participants (credit institutions, ancillary systems)	Inclusion and exclusion of participants
	Monitoring of the activities of their participants
	Provision of intraday liquidity necessary for the smooth running of the system
	Initiating payments on behalf of their own or on behalf of their participants
	Billing to their participants
	Handling of local contingency
	Minimum reserve management
	Processing of the standing facilities

Table 89 - Tasks of the CBs

Any payment order 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 CLM Account Holder. In practical terms, this means that each CB must be

- directly addressable in CLM in order to receive payments from other participants,
- able to submit payments on its own or on behalf of its customers in CLM.

9.2 Settlement of payments - specific functions for CBs

9.2.1 Payments linked to monetary policy operations

In CLM the following monetary policy operations are submitted by a CB and settled on the MCA of the CLM Account Holder.

- overnight deposits
- marginal lending facility
- minimum reserve requirements
- open market operations
- payment of interests
- debit of invoices
- update of credit line

All monetary policy operations are settled with priority and are either fully executed or queued, i.e. payments linked to monetary policy operations are never settled partially.

Depending on the type of operation, the CB can send payment orders to either debit or credit the participant's MCA with a CBs account as counterpart.

Monetary policy operations can be initiated by the CB in A2A or in U2A mode.

The CB can send the above mentioned monetary policy operations (depending on the underlying business case) as:

- pacs.009 FinancialInstitutionCreditTransfer
- pacs.010 FinancialInstitutionDirectDebit
- camt.050 LiquidityCreditTransfer

The CB can send credit transfers and/or direct debits also as connected payments. They are called "connected payments" due to the link between the payment (an immediate debit/credit of its MCA) and a corresponding change of a credit line. For further details please refer to chapter [Connected payment](#) [242].

Within the payment, CBs have the possibility to define the execution time ([Definition of execution time](#) [▶ 82]). It is possible to set

- an “earliest debit time indicator” (FROTIME) and
- a “latest debit time indicator” (REJTIME).

Furthermore, payments can be submitted as “warehoused payments” which means that the CBO is sent up to ten calendar days in advance. In this case, the payment is warehoused until CLM opens for the settlement on the intended settlement day (see chapter [Revalidate warehoused payments at SoD](#) [▶ 83]).

9.2.2 Cash withdrawals

Cash withdrawals are a part of CBOs.

It is the possibility for CBs to provide cash requested by credit institutions.

Retrieved cash withdrawals are debited from the MCA with a pacs.009 message. If liquidity has been reserved for CBOs on the MCA, cash withdrawals are taken from this reservation.

Apart from that the usual procedures for payments linked CBOs apply (see chapter [Settlement of payments linked to CBOs](#) [▶ 81] and [Payments linked to monetary policy operations](#) [▶ 235]).

9.3 Credit line management

9.3.1 Credit line update

9.3.1.1 Overview

Credit lines can be defined, modified or deleted in U2A mode (via dedicated screen) or A2A mode (by sending a ModifyCreditLine camt.998 XML message). Furthermore it is possible to modify or delete the credit line via a connected payment (see chapter [Connected payment](#) [▶ 242])

Two kinds of order types for modify credit line requests are possible.

- fixed amount credit line orders containing the new value of the credit in absolute figure
- delta amount credit line orders containing the delta between the new and the old credit line value

Both types should generally not be used in parallel. Otherwise the following restrictions apply for parallel usage.

- Pending fixed amount decreases are rejected if any kind of new credit line order (fixed or delta) is submitted and

Several delta amount orders can be pending in parallel. It means these requests are accepted and are placed in the queue once behind, but all pending delta amount orders for one account are rejected if a fixed amount order is submitted.

If the credit line is already used, orders to reduce the credit line is pending in case of insufficient available liquidity. Credit line orders automatically have the highest priority class in CLM without indicating a special priority within the message. In case of connected payments the credit line change is also placed in the queue.

Payment order processing principles do not apply to credit lines: e.g. credit lines bypass FIFO principle and are placed on the top of the queue, pending credit lines cannot be moved from the first position in the payment queue to the last one.

9.3.1.2 Credit line update process

Preconditions

To be able to modify the credit line a participant needs to have an active MCA in CLM.

Triggers and cut-off times

A credit line update is requested by the CLM Account Holder through the CB in charge or directly by the CB, e.g. the participant decreases the volume of its collateral portfolio and there is the need for the CB to decrease the credit line as well. The CB either enters it manually via U2A screen or it sends a modification order (camt.998 Modify Credit Line) to CLM through its collateral management system (via ECMS after launch in November 2022).

Changes to the credit line are in general possible throughout the whole business day. The only exceptions are the times between CB general cut-off for the use of standing facilities (i.e. 18:40) and the start of the provisioning of liquidity for the new business day (i.e. 19:00), as well the time during the maintenance window.

9.3.1.2.1 Increase of credit line

Process flow

If the modification order to increase the credit line is has successfully passed the technical and business validation, the credit line update shall be executed immediately.

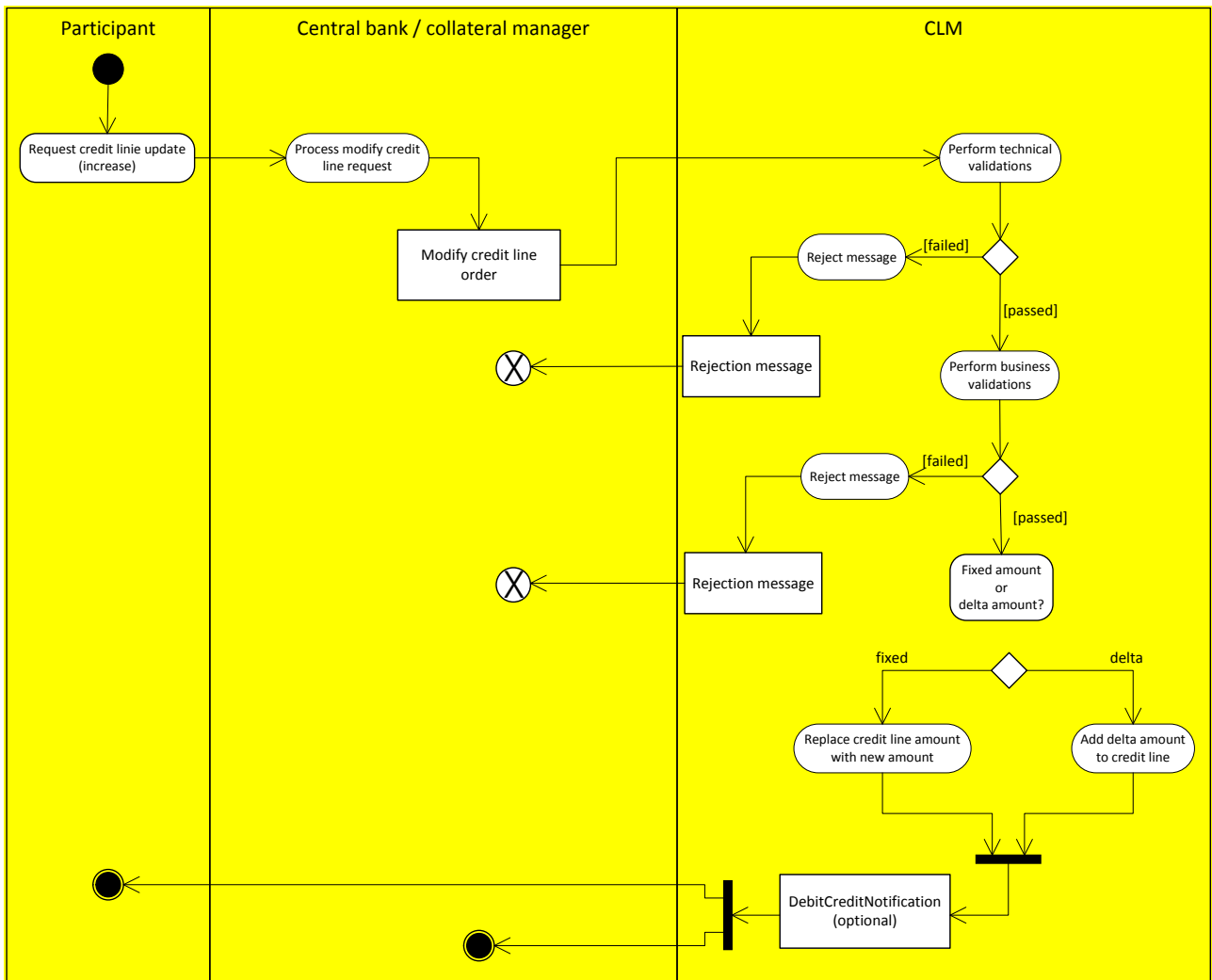


Figure 48 - Modify credit line process - increase

Process description

Step	Actor(s)	Description
1	CB	The CB submits an order via U2A or the camt.998 ModifyCreditLine message to increase the credit line of the CLM Account Holder's MCA. In A2A mode a receipt is pushed (camt.025) to inform about the status (settled, queued, rejected) of the request. Via U2A the status is displayed in the screen. It is either a fixed amount order with higher value than the actual credit line or a delta amount order to add the given amount to the actual credit line.
2	CB	After successful validation, the credit line on the MCA is updated. As a result, the liquidity available on the MCA (balance + credit line) is increased.
3	CB	The result of the update of the credit line is made available in A2A mode (if the CB has subscribed to it) via camt.054 or in U2A mode via success message on the dedicated screen.

Step	Actor(s)	Description
4	CLM Account Holder	<p>The increased credit line is also made transparent for the CLM Account Holder.</p> <p>On request the new amount of the available liquidity on its MCA is accessible via query in U2A mode. Moreover, the CLM Account Holder can be informed about the increase of the credit line through a camt.054 message (if the CLM Account Holder has subscribed to it).</p>

Table 90 - Process description for increasing credit line

Used messages

- | [Receipt \(camt.025\)](#) [397]
- | [ModifyCreditLine \(camt.998\)](#) [479]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [442]

9.3.1.2.2 Decrease of credit line

Process flow

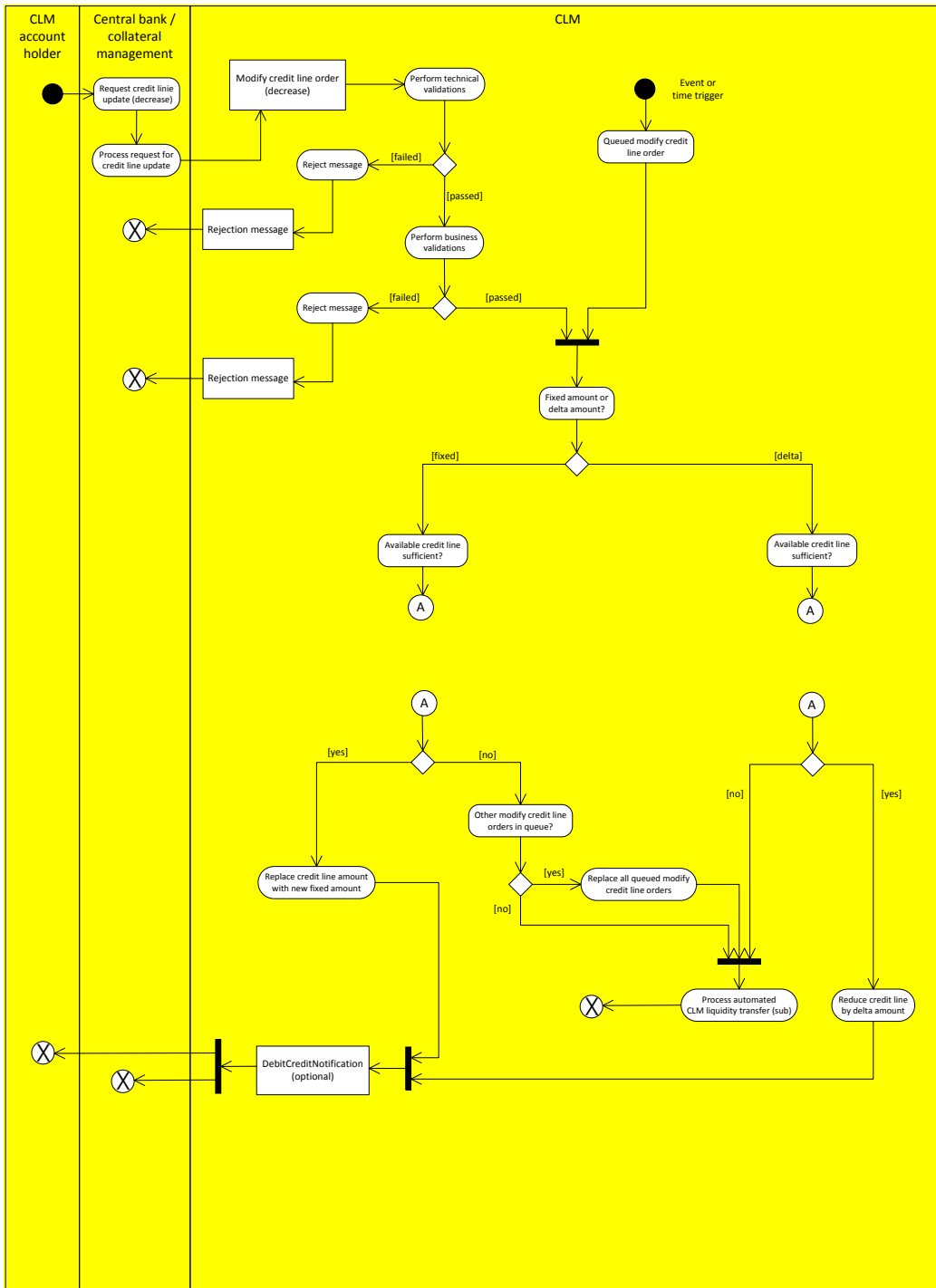


Figure 49 - Modify credit line process - decrease

Process description

Step	Actor(s)	Description
------	----------	-------------

Step	Actor(s)	Description
1	CB	<p>The CB submits an order via U2A or the camt.998 ModifyCreditLine message to decrease the credit line of the MCA of the CLM Account Holder. In A2A mode a receipt is pushed (camt.025) to inform about the status (settled, queued, rejected) of the request. Via U2A the status is displayed in the screen.</p> <p>It is either a fixed amount order with lower value than the current credit line or a delta amount order to subtract the given amount from the current credit line.</p>
2	CB	<p>After successful validation and under the condition that the credit line is not used, the credit line is updated.</p> <p>As a result, the liquidity available on the MCA (balance + credit line) is decreased.</p> <p>Note:</p> <p>In case there is not sufficient liquidity available on the CLM Account Holders MCA the order to decrease the credit line is stored with the status "pending" on the top of the queue (in case of several pending delta amount credit line orders they are ordered by amount - with the smallest value on top) until sufficient liquidity is available on the MCA to decrease the credit line. The credit line decrease is continuously attempted to settle whenever additional liquidity is available on the involved MCA.</p>
3	CB	<p>The result of the update of the credit line is made available in A2A mode (if the CB has subscribed to it) via camt.054 or in U2A mode via notification on the respective screen.</p>
4	CLM Account Holder	<p>The decreased credit line is also made transparent to the CLM Account Holder.</p> <p>On request the new amount of the available liquidity on its MCA is accessible in U2A mode. Moreover, the CLM Account Holder can be informed about the decrease of the credit line through a camt.054 message (if the CLM Account Holder has subscribed to it).</p>

Table 91 - Process description for decreasing credit line

Used messages

[Receipt \(camt.025\) \[397\]](#)

[ModifyCreditLine \(camt.998\) \[479\]](#)

[BankToCustomerDebitCreditNotification \(camt.054\) \[442\]](#)

9.3.2 Connected payment

9.3.2.1 Overview

A connected payment is a payment initiated by a CB system or CB operator that triggers a change in the credit line of the CLM Account Holder and debit/credit of its account simultaneously to compensate the change in its credit line. Therefore the CLM Account Holder needs a MCA.

The processing of connected payments is not possible between the CB general cut-off for the use of standing facilities (i.e. 18.40 on normal business day) and the start of the provisioning of liquidity for the new business day (i.e. 19.00 on normal business day), as well as during the maintenance window.

A connected payment leads to the increase or decrease of the CLM Account Holder's credit line and at the same time to a corresponding debit or credit booking on its MCA. (**Note:** The connected payment is processed on all or nothing basis). Connected payments are not queued and can therefore not be revoked. In case of insufficient liquidity this payment type is immediately rejected.

To decrease a credit line and credit the MCA a [FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491] message is used.

To increase a credit line and debit the MCA a [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 496] message is used.

9.3.2.2 Connected payment process

The following payment flow illustrates a connected payment with positive validation and settlement on the basis of a [FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491] .

Message flow

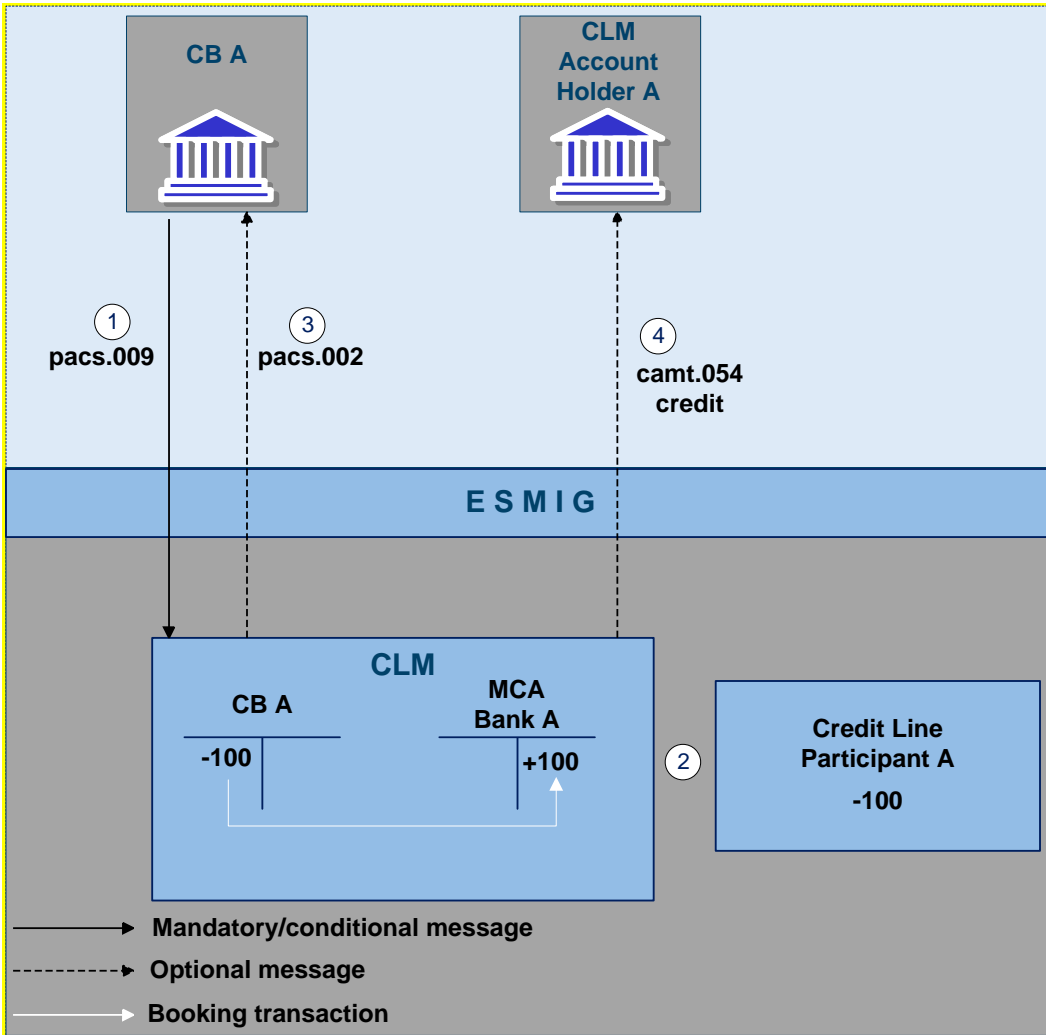


Figure 50 - pacs.009 connected payment

Process description

Step	Processing in/between	Description
1	CB via ESMIG to CLM	The CB sends a pacs.009 including message element CONPAY via ESMIG to CLM
2	CLM	CLM check and validation positive Debit CB account and credit MCA of Account Holder A simultaneously decrease credit line for account holder A (settlement amount is not necessarily equal to credit line change) if business validation positive
3	CLM via ESMIG to CB	Creation and forwarding of pacs.002 by the CLM (optional) via ESMIG to CB

Step	Processing in/between	Description
4	CLM via ESMIG to CLM Account Holder	Creation and forwarding of camt.054 (credit) by the CLM via ESMIG to CLM Account Holder A (optional)

Table 92 - Connected payment (pacs.009) – decrease of a credit line

Used messages

FinancialInstitutionCreditTransfer (COR) (pacs.009) [491]

PaymentStatusReport (pacs.002) [487]

BankToCustomerDebitCreditNotification (camt.054) [442]

The following payment flow illustrates a connected payment with positive validation and settlement on the basis of a [FinancialInstitutionDirectDebit \(pacs.010\)](#) [496] .

Message flow

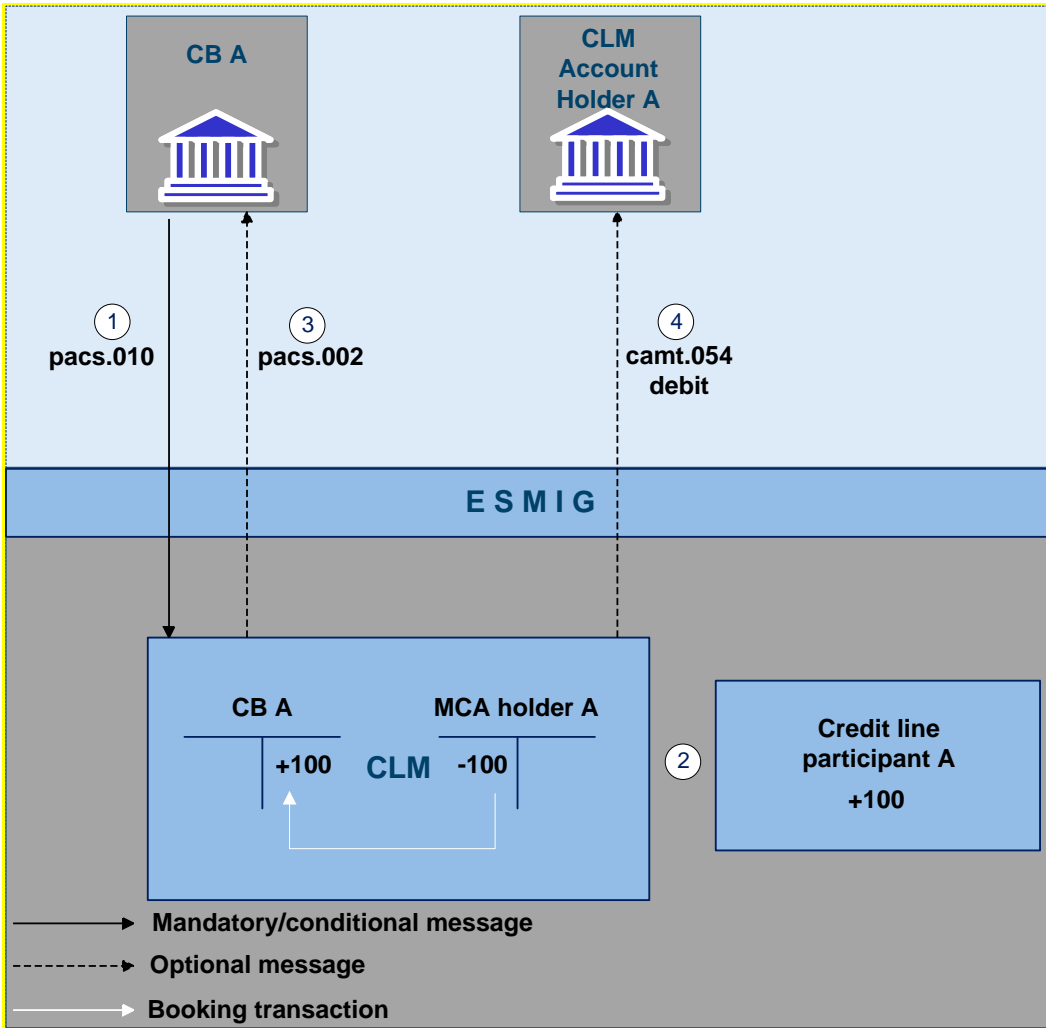


Figure 51 - pacs.010 connected payment

Process description

Step	Processing in/between	Description
1	CB via ESMIG to CLM	The CB sends a pacs.010 with message element CONPAY and the credit line change via ESMIG to CLM
2	CLM	CLM check and validation positive Credit CB account and debit MCA of account holder A simultaneously increase credit line for account holder A (settlement amount is not necessarily equal to credit line change) if business validation positive
3	CLM via ESMIG to CB	Creation and forwarding of pacs.002 by the CLM (optional) via ESMIG to CB

Step	Processing in/between	Description
4	CLM via ESMIG to CLM Account Holder	Creation and forwarding of camt.054 (debit) by the CLM via ESMIG to CLM Account Holder A (optional)

Table 93 - Connected payment (pacs.010) – increase of a credit line

Used messages

- | [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 496]
- | [PaymentStatusReport \(pacs.002\)](#) [▶ 487]
- | [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442]

9.4 General ledger handling

9.4.1 CLM general ledgers

9.4.1.1 CLM general ledgers production

During the EoD procedure, the general ledger processing in CLM is twofold.

- | handling of general ledger data from other services/components (e.g. RTGS, TIPS) to be forwarded to the responsible CBs
- | handling of general ledger data from CLM for CLM cash accounts on the basis of component–internal processing and taking into account the service-based general ledger data

Both processes run in parallel as far as possible, since receipt, checking and forwarding of the service-specific general ledger files is independent of using them in overall calculations of closing entries in the CLM general ledger output. Please refer to chapter [General ledger provision](#) [▶ 248] for details on the general ledger provision to the CBs.

Handling of general ledger data for other service’s cash accounts

CLM receives the general ledger data from the other services/components (one file per service) containing

- | SoD and EoD balances of the respective service’s accounts and subaccounts and
- | turnover on the respective service’s accounts and subaccounts.

After receiving the general ledger files, CLM performs the following consistency checks.

- | check the dedicated transit accounts between CLM and the settlement services on a per-service basis: addition of the balances of the respective accounts held in CLM and in the settlement service must give a zero result

- each published EoD balance is checked by adding all turnovers on the respective account and the SoD balance, which must again give the EoD balance

- all calculated EoD balances are added and must give a zero result

After the consistency checks are passed successfully, the general ledger files (per CB and service) are provided to the responsible CBs.

In case a service/component cannot provide the general ledger file or the respective file is inconsistent, an alert message is sent to the TARGET Service desk and, if dedicated transit accounts are affected, to the account owner(s).

Further, the EoD processing in CLM and the forwarding of general ledger files is stopped, while the following measures are possibly taken by the operational team in accordance with the incident management.

Case	Consequence
Delayed receipt by CLM after EoD processing due to a problem (e.g. service/component cannot provide a file during CLM EoD processing)	<ul style="list-style-type: none"> The general ledger files are not forwarded at the usual point of time. But they are immediately forwarded to the CBs after being available. “EoD settlement on CB’s ECB account” takes place without consideration of the balances of this service/component. Reserve management and standing facilities processes cannot consider the balances of this service/component for their calculation. CLM general ledger files are incorrect due to missing service/component positions.
The service/component cannot provide a general ledger file at all.	<ul style="list-style-type: none"> There are no general ledger files for the CBs available. “EoD settlement on CB’s ECB account” takes place without consideration of the balances of this service/component. Reserve management and standing facilities processes cannot consider the balances of this service/component for their calculation. CLM general ledger files are incorrect due to missing service/component positions.
The general ledger file provided by the service/component is inconsistent.	<ul style="list-style-type: none"> Inconsistent general ledger files are forwarded to the CBs. “EoD settlement on CB’s ECB account” takes places with inconsistent service/component positions. Reserve management and standing facilities processes cannot consider the balances of this service/component for their calculation. In some cases a calculation without consideration of balances of this service/component could be possible.

Incident management may decide to restart the EoD processing if files are missing or inconsistent.

Handling of general ledger data for CLM cash accounts

During EoD process (see chapter [End of day](#) [▶ 73]), all EoD settlement processes are finalised. In a next step CLM processes all remaining consistency checks.

Afterwards CLM builds the general ledger files for CLM business per CB. Finally it sends the general ledger files to the CBs using the structured ISO format [BankToCustomerStatement \(camt.053\)](#) [▶ 425].

9.4.1.2 CLM general ledgers content

The CLM general ledger file contains all cash accounts held in the CLM component. For a comprehensive description of accounts, please refer to chapter [Accounts structure and functionalities](#) [▶ 51]. The file includes the

- SoD and EoD balance,
- sum of credits and sum of debits for all included CLM cash accounts.

In case of CB accounts CLM provides the total debits and credits on national level and per defined CB (i.e. cross border level).

In case of CBs ECB accounts CLM provides the sums of debits and the sums of credits towards each CB. For TIPS, an additional entry is reported in case there have been cross-border payments during this business day (see chapter [Calculating the positions of CBs vis-à-vis other CBs](#) [▶ 76]), which also gives extensive numeric examples).

CLM delivers general ledger data that fulfil the following consistency condition. A single balance check is performed per account, where the EoD balance is verified by adding the account turnovers to the SoD balance, which must again result in the EoD balance.

9.4.2 General ledger provision

The sending of general ledger files to CBs is mandatory, therefore no report configuration is maintained in CRDM.

However, the general ledger file provision follows the defined standards of report management, only the activation is different. See chapter [Report generation process](#) [▶ 175] for the details on report management.

CLM generates one general ledger report per CB and service, meaning that a given CB receives as many files as it uses services. The provision starts immediately after the finalisation of the consistency checks for the respective service. CLM does not expect these files to arrive in a particular order.

Finally, CLM needs and uses the data from all services for the preparation of its own general ledger report. Hence this report can only be provided if all services have contributed their data.

In case a service/component cannot provide the general ledger file or the file is inconsistent, the respective handling is described in chapter [CLM general ledgers production](#) [246].

9.5 Query management - CB specific queries

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 CLM, CRDM, scheduler and billing](#) [177]. As regards the processing the description in chapter [Execute query](#) [299] also applies for all queries irrespective of their access limitations.

Query type	Initiation via GUI (U2A mode)	Initiation via XML message (A2A mode)
Aggregated available liquidity in CLM for the whole banking community query	X	-
Aggregated liquidity for all cash accounts query (only for crisis managers)	X	-
Balances of all dedicated transit accounts query	X	-
Liquidity on Banking Group level query	X	-
Liquidity on Banking Group level query (activated only upon crisis managers decision)	X	-
Minimum reserve requirements per participant query	X	X
Minimum reserve of a banking community query	X	X
Payments per status for the whole banking community query	X	-
Penalty query	X	X
Standing facilities transaction of the respective banking community query	X	X
Usage of marginal lending query	X	-
Usage of overnight deposit query	X	-

Table 94 - List of CB specific queries

As regards the queries in U2A, please refer to the CLM user handbook for further details about search parameters and query results. As regards queries in A2A, please see message specification in following chapters and on MyStandards.

9.6 Business/liquidity monitoring for CBs

Monitoring consists in providing data on the system according to an organised and human-readable form in order to allow the detection of potential problems in an early and accurate manner and to carry out the helpdesk activities.

In general, monitoring is understood as the display of aggregated information stemming from the different services/components. It is not the main objective of monitoring to provide detailed information (e.g. detailed information about a single transaction). These are provided by the different services/components themselves.

The main objectives for monitoring are

- to verify the correct functioning of the technical infrastructure,
- to provide aggregated information up-to-date in case of needs and
- to give the CB an overview of the liquidity situation of their participants.

Business/liquidity monitoring

The purpose of monitoring is to give the CB an overview of all the business running in TARGET Services. The detection of liquidity problems of the current business day for their participants => real-time monitoring of overall situation is also part of monitoring.

The aim of monitoring is to have a closer look on the payment activities of the participants (single participants and aggregated levels) and monitoring of liquidity in order to avoid gridlocks.

Origin of business/liquidity monitoring information

The following services/components are covered by the business/liquidity monitoring function.

- CLM
- RTGS
- T2S
- TIPS

The CRDM common component does not carry out operations that are relevant from a business/liquidity point of view. Therefore it is not included into the monitoring.

Information which can be monitored

Details of the displayed information are provided in the user handbook.

- Liquidity:** the liquidity of all MCAs on CLM Account Holders level can be displayed. Number and amount of rejected and revoked transactions are also shown

Early detection of gridlock situations: information on queued payments in CLM is displayed. For every account holder it is possible to display the amount (and number) of queued payments

Monitoring of payment behavior: the amount and number of sent and received payments and liquidity transfers including the percentage of how many sent and received payments and liquidity transfers are settled is shown in CLM and for individual account holders.

Reservations: information on reservations for CB operations are displayed

Queued payments: display of all queued incoming and outgoing transactions

Rejected payments: shows the number and amount of rejected transactions

Revoked payments: shows the number and amount of revoked payments

Warehoused payments: a list of CLM Account Holders who have warehoused payments for future value dates shown as a whole and grouped in value dates on their MCAs.

Standing facilities: the amount for marginal lending account balance (split on automated and on-request) for the current business day as well as the amount of the overnight deposit accounts is displayed

Reserve management: information about the ongoing fulfilment of mandatory minimum reserves is shown.

A list of CLM Account Holders MCAs with amount of required minimum reserve plus dates of the minimum reserve period which are the EoD balances, running average and adjustment balance are displayed.

9.7 Reserve management - specific functions for CBs

Following the processes described in chapter [Reserve management](#) [143] CBs have additional functions available to handle specific minimum reserve needs.

9.7.1 Input minimum reserve requirement

CBs submit the calculated minimum reserve requirement for each of their connected institutions subject to the Eurosystem's minimum reserve system to CLM. The minimum reserve requirement can be submitted by manual input in U2A mode or via a message in A2A mode⁴⁵.

The minimum reserve requirements can be submitted for the upcoming minimum reserve period at any time before the period starts.

⁴⁵ The information is submitted by the CBs collateral management system (by ECMS respectively after it's been launched).

Message name	Message code	Type
InsertCumpolsoryReserve_RM (CB only)	camt.998	InsertMinimumReserve

Table 95 - Message sent by the CB to the CLM component

9.7.2 Input additional balances

CBs can send information about the EoD balance of accounts held with them outside the TARGET Services on a U2A and A2A basis. These EoD balances are then considered in the minimum reserve calculation.

Additional balances need to be submitted before the calculation of the aggregated balance and the moving average per institution subject to the Eurosystem's minimum reserve system.

Message name	Message code	Type
InsertPHABalance_RM (CB only)	camt.998	InsertPHABalance

Table 96 - Message sent by the CB to the CLM component

9.7.3 Authorise penalty

In case there is a minimum reserve infringement and penalties are due to be paid by the affected institutions subject to the Eurosystem's minimum reserve system, CBs have to authorise such penalties before the payment order is created. Authorisation of penalties can be done in U2A and in A2A mode.

Message name	Message code	Type
AuthorizePenalty_RM (CB only)	camt.998	AuthorizePenalty

Table 97 - Message sent by the CB to the CLM component

9.8 Standing facilities - specific functions for CBs

As all processes above are CBOs, only the CB or its collateral manager respectively, is able to submit the related messages to CLM. The CB processes standing facilities if requested by a CLM Account Holder.

For details about the respective processes and related messages please refer to chapter [Standing facilities management](#) [147].

9.9 Billing - specific functions for CBs

Will be completed in v2.0.

9.10 Contingency services - specific functions for CBs

Will be completed in v2.0.

9.11 Specific requirements for CBs of "out" countries

Will be completed in v2.0.

Part II - Dialogues with the CLM participant

10 Processes with CLM

The purpose of part II of this UDFS is to describe the messages that CLM and the business application of a CLM Actor exchange for a given business scenario (use case). It provides a formalised description of the A2A interfaces in order to enable CLM Actors to adapt their business applications to interact with CLM. Part II of this UDFS does not enter into any description regarding the required behaviour of the business application(s) of CLM Actors, as this determination remains in the remit of the respective CLM Actor.

This chapter uses activity diagrams according to Unified Modelling Language conventions to present the processes and actions in CLM that result in message exchanges with the CLM Actor(s). This chapter describes the behaviour of CLM from the perspective of a technically directly connected CLM Actor. The descriptions in this chapter document only the CLM activities that process an inbound communication or trigger a possible outgoing communication to a CLM 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 CLM and the relevant CLM Actor(s) for every use case, as per examples in diagrams [insert reference here].

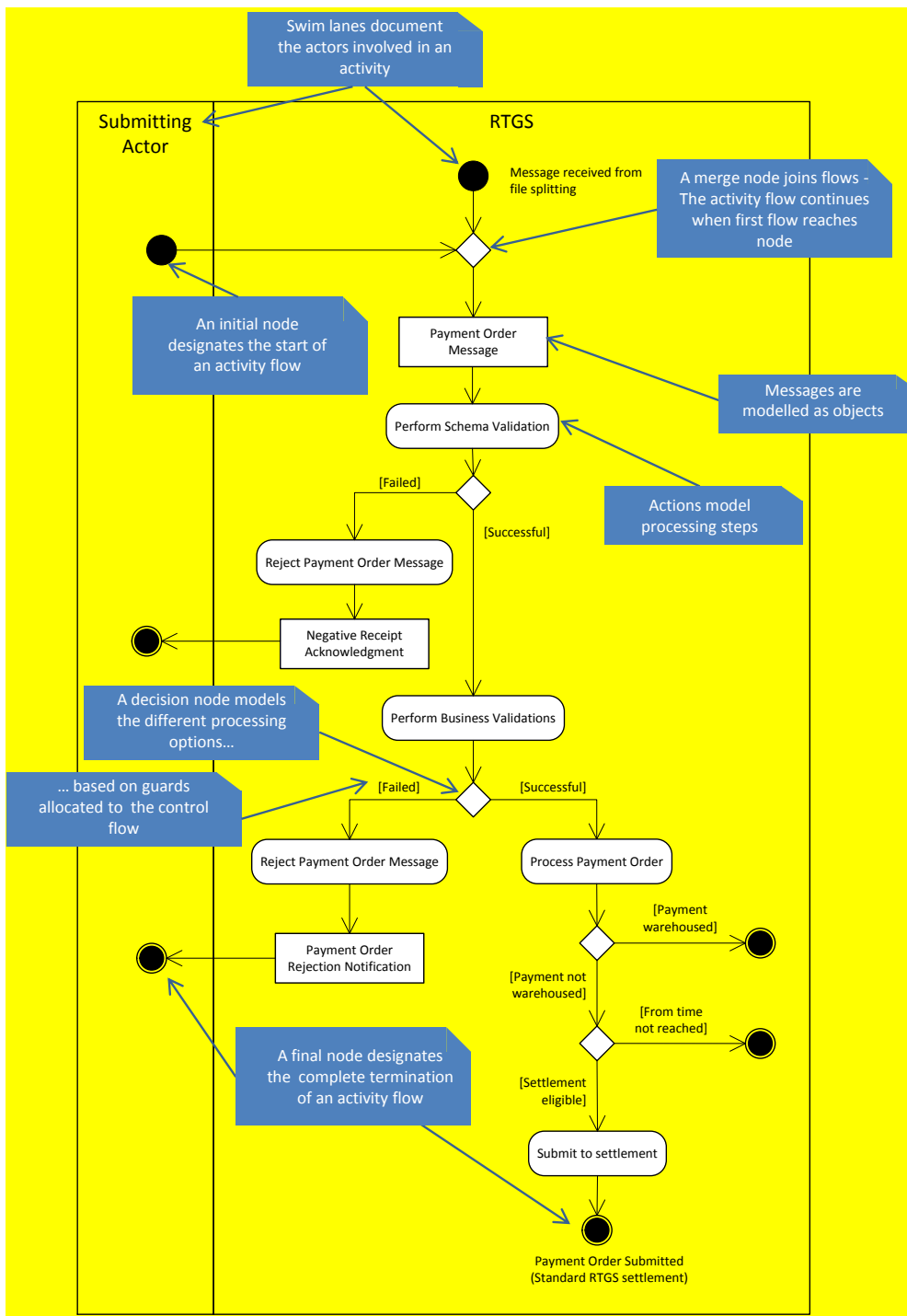


Figure 52 - UML conventions – example I

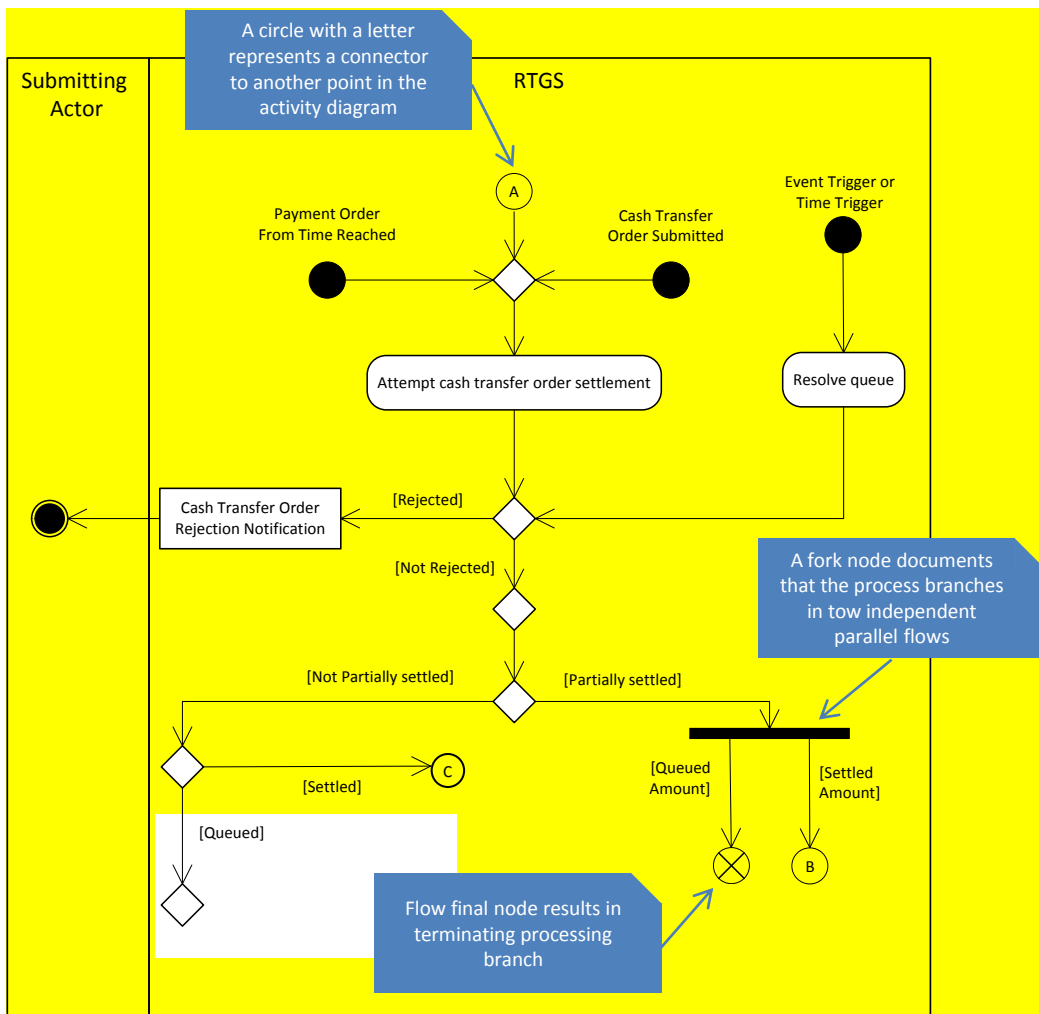


Figure 53 - UML conventions - example II

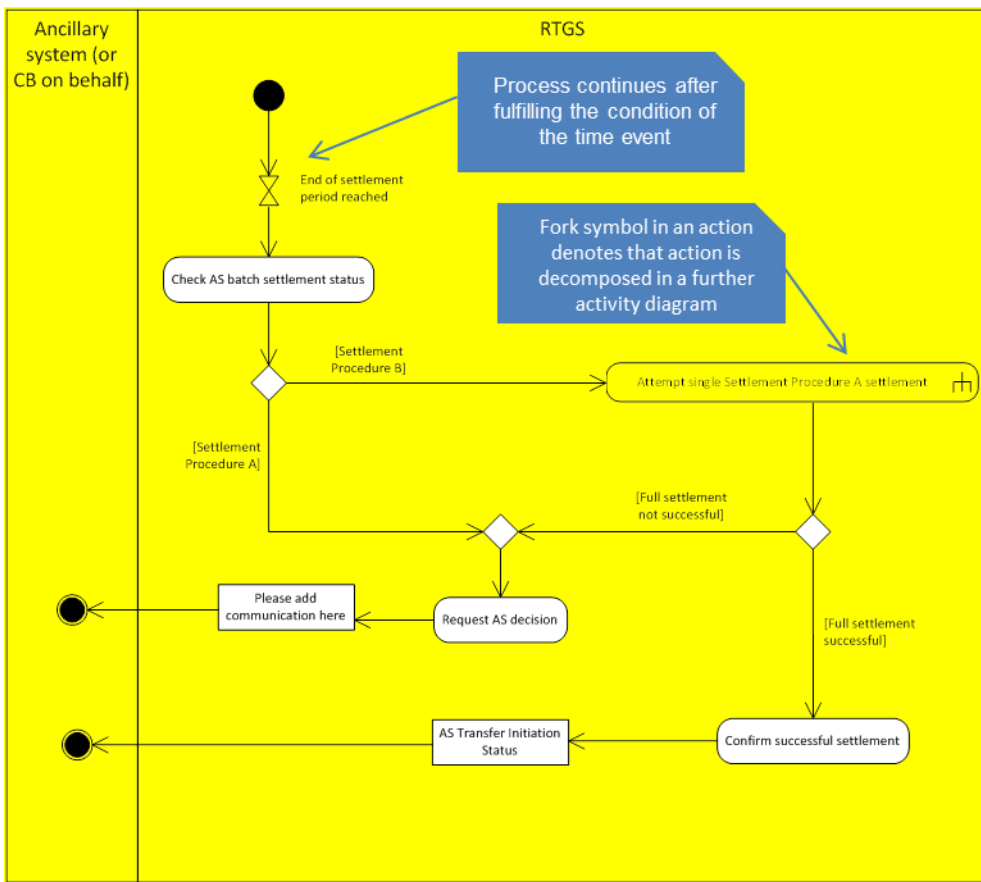


Figure 54 - UML conventions - example III

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

- I decomposed to provide diagrams on the level of its sub-processes,
- I 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.

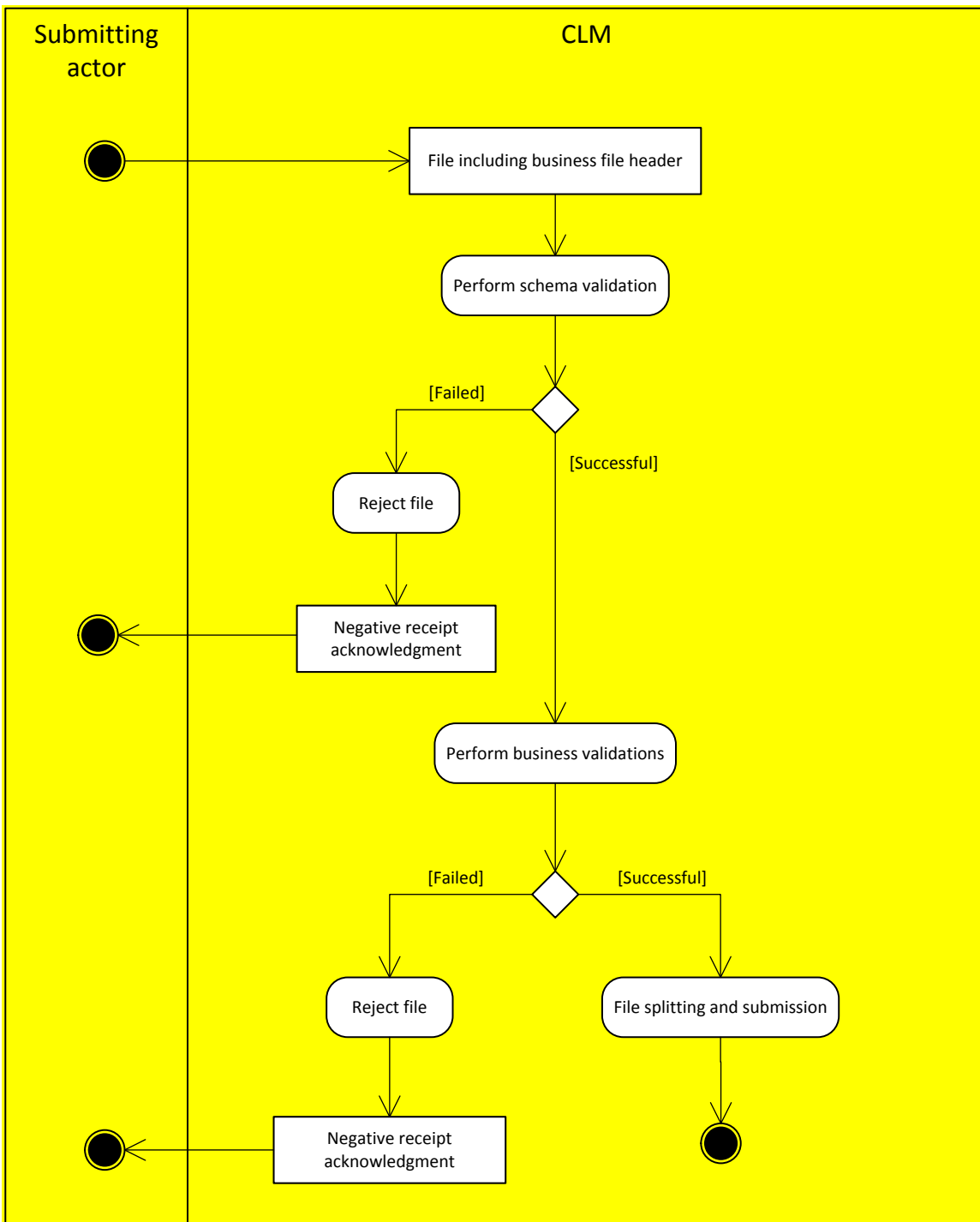


Figure 55 - Send file

Schema validation

As a first step, 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 recognizable).

[Failed] The process “Reject File” sends a [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347] message [Negative receipt acknowledgment] 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 \(admi.007\)](#) [▶ 347] message 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 Cash transfer order processing

10.2.1 Send cash transfer order

This process starts

when the submitting actor sends one of the following messages via ESMIG to CLM.

Message	Message name
FinancialInstitutionCreditTransfer (COR) (pacs.009) [▶ 491]	FinancialInstitutionCreditTransfer
FinancialInstitutionDirectDebit (pacs.010) [▶ 496]	FinancialInstitutionDirectDebit
LiquidityCreditTransfer (camt.050) [▶ 418]	LiquidityCreditTransfer

Table 98 - Messages sent by the submitting actor to CLM

when CLM receives a message from the file splitting process (refer to interface process [Send file](#) [▶ 257]).

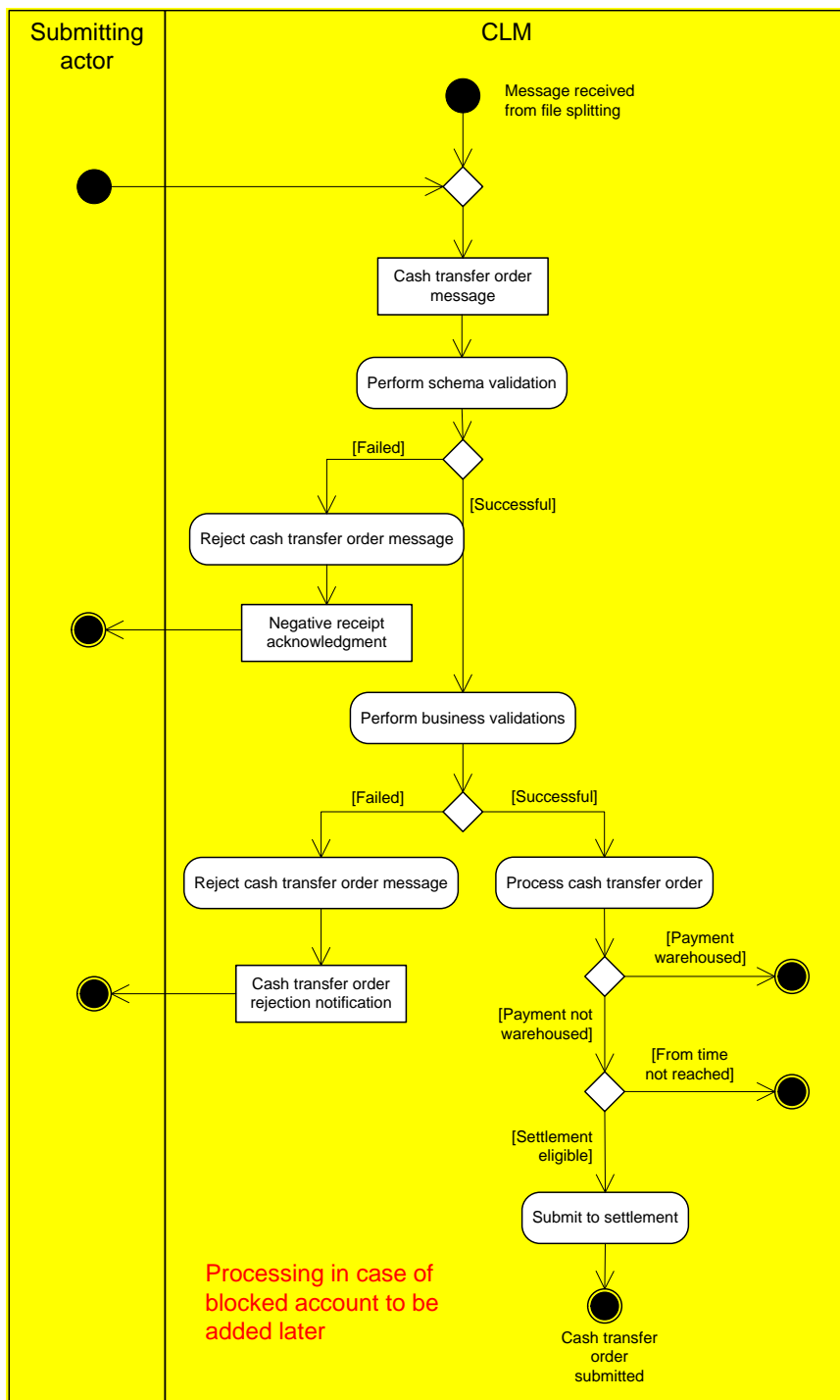


Figure 56 - Send CLM cash transfer order

Schema validation

In the first step, CLM performs the schema validation of the cash transfer order message.

- I **[Failed]** In case the schema validation fails, CLM rejects the cash transfer order message and the submitting actor receives a “*Negative receipt acknowledgement*” [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347].

Note: CLM 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, CLM continues with the business validation.

Business validation

In the second step, CLM performs the business validation with possible outcomes being:

[Failed] In case the business validation fails, CLM rejects the cash transfer order message and the submitting actor receives a “Cash transfer order rejection notification” [PaymentStatusReport \(pacs.002\)](#) [▶ 487] or [Receipt \(camt.025\)](#) [▶ 397].

Note: CLM 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.

[Successful] In case the business validation is successful, CLM continues with the processing of the cash transfer order.

As part of this processing step, CLM 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 CLM settlement](#) [▶ 266] process when it is directly eligible for settlement.

10.2.2 Revoke/cancel payment

A submitted payment can be revoked using a [PaymentRevocationRequest \(FIToFIPaymentCancellationRequest \(camt.056\)\)](#) [▶ 452] in A2A). A revocation of a payment is only possible as long as the payment is not settled on the CLM MCA. It is also possible to revoke warehoused payments. In case the payment is already settled, the CB receives a revocation reject message ([ResolutionOfInvestigation \(camt.029\)](#) [▶ 406]).

For further details about revocation/cancellation of payments please refer to chapter [Revocation of payments](#) [▶ 100].

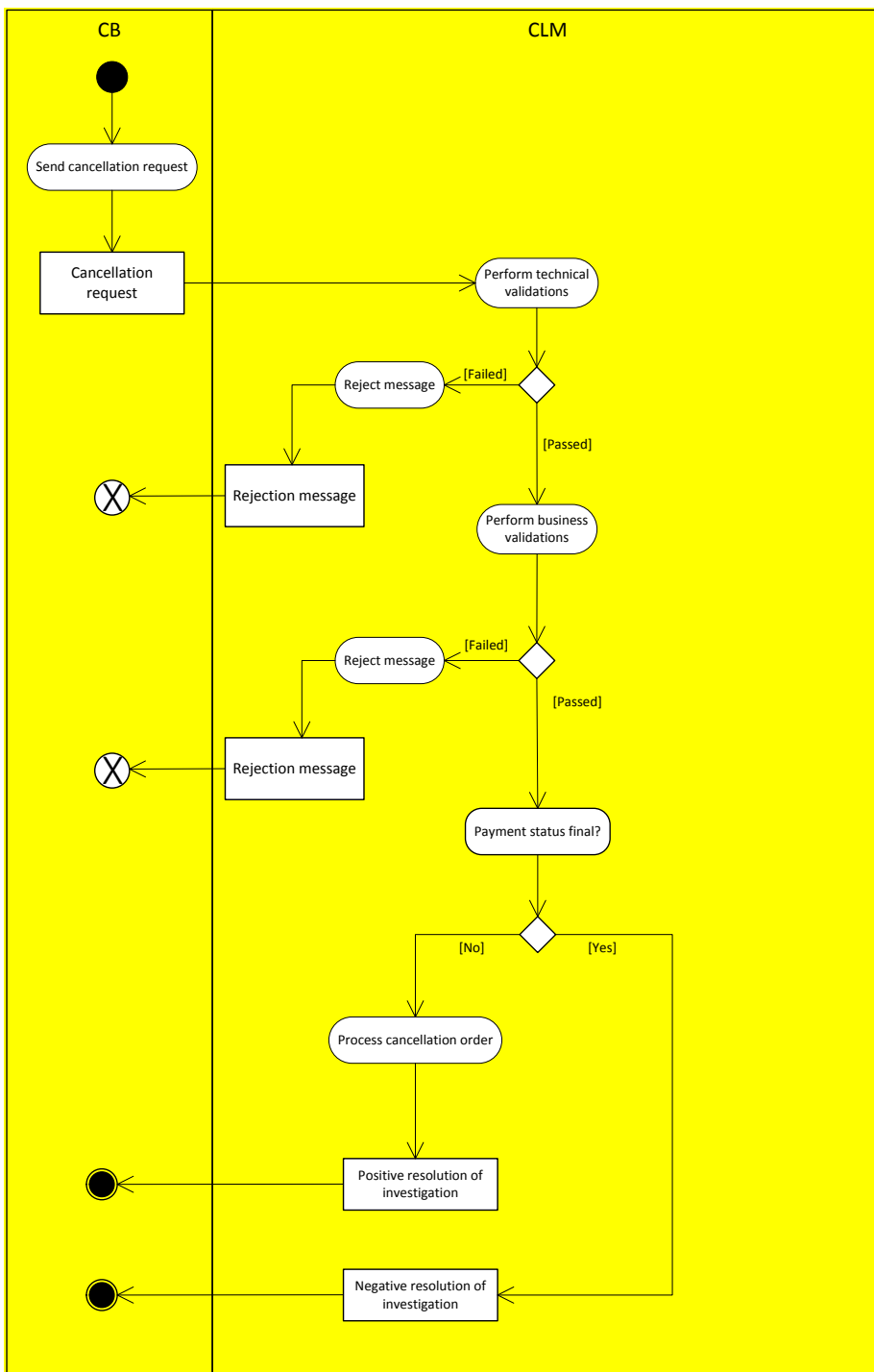


Figure 57 - Revoke/cancel payment order

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

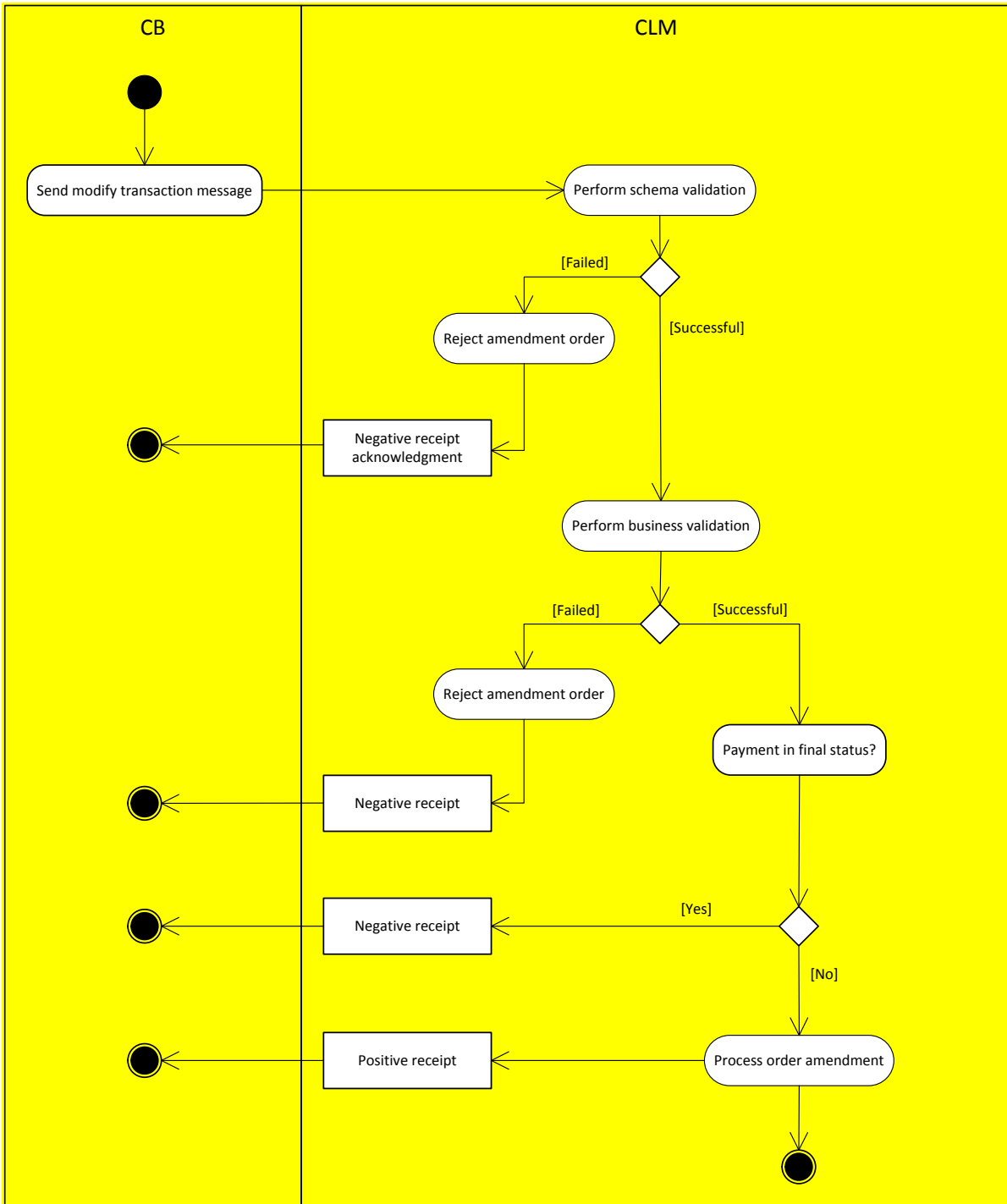


Figure 58 - Amend payment order

Step	Actor	Description
1	CB	CB sends a request to CLM to amend a payment.
2	CB	CLM carries out the technical and business validations.
3	CB	When validations are passed, CLM checks if the requested payment is settled.

Step	Actor	Description
4	CB	In case the payment is already settled, CLM sends a negative receipt to the CB.
5	CB	In case the payment is not settled yet, CLM processes the amendment order.
6	CB	CLM sends a positive receipt to the CB.

Table 99 - Process description amend payment

For further details about the amendment of payments please refer to chapter [Amendment of payments](#) [▶ 94].

10.2.4 Execute CLM standing order

CLM standing orders are instructions of a CLM Account Holder to transfer regularly a fixed amount from

- its MCA to another MCA,
- its MCA to an RTGS DCA,
- its MCA to a DCA of another service.

Standing orders are stored and maintained in CRDM and have been successfully transferred with the former CRDM data propagation to CLM.

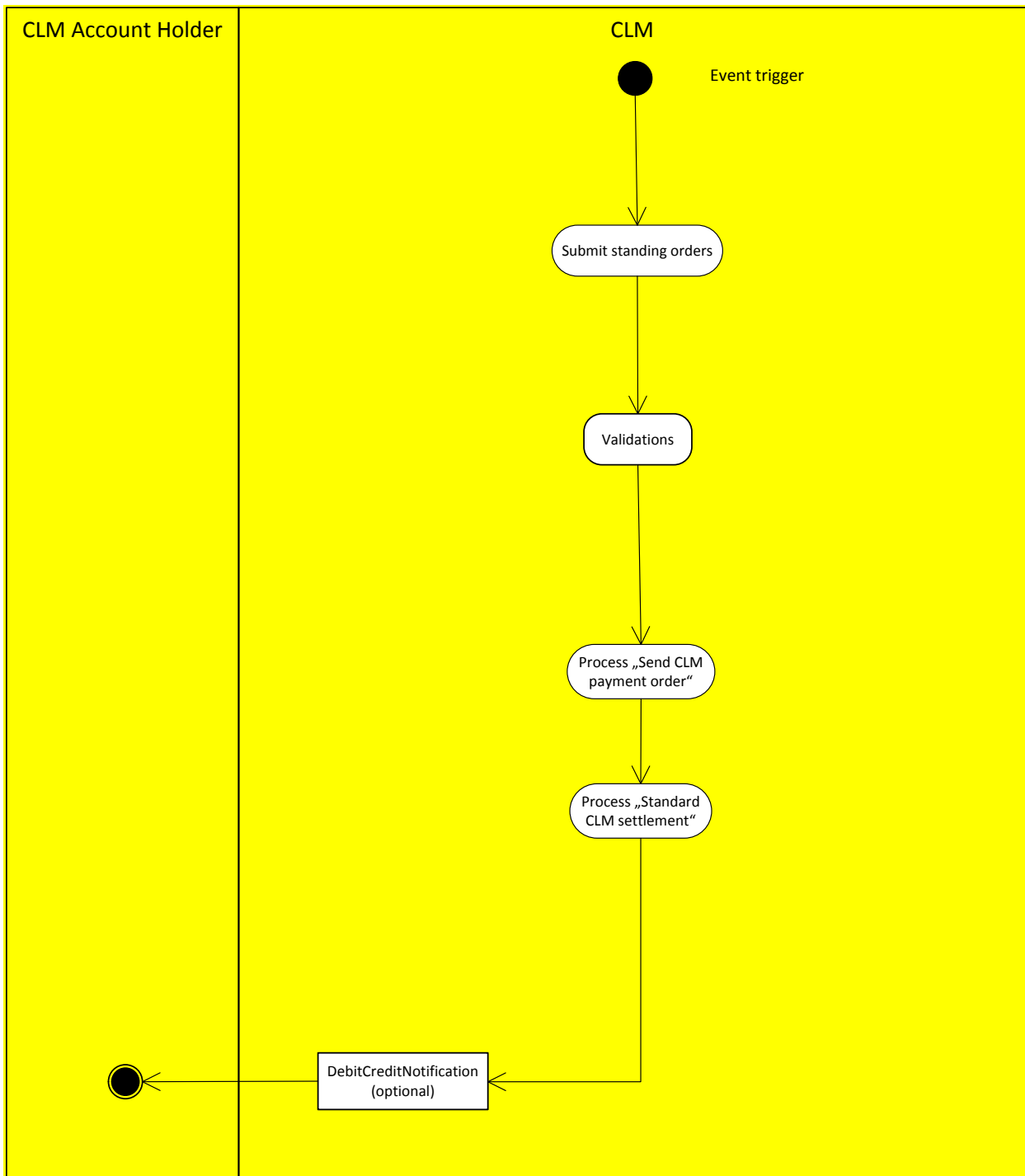


Figure 59 - Execute CLM standing order

Step	Actor	Description
1	CLM	Business day event "begin of provisioning of liquidity" initiates the process of submitting standing orders. (Normally at 19:30).
2	CLM	CLM carries out the validations.

Step	Actor	Description
3	CLM	In case of positive validation: CLM determines the type of standing order.
4	CLM	In case of standing order for liquidity transfer CLM starts the processes "Send CLM settlement order" and "Standard CLM settlement".
5	CLM	CLM sends a CreditDebitNotification to the debited and/or credited CLM Account Holder if he subscribed for it.

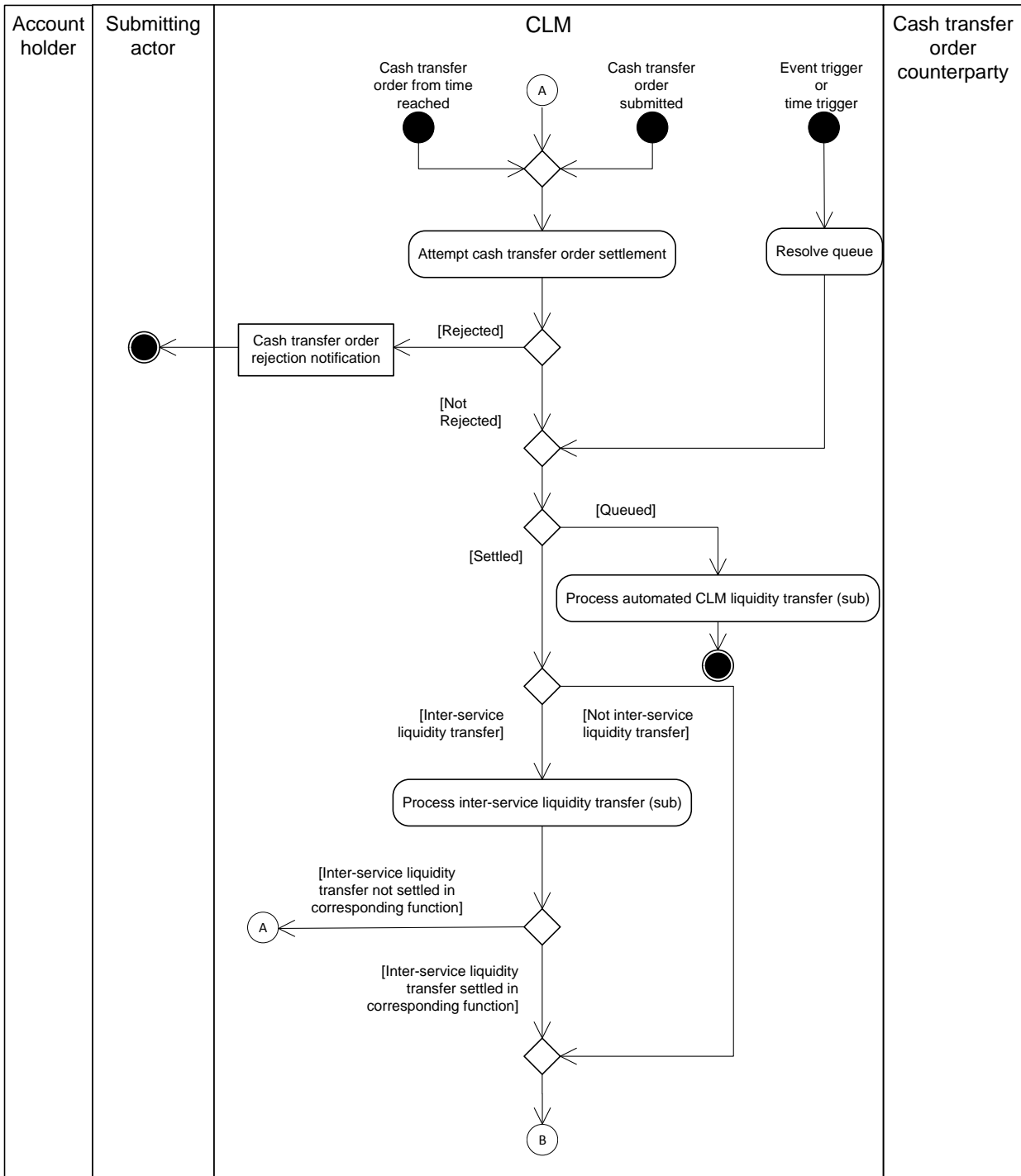
Table 100 - Process description execute CLM standing order

10.3 Settle CLM cash transfer orders

10.3.1 Standard CLM 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 CLM component could not be successfully booked in the other service or components and the amount needs to be credited back to the MCA [A] or
- | for a successfully validated cash transfer order that specifies "From time" and the "From time" has been reached [Cash transfer order from time reached]



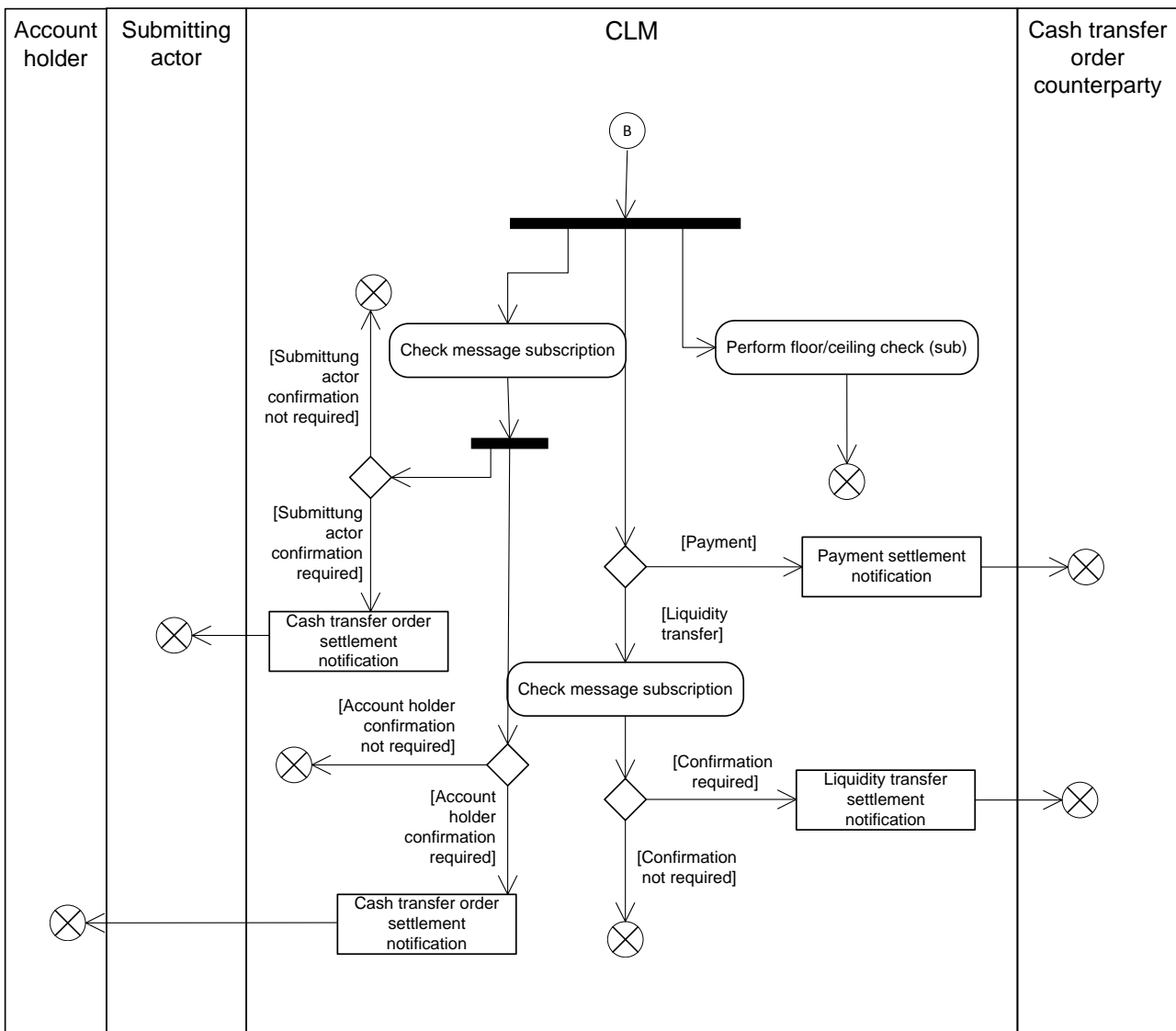


Figure 60 - Standard CLM settlement

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.

- l **[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” [camt.025 Receipt](#) [▶ 397] to the submitter of the original incoming [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418] .
- l **[Not Rejected]**

 - the payments (CBOs) settle or queue,
 - the liquidity transfers settle.

In the second step

- l for all accepted (not rejected) cash transfer orders,

- I as well as for all queued payments forwarded to the process “*Resolve queue*” in case of an event trigger.

the result of the process can be

- I **[Queued]** the queuing of CBOs which cannot settle, triggering the sub-process “*Automated CLM liquidity transfer*”.

Note: Queuing of liquidity transfers never takes place in CLM. Contrary to RTGS, queuing of liquidity transfers does not occur in the case of an automatically triggered inter-service liquidity transfer from RTGS.

- I **[Settled]** After successful settlement the “*Cash transfer order counterparty*” receives the following messages in case of

- CBOs:

- a “*Payment settlement notification*” [camt.054 BankToCustomerDebitCreditNotification](#) [▶ 442] provided that a respective message subscription configuration has been set-up in advance.

- liquidity transfers:

- a “*Liquidity transfer settlement notification*” [camt.054 BankToCustomerDebitCreditNotification](#) [▶ 442] provided that a respective message subscription has been set-up in advance.

Note: CLM treats inter-service liquidity transfers that another component initiates as any other intra-service liquidity transfer.

- I The *submitting actor* receives the following messages in case of

- liquidity transfers initiated via [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]:

- a “*Cash transfer settlement notification*” [camt.025 Receipt](#) [▶ 397] provided that a respective message subscription configuration has been set-up in advance.

- inter-service liquidity transfers initiated via [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418] in CLM:

- a “*Cash transfer settlement notification*” [camt.025 Receipt](#) [▶ 397] only after successful settlement in the other service or component provided that a respective message subscription configuration has been set-up in advance.

Note: In case the other service or component could not successfully book the settlement, CLM sends a negative “*Cash transfer settlement notification*” [camt.025 Receipt](#) [▶ 397] including the first error code reported by the other service or component.

- CBOs:

- a “*Cash transfer settlement notification*” [pacs.002 PaymentStatusReport](#) [▶ 487] 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 [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418]:

-
- a “Cash transfer settlement notification” [camt.054 BankToCustomerDebitCreditNotification](#) [▶ 442] provided that a respective message subscription configuration has been set-up in advance.
 - inter-service liquidity transfers initiated via [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418] in CLM:
 - a “Cash transfer settlement notification” [camt.054 BankToCustomerDebitCreditNotification](#) [▶ 442] only after successful settlement in the other service provided that a respective message subscription configuration has been set-up in advance.
 - CBOs:
 - a “Cash transfer settlement notification” [camt.054 BankToCustomerDebitCreditNotification](#) [▶ 442] 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 CBOs on the MCA.

Note: The settlement of liquidity transfers on MCAs trigger no floor/**ceiling** processing.

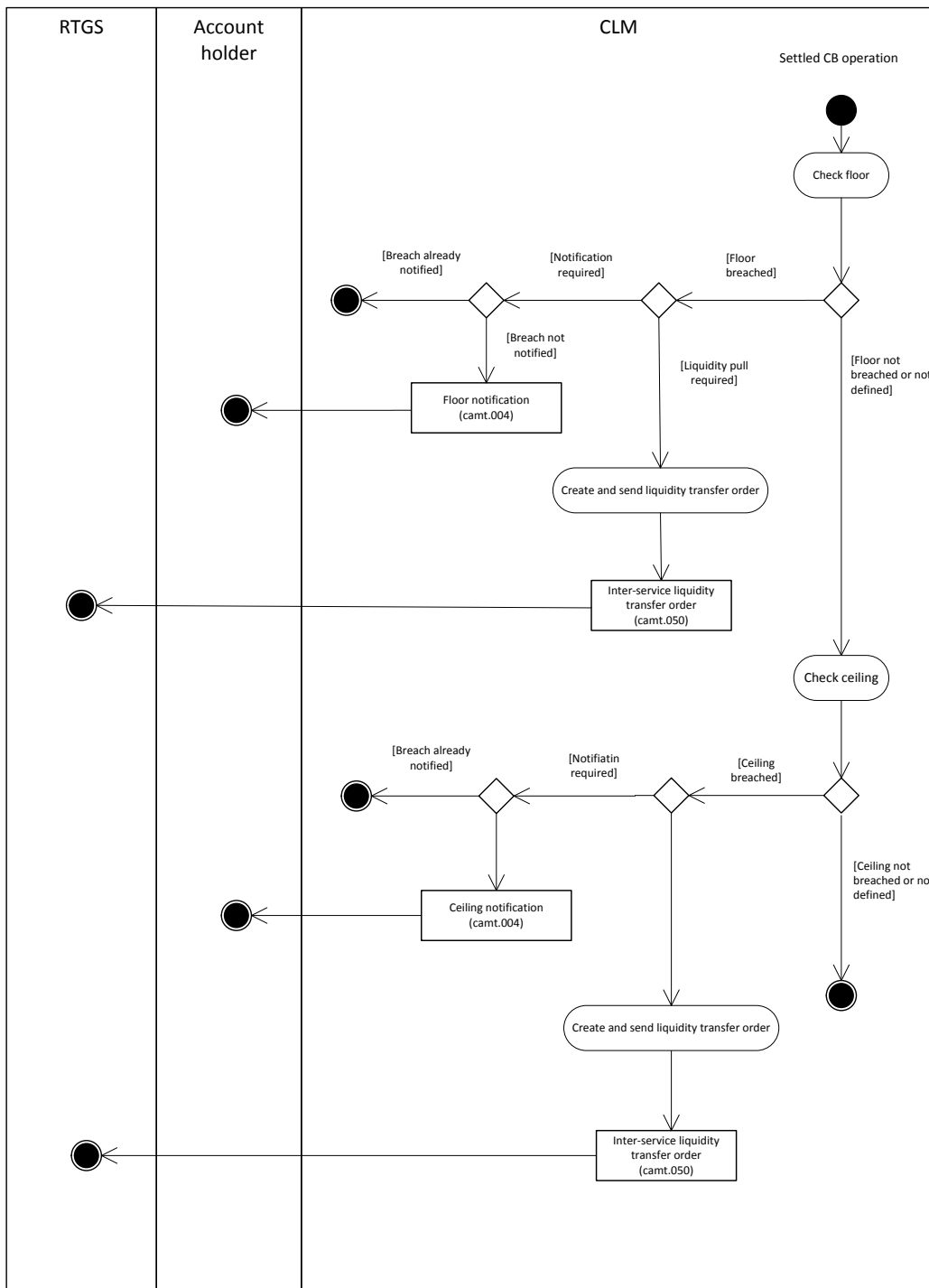


Figure 61 - Floor and ceiling processing

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 MCA holder receives a “Floor notification” [camt.004 \(ReturnAccount\)](#) [▶ 355].

- I 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 RTGS DCA has been set-up in advance

CLM sends to the RTGS an inter-service liquidity transfer order, [camt.050 \(LiquidityCreditTransfer\)](#) [▶ 418] , in order to pull liquidity up to the targeted floor amount.

CLM treats inter-service liquidity transfers that another component initiates as any other intra-service liquidity transfer.

Ceiling processing

- I 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 MCA holder receives a “Ceiling notification” [camt.004 \(ReturnAccount\)](#) [▶ 355].

- I 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 RTGS DCA has been set-up in advance

CLM sends to the RTGS an inter-service liquidity transfer order as [camt.050 \(LiquidityCreditTransfer\)](#) [▶ 418] in order to push liquidity to reach the predefined target ceiling amount.

10.3.1.2 Process automated liquidity transfer

This process starts when a CBO does not settle and, therefore is queued.

Note: This process does not apply to liquidity transfers in CLM.

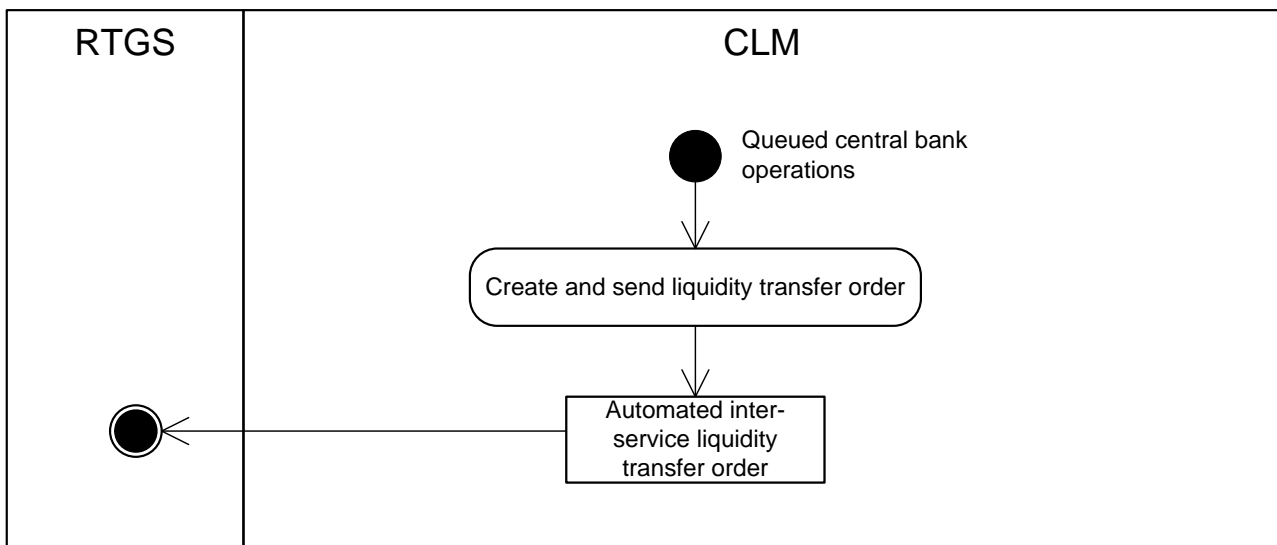


Figure 62 - Automated liquidity transfer

CLM automatically creates a new automated inter-service liquidity transfer order and sends a [camt.050 \(LiquidityCreditTransfer\)](#) [▶ 418] to RTGS to pull the missing liquidity that the settlement of a CBO requires.

10.3.2 Process reject time instructions

Payments can include a “latest credit/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 “reject time”.

The process starts after two entry points:

- the submission of a new payment order ([FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491], [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 496]) or
- new settlement attempt of a queued payment order.

CLM carries out the technical validations and sends a negative [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347] to the CB in case the validation fails.

If technical validations are passed, CLM performs the business validations and sends a negative [Receipt \(camt.025\)](#) [▶ 397] to the CB in case the validation fails. If business validations pass without error, CLM continues processing the payment order.

- In case a reject time is not present, the payment is further processed in *Standard CLM settlement* (please refer to chapter [Standard CLM settlement](#) [▶ 266]).
- In case a reject time is present, CLM checks whether the given time is reached.
- If the reject time has not been reached yet, the payment is further processed in *Standard CLM settlement* (please refer to chapter [Standard CLM settlement](#) [▶ 266]).
- If the reject time has been reached, CLM rejects the payment and send a [PaymentStatusReport \(pacs.002\)](#) [▶ 487] to the CB.

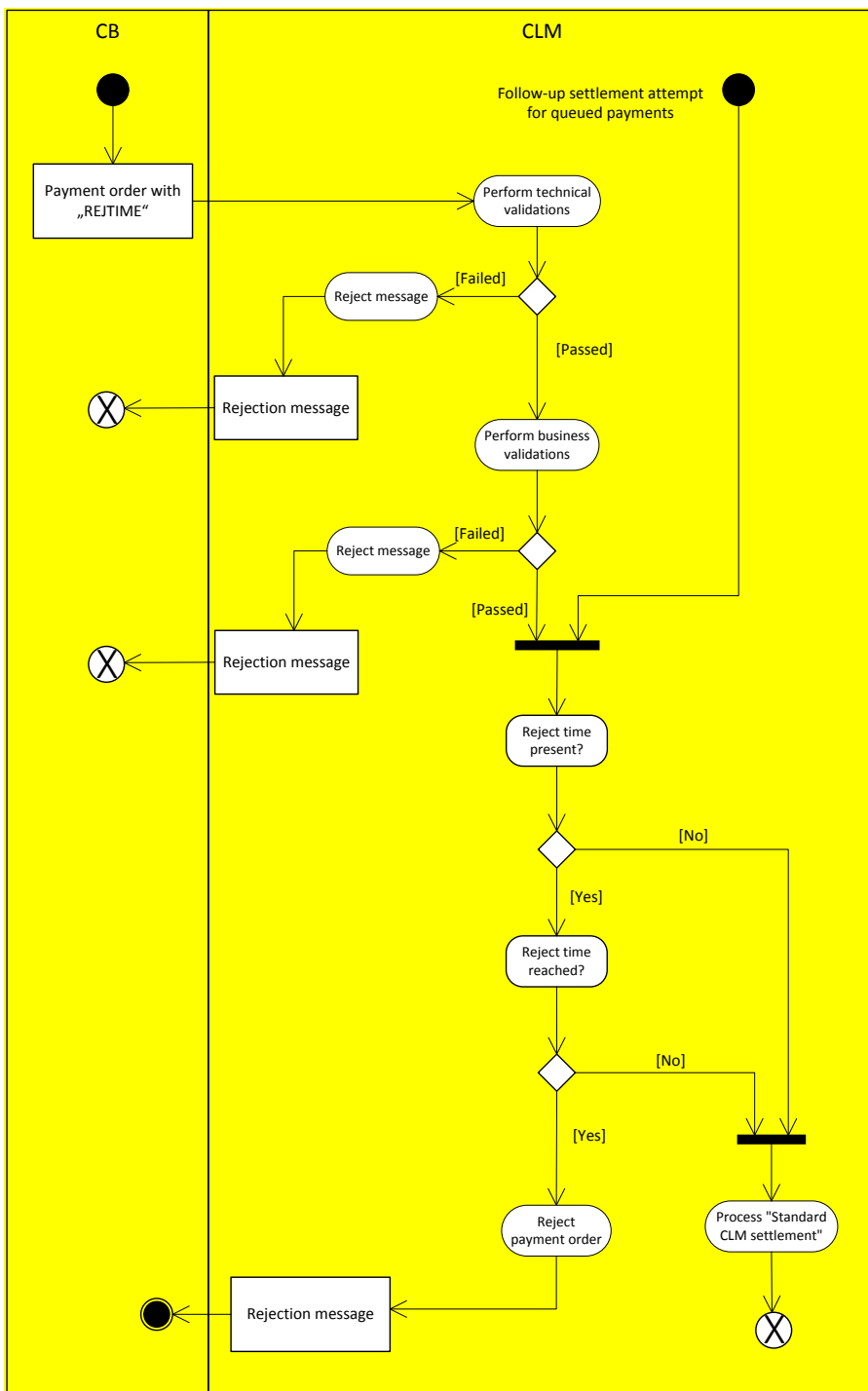


Figure 63 - Process reject time instructions

10.3.3 Settle connected payments

Description of process flow for connected payments

- CLM carries out the technical validations and sends a negative [ReceiptAcknowledgement \(admi.007 \[347\]](#) to the CB in case the validation fails.

If technical validations are passed, CLM performs the business validations and sends a negative [Receipt \(camt.025\)](#) [397] in case the validation fails. If business validations pass without error, CLM continues processing the modify credit line order. a connected payment order is processed as follows:

In case the MCA is credited, CLM continues with the processes “*Standard CLM settlement*” and “*Modify credit line*”decrease.

In case the MCA is debited, CLM continues with the processes “*Standard CLM settlement*” and “*Modify credit line*”increase.

Note: The amount booked on the MCA is not necessarily equal to the credit line change.

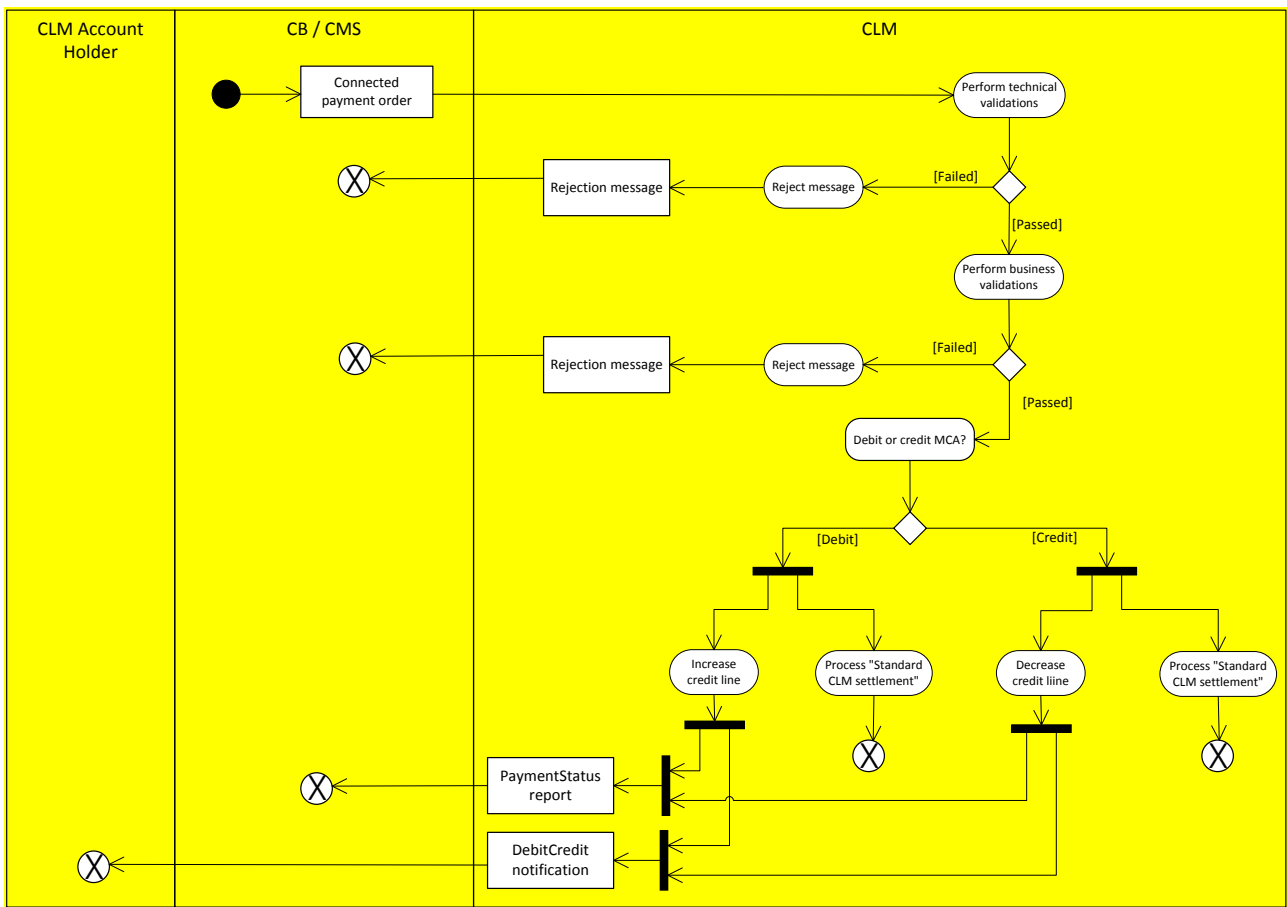


Figure 64 - Settle connected payment

10.3.4 Modify credit line

Process flow for increasing the credit line

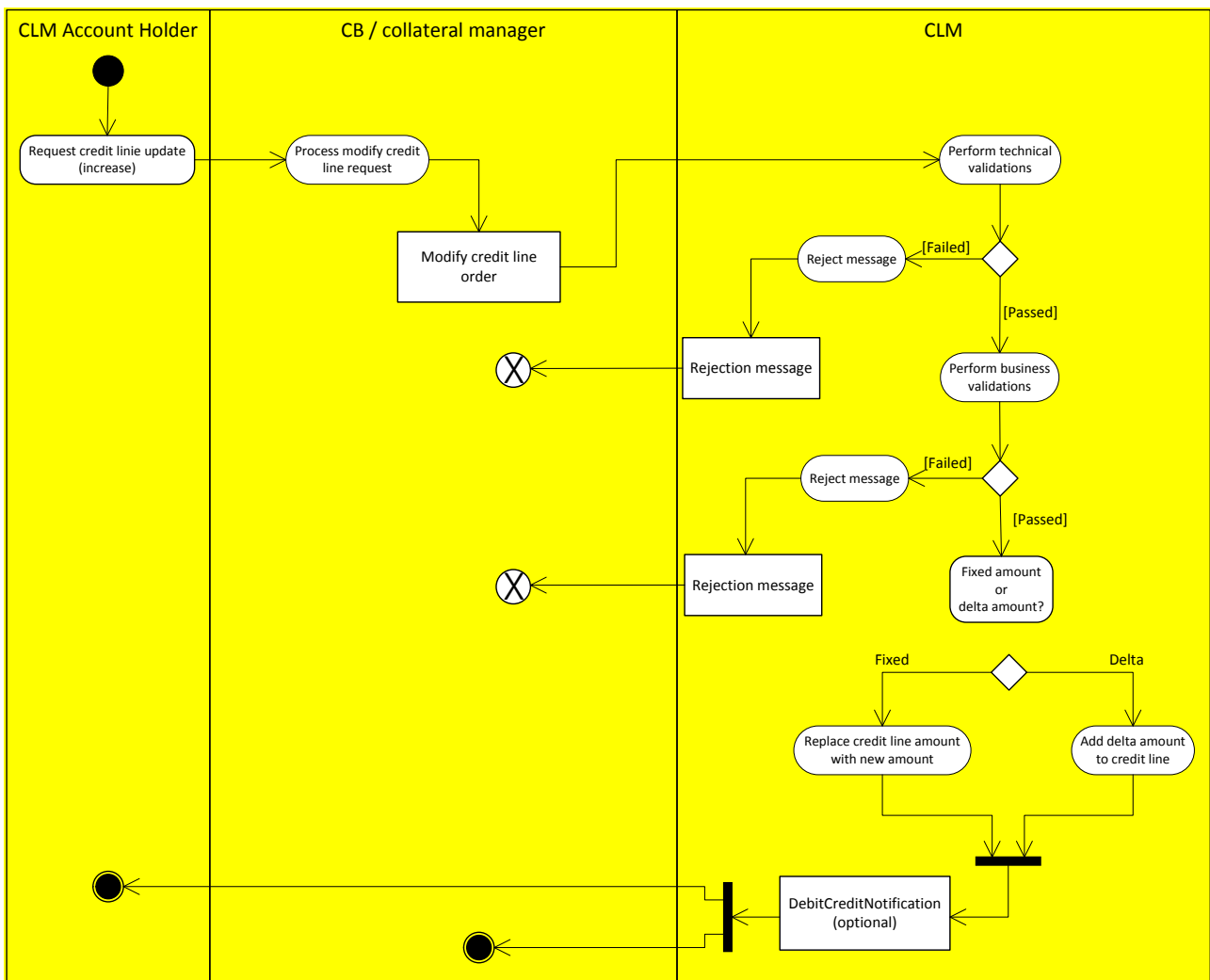


Figure 65 - Increase credit line

Description of process flow for increasing the credit line

- I The CLM Account Holder sends a request to its CB to update the credit line.
- I The CB submits the request via the collateral management system ⁴⁶ to CLM.
- I CLM carries out the technical validations and sends a negative [ReceiptAcknowledgement \(admi.007\)](#) [347] in case the validation fails.
- I If technical validations are passed, CLM performs the business validations and sends a negative [Receipt \(camt.025\)](#) [397] in case the validation fails. If business validations pass without error, CLM continues processing the modify credit line order.
- I CLM determines whether the credit line is updated by a fixed amount or by a delta amount.
 - In case it is updated by a fixed amount, CLM replaces the current credit line amount with the new one.

46 The national collateral management systems is replaced by ECMS as of end of 2022.

– In case it is updated by a delta amount, CLM adds the delta amount to the current credit line amount.

The CB receives a [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442] if it has opted for it in CRDM.

The respective participant receives a [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442] if he opted for it in CRDM.

Process flow for decreasing the credit line

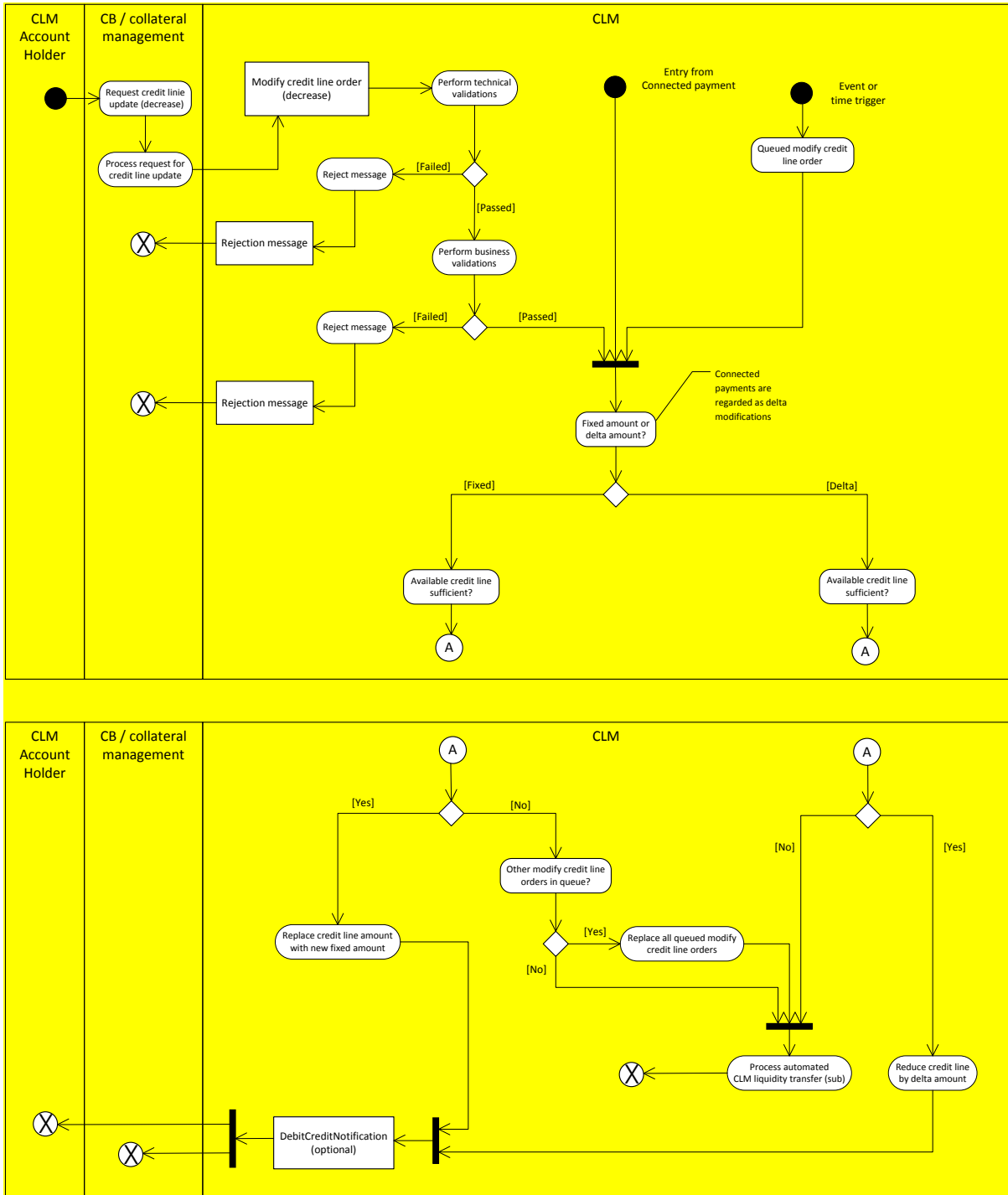


Figure 66 - Decrease credit line

Description of process flow for decreasing credit line

The CLM Account Holder sends a request to its CB to update the credit line.

- I The CB submits the request via the collateral management system ⁴⁷ to CLM
- I CLM carries out the technical validations and sends a negative [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347] in case the validation fails.
- I If technical validations are passed, CLM performs the business validations and sends a negative [Receipt \(camt.025\)](#) [▶ 397] in case the validation fails. If business validations pass without error, CLM continues processing the modify credit line order.
- I In case the credit line is modified to a new fixed amount, CLM checks if the available credit line is sufficient.
 - In case the available credit line is sufficient, the credit line is replaced with the new fixed amount and sends a positive [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442] to the CB/collateral management as well as to the CLM Account Holder if they have opted for it in CRDM.
 - In case the available credit line is not sufficient, CLM checks for other pending modify credit line orders.
 - If earlier modify credit line orders are already pending, CLM replaces all pending orders with the new fixed amount and continues with the process “automated CLM liquidity transfer”.
 - If no earlier modify credit line orders are pending, CLM continues with the process “automated CLM liquidity transfer”.
- I In case credit line is modified by a delta amount, CLM checks if the available credit line is sufficient.
 - In case the available credit line is sufficient, CLM subtracts the delta amount from the current credit line amount and sends a positive [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442] to the CB/collateral management as well as to the CLM Account Holder if they have opted for it in CRDM.
- I In case the available credit line is not sufficient, CLM continues the with the process “automated CLM liquidity transfer”.

⁴⁷ The national collateral management systems is replaced by ECMS as of end of 2022.

10.3.5 Settle marginal lending on request

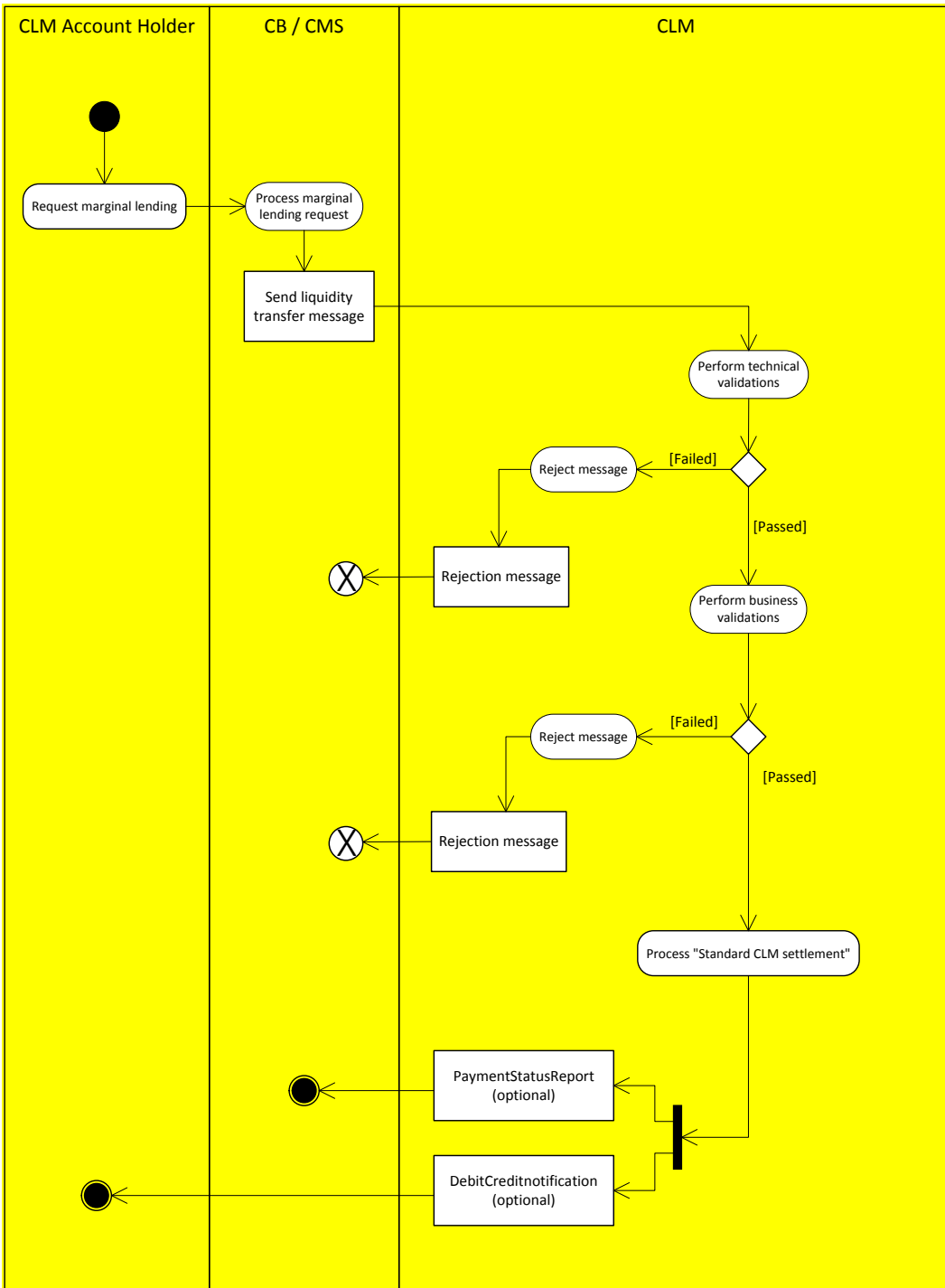


Figure 67 - Setup marginal lending on request

Description of process flow for setting up marginal lending on request

The CLM Account Holder requests a marginal lending at his respective CB. The request is not done by a message, but rather via phone, fax or other appropriate means.

-
- I The CB processes the request and sends a [FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491] via its collateral management system to CLM.
 - I CLM carries out the technical validations and sends a negative [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347] to the CB in case the validation fails.
 - I If technical validations are passed, CLM performs the business validations and sends a negative [Receipt \(camt.025\)](#) [▶ 397] to the CB in case the validation fails.
 - If business validations pass without error, CLM continues processing the payment order with the standard CLM settlement.
 - I The CB receives a [PaymentStatusReport \(pacs.002\)](#) [▶ 487] if it has opted for it in CRDM.
 - I The respective CLM Account Holder receives a [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442] if he opted for it in CRDM.

Process flow for setting up marginal lending on request with ECMS

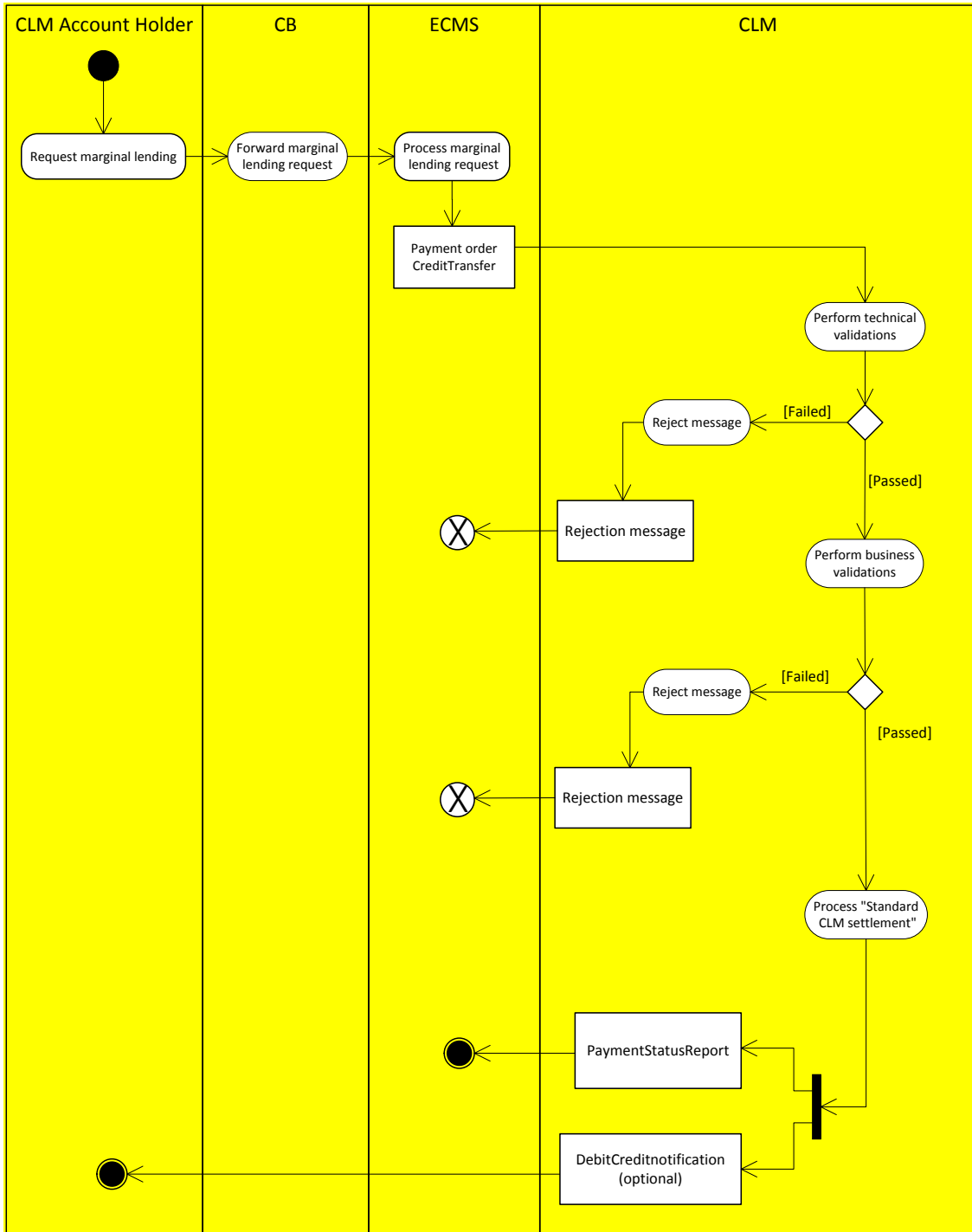


Figure 68 - ECMS - setup marginal lending on request

Description of process flow for setting up marginal lending on request with ECMS

- I The CLM Account Holder requests a marginal lending at his respective CB via phone, fax or other appropriate means.
- I The CB processes the request and sends a liquidity transfer message via ECMS to CLM.

CLM carries out the technical validations and sends a negative [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347] to ECMS in case the validation fails.

If technical validations are passed, CLM performs the business validations and sends a negative [Receipt \(camt.025\)](#) [▶ 397] to ECMS in case the validation fails.

- If business validations pass without error, CLM continues processing the payment order with the standard CLM settlement.

ECMS receives a [PaymentStatusReport \(pacs.002\)](#) [▶ 487] if it has opted for it in CRDM.

The respective account holder receives a [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442] if he opted for it in CRDM.

10.3.6 Reservation management

Description of process flow for reservation management

There are three possible entry points for reservation orders to CLM.

Entry point 1: The CLM Account Holder sends a [ModifyReservation \(camt.048\)](#) [▶ 414] or a [DeleteReservation \(camt.049\)](#) [▶ 416] to CLM

Entry point 2: CLM generates a standing order reservation which was set-up in CRDM by the account holder

Entry point 3: Queued modify reservation orders

CLM carries out the technical validations and sends a negative [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347] to the CLM Account Holder in case the validation fails. Technical validations are neither carried out for standing order reservation nor for queued reservation orders.

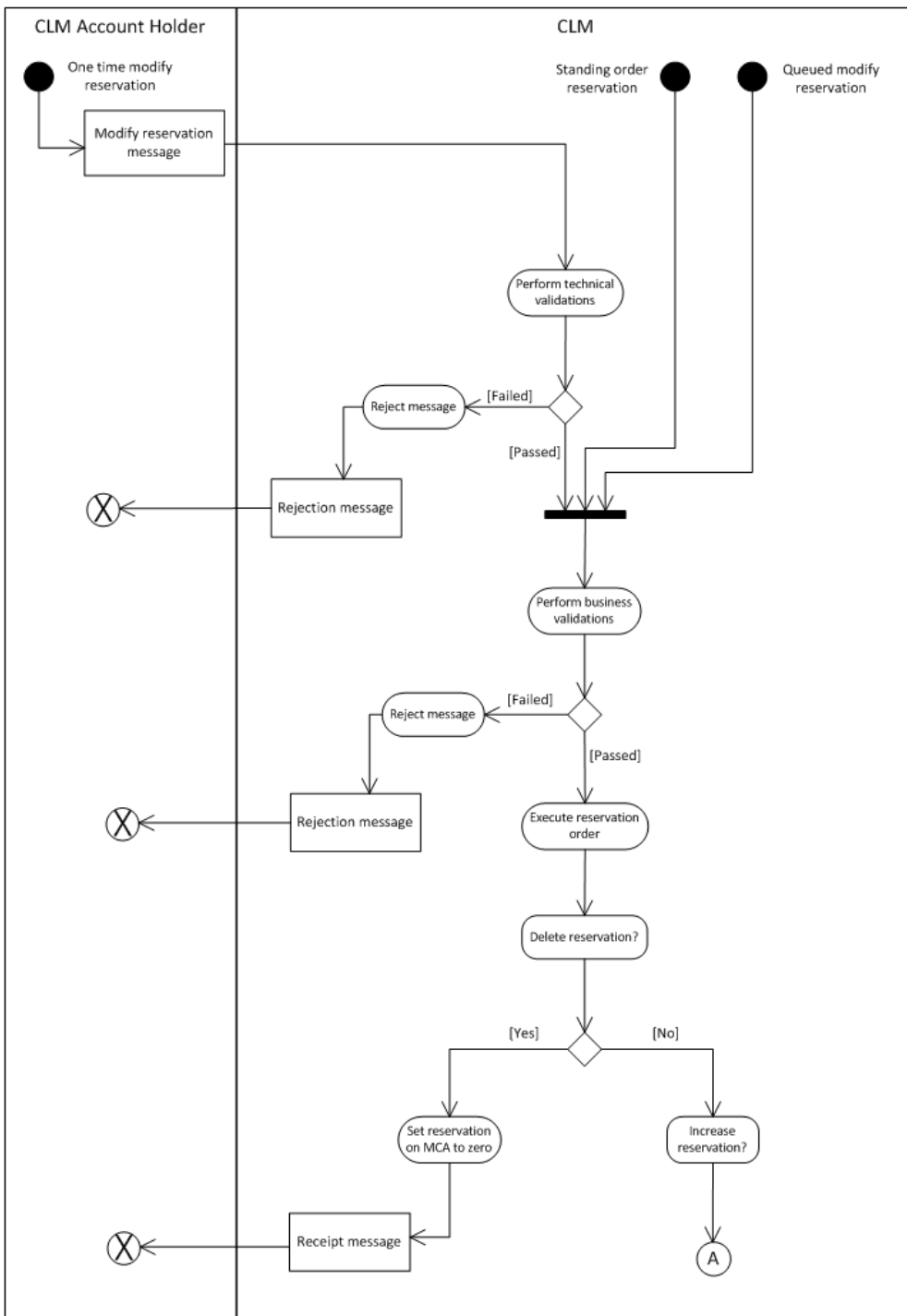
If technical validations are passed, CLM performs the business validations and sends a negative [Receipt \(camt.025\)](#) [▶ 397] to the CLM Account Holder in case the validation fails. If business validations pass without error, CLM continues processing the reservation management order.

In case of a [DeleteReservation \(camt.049\)](#) [▶ 416], CLM sets the respective reserved liquidity on the MCA to zero.

In case of a [ModifyReservation \(camt.048\)](#) [▶ 414], there are two possible variations.

- In case the CLM Account Holder wants to increase the reservation amount, CLM checks the available unreserved liquidity on the MCA.
- If the unreserved liquidity is sufficient, CLM increases the reserved liquidity on the MCA by the requested amount and sends a [Receipt \(camt.025\)](#) [▶ 397] to the CLM Account Holder.
- If the unreserved liquidity is not sufficient, CLM increases the reserved liquidity on the MCA by the available amount and queues the remaining part. CLM sends a [Receipt \(camt.025\)](#) [▶ 397] for the settled part to the CLM Account Holder.

- In case the CLM Account Holder wants to decrease the reservation amount, CLM checks the available reserved liquidity on the MCA.
- If the reserved liquidity is sufficient, CLM decreases the reserved liquidity on the MCA by the requested amount and send a [Receipt \(camt.025\)](#) [▶ 397] to the CLM Account Holder.
- If the reserved liquidity is not sufficient, CLM decreases the reserved liquidity on the MCA by the available amount and queues the remaining part. CLM sends a [Receipt \(camt.025\)](#) [▶ 397] for the settled part to the CLM Account Holder.



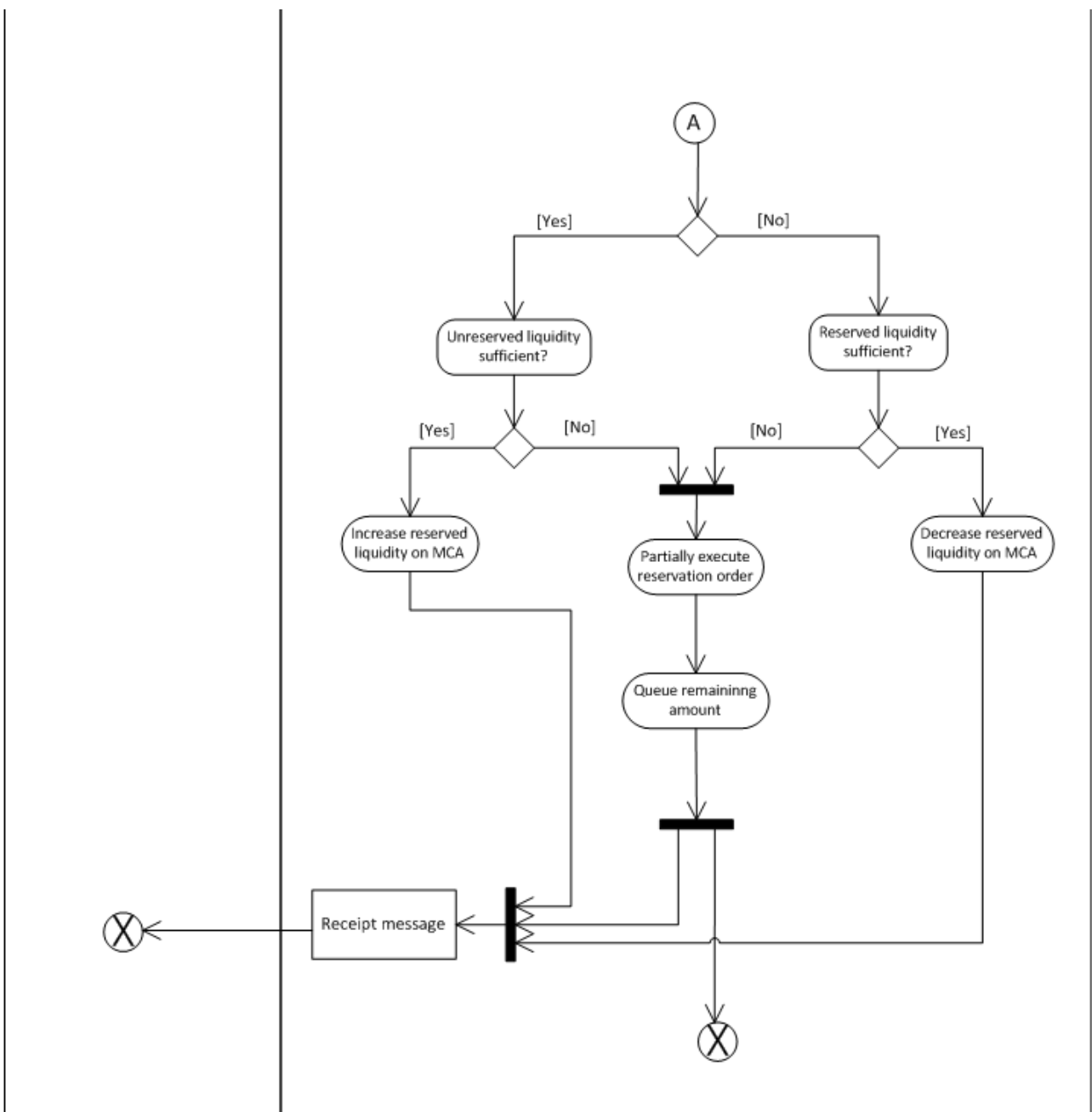


Figure 69 - Reservation management

10.4 Run CLM EoD processing

10.4.1 Reject payments (EoD)

If payments are still queued by the EoD due to lack of available liquidity, these payments are rejected during the EoD processing (with the exception of standing facilities that shall be executed before their dedicated cut-off).

The sending CB of the related inbound payment message is notified by negative [PaymentStatusReport \(pacs.002\)](#) [- 487].

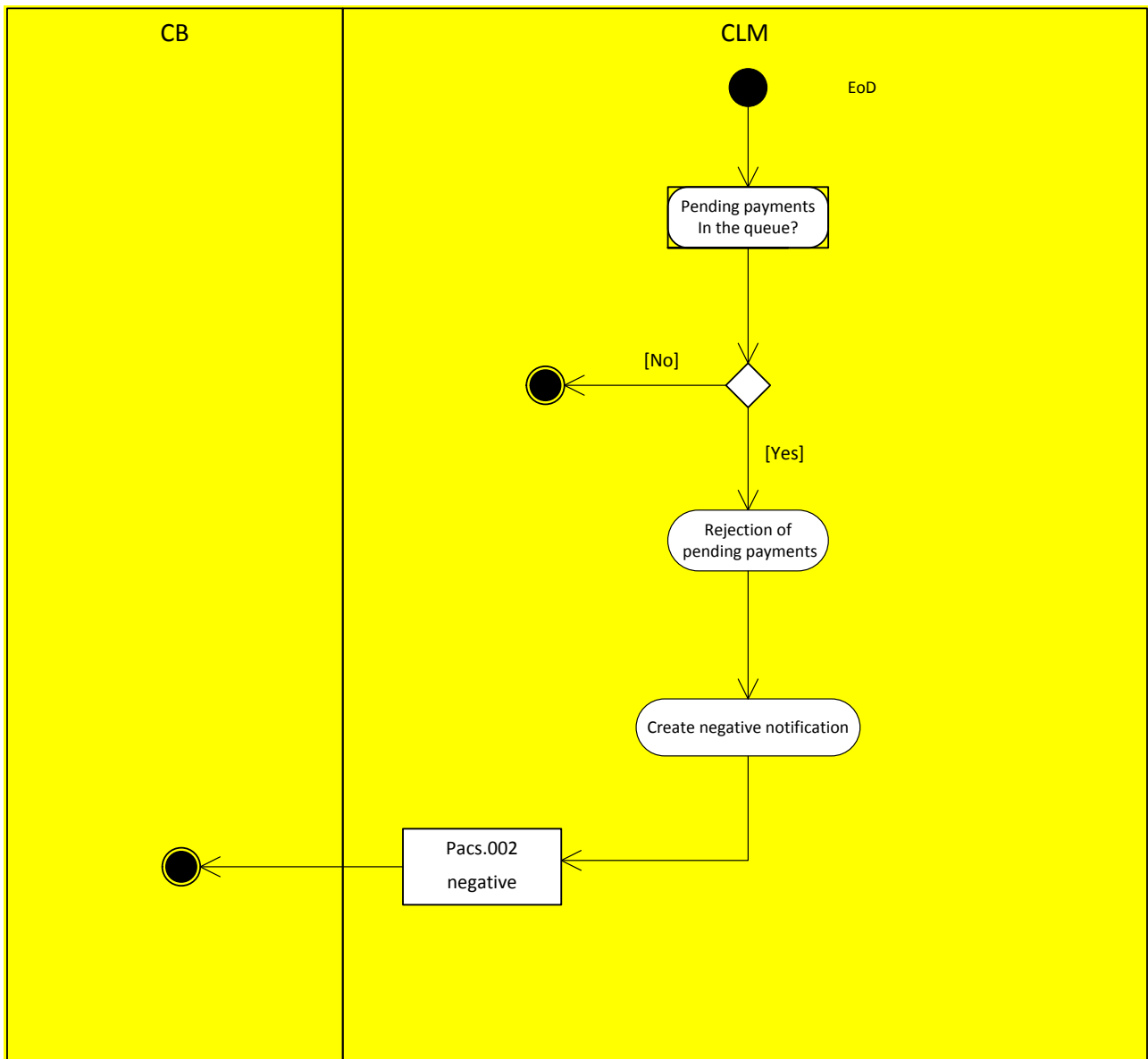


Figure 70 - Reject payments (EoD)

10.4.2 Process automated marginal lending

- | The process starts with the availability of all general ledger files from all other services/components respectively.
- | CLM calculates the aggregated EoD balance for each of their account holders.
 - In case the balance is zero or positive the process ends without further action.
- | In case the balance is negative CLM checks whether the account holder is eligible to the marginal lending facility.

-
- In case the participant is NOT eligible, the CLM sends a [ReturnAccount \(camt.004\)](#) [355] - for CBs only - to inform the CB/collateral management system about the negative account balance.
 - The CB starts the penalty procedure.
 - | In case the participant is eligible
 - CLM creates an internal connected payment ⁴⁸ for further processing.
 - CLM starts the process "[Settle connected payments](#) [▶ 274]".

⁴⁸ "Internal" reflects the fact that no messages are created or sent, but the flow is triggered internally.

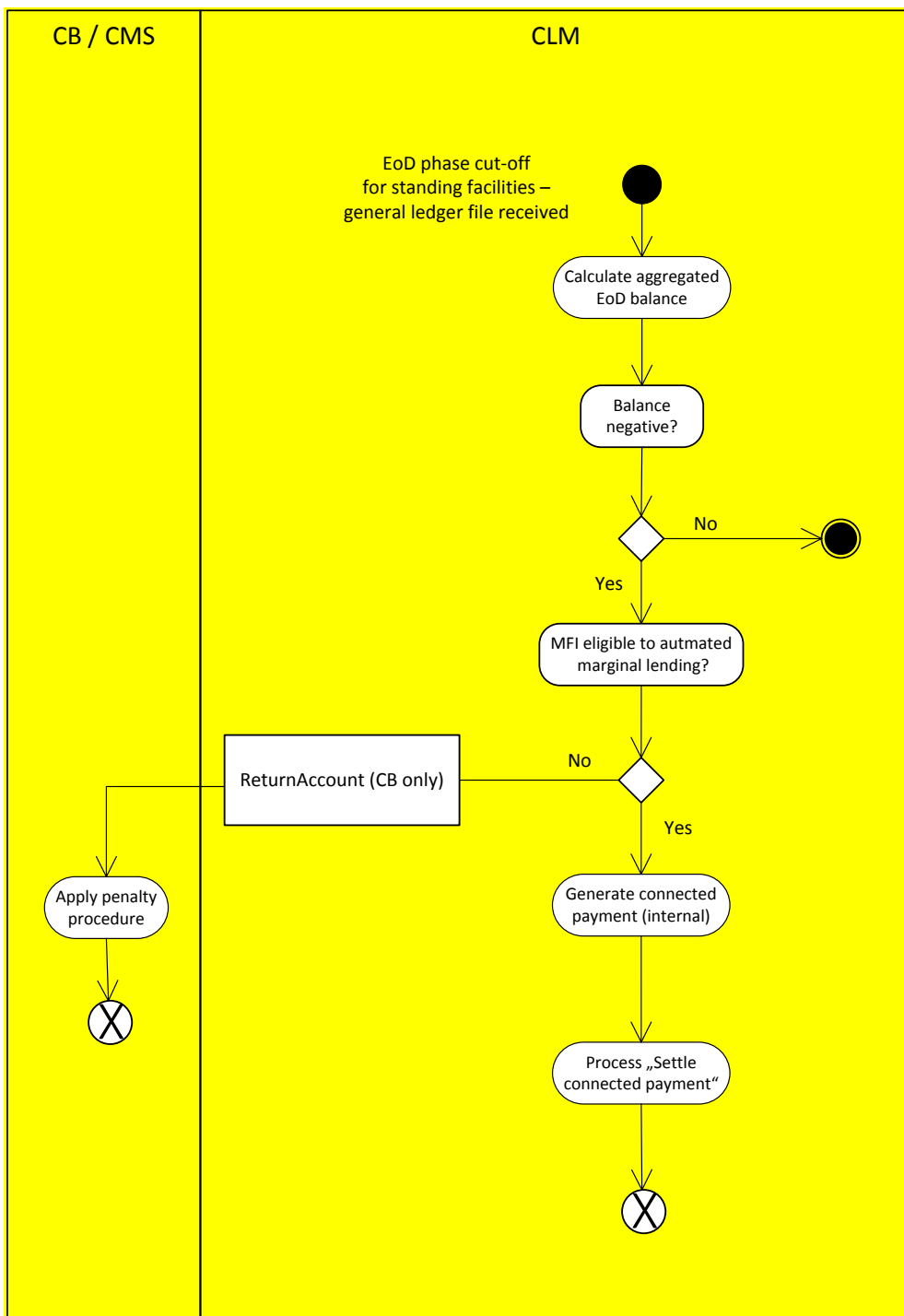


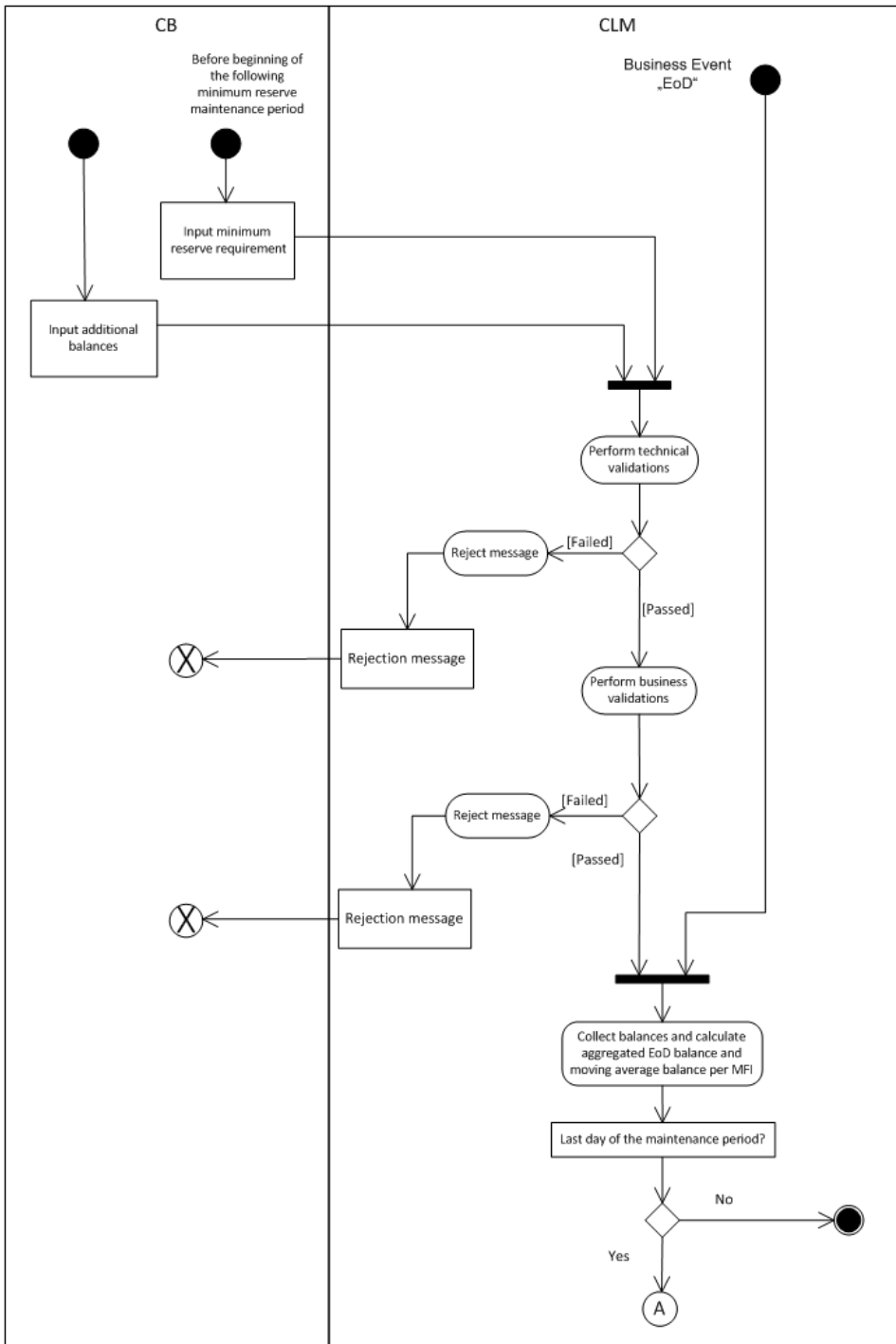
Figure 71 - Process automated marginal lending

10.4.3 Process minimum reserve

This is a general description of the CLM process “minimum reserve processing”. CLM calculates the aggregated EoD balances (including any additional balances sent by the CB) on a daily basis and periodically verifies the minimum reserve fulfillment for each MFI. The fulfillment is based on the minimum reserve re-

quirement, which was sent to CLM by the CB before the beginning of the current minimum reserve maintenance period.

The chapter [Reserve management](#) [143] describes the respective business scope.



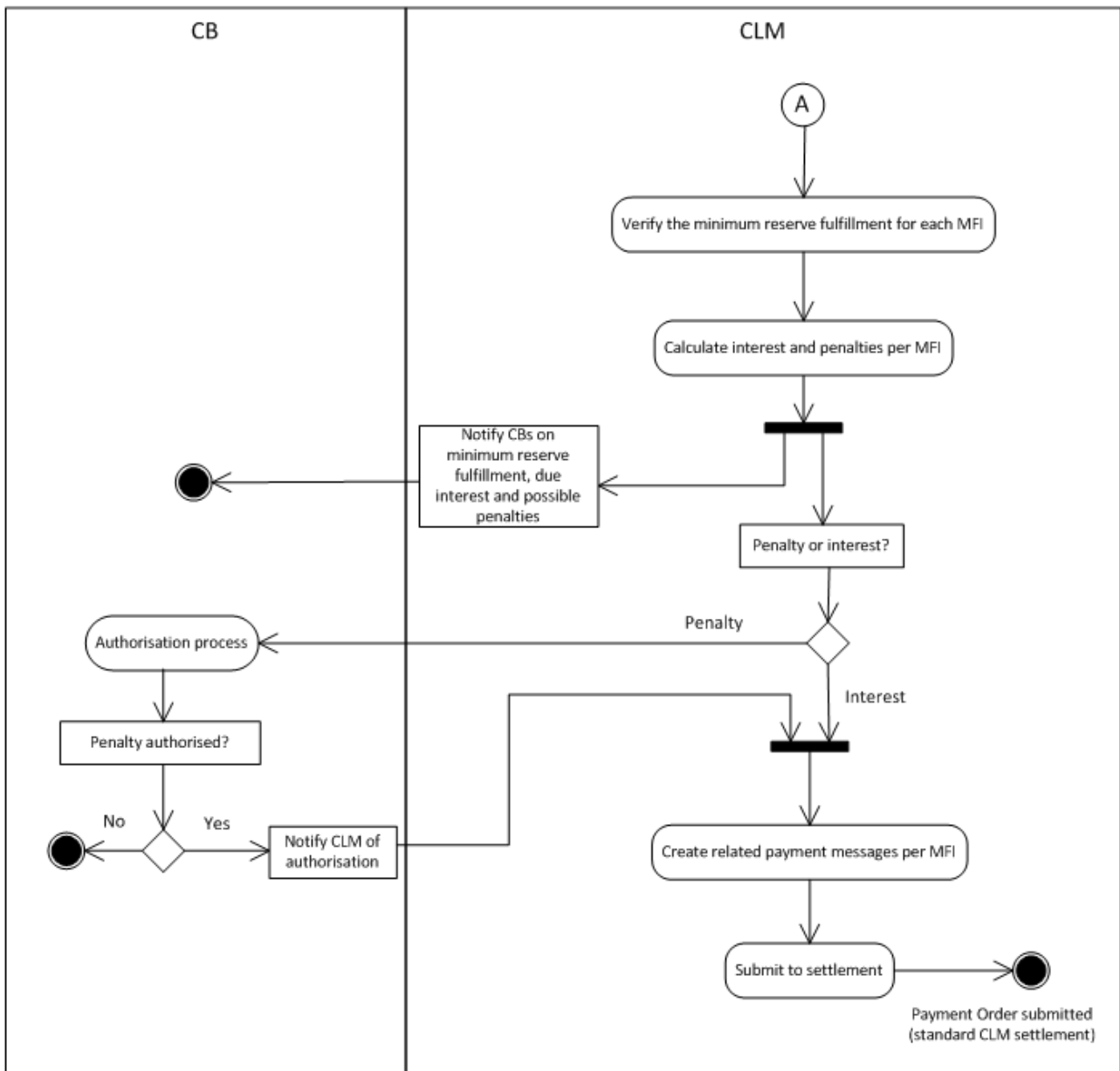


Figure 72 - Process minimum reserve

CLM reserve management calculation

The minimum reserve requirement is submitted to CLM by the CB per institution subject to the Eurosystem’s minimum reserve system before the beginning of the following minimum reserve maintenance period with a [InsertBalance_RM \(camt.998\)](#) [▶ 481]. Additional balances are optionally (i.e. when applicable) submitted to CLM by the CB with a [InsertBalance_RM \(camt.998\)](#) [▶ 481]. Please refer to chapter [Reserve management - specific functions for CBs](#) [▶ 251] for more information.

CLM carries out the technical validations and returns a negative [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347] in case the validation fails.

If technical validations are passed, CLM performs the business validations and returns a negative [Receipt \(camt.025\)](#) [397] in case the validation fails. If business validations pass without error, CLM continues processing the respective reserve management message.

The defined business event EoD triggers on a daily basis the process “Collect balances and calculate aggregated EoD balance and moving average balance per MFI”. On the last day of the reserve maintenance period CLM verifies the minimum reserve fulfillment for each MFI”, calculates interest and penalties per MFI and notifies the CB on its minimum reserve fulfillment, due interest and possible penalties. Out of the resulting interest and penalties per MFI, CLM “creates a related payment message per MFI” and submits it to settlement. Penalties have to be authorised by the CB prior to the payment order being created and submitted to settlement.

Message name	Code	Type
SendPeriodicFlow_RM	camt.998 - SendPeriodicFlow_RM [533]	SendPeriodicFlow

Table 101 - Notification message sent by the CLM component to CB

10.5 CLM SoD processing

10.5.1 Reimburse marginal lending

The reimbursement of marginal lending consists of two integrated processes:

1. the actual reimbursement of the lending amount and
2. the calculation and posting of interest

The process starts automatically with the business day phase “Begin of CB operations” at 19:00. CLM calculates the interest and creates an internal payment for further processing in the “*Standard CLM settlement*” process (see chapter [Standard CLM settlement](#) [266]).

The reimbursement of the lending amount is handled in two different ways, depending on the origin of the marginal lending.

- In case of a marginal lending on request, CLM creates an internal payment order for further processing in the “*Standard CLM settlement*” process (see chapter [Standard CLM settlement](#) [266]).
- In case of an automated marginal lending, CLM creates an internal connected payment order for further processing in the “[Settle connected payments](#) [274]”.

To inform the CB or its collateral management respectively about the reimbursed marginal lending, it receives a [ReturnAccount \(camt.004\)](#) [355] (push, for CBs only).

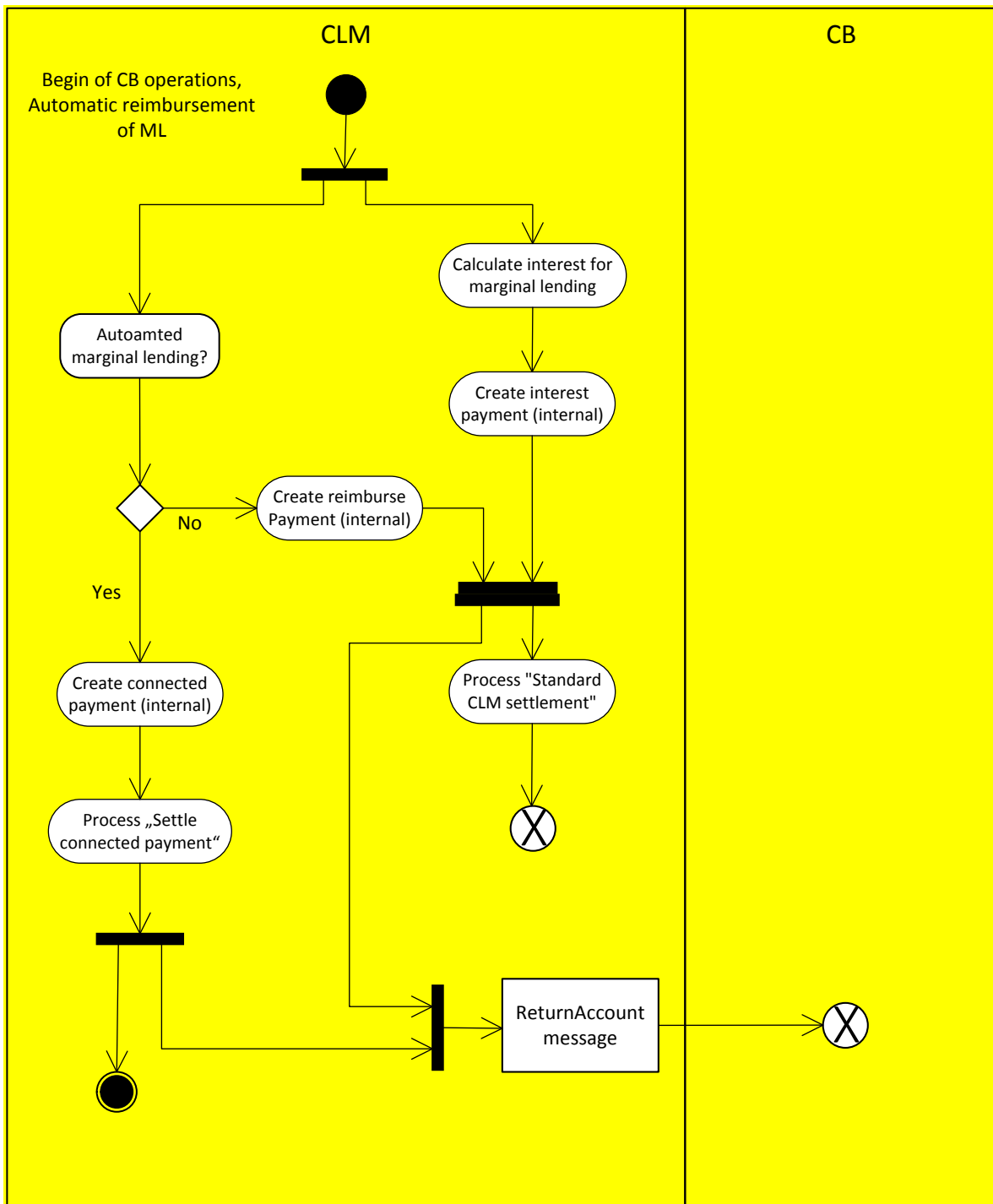


Figure 73 - Reimburse marginal lending

10.5.2 Process overnight deposit

- I An overnight deposit is initiated by the CLM Account Holder by sending a [LiquidityCreditTransfer \(camt.050\)](#) [418] to CLM.
- I CLM carries out the technical validations and sends a negative [ReceiptAcknowledgement \(admi.007\)](#) [347] to the CB in case the validation fails.

If technical validations are passed, CLM performs the business validations and sends a negative [Receipt \(camt.025\)](#) [▶ 397] to the CB in case the validation fails. If business validations pass without error, CLM continues processing the overnight deposit.

CLM checks whether the CLM Account Holder is eligible for the overnight deposit facility.

- In case the participant is not eligible, CLM rejects the liquidity transfer order and sends a negative [receipt \(camt.025\)](#) [▶ 397] to the account holder.
- In case the participant is eligible, CLM proceeds with the “*Standard CLM settlement*” process (please refer to chapter [Standard CLM settlement](#) [▶ 266])

CLM sends optional (if opted for in CRDM) [BankToCustomerDebitCreditNotification \(camt.054\)](#) [▶ 442] to the account holder and to the CB.

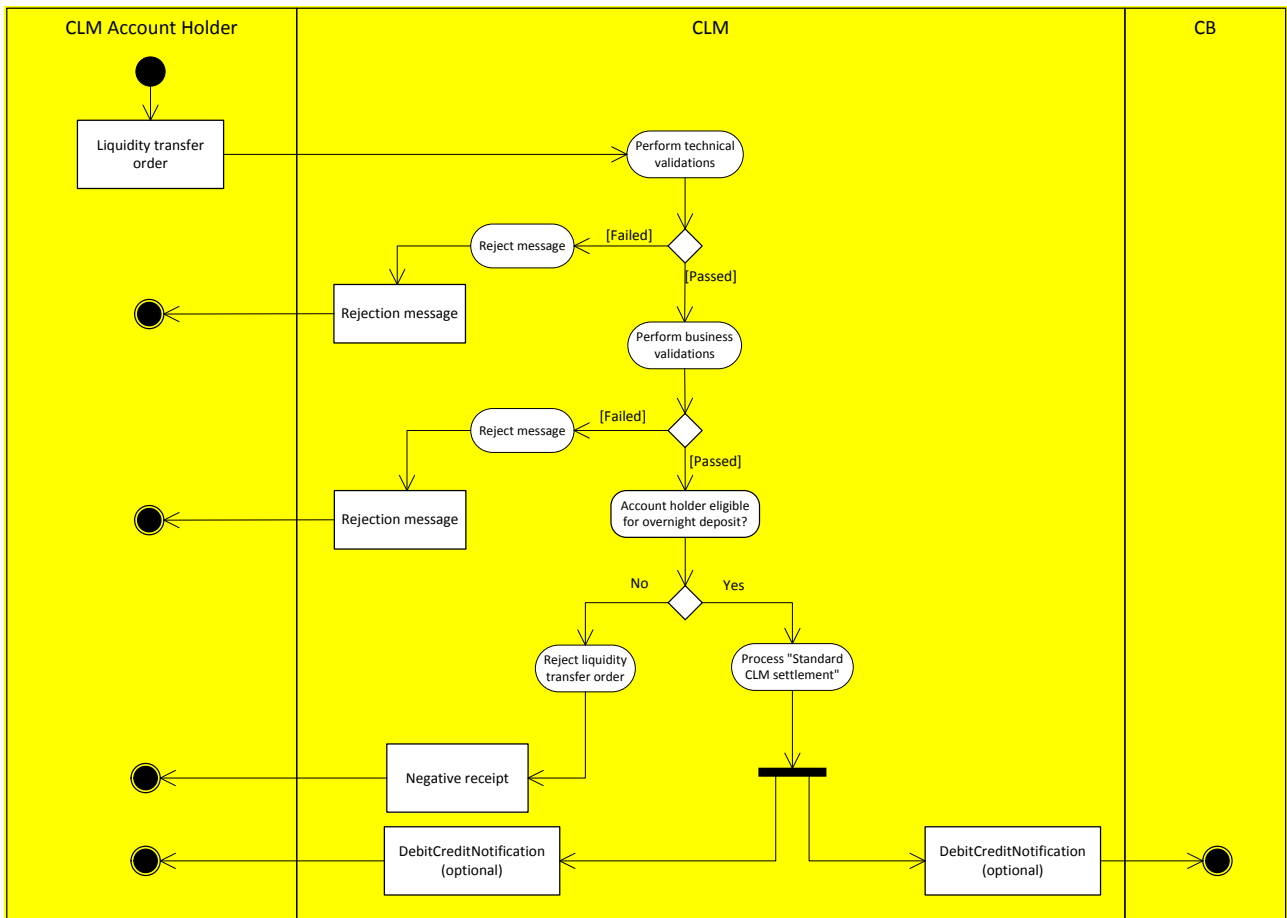


Figure 74 - Process overnight deposit

10.5.3 Revalidate warehoused payments at SoD

Basics

Warehoused payments are stored in CLM with a certain payment order 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.

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 authorization on the involved accounts still exist
- | 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](#) [82])

Further checks as described in chapter [Entry disposition](#) [▶ 106]

Technical validations like schema validations are only carried out on message level on the submission day. The same is valid for the duplicate payment order check. They are not repeated at SoD.

Processing on the intended settlement day

On the intended settlement date with the start of the processing time the warehoused payments are processed like described in Standard CLM settlement (see chapter [Standard CLM settlement](#) [▶ 266]). Exception: warehoused payments with a set execution from time indicator which has not been reached are set to status "Earmarked".

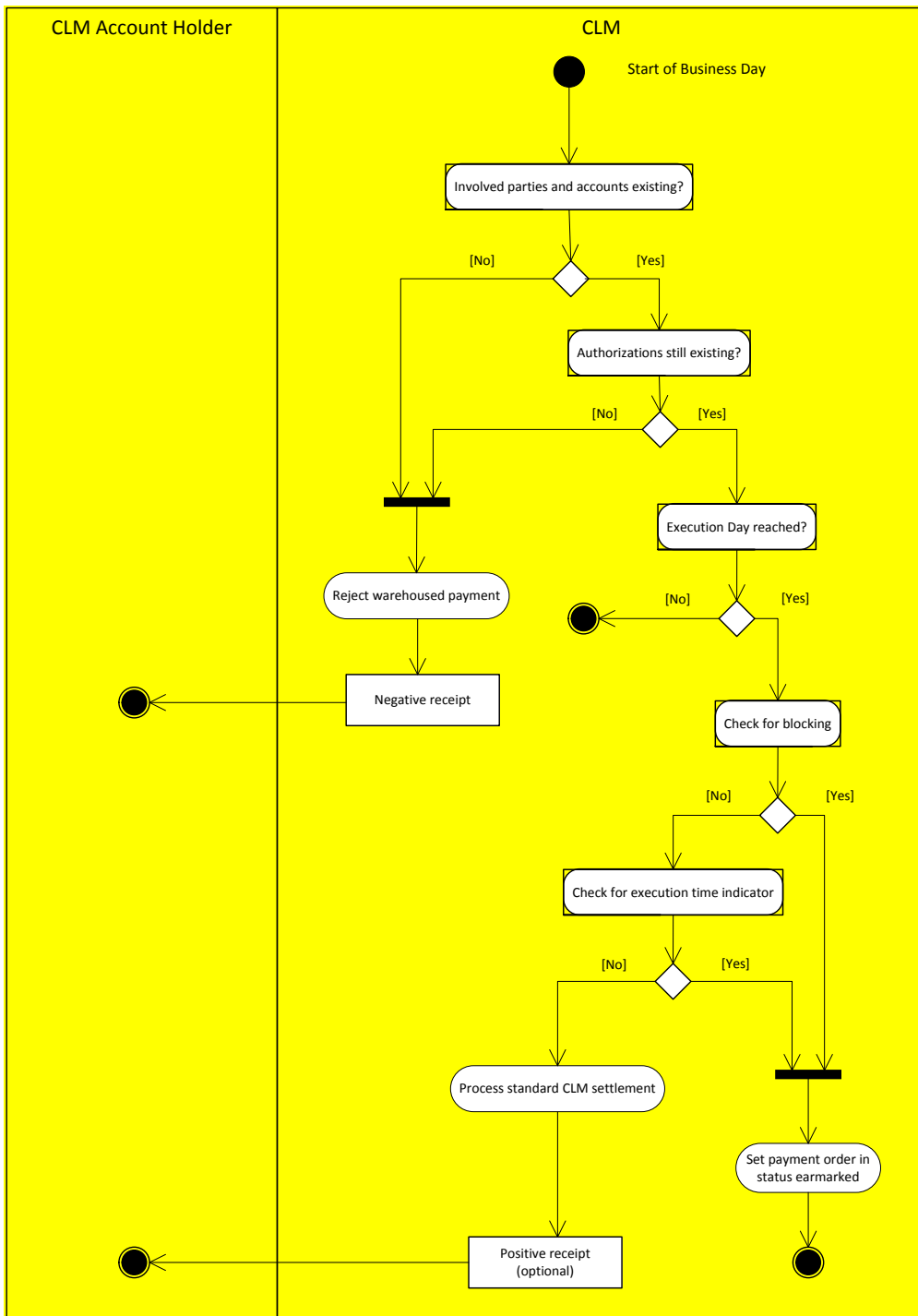


Figure 75 - Revalidate warehoused payments at SoD

10.6 Reference data management

10.6.1 Maintain local reference data object - maintain reservation

This is a general description of the CLM process "maintain current reservation". The submitting actor sends a

modify reservation request ([ModifyReservation \(camt.048\)](#) [▶ 414]) to CLM in which he instructs CLM to create a new or modify an existing reservation or a

delete reservation request ([DeleteReservation \(camt.049\)](#) [▶ 416]) in which he instructs CLM to delete an existing reservation.

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 ([ReceiptAcknowledgement \(admi.007\)](#) [▶ 347]) 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 \(camt.025\)](#) [▶ 397]) is sent to the submitting actor on mandatory basis.

[Ok] If validation was successful, the request is being processed by CLM.

Process create/modify/delete reservation

The request is processed by CLM.

[Ok] In case of successful processing a respective confirmation message ([camt.025](#) [▶ 397]) is sent to the submitting actors.

Note: Please see chapter [Liquidity reservation](#) [▶ 131] for details on the reservation feature.

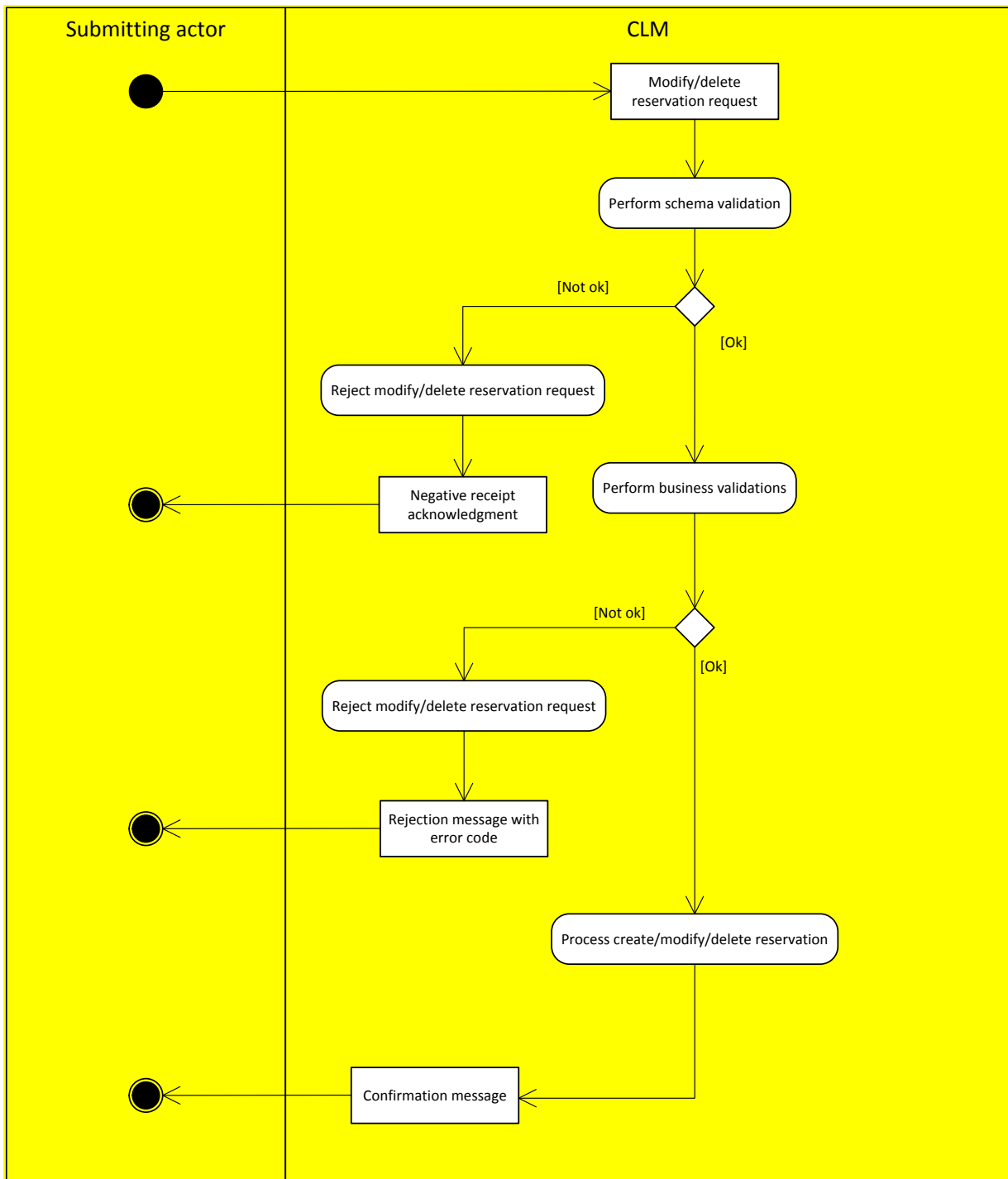


Figure 76 - Maintain reservation CLM

10.7 Information services

10.7.1 Execute query

This is a general process description for query requests to CLM in A2A mode, which is similar in all components. 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 CLM, CRDM, scheduler and billing](#) [▶ 177] describes the respective business scope. Concerning information on CB specific queries please see chapter [Query management - CB specific queries](#) [▶ 249].

The following activity diagram provides respective processes in the context of CLM.

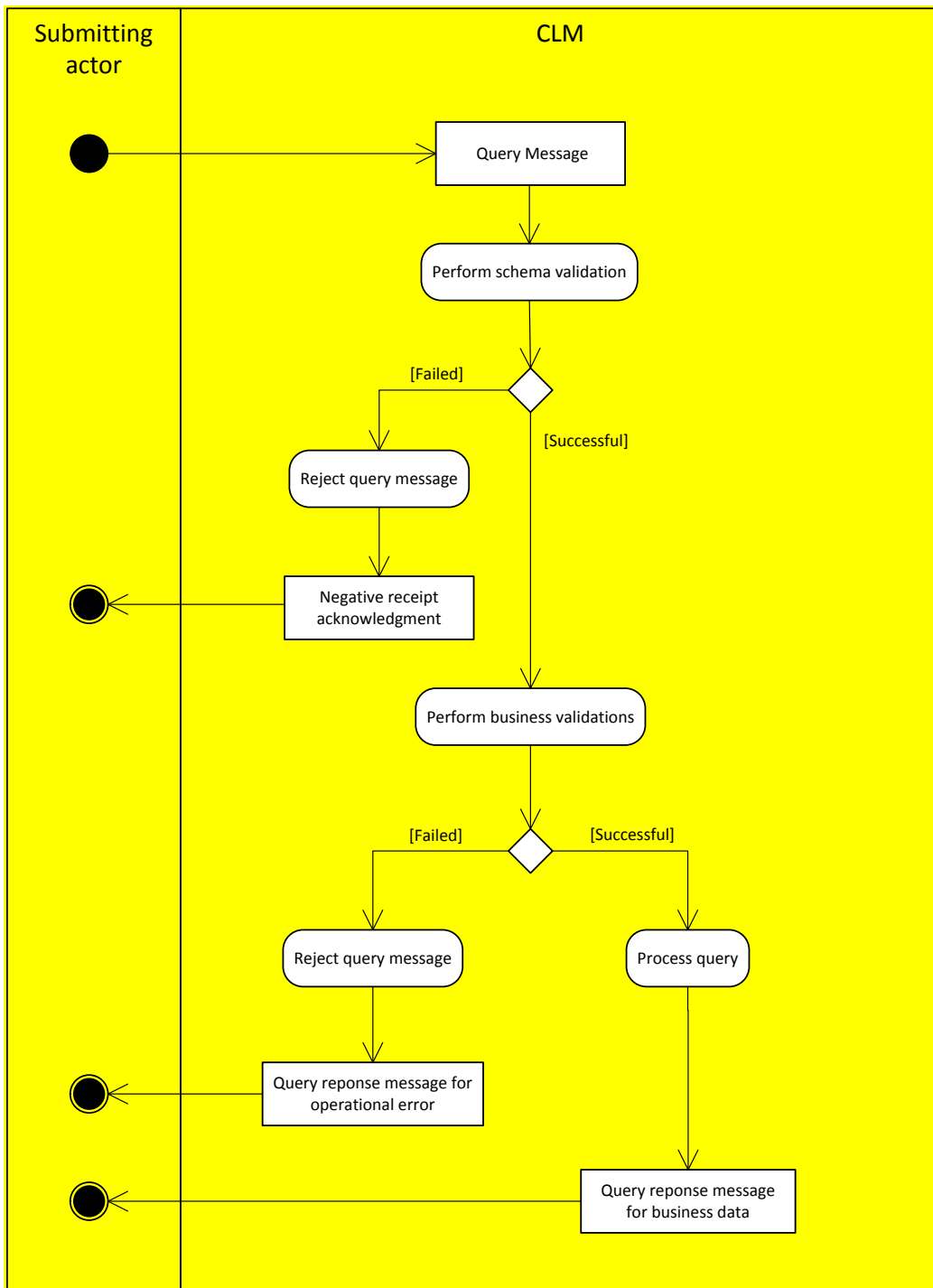


Figure 77 - CLM execute query

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 \(admi.007\)](#) [▶ 347] message [Negative Receipt Acknowledgment] 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 the following table “A2A messages for query processing” below) 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 statement query	ReportQueryRequest (admi.005) [▶ 345]	ReceiptAcknowledgement (admi.007) [347]	BankToCustomerStatement (camt.053) [425]
Audit trail for CLM query	AuditTrailQuery (camt.097) [468]	AuditTrailReport (camt.098) [▶ 470]	AuditTrailReport (camt.098) [▶ 470]
Available liquidity CLM query	GetAccount (camt.003) [350]	ReturnAccount (camt.004) [▶ 355]	ReturnAccount (camt.004) [▶ 355]
Cash transfer query	GetTransaction (camt.005) [368]	ReturnTransaction (camt.006) [378]	ReturnTransaction (camt.006) [378]
Current reservations query	GetReservation (camt.046) [409]	ReturnReservation (camt.047) [411]	ReturnReservation (camt.047) [411]
Event query	GetBusinessDayInformation (camt.018) [▶ 386]	ReturnBusinessDayInformation (camt.019) [388]	ReturnBusinessDayInformation (camt.019) [388]
Minimum reserve query	GetAccount (camt.003) [350]	ReturnAccount (camt.004) [▶ 355]	ReturnAccount (camt.004) [▶ 355]
Minimum reserve of a banking community query (CB only)	camt.998 - GetCompulsoryReserve_RM [▶ 528]	camt.998 - ReturnCompulsoryReserve_RM [530]	camt.998 - ReturnCompulsoryReserve_RM [530]

Query type	Query request message	Query response message for operational error	Query response message for business data
Minimum reserve requirements per participant query(CB only)	GetAccount (camt.003) [350]	ReturnAccount (camt.004) [355]	ReturnAccount (camt.004) [355]
Penalty query(CB only)	camt.998 - ReturnPenalty_RM [523]	camt.998 - ReturnPenalty_RM [523]	camt.998 - ReturnPenalty_RM [523]
System time query	GetBusinessDayInformation (camt.018) [386]	ReturnBusinessDayInformation (camt.019) [388]	ReturnBusinessDayInformation (camt.019) [388]
Standing facilities transaction of the respective banking community query(CB only)	GetTransaction (camt.005) [368]	ReturnTransaction (camt.006) [378]	ReturnTransaction (camt.006) [378]

Table 102 - A2A messages for query processing

10.7.2 Receive report

This is a general description of the CLM process “Receive report” in push mode. CLM uses reports to periodically provide CLM Actors with a defined set of data according to their data scope and access rights.

The chapter [CLM report generation](#) [174] describes the respective business scope.

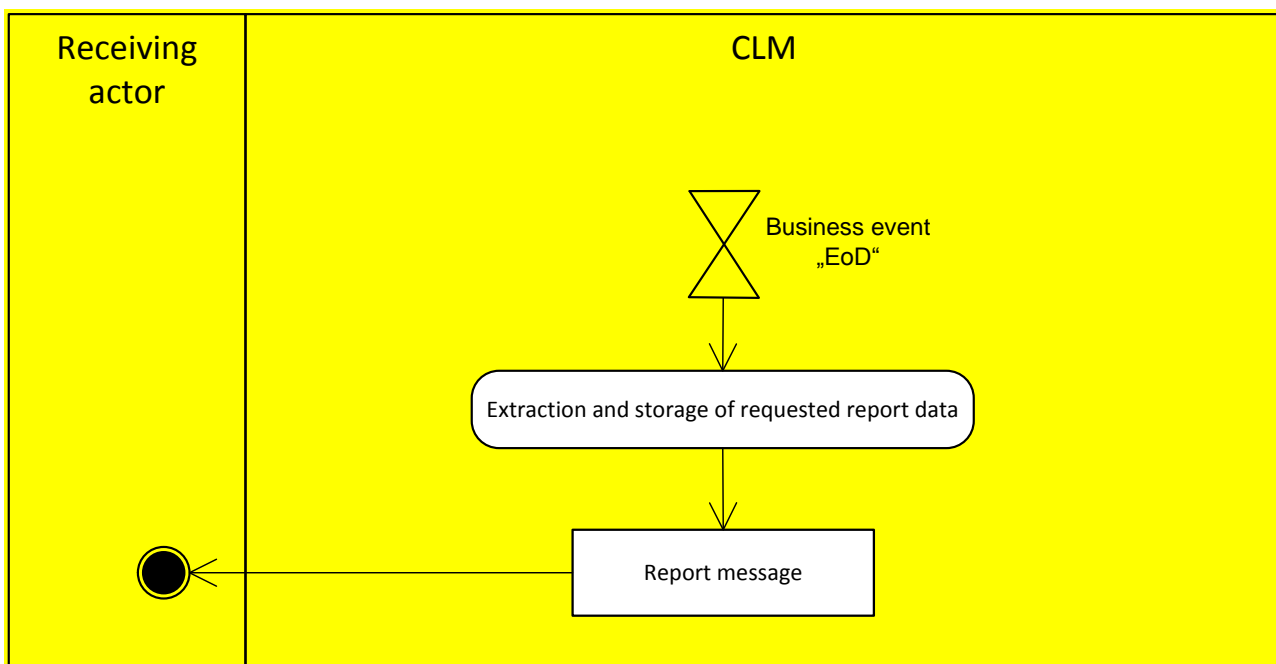


Figure 78 - Receive report

The defined business event 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 CLM cash account. For general ledger files it provides the report on mandatory basis to the CBs (see chapter [General ledger handling](#) [246]). CLM creates the report, including the execution of necessary calculations and storing the report for further processing. CLM 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) [425]
General ledger (CB only)	BankToCustomerStatement	BankToCustomerStatement (camt.053) [425]

Table 103 - Receive report

10.7.3 Receive system notification

This is a general description of the CLM process “Receive system notification”. CLM uses system notifications to regularly provide CLM Actors with a defined set of business events.

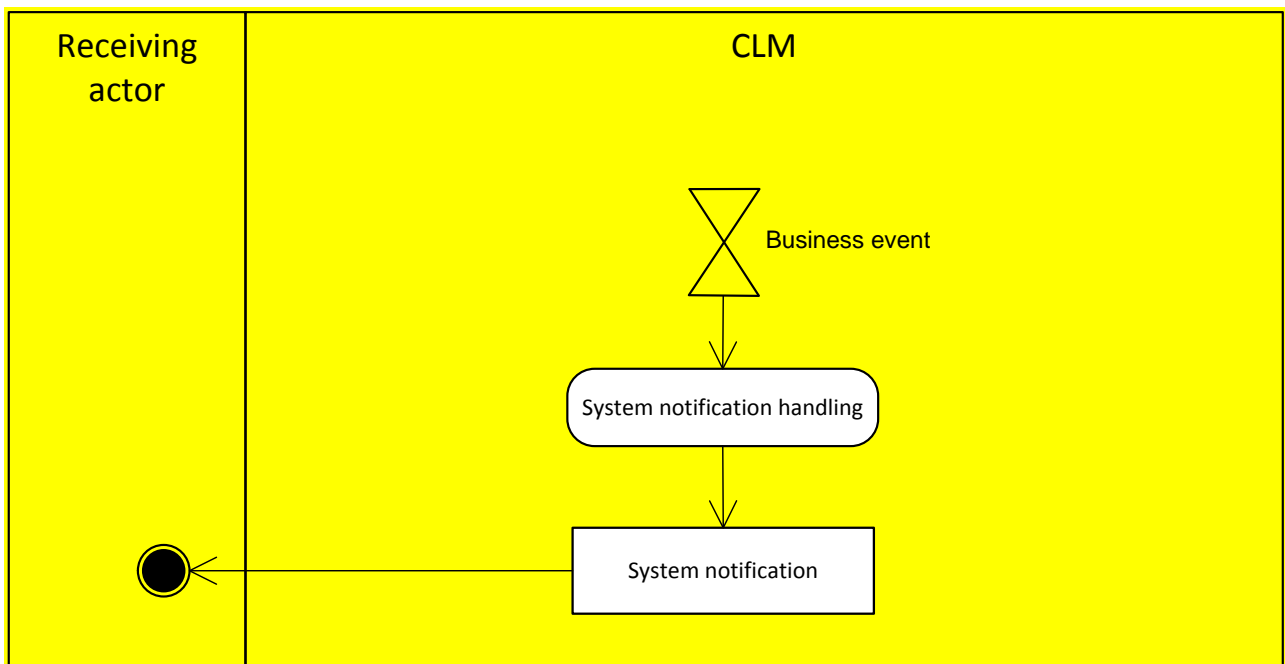


Figure 79 - CLM receive system notification

The defined business events trigger the process “System notification handling”. CLM 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) [388]

Table 104 - Receive system notification

10.7.4 Enter broadcast

Will be completed in v2.0.

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](#) [▶ 306] describes the interactions for the maintenance and query of common reference data using the A2A channel. Chapter [DMT file upload](#) [▶ 311] 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.

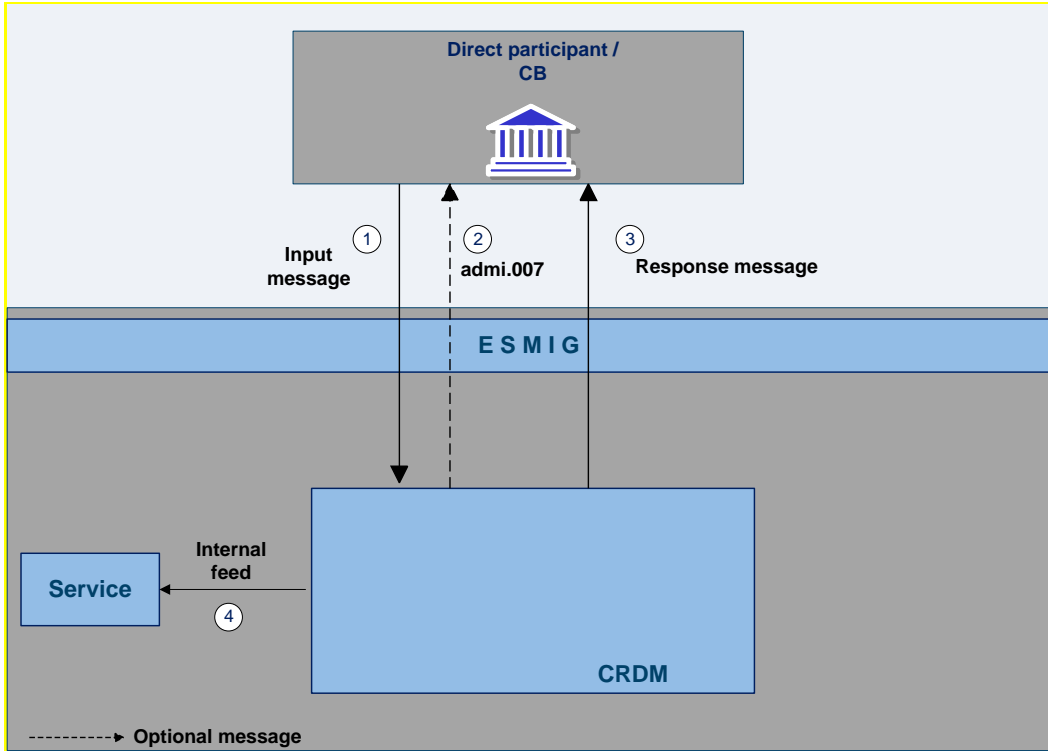


Figure 80 - Common reference data maintenance process

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 105 - 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 106 - 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.

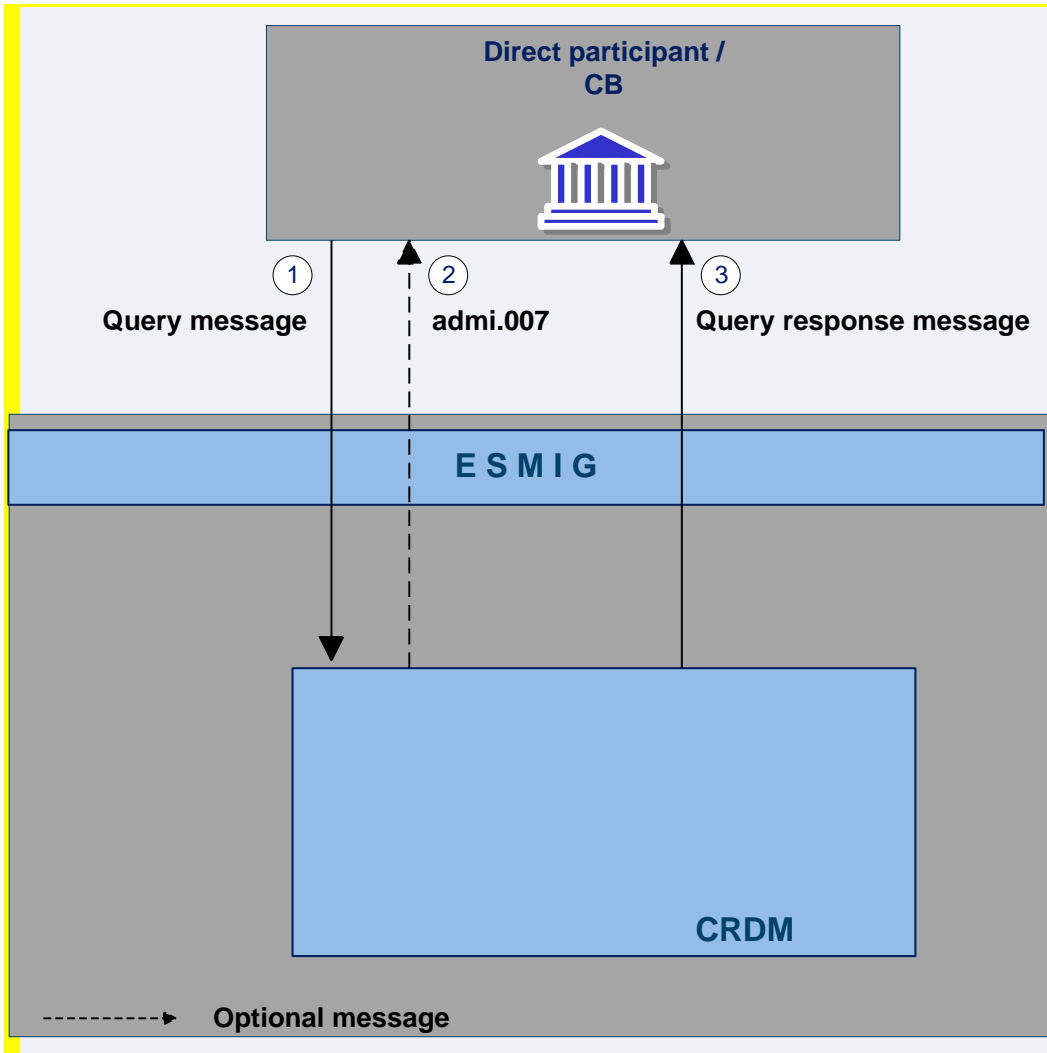


Figure 81 - Common reference data query process

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 107 - 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 108 - 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.

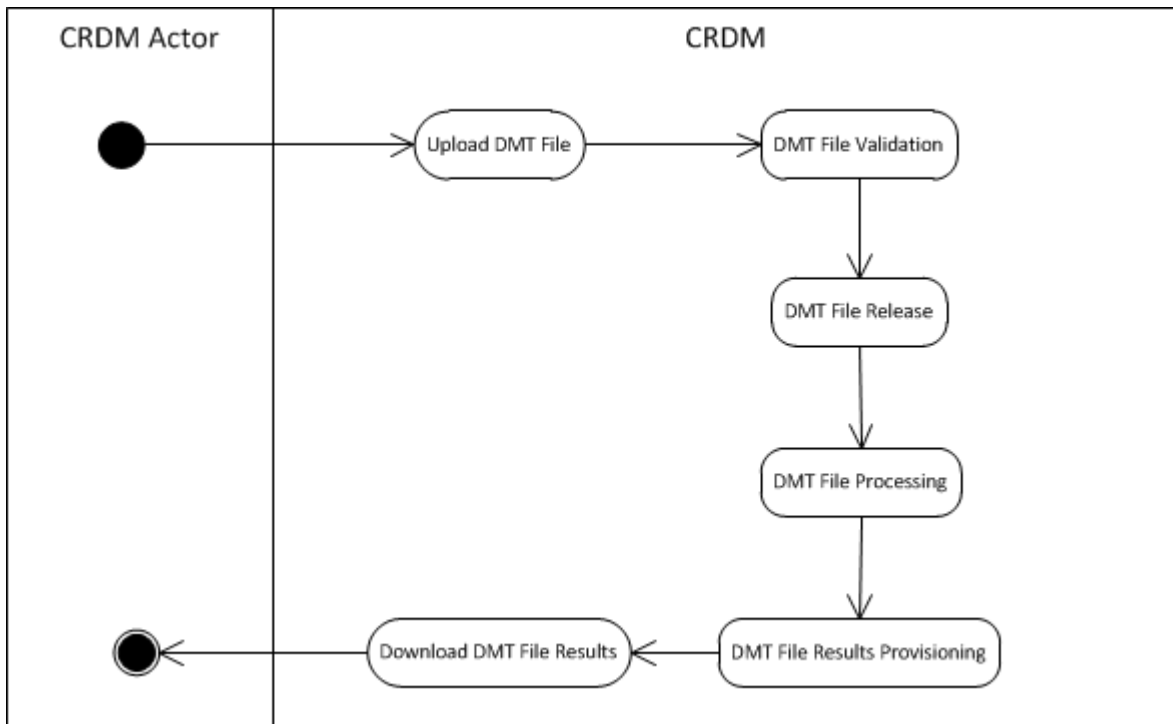


Figure 82 - DMT file upload process

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 109 - DMT files specifications

11.2 Dialogues and processes between ESMIG and participant

11.2.1 Communication processing

11.2.1.1 Introduction

11.2.1.2 Schema validation

Will be completed in v2.0.

11.2.1.3 Technical message validation

Will be completed in v2.0.

11.2.1.4 Inbound and Outbound messages

11.2.1.4.1 Inbound messages

Will be completed in v2.0.

11.2.1.4.2 Outbound Messages

Will be completed in v2.0.

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 CLM 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 A2A 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” and a detailed appendix, followed by “Specific messages for CBs”.

The introductory section “General information” provides general information on the concept of messaging or/and information applicable to all messages in CLM. 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 CLM. 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 CLM 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 CLM 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 CLM 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 CLM 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 comprehensive 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 version⁴⁹ 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.

49 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 are 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.


```
<?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>
```

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.

Simple types serve as a prescription on how to fill the respective message element in the message instance.

Example:

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>
      <xs:element name="FaceAmt" type="RestrictedFINImpliedCurrencyAndAmount"> </xs:element>
      <xs:element name="AmtsdVal" type="RestrictedFINImpliedCurrencyAndAmount"> </xs:element>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

ISO 2022 groups

ISO 2022 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 2022 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 2022 data dictionary).

13.1.2 CLM-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 CLM.

The customisation of the schema files used in CLM followed a particular approach which combines the needs of the CLM Actors to have a coherent logic across the messages and the need within CLM to have a usable and efficient schema definition. CLM derived this approach from the following customisation principles.

- | Customised CLM schema files are compliant with the initial ISO 2022 schema files.
- | When possible, CLM customisation drops all the message elements with no direct connection to the user requirements of CLM.
- | When possible, CLM customisation restricts element types to the CLM-specific usage.

- | CLM customisation defines the necessary content of mandatory fields which cannot be pruned (i.e. “removed”) from the ISO schema files.
- | CLM customisation restricts the list of possible code values to the sole codes allowed in CLM.
- | CLM customisation sets the length of the values to the length applicable in CLM.
- | CLM customisation sets the occurrence of message elements to the occurrence applicable in CLM.
- | CLM customisation makes optional message elements mandatory if their usage in CLM is always compulsory.
- | CLM customisation restricts the allowed characters to those used in CLM with a pattern.
- | CLM customisation restricts numeric fields applicable to CLM (e.g. for amounts).

Based on the chosen approach four scenarios apply to the customisation for CLM purposes.

1. A (part of a) message only contains elements which are supported by CLM and there is hence no need for any pruning.
2. CLM does not need a certain element but it cannot be pruned in the message because of a particular customer need.
3. Neither CLM nor CLM actors need a certain element and therefore it is pruned.
4. Neither CLM 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 CLM.

For the scenarios 1, 3 and 4, CLM only allows message elements according to the customised schema file. CLM rejects any inbound message containing message elements which are not part of the CLM customised schema file. Message elements under the scope of scenario 4 are not subject to further processing in CLM. CLM 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 CLM customised schema file although the message element is per se dispensable, CLM nevertheless processes the message. For these message elements only schema validations are applicable. CLM does not validate these elements against its business rules.

However, for all messages, CLM prunes elements which are not within the general scope of its functionalities.

CLM rejects messages during schema validation in cases where actors

- | use elements in the message which are not present in the CLM customised schema file,
- | use values in allowed elements but do not respect the restrictions of these values foreseen in the CLM customised schema.

For CLM outbound messages the logic for filling message elements customised to be optional is derived from the concrete circumstances and purposes of the concerned messages.

- l For query response messages the filled message elements for outbound messages are those necessary to convey the information requested by the corresponding query message.
- l For report messages the same applies, in accordance to the concrete configuration for the subscribed reports.

For any other CLM outbound message the filling of optional fields also depends either on

- l the corresponding inbound message with its specific intention
- l or the purpose of the CLM-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 ⁵⁰.

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).

⁵⁰ 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.

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: Abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789 /?:()., '+ CrLf Space												
For Initiating Party, Debtor, Ultimate Debtor, Creditor, Ultimate Creditor, Related Remittance Information and Remittance	Use of FIN X-Character set (see above), plus !#\$%&'*+/-=?^_`{ }~"(),:;<>@[]. Note: Five characters need to be escaped:												
	<table border="1"> <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												
"	"												
'	'												
<	<												
>	>												
&	&												

Table 110 - FIN X-Character set

13.1.3.1 Schema validation

All ISO 20022 messages which arrive at the CLM 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 CLM.

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.

- l The specific syntax characters (e.g. "<" and "&") are not used in the message except in their function as mark-up delineation.
- l 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.
- l 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 CLM undergoes a validity check according to the rules contained in the enriched CLM schema files. These CLM enriched schemas make the structure of the message visible to the user and provide all necessary explanations on the validations the message undergoes.

The CLM 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.

CLM provides the CLM 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>
      <NbOfTx>1</NbOfTx>
      <StlmInf>
        <StlmMtd>CLRG</StlmMtd>
      </StlmInf>
    </GrpHdr>
    <CdtTrfTxInf>
      <PmtId>
        <InstrId>INSTRIDpacs.009</InstrId>
        <EndToEndId>NOTPROVIDED</EndToEndId>
        <TxId>TXIDpacs.009</TxId>
      </PmtId>
      <IntrBkStlmAmt Ccy="EUR">1000000</IntrBkStlmAmt>
      <IntrBkStlmDt>2018-08-24</IntrBkStlmDt>
      <StlmPrty>NORM</StlmPrty>
      <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 CLM enriched schema, the CLM 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 CLM also conducts validations which are based on the business context CLM operates in.

This business validation in CLM 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 CLM verifies the validity of the transmitted message content against its reference data repository.

In case of violations against existing business rules, CLM transmits them to the relevant CLM actors directly via an outbound message. This message contains all the information the CLM 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 (CLM_Receipt_Response camt.025 to camt.024 Create Standing Order CLM to CLM Account Holder, rejected)

```
<RctDtls>
  <OrgnlMsgId>
    <MsgId>MSGIDcamt.024</MsgId>
  </OrgnlMsgId>
  <ReqHdlg>
    <StsCd>REJT</StsCd>
    <Desc>rejected</Desc>
  </ReqHdlg>
</RctDtls>
```

13.2 Communication infrastructure

13.2.1 Envelope messages

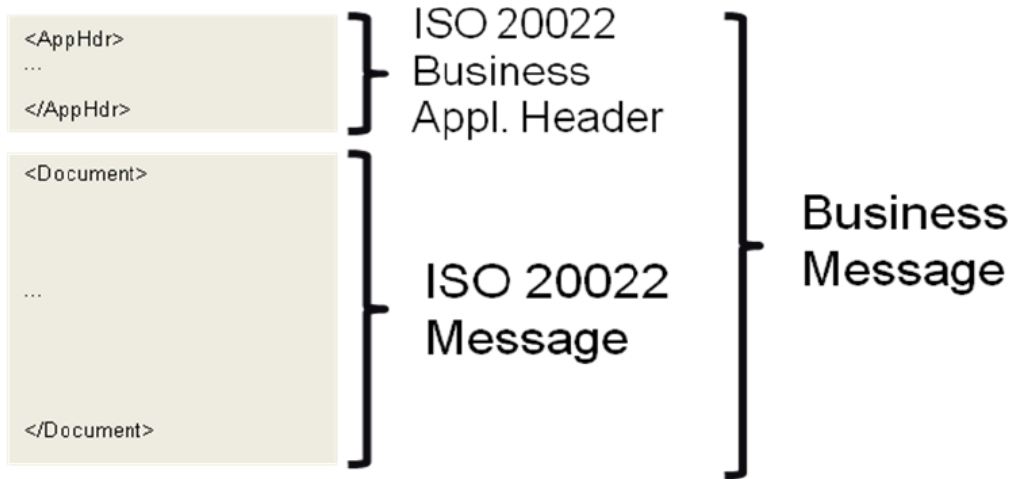
13.2.1.1 Business Application Header

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 CLM.

The BAH is not applicable when

- referring to the acknowledgement of the receipt (admi.007) of a message within CLM;
- technical validation errors identified during the "A2A Business File Validation and Splitting process" are answered from CLM by a ReceiptAcknowledgement (admi.007).

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.



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. 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>NCBPARTBIC</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>
```

The “Request Payload” stands for the whole communication data which is exchanged between and with CLM.

BAH and business message (XML message instance) are part of this payload.

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 CLM supports the exchange of message batches. Therefore, it is possible for the T2 Actor in CLM to send a file composed of several messages. CLM 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 application header 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 CLM entail a receipt confirmation from CLM. After the successful authentication check CLM divides the file into single messages. Every message undergoes a separate validation (schema validation). CLM reports errors on message level either by the corresponding response message or by a status message.

To communicate a user or an application 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 (BFH) and Business message see below.

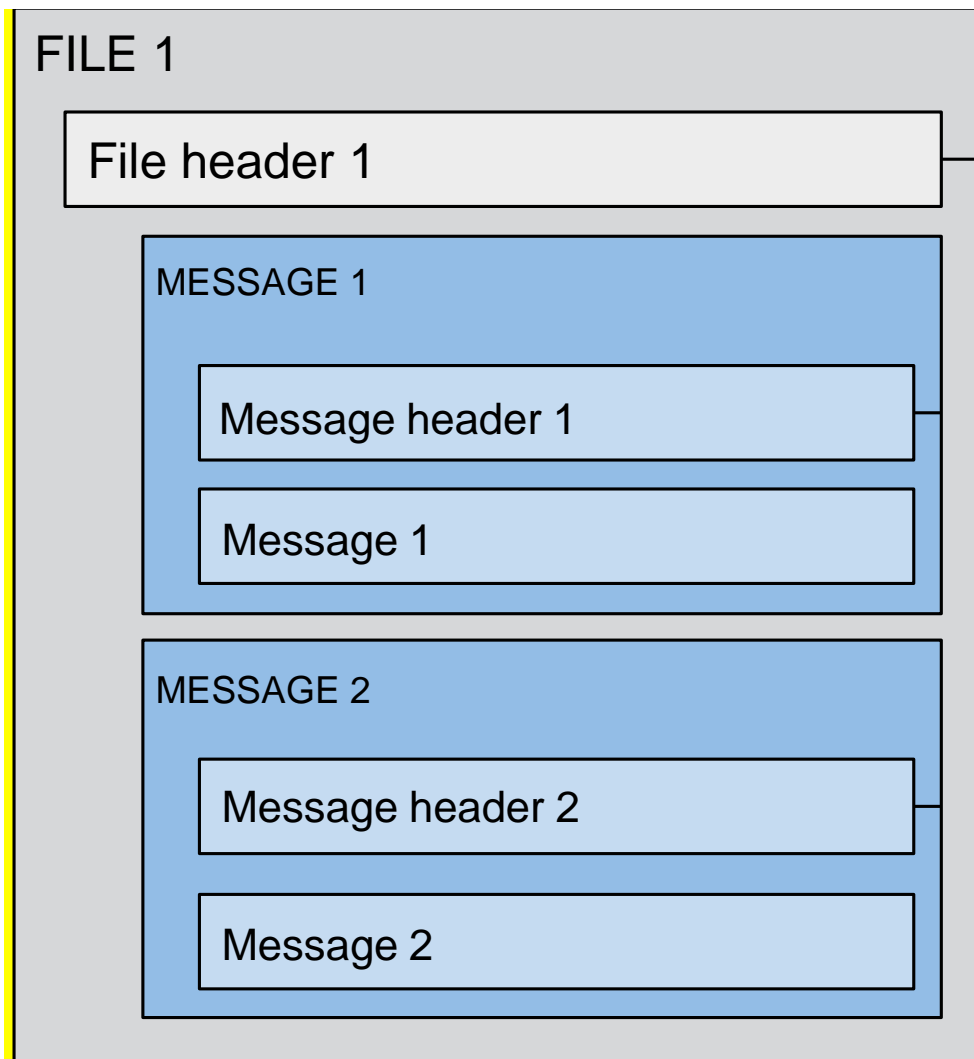


Figure 83 - Business file header

“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 XAdES ⁵¹ standard.

51 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.

The (NRO)⁵² 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](#) [▶ 574].

13.2.1.4 Time zones

Messages exchanged between CLM 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 CLM is CET or Central European Summer Time (CEST). All indications contained in the payload of CLM messages (based on given timestamps e.g.) refer to CET/CEST. The attribution of timestamps in CLM solely occurs on CET/CEST basis. All possible information related to time within the payload of messages sent to CLM must refer to CET/CEST. The CLM calendar as the relevant framework for all operational issues of CLM contains CET/CEST only.

Due to the ISO definition of the BAH the time indications within the BAH refer to Zulu time. CLM users must take into account the difference between the two time formats when exchanging messages with CLM.

Example:

A message sent to CLM on 17 December 2021 at 09:30:47 CET/CEST would need to contain the following field in the BAH ("ZULU time"⁵³).

```
<CreDt>2021-12-17T09:30:47Z</CreDt>
```

52 Non-repudiation of origin is intended to protect against the originator's false denial of having sent the message.

53 Zulu time is the used format for the time indication.

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"):

```
<CreDtTm>2015-12-17T10:30:47<CreDtTm>
```

13.2.1.5 Outbound traffic exceeding given size limitations

Traffic sent to or from CLM 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 CLM allow for adequate processing of the messages and the contained information. The solution envisaged differs according to CLM inbound and outbound traffic.

For CLM inbound traffic there is no need for the CLM 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 CLM Actor can send two single (small) messages. In contrast to outgoing messages there is no need to see them as "one unit".

For CLM 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, CLM 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). CLM 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 CLM outbound file, a technical solution is implemented to split this file technically in several parts.

In case the size of a CLM 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 CLM 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) [339]	acmt.025	AccountQuery
AccountListReport (acmt.026) [341]	acmt.026	AccountReport
Administration (admi)		
ReportQueryRequest (admi.005) [345]	admi.005	ReportQueryRequest
ResendRequest (admi.006) [346]	admi.006	ResendRequest
ReceiptAcknowledgement (admi.007) [347]	admi.007	ReceiptAcknowledgement
Cash Management (camt)		
GetAccount (camt.003) [350]	camt.003	GetAccount
ReturnAccount (camt.004) [355]	camt.004	ReturnAccount
GetTransaction (camt.005) [368]	camt.005	GetTransaction
ReturnTransaction (camt.006) [378]	camt.006	ReturnTransaction
ModifyTransaction (camt.007) [384]	camt.007	ModifyTransaction
GetBusinessDayInformation (camt.018) [386]	camt.018	GetBusinessDayInformation
ReturnBusinessDayInformation (camt.019) [388]	camt.019	ReturnBusinessDayInformation
ModifyStandingOrder (camt.024) [392]	camt.024	ModifyStandingOrder
Receipt (camt.025) [397]	camt.025	Receipt
ResolutionOfInvestigation (camt.029) [406]	camt.029	ResolutionOfInvestigation
GetReservation (camt.046) [409]	camt.046	GetReservation
ReturnReservation (camt.047) [411]	camt.047	ReturnReservation
ModifyReservation (camt.048) [414]	camt.048	ModifyReservation
DeleteReservation (camt.049) [416]	camt.049	DeleteReservation

Chapter	Message code	Message name
LiquidityCreditTransfer (camt.050) [▶ 418]	camt.050	LiquidityCreditTransfer
BankToCustomerStatement (camt.053) [▶ 425]	camt.053	BankToCustomerStatement
BankToCustomerDebitCreditNotifica- tion (camt.054) [▶ 442]	camt.054	BankToCustomerDebitCreditNotifica- tion
FIToFIPaymentCancellationRequest (camt.056) [▶ 452]	camt.056	FIToFIPaymentCancellationRequest
GetStandingOrder (camt.069) [▶ 457]	camt.069	GetStandingOrder
ReturnStandingOrder (camt.070) [▶ 459]	camt.070	ReturnStandingOrder
DeleteStandingOrder (camt.071) [▶ 466]	camt.071	DeleteStandingOrder
BillingReportRequest (camt.076) [▶ 468]	camt.076	BillingReportRequest
BillingReport (camt.077) [▶ 468]	camt.077	BillingReport
AuditTrailQuery (camt.097) [▶ 468]	camt.097	AuditTrailQuery
AuditTrailReport (camt.098) [▶ 470]	camt.098	AuditTrailReport
DirectDebitMandateQuery (camt.099) [▶ 474]	camt.099	DirectDebitMandateQuery
DirectDebitMandateReport(camt.100) [▶ 476]	camt.100	DirectDebitMandateReport
ModifyCreditLine (camt.998) [▶ 479]	camt.998	ModifyCreditLine
InsertBalance_RM (camt.998) [▶ 481]	camt.998	InsertBalance_RM
Headers (head)		
BusinessApplicationHeader (head.001) [▶ 482]	head.001	BusinessApplicationHeader
BusinessFileHeader (head.002) [▶ 485]	head.002	BusinessFileHeader
Payments Clearing and Settlement (pacs)		
PaymentStatusReport (pacs.002) [▶ 487]	pacs.002	PaymentStatusReport

Chapter	Message code	Message name
FinancialInstitutionCreditTransfer (COR) (pacs.009) [▶ 491]	pacs.009	FinancialInstitutionCreditTransfer
FinancialInstitutionDirectDebit (pacs.010) [▶ 496]	pacs.010	FinancialInstitutionDirectDebit
Reference Data (reda)		
PartyQuery (reda.015) [▶ 499]	reda.015	PartyQuery
PartyReport (reda.017) [▶ 501]	reda.017	PartyReport
CashAccountAuditTrailQuery (reda.039) [▶ 504]	reda.039	CashAccountAuditTrail Query
CashAccountAuditTrailReport (reda.040) [▶ 506]	reda.040	CashAccountAuditTrailReport
PartyAuditTrailQuery (reda.042) [▶ 509]	reda.042	PartyAuditTrailQuery
PartyAuditTrailReport (reda.043) [▶ 510]	reda.043	PartyAuditTrailReport
CalendarQuery (reda.064) [▶ 514]	reda.064	CalendarQuery
CalendarReport (reda.065) [▶ 515]	reda.065	CalendarReport

Table 111 - 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 \(acmt.026\)](#) [▶ 341] 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.

AccountServiceIdentification

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 112 - 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 \(acmt.025\)](#) [▶ 339] 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/CingNtfctnAmt	ImpliedCurrencyAndAmount	Ceiling notification amount
ClosingDate Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/ClsgDt	ISODate	Closing Date
Restriction Docu- ment/AcctListRpt/RptOrErr/AcctRpt/Ac ct/Rstrctn	Restriction1	Account restriction

Message item	Data type/code	Utilisation
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 113 - 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 114 - 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 CLM Account Holder (or a party authorised by them) to CLM. It is used to query the latest available report data of a specific report type.

Within CLM, 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 \(admi.007\)](#) [▶ 347] 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 CLM-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/CLM/admi.005.001.01_CLM

Business rules applicable to the schema

For business rules applicable to *ReportQueryRequest* please refer to chapter [Index of business rules and error codes](#) [▶ 537].

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 \(camt.053\)](#) [▶ 425] 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	MCA 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 115 - ReportQueryRequest (admi.005) - usage case Account Statement Query

Usage case example: admi.005_CLM_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 CLM 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 CLM component.

The *ResendRequest* message supports resend requests for messages from the CLM 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 CLM-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/CLM/admi.006.001.01_CLM

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 CLM component to a CLM Account Holder. It is used to reject the reception of a previously sent message. The CLM component generates this message after a negative authentication process. It can be also sent as an error reporting response to a report query. Within CLM, the *ReceiptAcknowledgement* message has the following usages:

- | Negative Receipt Acknowledgement (e. g. schema validation rejection, technical validation)

The *ReceiptAcknowledgement* 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 receipt acknowledgement 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 CLM-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/CLM/admi.007.001.01_CLM

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:

- | Reject Payment Order Message
- | Reject Amendment payment order

- | Reject Modify/delete reservation request (after schema validation)
- | Reject CLM Query Message

Usage case: NegativeReceiptAcknowledgement SchemaValidation for all message functions

In the above mentioned usage case, the recipient is being informed that a message previously received from CLM does not comply with CLM technical rules and is not processable for CLM.

Specific message content

Message item	Data type/code	Utilisation
Message Identification RctAck/MsgId/	CLM_RestrictedFINXMax35Text	Value NONREF as the message ID is already part of the BAH
Related Reference RltdRef/Ref	CLM_RestrictedFINXMax35Text	Reference given by the original message: MsgID of the incoming message this receipt acknowledgement is sent for. In case the MsgID 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 116 - ReceiptAcknowledgement(admi.007) – usage case Negative Receipt Acknowledgement

Usage case example 1: admi.007_CLM_ReceiptAcknowledgement_Reject Payment Order Message_Example_1.xml

In this sample a [LiquidityCreditTransfer \(camt.050\)](#) [▶ 418] is sent to CLM for further processing, but rejected due as the same BizMsgIdr and the same Business sender "From" at the same day was detected as a duplicate.

Usage case example 2: admi.007_CLM_Reject Amendment payment order_Example_2.xml

In this sample a [GetTransaction \(camt.005\)](#) [▶ 368] is sent to CLM for to modify one liquidity transfer order on the CLM Participant's MCA. In the end it was rejected as the business sending user was not known within CLM.

Usage case example 3: admi.007_CLM_ModifyReservation request (after schema validation) Example_3.xml

In this sample a [ModifyReservation \(camt.048\)](#) [▶ 414] is sent by a CLM Account Holder to CLM, but rejected as the format of one field does not comply with the XSD scheme.

Usage case example 4: admi.007_CLM Query Message _Example_4.xml

In this sample a [GetAccount \(camt.003\)](#) [▶ 350] is sent by a CLM Account Holder to CLM, 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 CLM Account Holder (or on their behalf by an authorised party) or a CB to CLM. It is used to request balances, including credit line, of one CLM MCA.

The *GetAccount* message contains the criteria which is used to select the response information.

Within CLM, the *GetAccount* message has the following usages:

- | Available Liquidity CLM Query
- | Minimum Reserve Query
- | Query Minimum Reserve Requirements per Participant (CB only)

In response to the *GetAccount* message, a [ReturnAccount \(camt.004\)](#) [▶ 355] 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 CLM-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/CLM/camt.003.001.06> CLM

Business rules applicable to the schema

For business rules applicable to *GetAccount* please refer to chapter [Index of business rules and error codes](#) [► 537].

14.3.1.3 The message in business context

Usage case: Available Liquidity CLM Query

In this usage case, the sender requests information regarding all liquidity available on CLM MCA 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 *GetAccount* only includes message 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.

Table 117 - GetAccount (camt.003) – usage case Available Liquidity CLM Query

Request for specific sub-account, *GetAccount* only includes 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.

Table 118 - GetAccount (camt.003) – usage case Available Liquidity CLM Query

Request on behalf of third party (AccountOwner) e.g. BIC of NCB or group of account manager is the sender of the *GetAccount* message. Sending party and AccountOwner are different then, in the *GetAccount* message identification and account owner are 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 she is responsible for. AnyBICIdentifier/BIC used on behalf of third parties Account ID or account owner must be present, but not both.

Table 119 - GetAccount (camt.003) – usage case Available Liquidity CLM Query

Usage case example: camt.003_CLM_GetAccount_AvailableLiquidityCLMQuery_Example.xml

In this example, a *GetAccount* is instructed by a third party. It illustrates the mandatory elements in the message.

Usage case: Minimum Reserve Query

In this usage case, the sender requests information regarding the minimum reserve amount set on CLM MCA within their query criteria.

Specific message requirements

All content must comply with the business rules for the message.

camt.003_GetAccount_MinimumReserveQuery_MessageRequirements

*See above camt.003_GetAccount_AvailableLiquidityCLMQuery_MessageRequirements

Usage case example: camt.003_CLM_GetAccount_MinimumReserveQuery_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: Query Minimum Reserve Requirements per Participant (CB only)

In this usage case, the sending CB requests information regarding the minimum reserve amount set across CLM MCA for the participant 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/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: - BIC of the CLM MCA. <ul style="list-style-type: none"> In case the requestor is a normal CI, his DN must match to the stated BIC. A CB may only select accounts she is responsible for. AnyBICIdentifier/BIC used on behalf of third parties

Table 120 - GetAccount (camt.003) – usage case Query Minimum Reserve Requirements per Participant (CB only)

Usage case example:
camt.003_CLM_GetAccount_QueryMinimumReservePerParticipant_Example.xml

In this example, a *GetAccount* is instructed by the NCB to request information about the fulfilment of the minimum reserve of a participant. 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* message.

The *ReturnAccount* message is sent by CLM to a CLM Account Holder (or a party authorised by them). It is used to provide information on the balances, including credit line, of one CLM MCA.

Within CLM, the *ReturnAccount* message has the following usages:

- | Available Liquidity CLM Query (Data or Error response)
- | Minimum Reserve Query (Data or Error response)
- | Query Minimum Reserve Requirements per Participant (CBs only) (Data or Error response)
- | Floor Notification
- | Ceiling Notification
- | Reimbursed Marginal Lending

The *ReturnAccount* message is sent in response to a [GetAccount \(camt.003\)](#) [▶ 350] message, which requested the information. The floor, ceiling and reimbursed margin notifications are sent based upon activity within CLM.

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 CLM-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/CLM/camt.004.001.07_CLM

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: Available Liquidity CLM Query (Data response)

In this usage case, the recipient of the message is being informed regarding all liquidity available on the CLM MCA within their query criteria.

Specific message content

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/OrgnIBizQry/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) [350] copied from the BAH
Account Identification Docu-	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account owner and the account servicer.

Message item	Data type/code	Utilisation
ment/RtrAcct/RptOrErr/AcctRpt/AcctId/Othr/Id		
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.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp	SystemBalanceType1Code__1: OPNG CRRT CLSG BLCK DLOD ADJT PRAV Proprietary Code: FLOR CEIL	Specifies the nature of a balance which is being reported.
Multilateral Balance Status Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Sts	PDNG STLD	Current status of a cash balance which is being reported.

Table 121 - ReturnAccount (camt.004) – usage case Available Liquidity CLM Query (Data response)

Usage case example: `camt.004_CLM_ReturnAccount-
_AvailableLiquidityCLMQueryData_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 CLM MCA is sent to the requesting party.

Usage case: Available Liquidity CLM 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 \(camt.003\)](#) [▶ 350]

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/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) [350] 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 122 - ReturnAccount (camt.004) – usage case Available Liquidity CLM Query (Error response)

Usage case example : **camt.004_CLM_ReturnAccount_AvailableLiquidityCLMQueryError_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: Minimum Reserve Query (Data response)

In this usage case, the recipient of the message is being informed regarding the minimum reserve amount set on CLM MCA within their query criteria.

If a problem is encountered while retrieving this information, the error information is reported instead.

Specific message content

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/OrgnIBizQry/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) [350] 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.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp	SystemBalanceType1Code__1: BLCK ADJT PRAV	Specifies the nature of a balance which is being reported.
Multilateral Balance Status Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Sts	PDNG STLD	Current status of a cash balance which is being reported.

Table 123 - ReturnAccount (camt.004) – usage case Minimum Reserve Query (Data response)

Usage case example : camt.004_CLM_ReturnAccount_MinimumReserveQueryData_Example.xml

In this example a *ReturnAccount* containing a reference to an incoming message with the ID “MSGIDcamt.003”, and the available reserve information on the CLM MCA is sent to the requesting party.

Usage case: Minimum Reserve 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 Minimum Reserve Query (camt.003).

The identification of the previously sent query message is included in this error response for reconciliation purposes.

Specific message content

For specific content and example, please see the above usage case:

Available Liquidity CLM Query (Error response)

Usage case: Query Minimum Reserve Requirements per Participant (Data response)

In this usage case, the CB is being informed regarding the minimum reserve amount set across CLM MCA for a single CLM Account Holder, contained within a their query criteria.

Specific message content

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) [350] 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.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	CACC	CLM-use: Only cash 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.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp	SystemBalanceType1Code__1: BLCK ADJT PRAV	Specifies the nature of a balance which is being reported.
Multilateral Balance Status Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Sts	PDNG STLD	Current status of a cash balance which is being reported.

Table 124 - ReturnAccount (camt.004) – usage case Query Minimum Reserve Requirements per Participant (Data response)

Usage case example : **camt.004_CLM_ReturnAccount-
_QueryMinimumReservePerParticipantData_Example.xml**

In this example a *ReturnAccount* containing a reference to an incoming message with the ID “MSGIDcamt.003”, and a response to a request of information about the fulfilment of the minimum reserve of a participant, is sent to the requesting CB.

Usage case: Query Minimum Reserve Requirements per Participant (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 Query Minimum Reserve Requirements per Participant (camt.003).

The identification of the previously sent query message is included in this Error response for reconciliation purposes.

Specific message content

For specific content and example, please see the above usage case:

Available Liquidity CLM Query (Error response)

Usage case: Floor Notification

In this usage case, the recipient of the message is being informed that the balance on one of their CLM MCAs has fallen below the pre-defined floor threshold for the account.

Specific message content

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/OrgnIBizQry/Msg Id	RestrictedFINXMax35Text	CLM-use: For floor and ceiling notifica- tions, value "NONREF" as the <i>Re- turnAccount</i> message is created in push mode.
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	CACC	CLM-use: Only cash 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 own- ing 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.
Multilateral Balance Type Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Tp	SystemBalanceType1Code__1: CRRT FLOR	Specifies the nature of a balance which is being reported.
Multilateral Balance Status Document/RtrAcct/RptOrErr/AcctRpt/AccOrErr/Acct/MulBal/Sts	PDNG STLD	Current status of a cash balance which is being reported.

Table 125 - ReturnAccount (camt.004) – usage case Floor Notification

Usage case example : camt.004_CLM_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 CLM MCAs has risen above the pre-defined ceiling threshold for the account.

Specific message content

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 Document/RtrAcct/MsgHdr/OrgnlBizQry/MsgId	RestrictedFINXMax35Text	CLM-use: For floor and ceiling notifications, value "NONREF" as the <i>ReturnAccount</i> message is created in push mode.
Account Identification Docu-	RestrictedFINXMax34Text	Unique and unambiguous identification for the account between the account

Message item	Data type/code	Utilisation
ment/RtrAcct/RptOrErr/AcctRpt/AcctId/Othr/Id		owner and the account servicer.
Account Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Tp/Cd	CACC	CLM-use: Only cash 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/Orgld/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.
Multilateral Balance Credit Debit Indicator Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/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.
Multilateral Balance Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Tp	SystemBalanceType1Code__1: CRRT CEIL	Specifies the nature of a balance which is being reported.
Multilateral Balance Status Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Sts	STLD	Current status of a cash balance which is being reported.

Table 126 - ReturnAccount (camt.004) – usage case Ceiling Notification

Usage case example : camt.004_CLM_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: Reimbursed Marginal Lending

In this usage case, the recipient of the message is being informed that the marginal lending on one of their CLM MCAs has been reimbursed.

Specific message content

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	CLM-Use: For refunding of the marginal lending and marginal lending reverse transactions, value "NONREF" as the <i>ReturnAccount</i> message is created in push mode.
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	MGLD	Marginal lending account
Currency Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/Ccy	ActiveOrHistoricCurrencyCode	Specifies the currency of the cash account or sub-account being reported.
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.
Multilateral Balance Credit Debit Indi-	CRDT	Indicates whether the balance is a

Message item	Data type/code	Utilisation
cator Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/CdtDbtInd	DBIT	credit or a debit balance. A zero bal- ance is considered to be a credit bal- ance.
Multilateral Balance Type Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Tp	SystemBalanceType1Code__1: CRRT	Specifies the nature of a balance which is being reported.
Multilateral Balance Status Docu- ment/RtrAcct/RptOrErr/AcctRpt/AccOr Err/Acct/MulBal/Sts	STLD	Current status of a cash balance which is being reported.
Value Date	DateAndDateTimeChoice__1	Date and time at which the balance is or will be available.

Table 127 - ReturnAccount (camt.004) – usage case Reimbursed Marginal Lending

Usage case example : camt.004_CLM_ReturnAccount_ReimbursedMarginalLending_Example.xml

In this example a *ReturnAccount* is sent as a push notification to the collateral manager with information about the refunding of the marginal lending or the marginal lending reverse transaction.

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 CLM Account Holder (or on their behalf by an authorised party) to CLM. It is used to request information about liquidity transfer orders, liquidity transfers, payment orders and payments.

The *GetTransaction* message can be used to request the above information based upon multiple criteria.

Within CLM, the *GetTransaction* message has the following usages:

- | CLM Payment Order Query
- | Query Standing Facilities Transaction (CBs only)

In response to the *GetTransaction* message, a [ReturnTransaction \(camt.006\)](#) [▶ 378] 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: requested execution date, payment identification, status, instructed amount, instructed currency, debit/credit indicator, interbank settlement amount, interbank settlement currency, payment method, payment type, processing validity time
- | account identification
- | entry date

References/links

The CLM-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/CLM/camt.005.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *GetTransaction* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.3.3 The message in business context

Usage case: CLM Payment Order Query

In this usage case, the sender requests information regarding the details of all payment orders 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	CLM_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 Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtTo/Ctry	CountryCode	Search on country to which the payment is sent
RequestedExecutionDate	DatePeriodSearch1Choice	Search on RequestedExecutionDate

Message item	Data type/code	Utilisation
Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/ReqdExctnDt/DtSch		range
LongBusinessIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Pmtld/LngBizld	LongPaymentIdentification1	Search on payment LongBusinessIden- tification
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- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Sts/PmtInstrSts/PdgAnd FnlSts	CashPaymentStatus2Code	Search on final or pending status (FINL, PDNG)
PaymentInstructionStatusDateTime Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Sts/PmtInstrStsDtTm	DateTimePeriod1Choice	Search on the payment status date time range
(Rejection) Status Reason Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Sts/PrtryStsRsn	Max4AlphaNumericText	Search on the reason that has been used by the system to reject the trans- action.
InstructedAmount Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/InstdAmt/CcyAndAmtRg	ActiveOrHistoricCurrencyAndAmountRange2	Search on instructed amount with cur- rency, amount range, debit/credit indi- cator
InstructedAmountCurrency	ActiveOrHistoricCurrencyCode	Search on the instructed amount cur-

Message item	Data type/code	Utilisation
Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/InstdAmtCcy		rency
CreditDebitIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/CdtDbtInd	CreditDebitCode	Search on payment in debit or credit
InterbankSettlementAmount Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/IntrBkStlmAmt/CcyAnd AmtRg	ActiveCurrencyAndAmountRange3	Search on settlement amount with currency, amount range, debit/credit indicator
InterbankSettlementAmountCurrency Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/IntrBkStlmAmtCcy	ActiveCurrencyCode	Search on interbank settlement amount currency
PaymentMethod Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/PmtMtd	PaymentOrigin1Choice	Search on XML message name carry- ing the payment OR on proprietary codes (ASXML, INTERN, ORDER)
PaymentType Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/PmtTp	PaymentType4Choice	Search on code (LIQ, MGL, OND) or proprietary code (ASTI, BIDB, CWD, CONP, CREDITLINE, LIQP, PDDB)
ProcessingValidityTime Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/PrcgVldtyTm	DateTimePeriod1Choice	Search on date time range for the pro- cessing validity time
Debtor BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S	CLM_BIC11Text	Search on debtor's BIC

Message item	Data type/code	Utilisation
chCrit/PmtSch/Pties/Dbtr/FinInstnId/BICFI		
DebtorAgent BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Pties/DbtrAgt/FinInstnId/ BICFI	CLM_BIC11Text	Search on debtor agent's BIC
IntermediaryAgent BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Pties/IntrmyAgt/FinInstn Id/BICFI	CLM_BIC11Text	Search on intermediary agent's BIC
CreditorAgent BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Pties/CdtrAgt/FinInstnId/ BICFI	CLM_BIC11Text	Search on creditor agent's BIC
Creditor' BIC Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/PmtSch/Pties/Cdtr/FinInstnId/BI CFI	CLM_BIC11Text	Search on creditor's BIC
AccountIdentification Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/AcctNtrySch/AcctId/EQ/Othr/Id	Max34Text	Search on the cash entry account identification
Entry DateTime Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/S chCrit/AcctNtrySch/NtryDt/DtTmSch	DateTimePeriod1Choice	Search on the entry date time range
ReturnCriteria specify which information is requested in the ReturnTransaction (camt.006) [378]		
Pay- mentTo/MemberIdentificationIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/	RequestedIndicator	true, false If absent, default value is true

Message item	Data type/code	Utilisation
RtrCrit/PmtToRtrCrit/MmbldInd		
Pay- mentFrom/MemberIdentificationIndicat or Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtFrRtrCrit/MmbldInd	RequestedIndicator	true, false If absent, default value is true
EntryDateIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/AcctCshNtryRtrCrit/NtryDtInd	RequestedIndicator	true, false If absent, default value is true
InstructionIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/InstrInd	RequestedIndicator	
PaymentInstructionStatusIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/InstrStsRtrCrit/Pmtln strStsInd	RequestedIndicator	true, false If absent, default value is true
PaymentInstructionStatusDateTimeInd- icator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/InstrStsRtrCrit/Pmtln strStsDtTmInd	RequestedIndicator	true, false If absent, default value is true
PaymentInstructionStatusReasonIndi- cator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/InstrStsRtrCrit/Pmtln strStsRsnInd	RequestedIndicator	true, false If absent, default value is true
InstructedAmountIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/	RequestedIndicator	true, false If absent, default value is true

Message item	Data type/code	Utilisation
RtrCrit/PmtRtrCrit/InstdAmtInd		
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
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/PmtTpInd	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	RequestedIndicator	true, false

Message item	Data type/code	Utilisation
Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/PmtMtdInd		If absent, default value is true
DebtorIndicator Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/ RtrCrit/PmtRtrCrit/DbtrInd	RequestedIndicator	true, false If absent, default value is true
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 128 - GetTransaction (camt.005) – usage case CLM Payment Order Query

Usage case example 1: camt.005_CLM_GetTransaction_CLMPaymentOrderQuery_Example1.xml

Multiple criteria are defined in the query. The query purpose is to extract for the authorised user all the liquidity transfers via payment (LIQP) with a “Pending” status, an amount greater of equal to 15000 EUR, with a debit entry on the account number “Account123456”, and submitted to the system by using pacs.009.001.07. The return criteria are the respective default values except for “InstructionCopy” and “EntryDate” on the account which are not requested.

Usage case example 2: camt.005_CLM_GetTransaction_CLMPaymentOrderQuery_Example2.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.

Usage case example 3: camt.005_CLM_GetTransaction_CLMPaymentOrderQuery_Example3.xml

In this example, the query requests only the new matching items since the last similar query based on camt.005.001.07

Usage case: Query Standing Facilities Transaction

In this usage case, the CB requests information regarding the details of all standing facility transactions 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	CLM_RestrictedFINXMax35Text	Identification of the message
Document/GetTx/MsgHdr/MsgId		
SearchCriteria		
PaymentTo/MemberIdentification	RTGS_BIC11Text	Search on member's BIC to which the payment is sent
Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtTo/Mmbld/FinInstnId/BICFI		
PaymentType	PaymentType3Code	OND, MGL
Docu- ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/PmtTp/Cd		
ReturnCriteria		
Same as general description		

Table 129 - GetTransaction (camt.005) – usage case Query Standing Facilities Transaction

Usage case example:
camt.005_CLM_GetTransaction_QueryStandingFacilityTransaction_Example.xml

Request of information about all standing facilities transactions of a participant. The requestor is the NCB of a participant.

Search on payment type OND and MGL.

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 CLM to a CLM Account Holder (or a party authorised by them). It is used to provide information on the details of one or more liquidity transfer orders, liquidity transfers, payment orders and/or payments.

The *ReturnTransaction* message contains such information based upon CLM MCAs and upon the criteria provided in the request.

Within CLM, the *ReturnTransaction* message has the following usages:

- | CLM Payment Order Query (Data or Error response)
- | Query Standing Facilities Transaction (CBs only) (Data or Error response)

The *ReturnTransaction* message is sent in response to a [GetTransaction \(camt.005\)](#) [▶ 368] 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: status, instructed amount, interbank settlement amount, payment method, processing validity time, payment type, debtor, debtor agent, intermediary agent, creditor agent, creditor

References/links

The CLM-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/CLM/camt.006.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *ReturnTransaction* message.

14.3.4.3 The message in business context

Usage case: CLM Payment Order Query (Data response)

In this usage case, the recipient of the message is being informed regarding the details of all payment orders within their query criteria.

Specific message content

Message item	Data type/code	Utilisation
MessageIdentification Document/RtrTx/MsgHdr/MsgId	CLM_RestrictedFINXMax35Text	Identification of the message
OriginalBusinessQuery MessageIdentification Docu- ment/RtrTx/MsgHdr/OrgnBizQry/MsgId	CLM_RestrictedFINXMax35Text	Identification of the original business query message
QueryName Document/RtrTx/MsgHdr/QryNm	Max35Text	Name of the query allocated by the system in the return message

Message item	Data type/code	Utilisation
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/FinInstnId/BIC FI	CLM_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/FinInstnId/BICF I	CLM_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
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 Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Sts/DtTm/DtTm	ISODatetime	Date and time at which the status was 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	CLM_Max14_Max5DecimalAmount	Amount and currency instructed in the

Message item	Data type/code	Utilisation
Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/InstdAmt/AmtWthCcy		payment
InterbankSettlementAmount Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/IntrBkSttlmAmt/AmtWthC cy	CLM_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)
ProcessingValidityTime Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/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- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Tp	PaymentType4Choice	Code (LIQ, MGL, OND) or proprietary code (ASTI, BIDB, CWD, CONP, CREDITLINE, LIQP, PDDB)
TransactionIdentification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Txid	Max35Text	Unique identification, as assigned by the first instructing agent, to unambigu- ously identify the transaction that is passed on, unchanged, throughout the entire interbank chain.
InterbankSettlementDate Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/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.

Message item	Data type/code	Utilisation
EndToEndIdentification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/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/Tx OrErr/Tx/Pmt/Pties/Dbtr/FinInstnId/BIC FI	CLM_BIC11Text	BIC of the debtor
DebtorAgent's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Pties/DbtrAgt/FinInstnId/ BICFI	CLM_BIC11Text	BIC of the debtor agent
IntermediaryAgent's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Pties/IntrmyAgt/FinInstnId/ BICFI	CLM_BIC11Text	BIC of the intermediary agent
CreditorAgent's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Pties/CdtrAgt/FinInstnId/ BICFI	CLM_BIC11Text	BIC of creditor agent
Creditor's BIC Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/Pmt/Pties/Cdtr/FinInstnId/BIC FI	CLM_BIC11Text	BIC of creditor
Entry Account Identification Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/AcctNtry/AcctId/Othr/Id	Max34Text	Account to or from which a cash entry is made.
Entry Date Time	ISODateTime	Date and time of the posting cash entry

Message item	Data type/code	Utilisation
Docu- ment/RtrTx/RptOrErr/BizRpt/TxRpt/Tx OrErr/Tx/AcctNtry/Ntry/Dt/DtTm		on the account
Operational Error		
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 for- mat.

Table 130 - ReturnTransaction (camt.006) – usage case CLM Payment Order Query (Data response)

Usage case example: camt.006_CLM_ReturnTransaction_CLMPaymentOrderQueryData_Example.xml

The message returns the 2 transactions meeting the search criteria defined in the query message (camt.005_CLM_GetTransaction_CLMPaymentOrderQuery_Example1).

Usage case: CLM Payment Order 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 \(camt.005\)](#) [368].

Specific message content

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 for- mat.

Table 131 - ReturnTransaction (camt.006) – usage case CLM Payment Order Query (Error response)

Usage case example:
camt.006_CLM_ReturnTransaction_CLMPaymentOrderQueryError_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.

Usage case: Query Standing Facilities Transaction (Data response)

In this usage case, the recipient of the message is being informed regarding the details of all standing facility transactions within their query criteria.

Specific message content

Message item	Data type/code	Utilisation
Same as general description		

Table 132 - ReturnTransaction (camt.006) – usage case Query Standing Facilities Transaction (Data response)

Usage case example:
camt.006_CLM_ReturnTransaction_QueryStandingFacilitiesTransactionData_Example.xml

In the example, 2 transactions are meeting the search criteria and information is returned as per returned criteria. The first transaction is an overnight deposit and has been rejected. The second transaction is a marginal lending and has been settled.

Usage case: Query Standing Facilities Transaction (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 \(camt.005\)](#) [368].

Specific message content

For specific content and example, please see the above usage case:

CLM Payment Order Query (Error response)

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 CLM Account Holder (or on their behalf by an authorised party) to CLM. It is used to modify one liquidity transfer order or one payment order on the CLM participant's CLM MCA.

The *ModifyTransaction* message may only be used for an order which is in a transient status (i.e. it has not reached a final status such as rejected, revoked or settled).

The *ModifyTransaction* message contains the new value that the CLM Account Holder wants to be applied to the relevant feature of the order identified in the message. Only one feature, of one order, may be changed in a single *ModifyTransaction* message.

Within CLM, the *ModifyTransaction* message has the following usages:

- I TBC1

In response to the *ModifyTransaction* message, a [Receipt \(camt.025\)](#) [▶ 397] 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 order and the modification to be executed. The modifiable attributes are:

- I priority
- I processing validity time

References/links

The CLM-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/CLM/camt.007.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *ModifyTransaction* please refer to chapter [Index of business rules and error codes](#) [▶ 537].

14.3.5.3 The message in business context

Usage case: TBC1

XXXXXXXXXX

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
	TBC	

Table 133 - ModifyTransaction (camt.007) – usage case TBC1

Usage case example: camt.007_CLM_ModifyTransaction_<TBC1>_Example.xml

14.3.6 GetBusinessDayInformation (camt.018)

14.3.6.1 Overview and scope of the message

This chapter illustrates the *GetBusinessDayInformation* message.

The *GetBusinessDayInformation* message is sent by a CLM Account Holder (or on their behalf by an authorised party) to CLM. It is used to request information on different types of administrative data linked to.

Within CLM, the *GetAccount* message has the following usages:

- | Event Query
- | System Time Query

In response to the *GetBusinessDayInformation* message, a [ReturnBusinessDayInformation \(camt.019\)](#) [▶ 388] message containing the requested information is returned.

14.3.6.2 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.

BusinessDayInformationQueryDefinition

This building block is mandatory. It contains detailed information related to the business query criteria about the business day information. It includes the following elements:

- | query type
- | query name
- | system identification

References/links

The CLM-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/CLM/camt.018.001.04_CLM

Business rules applicable to the schema

For business rules applicable to *GetBusinessDayInformation* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.6.3 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 CLM.

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 134 - GetBusinessDayInformation (camt.018) – usage case Event Query

Usage case example: camt.018_CLM_GetBusinessDayInformation_EventQuery_Example.xml

Usage case: System Time Query

In this usage case, the sender is requesting to be informed of the CLM 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 CLM system time query - TBC

Table 135 - GetBusinessDayInformation (camt.018) – usage case System Time Query

Usage case example: camt.018_CLM_GetBusinessDayInformation_SystemTimeQuery_Example.xml

14.3.7 ReturnBusinessDayInformation (camt.019)

14.3.7.1 Overview and scope of the message

This chapter illustrates the *ReturnBusinessDayInformation* message.

The *ReturnBusinessDayInformation* message is sent by the CLM to a CLM 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 CLM.

The *ReturnBusinessDayInformation* message contains such administrative data information based upon the criteria provided in the request.

Within CLM, 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 \(camt.018\)](#) [▶ 386] message, which requested the information. The system notification usage is sent in push mode, based upon operational settings with the CLM.

14.3.7.2 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
- | event types per currency
- | event types scheduled and effective date/times

References/links

The CLM-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/CLM/camt.019.001.06_CLM

Business rules applicable to the schema

No business rules are applicable to a *ReturnBusinessDayInformation* message.

14.3.7.3 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 CLM.

Specific message content

Message item	Data type/code	Utilisation
System ID Docu- ment/RtrBizDayInf/RptOrErr/BizRpt/Sy sId/MktInfrstrctrId/Cd	ExternalMarketInfrastructureCode	CLM
Business date Docu- ment/RtrBizDayInf/RptOrErr/BizRpt/Biz DayOrErr/BizDayInf/SysDt	ISODate	Current business date of CLM

Table 136 - ReturnBusinessDayInformation (camt.019) – usage case System Time Query (Data response)

Usage case example:
camt.019_CLM_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 \(camt.018\)](#) [▶ 386]).

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
Proprietary Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Prtry/	Max4Text	CLM code for the problem being in- formed.
Description Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Desc	Max140Text	Description of the problem being in- formed.

Table 137 - ReturnBusinessDayInformation (camt.019) – usage case System Time Query (Error response)

Usage case example:
camt.019_CLM_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 CLM processing event.

Specific message content

Message item	Data type/code	Utilisation
Proprietary Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Prtry/	Max4Text	CLM code for the problem being in- formed
Description Docu- ment/RtrBizDayInf/RptOrErr/OprlErr/Err /Desc	Max140Text	Description of the event being informed

Table 138 - ReturnBusinessDayInformation (camt.019) – usage case Event Query (Data response)

Usage case example: camt.019_CLM_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 \(camt.018\)](#) [▶ 386]).

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
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 in- formed

Table 139 - ReturnBusinessDayInformation (camt.019) – usage case Event Query (Error response)

Usage case **example:**
camt.019_CLM_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 CLM component.

Specific message content

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 140 - ReturnBusinessDayInformation (camt.019) – usage case System Notification

Usage case **example:**
camt.019_CLM_ReturnBusinessDayInformation_SystemNotification_Example.xml

14.3.8 ModifyStandingOrder (camt.024)

14.3.8.1 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:

- | RTGS modify standing order
- | ASI6 RTGS modify standing order
- | CLM modify standing order

The *ModifyStandingOrder* message is replied by a [Receipt \(camt.025\)](#) [▶ 397] to return a positive technical response to the sender of the message or to provide detailed information in case of an error.

14.3.8.2 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>

14.3.8.3 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 Modifiy 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 141 - 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/FinInstnId/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 142 - 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 143 - 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”.

14.3.9 Receipt (camt.025)

14.3.9.1 Overview and scope of the message

This chapter illustrates the *Receipt* message.

The *Receipt* message is sent by the CLM to a CLM Account Holder (or a party authorised by them). It is used to reply to a previously sent liquidity transfer order, payment order, order-related activity or CRDM request.

The *Receipt* message returns a positive response to the sender of the previous message or provides detailed information in case of an error.

Within CLM, 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.

- I Usage category – Settlement

- Payment Order Rejection Notification
- Payment Order Settlement Notification (Liquidity Transfer)
- Payment Order Settlement Notification (Inter-Service Liquidity Transfer)
- Negative Payment Order Settlement Notification (Inter-Service Liquidity Transfer)
- I Usage category – CLM Status
 - Receipt Acknowledgement (Amend Payment Order)
 - Positive Receipt Message (Reservation Management)
 - Negative Receipt (Overnight Deposit)
- I Usage Category – Liquidity Management
 - Reject Modify/Delete Reservation Request
 - Confirm Successful Modify/Delete Reservation Request
- I Usage Category – CRDM
 - Create/Modify Standing Order
 - Delete Standing Order
 - Modify Limit
 - Delete Limit
 - Modify Standing Order for Reservation
 - Delete Standing Order for Reservation

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.

14.3.9.2 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:

- I original message identification

- | status code
- | description

References/links

The CLM-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/CLM/camt.025.001.04_CLM

Business rules applicable to the schema

No business rules are applicable to a *Receipt* message.

14.3.9.3 The message in business context

Usage category – Settlement

All usage cases in this category 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 is not processed further. A rejection code is given and, in most cases, a reason code and reason text are provided as well.

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	CLM rejection

Table 144 - Receipt (camt.025) – usage category case Settlement – Rejected

Usage category case example: camt.025_CLM_Receipt_SettlementRejected_Example.xml

Usage category case – Settlement – Settled

In this usage category case, the recipient of the message is 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 145 - Receipt (camt.025) – usage category case Settlement – Settled

Usage category case example: camt.025_CLM_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 is given and, in most cases a reason text are provided as well.

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 146 - Receipt (camt.025) – usage category case Settlement – Unsettled

Usage case example: camt.025_CLM_Receipt_SettlementUnsettled_Example.xml

Usage category – CLM Status

All usage cases in this category 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 – CLM 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 is not processed further. A rejection code is given and, in most cases, a reason code and reason text are provided as well.

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	RRJT
Description Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	CLM rejection

Table 147 - Receipt (camt.025) – usage category case CLM Status – Rejected

Usage category case example: camt.025_CLM_Receipt_CLMStatusRejected_Example.xml

Usage category case – CLM 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/OrgnMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ StsCd	Max4AlphaNumericText	RCON
Description Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	CLM confirmation

Table 148 - Receipt (camt.025) – usage category case CLM Status – Confirmed

Usage category case example: camt.025_CLM_Receipt_CLMStatusConfirmed_Example.xml

Usage category case – CLM 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 is given and, in most cases a reason text are provided as well.

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 [537]

Table 149 - Receipt (camt.025) – usage category case CLM Status – Validation Error

Usage case example: camt.025_CLM_Receipt_CLMStatusValidationError_Example.xml

Usage category – Liquidity Management

All usage cases in this category 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	SSET
Description Docu- ment/Rct/RctDtIs/OrgnMsgId/ReqHdlg/ Desc	RestrictedFINXMax140Text	settled

Table 150 - Receipt (camt.025) – usage category case Liquidity Management – Approved

Usage category case example: camt.025_CLM_Receipt_LiquidityManagementApproved_Example.xml

Usage category – CRDM

All usage cases in this category 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 – CRDM – Rejected

In this usage category case, the recipient of the message is informed that a previously sent message in this category sent by them (or on their behalf) has been rejected and is not processed further. A rejection code is given and, in most cases, a reason code and reason text are provided as well.

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 151 - Receipt (camt.025) – usage category case CRDM – Rejected

Usage category case example: camt.025_CLM_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 152 - Receipt (camt.025) – usage category case CRDM – Completed

Usage category case example: camt.025_CLM_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 is given and, in most cases a reason text are provided as well.

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 153 - Receipt (camt.025) – usage category case CRDM – Queued

Usage case example: camt.025_CLM_Receipt_CRDMQueued_Example.xml

14.3.10 ResolutionOfInvestigation (camt.029)

14.3.10.1 Overview and scope of the message

This chapter illustrates the *ResolutionOfInvestigation* message.

The *ResolutionOfInvestigation* message is sent by CLM to a CLM Account Holder (or a party authorised by them). It is used to inform of the status of a previously requested liquidity transfer order cancellation.

The *ResolutionOfInvestigation* message only concerns the cancellation of one liquidity transfer order or one payment order.

Within CLM, the *ResolutionOfInvestigation* message has the following usages:

- I Rejection of Payment Order Cancellation Request (Negative Resolution to Investigation)

The *ResolutionOfInvestigation* message is sent in response to a [FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 452] message.

14.3.10.2 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 transaction being cancelled.

References/links

The CLM-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/CLM/camt.029.001.08_CLM

Business rules applicable to the schema

No business rules are applicable to a *ResolutionOfInvestigation* response message.

14.3.10.3 The message in business context

Usage case: Rejection of Payment Order Cancellation Request

In this usage case, the recipient is being informed that a previously sent request to cancel a payment order ([FIToFIPaymentCancellationRequest \(camt.056\)](#) [▶ 452]), has been rejected. The payment order will still go ahead settle, or has already settled.

Specific message content

Message item	Data type/code	Utilisation
Id RsltnOfInvstgtn/ Assgnmt/Id	CLM_RestrictedFINXMax35Text	PDCR
BICFI RsltnOfInvstgtn/ Assgnr/Agt/FinInstnId/ BICFI	CLM_BIC11Text	BIC
BICFI RsltnOfInvstgtn/ Assgne /Agt/FinInstnId/ BICFI	CLM_BIC11Text	BIC
Confirmation RsltnOfInvstgtn/ Conf	ExternalInvestigationExecutionConfir- mation1Code	RJCR
Cancellation Status Identification RsltnOfInvstgtn/ CxlDtls/TxInfAndSts/ CxlStsId	CLM_RestrictedFINXMax35Text	The cancellation status identification can be used for reconciliation or to link tasks relating to the cancellation re- quest
Original Message Identification RsltnOfInvstgtn/ CxlDtls/TxInfAndSts/ OrgnlMsgId	CLM_RestrictedFINXMax35Text	If present in underlying FIToFIPay- mentCancellationRequest (camt.056) [452] , the original instruction identifi- cation is recommended to be trans- ported in the <i>ResolutionOfInvestigation</i> under original instruction identification
Original Message Name Identification RsltnOfInvstgtn/ CxlDtls/TxInfAndSts/ OrgnlGrpInf/ OrgnlMsgNmId	CLM_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) [▶ 452] , 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 154 - ResolutionOfInvestigation (camt.029) – usage case Rejection of Payment Order Cancellation Request

Usage case example:
camt.029_CLM_ResolutionOfInvestigation_RejectionOfPaymentOrderCancellationRequest_Example.xml

14.3.11 GetReservation (camt.046)

14.3.11.1 Overview and scope of the message

This chapter illustrates the *GetReservation* message.

The *GetReservation* message is sent by a CLM Account Holder (or on their behalf by an authorised party) to CLM. It is used to request reservation information on CBOs and cash withdrawals set by the CLM 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 CLM, the *GetReservation* message has the following usages:

- Current Reservations Query

In response to the *GetReservation* message, a [ReturnReservation \(camt.047\)](#) [▶ 411] message containing the requested information is returned.

14.3.11.2 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 CLM-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/CLM/camt.046.001.04_CLM

Business rules applicable to the schema

For business rules applicable to *GetReservation* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.11.3 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 CLM MCAs 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	CLM Account Holder BIC
Identification Document/GetRsvatn/ RsvatnQryDef/ RsvatnCrit/ NewCrit/ SchCrit/ AcctId / Othr / ID	Max35Text	CLM MCA ID

Table 155 - GetReservation (camt.046) – usage case Current Reservations Query

Usage case example: camt.046_CLM_GetReservation_CurrentReservationsQuery_Example.xml

14.3.12 ReturnReservation (camt.047)

14.3.12.1 Overview and scope of the message

This chapter illustrates the *ReturnReservation* message.

The *ReturnReservation* message is sent by CLM to a CLM 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 CLM Account Holder (or on their behalf by an authorised party).

Within CLM, the *ReturnReservation* message has the following usages:

- Current Reservations Query (Data or Error response)

The *ReturnReservation* message is sent in response to a [GetReservation \(camt.046\)](#) [▶ 409] message which requested the information.

14.3.12.2 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 CLM-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/CLM/camt.047.001.05_CLM

Business rules applicable to the schema

No business rules are applicable to a *ReturnReservation* response message.

14.3.12.3 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 CLM MCAs within their query criteria.

Specific message content

Message item	Data type/code	Utilisation
Code Docu- ment/RtrRsvatn/RptOrErr/BizRpt/CurR svatn/RsvatnId/Tp/Cd	ReservationType2Code	BLCK
BICFI Docu- ment/RtrRsvatn/RptOrErr/BizRpt/CurR svatn/RsvatnId/AcctOwnr/FinInstnId/BI CFI	BICFIIdentifier	CLM MCA owner
Amount with Currency Docu- ment/RtrRsvatn/RptOrErr/BizRpt/CurR svatn/RsvatnOrErr/Rsvatn/Amt/AmtWth Ccy	ActiveCurrencyAndAmount	Current reservation
Current reservation – status Docu- ment/RtrRsvatn/RptOrErr/BizRpt/CurR svatn/RsvatnOrErr/Rsvatn/Sts/Cd	ReservationStatusCode	ENAB REQD

Table 156 - ReturnReservation (camt.047) – usage case Current Reservations Query (Data response)

Usage case example:
camt.047_CLM_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 \(camt.046\)](#) [▶ 409]).

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
	TBC	

Table 157 - ReturnReservation (camt.047) – usage case Current Reservations Query (Error response)

Usage	case	example:
		<code>camt.047_CLM_ReturnReservation_CurrentReservationsQueryError_Example.xml</code>

14.3.13 ModifyReservation (camt.048)

14.3.13.1 Overview and scope of the message

This chapter illustrates the *ModifyReservation* message.

The *ModifyReservation* message is sent by a CLM Account Holder (or on their behalf by an authorised party) directly to CRDM. It is used to modify the details of one existing reservations set by the CLM Account Holder (or on their behalf by an authorised party).

The *ModifyReservation* message will contain the new value that the CLM Account Holder wants to be applied to the reservations identified in the message.

Within CLM, 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 \(camt.025\)](#) [397] message is sent, indicating the success or rejection/failure of the modification.

14.3.13.2 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:

- | start date & time
- | amount with currency

References/links

The CLM-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/CLM/camt.048.001.04_CLM

Business rules applicable to the schema

For business rules applicable to *ModifyReservation* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.13.3 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 CLM MCA 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/ModfyRsvatn/RsvatnId/Cur/Tp/Cd	ReservationTypeCode	BLCK
BICFI Docu- ment/ModfyRsvatn/RsvatnId/Cur/AcctO wnr/FinInstnId/BICFI	BICFIIdentifier	CLM MCA owner
New reservation amount Docu- ment/ModfyRsvatn/NewRsvatnValSet/ Amt/AmtWthCcy	ActiveCurrencyAndAmount	New reservation amount required

Table 158 - ModifyReservation (camt.048) - usage case Modify Reservation Request

Usage case example: camt.048_CLM_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 159 - ModifyReservation (camt.048) - usage case Modify Standing Order for Reservation

Usage case example: camt.048_CLM_ModifyReservation_ModifyStandingOrderFor ReservationRequest_Example.xml

14.3.14 DeleteReservation (camt.049)

14.3.14.1 Overview and scope of the message

This chapter illustrates the *DeleteReservation* message.

The *DeleteReservation* message is sent by a CLM Account Holder (or on their behalf by an authorised party) to CLM. It is used to request the deletion of one particular reservation set by the CLM Account Holder (or on their behalf by an authorised party).

The *DeleteReservation* message allows for the deletion of only one reservation facility.

Within CLM, the *DeleteReservation* message has the following usages:

- | Delete Reservation Request
- | Delete Standing Order for Reservation

In response to the *DeleteReservation* message, a [Receipt \(camt.025\)](#) [▶ 397] is sent, indicating the success or rejection/failure of the deletion.

14.3.14.2 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 CLM-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/CLM/camt.049.001.04_CLM

Business rules applicable to the schema

For business rules applicable to *DeleteReservation* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.14.3 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 CLM MCA 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	ReservationType2Code	HPAR/UPAR
CurRsvatn/Tp/Cd		
BICFI	BICFIIdentifier	Current reservation – CLM MCA owner
AcctOwnr/ FinInstnId/ BICFI		
Id	Max35Text	CLM Account Holder account ID
AcctId/ Othr/ Id		
Message item	Data type/code	Utilisation
	TBC	

Table 160 - DeleteReservation (camt.049) – usage case Delete Reservation Request

Usage case example: camt.049_CLM_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 161 - DeleteReservation (camt.049) – usage case Delete Standing Order for Reservation

Usage case example:
camt.049_CLM_DeleteReservation_DeleteStandingOrderForReservation_Example.xml

14.3.15 LiquidityCreditTransfer (camt.050)

14.3.15.1 Overview and scope of the message

This chapter illustrates the *LiquidityCreditTransfer* message.

The *LiquidityCreditTransfer* message is sent by a CLM Account Holder (or on their behalf by an authorised party) to CLM.

The *LiquidityCreditTransfer* message may also be sent by CLM itself to RTGS (on behalf of a CLM Account Holder), in order to manipulate liquidity at need and to maintain the floor and ceiling balances of CLM MCAs.

The *LiquidityCreditTransfer* message is used to request a transfer of funds

- | between two CLM MCAs belonging to the CLM Account Holder, or
- | from a CLM MCA to a DCA of another component, within the same liquidity group of MCAs, defined within CLM with each CLM MCA being identified using its BIC11.

Within CLM, the *LiquidityCreditTransfer* message has the following usages:

- | Automated Inter-Service Liquidity Transfer Order
- | Payment Order Message
- | Inter-Service Liquidity Transfer (Floor Processing)
- | Inter-Service Liquidity Transfer (Ceiling Processing)
- | Liquidity Transfer Order (Overnight Deposit)

In response to the *LiquidityCreditTransfer* message, a [Receipt \(camt.025\)](#) [▶ 397] message containing the status of the liquidity transfer is returned to the sending party.

14.3.15.2 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 CLM-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/CLM/camt.050.001.04_CLM

Business rules applicable to the schema

For business rules applicable to *LiquidityCreditTransfer* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.15.3 The message in business context

Usage case: Automated Interservice Liquidity Transfer Order

In this usage case, the debiting party of a payment order has insufficient balance in its CLM MCA to achieve settlement.

This message is sent by CLM to RTGS to execute a movement of funds from a RTGS DCA into the CLM MCA.

This will increase the balance of the CLM MCA to enable the payment order to be settled.

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 Document/ LqdyCdtTrf/LqdyCdtTrf/LqdyTrf/En dToEndId	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/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/SettlnDt	ISODate	Current RTGS business date

Table 162 - LiquidityCreditTransfer (camt.050) – usage case Automated Interservice Liquidity Transfer Order

Usage case example:
camt.050_CLM_LiquidityCreditTransfer_AutomatedInterServiceLiquidityTransferOrder_Example.xml

Usage case: Payment Order Message

In this usage case, the message is sent by a CLM Account Holder to instruct the CLM component to execute a movement of funds from one of its CLM MCAs.

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 Document/ LqdyCdtTrf/LqdyCdtTrf/LqdyTrfl/En dToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/CdtrAcct	RestrictedFINX2Max34Text	CLM MCA to be credited

Message item	Data type/code	Utilisation
/ID/Othr/ID		
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/SettImDt	ISODate	Current CLM business date

Table 163 - LiquidityCreditTransfer (camt.050) – usage case Payment Order Message

Usage case example: camt.050_CLM_LiquidityCreditTransfer_PaymentOrderMessage_Example.xml

Usage case: Inter-Service Liquidity Transfer (Floor Processing)

In this usage case, a CLM MCA balance has fallen below its pre-defined floor amount.

This message is sent by the CLM component to the RTGS component to execute a movement of funds from a RTGS DCA into the CLM MCA.

This will increase the balance of the CLM MCA 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/LqdyTrf/ InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification Document/ LqdyCdtTrf/LqdyCdtTrf/LqdyTrf/En dToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party

Message item	Data type/code	Utilisation
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 164 - LiquidityCreditTransfer (camt.050) – usage case Inter-Service Liquidity Transfer (Floor Processing)

Usage case example: camt.050_CLM_LiquidityCreditTransfer_InterServiceLTFloor_Example.xml

Usage case: Inter-Service Liquidity Transfer (Ceiling Processing)

In this usage case, a CLM MCA balance has risen above its pre-defined ceiling amount.

This message is sent by the CLM component to the RTGS component to execute a movement of funds into a RTGS DCA from the CLM MCA.

This will decrease the balance of the CLM MCA 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/LqdyTrfl d/InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification	RestrictedFINXMax16Text	Unique ID set by the initiating party

Message item	Data type/code	Utilisation
Document/ LqdyCdtTrf/LqdyCdtTrf/LqdyTrfId/EndToEndId		
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 165 - LiquidityCreditTransfer (camt.050) – usage case Inter-Service Liquidity Transfer (Ceiling Processing)

Usage case example: camt.050_CLM_LiquidityCreditTransfer_InterServiceLTceiling_Example.xml

Usage case: Liquidity Transfer Order (Overnight Deposit)

In this usage case, a CLM Account Holder is instructing a movement of funds between CLM MCAs in order to arrange its balances for the overnight period.

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/LqdyTrfId/InstrId	RestrictedFINXMax16Text	Not provided
EndToEndIdentification	RestrictedFINXMax16Text	Unique ID set by the initiating party

Message item	Data type/code	Utilisation
Document/ LqdyCdtTrf/LqdyCdtTrf/LqdyTrfId/EndToEndId		
Identification Docu- ment/LqdyCdtTrf/LqdyCdtTrf/CdtrAcct /ID/Othr/ID	RestrictedFINX2Max34Text	CLM overnight deposit 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	CLM MCA to be debited
SettlementDate Docu- ment/LqdyCdtTrf/LqdyCdtTrf/SettlmDt	ISODate	Current CLM business date
Message item	Data type/code	Utilisation
	TBC	

Table 166 - LiquidityCreditTransfer (camt.050) – usage case Liquidity Transfer Order (Overnight Deposit)

Usage case example:
camt.050_CLM_LiquidityCreditTransfer_LiquidityTransferOrderOvernightDeposit_Example.xml

14.3.16 BankToCustomerStatement (camt.053)

14.3.16.1 Overview and scope of the message

This chapter illustrates the *BankToCustomerStatement* message.

The *BankToCustomerStatement* message is sent by CLM to a CLM Account Holder (or a party authorised by them). It is used to inform of the entries booked to a CLM MCA and to provide account balance information at a given point in time as an account statement.

The *BankToCustomerStatement* message is also sent by CLM 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 liquidity transfers and payments that have been included in the entry.

Within CLM, the *BankToCustomerStatement* message has the following usages:

- | Query Response Message for Business Data
- | Statement of Accounts
- | CB General Ledger

The *BankToCustomerStatement* message is produced depending upon a party's reporting configurations.

14.3.16.2 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 CLM DCA. It may contain:

- | statement identification
- | report sequence information
- | creation timestamp
- | account identification
- | account balance/s
- | summary of transactions
- | details of each entry: entry reference, amount & currency, debit/credit indicator, status, booking date, value date, bank transaction code

References/links

The CLM-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/CLM/camt.053.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *BankToCustomerStatement* message.

14.3.16.3 The message in business context

Usage case: Query Response Message for Business Data

In this usage case, the CLM Account Holder has specifically requested to be informed of movements for CLM MCAs 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-	Max35Text	Reporting sequence –from.

Message item	Data type/code	Utilisation
ment/BkToCstmrStmt/Stmt/RptgSeq/Fr Seq		
Reporting sequence Docu- ment/BkToCstmrStmt/Stmt/RptgSeq/To Seq	Max35Text	Reporting sequence –to.
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 number.
Currency Docu- ment/BkToCstmrStmt/Stmt/Acct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS DCA.
Account owner Docu-	AnyBICIdentifier	Owner of the RTGS DCA.

Message item	Data type/code	Utilisation
ment/BkToCstmrStmnt/Stmnt/Acct/Ownr/ID/OrgId/AnyBIC		
Account owner country Docu- ment/BkToCstmrStmnt/Stmnt/Acct/Ownr/ID/CtryOfRes	CountryCode	Country of residence, of the owner of the RTGS DCA.
Multiple repetitions of Balance information		
Balance type Docu- ment/BkToCstmrStmnt/Stmnt/Bal/TP/CdOrPrty/Cd	CSLD_BalanceTypeCode	Type of balance.
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/Avlby	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.

Message item	Data type/code	Utilisation
Net sum of all entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/TtlNetNtry/Amt	NonNegativeDecimalNumber	Net total sum of all of entries on state- ment.
Credit debit ind Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/TtlNetNtry/CdtDbtInd	CreditDebitCode	Credit debit ind for the net total sum of all of entries on statement.
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	ISODateTime	Date and time the entry was booked.

Message item	Data type/code	Utilisation
Dt/DtTm		
Value date Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/ValDt/ Dt	ISODate	Value date.
Value datetime Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/ValDt/ DtTm	ISODateTime	Value date & time.
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 167 - BankToCustomerStatement (camt.053) – usage case Query Response Message for Business Data

Usage case example:
camt.053_CLM_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 CLM MCAs in its data scope. This report message is automatically generated by the CLM component in accordance with the reporting configuration settings applied by the CLM Account Holder.

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-	YesNoIndicator	Indicates the last page.

Message item	Data type/code	Utilisation
ment/BkToCstmrStmt/GrpHdr/MsgPgnt n/LastPgInd		
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 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	Max34Text	RTGS DCA number.

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmt/Stmt/Acct/ID/Oth r/ID		
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.
Multiple repetitions of Balance infor- mation		
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/Avlby	CashAvailability	Availability of balance. Might be need- ed for non-EURO – TBD.
Transactions summary		

Message item	Data type/code	Utilisation
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 state- ment.
Credit debit ind Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/TtlNetNtry/CdtDbtInd	CreditDebitCode	Credit debit ind for the net total sum of all of entries on statement.
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-	CreditDebitCode	Credit debit indicator for entry amount.

Message item	Data type/code	Utilisation
ment/BkToCstmrStmt/Stmt/Ntry/CrdDbt Ind		
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 & time.
Bank transaction code Docu- ment/BkToCstmrStmt/Stmt/Ntry/BkTxC d/Prtry/Cd	CSLD_BankTransactionCode	Transaction code.
Message ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/Refs/MsgId	Max35Text	Unique message reference of the in- structing 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	Max35Text	Transaction ID set by the instructing

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Refs/TxID		agent.
Transaction ID Docu- ment/BkToCstmrStmnt/Stmnt/Ntry/NtryDtl s/TxDtls/Amt	CSLD_Max14_Max2DecimalAmount	Entry detail amount
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 168 - BankToCustomerStatement (camt.053) – usage case Statement of Accounts

Usage case example:
camt.053_CLM_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 CLM component in accordance with the reporting configuration settings applied by the CB.

Specific message content

Message item	Data type/code	Utilisation
Statement ID Document/BkToCstmrStmnt/Stmnt/ID	Max35Text	Statement number.
Reporting sequence	Max35Text	Reporting sequence –from.

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmnt/Stmnt/RptgSeq/Fr Seq		
Reporting sequence Docu- ment/BkToCstmrStmnt/Stmnt/RptgSeq/To Seq	Max35Text	Reporting sequence –to.
Reporting sequence Docu- ment/BkToCstmrStmnt/Stmnt/RptgSeq/Fr ToSeq/FrSeq	Max35Text	Reporting sequence –range from.
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	AnyBICIdentifier	Owner of the RTGS DCA.

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmt/Stmt/Acct/Ownr/I D/OrgId/AnyBIC		
Account owner country Docu- ment/BkToCstmrStmt/Stmt/Acct/Ownr/I D/CtryOfRes	CountryCode	Country of residence, of the owner of the RTGS DCA.
Multiple repetitions of Balance infor- mation		
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/Avlby	CashAvailability	Availability of balance. Might be need- ed 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	DecimalNumber	Total sum of all of entries on statement.

Message item	Data type/code	Utilisation
/TtlNtries/Sum		
Net sum of all entries Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/TtlNetNtry/Amt	NonNegativeDecimalNumber	Net total sum of all of entries on state- ment.
Credit debit ind Docu- ment/BkToCstmrStmt/Stmt/TxsSummry /TtlNtries/TtlNetNtry/CdtDbtInd	CreditDebitCode	Credit debit ind for the net total sum of all of entries on statement.
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	ISODateTime	Date and time the entry was booked.

Message item	Data type/code	Utilisation
Docu- ment/BkToCstmrStmt/Stmt/Ntry/Bookg Dt/DtTm		
Value date Docu- ment/BkToCstmrStmt/Stmt/Ntry/ValDt/ Dt	ISODate	Value date.
Value datetime Docu- ment/BkToCstmrStmt/Stmt/Ntry/ValDt/ DtTm	ISODateTime	Value date & time.
Bank transaction code Docu- ment/BkToCstmrStmt/Stmt/Ntry/BkTxC d/Prtry/Cd	CSLD_BankTransactionCode	Transaction code.
Message ID Docu- ment/BkToCstmrStmt/Stmt/Ntry/NtryDtl s/TxDtls/Refs/MsgId	Max35Text	Unique message reference of the in- structing 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	CSLD_Max14_Max2DecimalAmount	Entry detail amount

Message item	Data type/code	Utilisation
s/TxDtls/Amt		
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 169 - BankToCustomerStatement (camt.053) – usage case CB General Ledger

Usage case example:
camt.053_CLM_BankToCustomerStatement_CentralBankGeneralLedger_Example.xml

14.3.17 BankToCustomerDebitCreditNotification (camt.054)

14.3.17.1 Overview and scope of the message

This chapter illustrates the *BankToCustomerDebitCreditNotification* message.

The *BankToCustomerDebitCreditNotification* message is sent by CLM to a CLM Account Holder (or a party authorised by them) or a CB. It is used to inform of the credit or the debit of a certain amount on one of their CLM MCAs. The *BankToCustomerDebitCreditNotification* message is sent by CLM when the CLM MCA owner was not the instructor of the movement.

The *BankToCustomerDebitCreditNotification* message is only concerned with one single debit or credit movement on one single CLM MCA.

Within CLM, the *BankToCustomerDebitCreditNotification* message has the following usages:

- | Liquidity Transfer Settlement Notification
- | Payment Order Settlement Notification (Intra-Service LTs)
- | Payment Order Settlement Notification (Inter-Service LTs)
- | Payment Order Settlement Notification (CBOs)

- | Settlement Notification (CLM Standing Order SoD)
- | Debit/Credit Notification related to:
 - Settlement of Connected Payments
 - Modification of Credit Line
 - Settlement of Requested Marginal Lending
 - Settlement of Automated Marginal Lending
 - Overnight Deposit

The *BankToCustomerDebitCreditNotification* message is sent in response to a debit/credit movement activity within CLM.

14.3.17.2 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 CLM MCA. It may contain:

- | identification
- | creation timestamp
- | account identification
- | amount
- | debit/credit indicator
- | status
- | booking date
- | value date
- | bank transaction code
- | amount details
- | related parties & agents

I local instrument

References/links

The CLM-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/CLM/camt.054.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *BankToCustomerDebitCreditNotification* response message.

14.3.17.3 The message in business context

Usage case: Liquidity Transfer Settlement Notification

In this usage case, the recipient is being informed of the successful execution of a credit or debit movement against one of its CLM MCAs, resulting from a LT.

Specific message content

Message item	Data type/code	Utilisation
Identification Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/I D	Max35Text	CLM booking ID
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	CLM MCA
Currency Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the CLM MCA

Message item	Data type/code	Utilisation
AnyBIC Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/A cct/Ownr/ID/Orgld/AnyBIC	CSLD_BIC11Text	Party owning the CLM MCA
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 entry was booked (business day + settlement time)
DateTime Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/ValDt/DtTm	ISODateTime	Time when the entry amount became available (business day + settlement time)
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 instruction which caused the entry
InstructionIdentification Docu-	Max35Text	Identification of the underlying instruction which caused the entry

Message item	Data type/code	Utilisation
Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Refs/InstrId		
EndToEndIdentification Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Refs/EndToEndId	Max35Text	End-to-end ID of the underlying instruction which caused the entry
TransactionIdentification Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Refs/TxId	Max35Text	Transaction ID of the underlying instruction which caused the entry
Amount Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Amt	CSLD_Max14_Max5DecimalAmount	Instructed amount
CreditDebitIndicator Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/CrdDbtInd	CreditDebitCode	CRDT or DBIT
Identification Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdPties/DbtrAcct/ID/Othr/ID	Max34Text	Debtor account in the underlying transaction
Identification Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdPties/CdtrAcct/ID/Othr/ID	Max34Text	Creditor account in the underlying transaction

Table 170 - BankToCustomerDebitCreditNotification (camt.054) – usage case Liquidity Transfer Settlement Notification

Usage case example:
camt.054_CLM_BankToCustomerDebitCreditNotification_LiquidityTransferSettlementNotification_Example.xml

In this example a confirmation of a credit movement on the “MAINCASHACCOUNT1” resulting from a liquidity transfer is sent to the corresponding party.

Usage case: Payment Order Settlement Notification (Intra-Service LTs)

In this usage case, the recipient is being informed of the successful execution of a credit or debit movement against one of its CLM MCAs, resulting from an intra-service liquidity transfer.

Specific message content

Message item	Data type/code	Utilisation
	TBC	

Table 171 - BankToCustomerDebitCreditNotification (camt.054) – usage case Payment Order Settlement Notification (Intra-Service LTs)

Usage case example:
**camt.054_CLM_BankToCustomerDebitCreditNotification_PaymentOrderSettlementNotificationIntraS
 LT_Example.xml**

Usage case: Payment Order Settlement Notification (Inter-Service LTs)

In this usage case, the recipient is being informed of the successful execution of a credit or debit movement against one of its CLM MCAs, resulting from an inter-service liquidity transfer.

Specific message content

Message item	Data type/code	Utilisation
	TBC	

Table 172 - BankToCustomerDebitCreditNotification (camt.054) – usage case Payment Order Settlement Notification (Inter-Service LTs)

Usage case example:
**camt.054_CLM_BankToCustomerDebitCreditNotification_PaymentOrderSettlementNotificationInterS
 LT_Example.xml**

Usage case: Payment Order Settlement Notification (Central Bank Operations)

In this usage case, the recipient is being informed of the successful execution of a credit or debit movement against one of its CLM MCAs, resulting from a CB operation process.

Specific message content

Message item	Data type/code	Utilisation
	TBC	

Table 173 - BankToCustomerDebitCreditNotification (camt.054) – usage case Payment Order Settlement Notification (CBOs)

Usage case **example:**
camt.054_CLM_BankToCustomerDebitCreditNotification_PaymentOrderSettlementNotificationCBO_Example.xml

Usage case: Settlement Notification (CLM Standing Order start-of-day)

In this usage case, the CLM component sends a confirmation of debit or credit movement to a CLM Account Holder or a CB if one of its CLM MCAs was credited or debited as the result of the settlement of a CLM standing order at SoD.

Specific message content

Message item	Data type/code	Utilisation
	TBC	

Table 174 - BankToCustomerDebitCreditNotification (camt.054) – usage case Settlement Notification (CLM Standing Order SoD)

Please refer to general description.

Usage case **example:**
camt.054_CLM_BankToCustomerDebitCreditNotification_SettlementNotificationCLMSOsod_Example.xml

The example illustrates a settlement notification resulting from a CLM standing order which transfers an amount from a participant’s MCA to DCA.

Usage case: Debit/Credit Notification

In this usage case, the CLM component sends a confirmation of credit or a debit movement to a participant’s, or a CB’s, CLM MCA.

Same example as above.

Usage case: Connected Payment Settlement Notification

Specific message content

Please refer to generic description except for the specifics below.

Message item	Data type/code	Utilisation
BankTransactionCode Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BnkTxCd/Prtry/Cd	CSLD_BankTransactionCode	PMNT
Transaction Details Amount Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Nt ry/NtryDtls/TxDtls/Amt	CSLD_Max14_Max5DecimalAmount	Credit line change
Local Instrument Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Nt ry/NtryDtls/TxDtls/LclInstrm/Prtry		CONPAY, CRDTLN

Table 175 - BankToCustomerDebitCreditNotification (camt.054) – usage case Connected Payment Settlement Notification

Usage case example:
camt.054_RTGS_BankToCustomerDebitCreditNotification_ConnectedPayment_Example.xml

The example illustrates a settlement notification for a connected payment with a change in credit line.

Usage case: Modify Credit Line Settlement Notification

Specific message content

Please refer to generic description except for the specifics below.

Message item	Data type/code	Utilisation
BankTransactionCode Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BnkTxCd/Prtry/Cd	CSLD_BankTransactionCode	PMNT
Local Instrument Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Nt ry/NtryDtls/TxDtls/LclInstrm/Prtry		CREDITLINE

Table 176 - BankToCustomerDebitCreditNotification (camt.054) – usage case Modify Credit Line Settlement Notification

Usage case example: camt.054_RTGS_BankToCustomerDebitCreditNotification_ ModifyCreditLine_Example.xml

Example illustrates a settlement notification including a modification of the credit line.

Usage case: Marginal Lending Interest (on request) Settlement Notification

Specific message content

Please refer to generic description except for the specifics below.

Message item	Data type/code	Utilisation
BankTransactionCode Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/BnkTxCd/Prtry/Cd	CSLD_BankTransactionCode	PMNT
Local Instrument Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtIs/TxDtIs/LclInstrm/Prtry		SFMLOINT

Table 177 - BankToCustomerDebitCreditNotification (camt.054) – usage case Marginal Lending Interest (on request) Settlement Notification

Usage case example: camt.054_RTGS_BankToCustomerDebitCreditNotification_ MarginalLendingInterestOnRequest_Example.xml

Example illustrates a settlement notification for marginal lending interest payment (on request).

Usage case: Marginal Lending Interest (automated) Settlement Notification

Specific message content

Please refer to generic description except for the specifics below.

Message item	Data type/code	Utilisation
BankTransactionCode Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BnkTxCd/Prtry/Cd	CSLD_BankTransactionCode	PMNT
Local Instrument Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Nt ry/NtryDtIs/TxDtIs/LclInstrm/Prtry		SFMLAINT

**Table 178 - BankToCustomerDebitCreditNotification (camt.054) – usage case Marginal Lending Interest (auto-
mated) Settlement Notification**

Usage case example:
camt.054_RTGS_BankToCustomerDebitCreditNotification_MarginalLendingInterestAutomated_Example.xml

Example illustrates a settlement notification for marginal lending interest automated payment.

Usage case: Overnight Deposit Interest Payment Settlement Notification

Specific message content

Please refer to generic description except for the specifics below.

Message item	Data type/code	Utilisation
BankTransactionCode Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/N try/BnkTxCd/Prtry/Cd	CSLD_BankTransactionCode	PMNT
Local Instrument Docu- ment/BkToCstmrDbtCdtNtfctn/NtFctn/Nt ry/NtryDtIs/TxDtIs/LclInstrm/Prtry		SFOVDINT

**Table 179 - BankToCustomerDebitCreditNotification (camt.054) – usage case Overnight Deposit Interest Pay-
ment Settlement Notification**

Usage case example: camt.054_RTGS_BankToCustomerDebitCreditNotification_OvernightDepositInterest_Example.xml

Example illustrates a settlement notification for overnight deposit interest payment.

14.3.18 FIToFIPaymentCancellationRequest (camt.056)

14.3.18.1 Overview and scope of the message

This chapter illustrates the *FIToFIPaymentCancellationRequest* message.

The *FIToFIPaymentCancellationRequest* message is sent by a CLM Account Holder (or on their behalf by an authorised party) or a CB to CLM. It is used to request the cancellation of an original payment order.

The *FIToFIPaymentCancellationRequest* message concerns only one original payment order.

Within CLM, the *FIToFIPaymentCancellationRequest* message has the following usages:

- | Cancel Payment Order

In response to the *FIToFIPaymentCancellationRequest* message, a [ResolutionOfInvestigation \(camt.029\)](#) [▶ 406] is sent, indicating the success or rejection/failure of the cancellation.

14.3.18.2 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:

- | cancellation identification
- | original group information
- | original: instruction identification, end-to-end identification, transaction identification, clearing system reference
- | original interbank settlement amount
- | original interbank settlement date
- | cancellation reason information

References/links

The CLM-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/CLM/camt.056.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *FIToFIPaymentCancellationRequest* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.18.3 The message in business context

Usage case: Cancel Payment Order

In this usage case, a CLM Account Holder or CB (or a party authorised by them) is requesting that a previously sent payment order should be cancelled.

If the previously sent payment order has not yet been settled, then this message intends that it will never reach settlement. If the previously sent payment order 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	CLM_RestrictedFINXMax35Text	Assignment identification
Docu- ment/FIToFIPmtCxlReq/Assgnmt/Assg nr/Agt/FinInstnId/BICFI	CLM_BIC11Text	Assigner BIC
Docu- ment/FIToFIPmtCxlReq/Assgnmt/Assg ne/Agt/FinInstnId/BICFI	CLM_BIC11Text	Assignee BIC
Docu- ment/FIToFIPmtCxlReq/Assgnmt/CreD tTm	ISODateTime	Creation timestamp

Message item	Data type/code	Utilisation
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlId	CLM_RestrictedFINXMax35Text	Cancellation ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlGrpInf/OrgnlMsgId	CLM_RestrictedFINXMax35Text	Original instruction message ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlGrpInf/OrgnlMsgNmId	CLM_XMLMessageNamePattern	Original instruction message name ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlGrpInf/OrgnlCreDtTm	ISODateTime	Original instruction message time
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 clearing system reference
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ OrgnlIntrBkSttlmAmt	CLM_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-	Max70Text	Postal address street

Message item	Data type/code	Utilisation
ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/PstlAdr/StrtNm		
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
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/OrgId/AnyBIC	AnyBICIdentifier	Identification by BIC
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/OrgId/Othr/Id	Max35Text	Identification by other ID
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/OrgId/Othr/SchmeNm/Cd	ExternalOrganisationIdentification1Code	Identification by other scheme
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/OrgId/Othr/SchmeNm/Prtry	Max35Text	Identification by other scheme-code
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/OrgId/Othr/Issr	Max35Text	Identification by other scheme issuer
Docu-	ISODate	Identification by private – birth date

Message item	Data type/code	Utilisation
ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/DtAndPlcOfBirth/BirthDt		
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/DtAndPlcOfBirth/PrvcOfBirth	Max35Text	Identification by private – birth place
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/DtAndPlcOfBirth/CityOfBirth	Max35Text	Identification by private – birth city
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/DtAndPlcOfBirth/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/SchmeNm/Cd	ExternalPersonIdentification1Code	Identification by private other - scheme
Docu- ment/FIToFIPmtCxlReq/Undrlyg/TxInf/ CxlRsnInf/Orgtr/Id/PrvtId/Othr/SchmeNm/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 SYAD

Table 180 - FIToFIPaymentCancellationRequest (camt.056) – usage case Cancel Payment Order

Usage **case** **example:**
camt.056_CLM_FIToFIPaymentCancellationRequest_CancelPaymentOrder_Example.xml

The example illustrates a request from a participant to the CLM system to cancel a [FinancialInstitution-CreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491] identified by several identifications from the original transaction. The request will not be forwarded to the next party if the transaction is already settled.

14.3.19 GetStandingOrder (camt.069)

14.3.19.1 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 \(camt.070\)](#) [▶ 459] to return the retrieved standing order information or to provide detailed information in case of an error (e.g. no rows retrieved).

14.3.19.2 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>

14.3.19.3 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 181 - 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.

14.3.20 ReturnStandingOrder (camt.070)

14.3.20.1 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 \(camt.069\)](#) [▶ 457].

14.3.20.2 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

- | CreditDebit 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>

14.3.20.3 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/StgOrdrl d/Id	RestrictedFINMax16Text	Identification
Account identification Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrl d/Acct/Id/Othr/Id	RestrictedFINMax34Text	Account identification
Account owner	BICFIIdentifier	Account owner

Message item	Data type/code	Utilisation
Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr/ d/AcctOwnr/FinInstnId/BICFI		
Amount Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/Amt/AmtWthtCcy	RestrictedFINImpliedCurrencyAndA- mount	Amount
CreditDebit indicator Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/CdtDbtInd	CreditDebitCode	CreditDebit indicator
Validity period Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/VldtyPrd	DatePeriodDetails1	Validity period
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 182 - 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/StgOrdrId/Id	RestrictedFINMax16Text	Identification
Account identification Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrId/AcctId/Othr/Id	RestrictedFINMax34Text	Account identification
Account owner Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdrId/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Technical account BIC
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	BICFIIdentifier	Responsible NCB

Message item	Data type/code	Utilisation
Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/RspnsblPty/FinInstnId/Bl CFI		
Execution type Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/ExctnTp/Evt/Cd	ExternalSystemEventType1Code	Execution type
Creditor TBD	TBD	Creditor BIC
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 183 - 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	RestrictedFINMax16Text	Identification
Account identification Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr/ Acct/Id/Othr/Id	RestrictedFINMax34Text	Account identification
Account owner Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr/ AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Account owner
Amount Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/Amt/AmtWthtCcy	RestrictedFINImpliedCurrencyAndA- mount	Amount
CreditDebit indicator Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/StgOrdr/CdtDbtInd	CreditDebitCode	CreditDebit indicator
Validity period Docu- ment/RtrStgOrdr/RptOrErr/Rpt/StgOrdr OrErr/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 party
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 184 - 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 185 - ReturnStandingOrder (camt.070) – usage case Error

14.3.21 DeleteStandingOrder (camt.071)

14.3.21.1 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 \(camt.025\)](#) [▶ 397] to return a positive technical response to the sender of the message or to provide detailed information in case of an error.

14.3.21.2 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>

14.3.21.3 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/ Id	RestrictedFINMax16Text	Standing order identification
Account identification Docu- ment/DelStgOrdr/StgOrdrDtls/StgOrdr/ Acct/Id/Othr/Id	RestrictedFINMax34Text	Account identification

Table 186 - 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”.

14.3.22 BillingReportRequest (camt.076)

Will be completed in v2.0.

14.3.22.1 Overview and scope of the message

14.3.22.2 Schema

14.3.22.3 The message in business context

14.3.23 BillingReport (camt.077)

Will be completed in v2.0.

14.3.23.1 The message in business context

14.3.23.2 Schema

14.3.23.3 The message in business context

14.3.24 AuditTrailQuery (camt.097)

14.3.24.1 Overview and scope of the message

This chapter illustrates the *AuditTrailQuery* message.

The *AuditTrailQuery* message is sent by a CLM Account Holder (or on their behalf by an authorised party) to CLM. It is used to request information on audit trails which have been logged to CLM.

The *AuditTrailQuery* message can be used to query audit trails relating to: transactions, credit lines or reservations.

Within CLM, the *AuditTrailQuery* message has the following usages:

- Audit Trail for CLM Query

In response to the *AuditTrailQuery* message, an [AuditTrailReport \(camt.098\)](#) [▶ 470] message containing the requested information is returned.

14.3.24.2 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, credit line, reservation)

References/links

The CLM-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/CLM/camt.097.001.01_CLM

14.3.24.3 The message in business context

Usage case: Audit Trail for CLM Query

In this usage case, the sender requests information regarding the audit trail of the transaction, credit line or reservation 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/AudtTrlQry/MsgHdr/MsgId	RestrictedFINMax35Text	Point to point reference, as assigned by the sender, to unambiguously identify the message
Account Identification Docu- ment/AudtTrlQry/SchCrit/AcctId/Othr/Id	RestrictedFINX2Max34Text	Identification assigned by an institution
Account Owner Docu- ment/AudtTrlQry/SchCrit/AcctOwnr/Finl nstnId	CLM_BIC11Text	Owner of the account which is being queried
Audit Trail Type Docu- ment/AudtTrlQry/SchCrit/AudtTrlTp	CRTL RSVT TRXN	Specifies which type of audit trails data must be returned.

Table 187 - AuditTrailQuery (camt.097) – usage case Audit Trail for CLM Query

Usage case example: camt.097_CLM_AuditTrailQuery_AuditTrailForCLMQuery_Example.xml

In this example, an Audit Trail Query is instructed by the Account Owner with credit line as target data. It illustrates the mandatory elements in the message.

14.3.25 AuditTrailReport (camt.098)

14.3.25.1 Overview and scope of the message

This chapter illustrates the *AuditTrailReport* message.

The *AuditTrailReport* message is sent by CLM to a CLM Account Holder (or a party authorised by them). It is used to provide audit trail information which has been logged by CLM.

The *AuditTrailReport* message may contain audit trail information relating to: transactions, credit lines or reservations.

Within CLM, the *AuditTrailReport* message has the following usages:

- I Audit Trail for CLM Query (Data or Error response)

The *AuditTrailReport* message is sent in response to an [AuditTrailQuery \(camt.097\)](#) [▶ 468] message, which requested the information.

14.3.25.2 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, credit line, reservation)
- | timestamp
- | approval status
- | processing status

References/links

The CLM-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/CLM/camt.098.001.01_CLM

14.3.25.3 The message in business context

Usage case: Audit Trail for CLM Query (Data response)

In this usage case, the recipient of the message is being informed regarding the audit trail of the transaction, credit line or reservation 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 approval 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 188 - AuditTrailReport (camt.098) – usage case Audit Trail for CLM Query (Data response)

Usage case example: camt.098_CLM_AuditTrailReport_AuditTrailForCLMQuery_Example.xml

In this example an Audit Trail Report containing a reference to an incoming message with the ID “MSGIDcamt.097”, and the available audit trail related to reservation changes is sent to the requesting party.

Usage case: Audit Trail for CLM 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 CLM Query ([AuditTrailQuery \(camt.097\)](#) [▶ 468]).

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
	TBC	

Table 189 - AuditTrailReport (camt.098) – usage case Audit Trail for CLM Query (Data response)

Usage case example: camt.098_CLM_AuditTrailReport_AuditTrailForCLMQueryError_Example.xml

In this usage case, a problem is encountered while retrieving the audit trail, the error information is reported instead of the audit trail information.

14.3.26 DirectDebitMandateQuery (camt.099)

14.3.26.1 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\(camt.100\)](#) [▶ 476] containing the requested information is returned.

14.3.26.2 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>

14.3.26.3 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/I d/AnyBIC	AnyBICIdentifier	Creditor
Responsible party Docu- ment/DrctDbtMndtQry/SchCrit/Cdtr/Rsp nsblPtyId/Id/AnyBIC	AnyBICIdentifier	NCB
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 190 - DirectDebitMandateQuery (camt.099) – usage case Direct debit mandate query

Usage case example: DirectDebitMandateQuery_example.xml

14.3.27 DirectDebitMandateReport(camt.100)

14.3.27.1 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 \(camt.099\)](#) [▶ 474] message.

14.3.27.2 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>

14.3.27.3 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-	Max34Text	Account

Message item	Data type/code	Utilisation
tRpt/MndtOrErr/DrctDbtMndt/CshAcct/Othr/Id		
Amount type Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDbtMnd- tRpt/MndtOrErr/DrctDbtMndt/MaxAmt/Tp/Cd	ExternalMaximumAmountType1Code	Amount type
Amount Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDbtMnd- tRpt/MndtOrErr/DrctDbtMndt/MaxAmt/Amt	ActiveCurrencyAndAmount	Amount
Direct debit mandate reference Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDbtMnd- tRpt/MndtOrErr/DrctDbtMndt/DrctDbtMndtRef	Max35Text	Direct debit mandate reference
Valid from Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDbtMnd- tRpt/MndtOrErr/DrctDbtMndt/VldFr/DtTm	ISODatetime	Valid from
Valid to Docu- ment/DrctDbtMndtRpt/RptOrErr/DrctDbtMnd- tRpt/MndtOrErr/DrctDbtMndt/VldTo/DtTm	ISODatetime	Valid from

Table 191 - 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/OprlErr /Err/Prtry	Max35Text	Specific error
Description Docu- ment/DrctDbtMndtRpt/RptOrErr/OprlErr /Desc	Max140Text	Textual description in addition to the reported error

Table 192 - DirectDebitMandateReport (camt.100) – usage case Error

Usage case example: DirectDebitMandateReportrReport_example.xml

14.3.28 ModifyCreditLine (camt.998)

14.3.28.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage ModifyCreditLine* message.

The *CashManagementProprietaryMessage* 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 *ModifyCreditLine* message.

The *ModifyCreditLine* usage is sent from a CB to CLM. It is used to enter or modify the credit line value available to credit institutions within its community.

Within CLM, the *ModifyCreditLine* message has the following usages:

- I TBC – version 2

The *ModifyCreditLine* message is sent as a result of processing by the CB.

14.3.28.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

Related

References a previously received message, from the same sender.

Previous

References a previously sent message, to the same receiver.

ProprietaryData

Type of the proprietary document enclosed and the actual *ModifyCreditLine* message itself.

References/links

The CLM-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/CLM/camt.998.001.03_CLM_ModifyCreditLine

Business rules applicable to the schema

For business rules applicable to *ModifyCreditLine* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.28.3 The message in business context

Usage case: TBC

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
TBC	TBC	TBC

Table 193 - CashManagementProprietaryMessage ModifyCreditLine (camt.998) – usage case TBC

Message

example:

camt.998_CLM_CashManagementProprietaryMessageModifyCreditLine_<TBC>_Example.xml

14.3.29 InsertBalance_RM (camt.998)

14.3.29.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage InsertBalanceRM* message.

The *CashManagementProprietaryMessage* 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 *InsertBalanceRM* message.

The *InsertBalanceRM* message is sent from a CB to the CLM component. It is used to enter a balance amount directly onto an RTGS DCA.

Within CLM, the *A InsertBalanceRM* message has the following usages:

- I TBC – version 2

The *InsertBalanceRM* message is sent as a result of processing by the CB.

14.3.29.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

Related

References a previously received message, from the same sender.

Previous

References a previously sent message, to the same receiver.

ProprietaryData

Type of the proprietary document enclosed and the actual *InsertBalanceRM* message itself.

References/links

The CLM-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/CLM/camt.998.001.03_CLM_InsertBalanceRM

Business rules applicable to the schema

For business rules applicable to *InsertBalanceRM* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.3.29.3 The message in business context

Usage case: TBC

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
TBC	TBC	TBC

Table 194 - CashManagementProprietaryMessage InsertBalanceRM (camt.998) – usage case TBC

Message

example:

camt.998_CLM_CashManagementProprietaryMessageInsertBalanceRM_<TBC>_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 (BAH)* 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 \(COR\) \(pacs.009\)](#) [▶ 491] 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.

Business service (optional)

Specifies the business service agreed between the sender and the receiver under which rules this message is exchanged. To be used when there is a choice of processing services or processing service levels. Example: E&I.

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 cancelling or amending.

References/links

The CLM-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/CLM/head.001.001.01_CLM

Business rules applicable to the schema

14.4.1.3 The message in business context

The BAH contains information to correctly process the message payload. Every message exchanged between the CLM 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_CLM_IncomingMessageWithin CLM _Example.xml
- | In this example the *BusinessApplicationHeader* (BAH) is used for an incoming message within CLM. It is sent from a NCB with parent BIC "NCBPARNTBIC" and party BIC "NCBPARTYBIC" to CLM. The BAH is filled with the corresponding digital signature.
- | **Message example 2:** head.001_CLM_OutgoingMessageBeingSentByCLMAsCopy_Example.xml
- | In this example the *BusinessApplicationHeader* is used for an outgoing message being sent by CLM 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_CLM_BankToCustomerStatementSentByCLM_Example.xml
- | In this example the *BusinessApplicationHeader* is used for a bank to customer statement sent by CLM to the account owner "NCBBICEUXXX". The BAH includes the 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 CLM component to receive several business messages within one file to the CLM component.

Under a single *BusinessFileHeader*, every message within the file has to be an ISO 20022 message together with its BAH (business message). A file can contain one or several business messages.

Within CLM, the *BusinessFileHeader* information is used for:

- | consistency and completeness checks

This usage is described below, in the chapter “The message in business context”.

In response to an incoming file which fails validation, the CLM component sends a [ReceiptAcknowledgement \(admi.007\)](#) [▶ 347] 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 document type 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 BAH.

References/links

The CLM-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/CLM/head.002.001.01_CLM

Business rules applicable to the schema

For business rules applicable to *BusinessFileHeader* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.4.2.3 The message in business context

Usage case: Consistency and completeness checks

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Table 195 - BusinessFileHeader (head.002) – usage case Consistency and completeness checks

Message example: head.002_CLM_IncomingMessageFileWithinCLM_Example.xml

In this example the *BusinessFileHeader* is used for an incoming file within CLM. The file payload contains a [GetAccount \(camt.003\)](#) [▶ 350] message. The file envelope includes a system user (SystemUserX1) and the corresponding digital signature.

Message example: head.002_CLM_OutgoingFileSentfromCLM_Example.xml

In this example the *BusinessFileHeader* is used for an outgoing file. The file includes a [PaymentStatusReport \(pacs.002\)](#) [▶ 487] message. The file envelope includes a system user (SystemUserX1) and 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 *FIToFIPaymentStatusReport* message.

The *FIToFIPaymentStatusReport* message is sent by the CLM component to a CLM Account Holder (or a party authorised by them). It is used to inform this party about the status of a previous payment order.

The *FIToFIPaymentStatusReport* message is treated as mandatory for all processing failure situations. To receive a *FIToFIPaymentStatusReport* message for normal successful processing situations, subscription is required.

Within CLM, the *FIToFIPaymentStatusReport* message has the following usages:

- | Payment Order Settlement Notification
- | Payment Order Rejection Notification
- | Payment Processing Notification related to:
 - Settlement of Connected Payments
 - Modification of Credit Line
 - Settlement of Requested Marginal Lending
 - Payment Rejection (EoD)
 - Settlement of Automated Marginal Lending

The *FIToFIPaymentStatusReport* message is sent in response to a previously sent payment order message ([FinancialInstitutionCreditTransfer \(COR\) \(pacs.009\)](#) [▶ 491] or [FinancialInstitutionDirectDebit \(pacs.010\)](#) [▶ 496]).

14.5.1.2 Schema

Outline of the schema.

The *FIToFIPaymentStatusReport* 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 CLM-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/CLM/pacs.002.001.09_CLM

Business rules applicable to the schema

No business rules are applicable to a *FIToFIPaymentStatusReport* message.

14.5.1.3 The message in business context

Usage case: Payment Order Settlement Notification

In this usage case, the recipient of the message is being informed that a previous payment order sent by them (or on their behalf) has been actioned successfully (i.e. payment order has been settled).

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

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 196 - PaymentStatusReport (pacs.002) – usage case Payment Order Settlement Notification

Usage case example pacs.002_CLM_FIToFIPaymentStatusReportSuccessful_Example.xml

In this example a PaymentStatusReport “Payment Order Settlement Notification” resulting from a Payment Order containing the CLM booking reference is sent to the corresponding party.

Usage case: Payment Order Rejection Notification

In this usage case, the recipient of the message is being informed that a previous payment order 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/Or gnlGrpInf/OrgnlMsgId	Max35Text	Message ID of original instruction
OriginalMessageNameIdentification Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/Or gnlGrpInf/OrgnlMsgNmId	RTGS_XMLMessageNamePattern	Message name of the original instruc- tion
OriginalInstructionIdentification	Max35Text	Identification of the original instruction

Message item	Data type/code	Utilisation
Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/OrgnlInstrId		
TransactionStatus Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/TxSts	RTGS_TransactionStatusCode	RJCT
Proprietary Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/StatusRsnInf/Rsn/Prtry	Max35Text	Status reason
AdditionalInformation Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/StatusRsnInf/AddtlInf	Max105Text	Detailed error description

Table 197 - PaymentStatusReport (pacs.002) – usage case Payment Order Rejection Notification

Usage case example: pacs.002_CLM_FIToFIPaymentStatusReportRejection_Example.xml

In this example a PaymentStatusReport “Payment Order Rejection Notification” containing a reference to an incoming message with the ID “INSTRIDpacs.009”, the error code “2862” and the description “Request out of cut-off time” is sent to the corresponding party.

Usage case: Payment Processing Notification

In this usage case, the recipient is informed about the status of payments moving through automated parts of the CLM component, which affect the balances of CLM MCAs within the scope of the recipient.

Specific message content

Message item	Data type/code	Utilisation
OriginalMessageIdentification Docu- ment/FIToFIPmtStsRpt/TxInfAndSts/OrgnlGrpInf/OrgnlMsgId	Max35Text	Message ID of original instruction
OriginalMessageNameIdentification Docu-	RTGS_XMLMessageNamePattern	Message name of the original instruction

Message item	Data type/code	Utilisation
Document/FIToFIPmtStsRpt/TxInfAndSts/OriginalGrpInf/OrgnlMsgNmId		
OriginalTransactionIdentification Document/FIToFIPmtStsRpt/TxInfAndSts/OriginalTxId	Max35Text	Transaction ID of the original instruction
TransactionStatus Document/FIToFIPmtStsRpt/TxInfAndSts/TxSts	RTGS_TransactionStatusCode	ACSC
ClearingSystemReference Document/FIToFIPmtStsRpt/TxInfAndSts/ClrSysRef	Max105Text	RTGS booking reference

Table 198 - PaymentStatusReport (pacs.002) – usage case Payment Processing Notification

Usage case example:
pacs.002_CLMS_PaymentStatusreport_PaymentProcessingNotification_Example.xml

14.5.2 FinancialInstitutionCreditTransfer (COR) (pacs.009)

14.5.2.1 Overview and scope of the message

This chapter illustrates the *FinancialInstitutionCreditTransfer* message.

This message type can be used for different CLM component activities:

- | liquidity transfers
- | CBOs

High value payments can be sent by a

- | CLM Account Holder
- | CBs as a participant or on behalf of a CLM Account Holder (mandated payments)

Credited and debited CLM accounts must be denominated in the same currency.

Within CLM, the *FinancialInstitutionCreditTransfer* message has the following usages:

- | Payment Order Message

In response to the *FinancialInstitutionCreditTransfer* message, a [PaymentStatusReport \(pacs.002\)](#) [▶ 487] message containing the status of the payment order is returned to the sending party.

In addition, if the payment order is successfully processed, the *FinancialInstitutionCreditTransfer* message is forwarded to the credited CLM Account Holder (or a party authorised by them).

14.5.2.2 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

This building block is mandatory and non-repetitive. It is a set of elements providing information specific to the individual credit transfer(s). It contains the following elements:

- | payment identification
- | payment type
- | interbank settlement amount
- | interbank settlement date
- | settlement priority
- | settlement time request
- | instructing and instructed agents
- | debtor and creditor parties

References/links

The CLM 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/CLM/pacs.009.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *FinancialInstitutionCreditTransfer* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.5.2.3 The message in business context

Usage case: Payment Order Message

In this usage case, the message provides the details required for the CLM component to execute 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
MessageIdentification Document/FICdtTrf/GrpHdr/MsgId	Max35Text	Identification of the message
Creation Date Time Document/FICdtTrf/GrpHdr/CreDtTm	ISODateTime	Date and time at which the message was created.
NumberOfTransactions Document/FICdtTrf/GrpHdr/NbOfTx	Max15NumericText	Always equals to "1"
SettlementMethod Docu- ment/FICdtTrf/GrpHdr/StlmInf/StlmMtd	SettlementMethod1Code	CLRG
InstructionIdentification Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/InstrId	Max35Text	Identify unambiguously the instruction
EndtoEndIdentification Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/EndToEndId	Max35Text	Identification provided by the initiating party and passed unchanged throughout the entire end-to-end chain
TransactionIdentification Docu- ment/FICdtTrf/CdtTrfTxInf/PmtId/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.
LocalInstrument Docu- ment/FICdtTrf/CdtTrfTxInf/PmtTpInf/Sv	ExternalLocalInstrument1Codes	Allowed codes are MANP, BACP, SBTI, ASTI.

Message item	Data type/code	Utilisation
cLvl/Cd		
LocalInstrument Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/LclInstrm/Prtry	Max35Text	Allowed code words : for connected payment : CON-PAY/AmountValue and ESCBSTAT 2I: used for setting up or reimbursement of repo operations with the CB for intraday credit. Other codewords to be used in case of Eurosystem's InterCB transactions (where the ordering institution and the beneficiary institution are CBs).
InterbankSettlementAmount Document/FICdtTrf/CdtTrfTxInf/IntrBkSttlmAmt	CLM_Max14_Max2DecimalAmount	Amount and currency to be settled
InterbankSettlementDate Document/FICdtTrf/CdtTrfTxInf/IntrBkSttlmDt	ISODate	Date of the settlement
SettlementPriority Document/FICdtTrf/CdtTrfTxInf/SttlmPrty	Priority3Code	URGT
SettlementTimeRequest	ISOTime	The user can indicate the time by when the settlement should take place (TillTime, TimeFrom, RejectTime)
InstructingAgent Document/FICdtTrf/CdtTrfTxInf/InstgAgt/FinlnstnId/BICFI	BICFIIdentifier	Identification of the instructing agent

Message item	Data type/code	Utilisation
InstructedAgent Docu- ment/FICdtTrf/CdtTrfTxInf/InstdAgt/FinInstnId/BICFI	BICFIIdentifier	Identification of the instructed agent
Debtor Docu- ment/FICdtTrf/CdtTrfTxInf/Dbtr/FinInstnId/BICFI	BICFIIdentifier	Financial institution that owes an amount of money to the financial institutional creditor.
Creditor Docu- ment/FICdtTrf/CdtTrfTxInf/Cdtr/FinInstnId/BICFI	BICFIIdentifier	Financial institution that receives an amount of money from the financial institutional debtor.

Table 199 - FinancialInstitutionCreditTransfer (pacs.009) – usage case Payment Order Message

Usage case example 1: pacs.009_CLM_FinancialInstitutionCreditTransfer_PaymentOrderMessage COR_Example1.xml

In this example, a COR payment is instructed by the debtor to the creditor. It illustrates the mandatory elements in the message.

Usage case example 2: pacs.009_CLM_FinancialInstitutionCreditTransfer_PaymentOrderMessage COR_Example2.xml

In this example, a COR payment is instructed by the debtor to the creditor. The code in local instrument is used to indicate it is a mandate payment (MANP)

Usage case example 3: pacs.009_CLM_FinancialInstitutionCreditTransfer_PaymentOrderMessage COR_Example3.xml

In this example, a COR payment is instructed by the debtor to the creditor. The local instrument/proprietary is used to indicate it is a connected payment (CONPAY) followed by the amount value (ISO amount format) preceded by a “/”.

Usage case example 4: pacs.009_CLM_FinancialInstitutionCreditTransfer_PaymentOrderMessage COR_Example4.xml

In this example, a COR payment is instructed by the debtor to the creditor. The local instrument/proprietary is used for setting up or reimbursement of repo operations with the CB for intraday credit (ESCBSTAT 2I).

14.5.3 FinancialInstitutionDirectDebit (pacs.010)

14.5.3.1 Overview and scope of the message

This chapter illustrates the *FinancialInstitutionDirectDebit* message.

The *FinancialInstitutionDirectDebit* message is sent by a CLM Account Holder (or on their behalf by an authorised party) to the CLM component. It is used to move an amount from the CLM MCA of another CLM Account Holder, to a CLM MCA of the sending CLM Account Holder.

The *FinancialInstitutionDirectDebit* message concerns only one direct debit movement.

Within CLM, the *FinancialInstitutionDirectDebit* message has the following usages:

- l Payment Order Message

In response to the *FinancialInstitutionDirectDebit* message, a [PaymentStatusReport \(pacs.002\)](#) [▶ 487] message containing the status of the movement is returned to the sending party.

In addition, if the movement is successfully processed, the *FinancialInstitutionDirectDebit* message is forwarded to the debited CLM Account Holder (or a party authorised by them).

14.5.3.2 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:

- l credit identification
- l instructing and instructed agents
- l creditor party
- l debit information: payment identification, payment type, interbank settlement amount, interbank settlement date, settlement priority, settlement time request, debtor party, remittance information

References/links

The CLM 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/CLM/pacs.010.001.02_CLM

Business rules applicable to the schema

For business rules applicable to *FinancialInstitutionDirectDebit* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

14.5.3.3 The message in business context

Usage case: Payment Order Message

In this usage case, the message provides the details required for the CLM 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 order
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	BICFIIdentifier	BIC of the instructing agent

Message item	Data type/code	Utilisation
Docu- ment/FIDrctDbt/CdtInstr/InstgAgt/FinIns tnId/BICFI		
Instructed Agent Docu- ment/FIDrctDbt/CdtInstr/InstdAgt/FinIns tnId/BICFI	BICFIIdentifier	BIC of the instructed agent
Creditor Docu- ment/FIDrctDbt/CdtInstr/Cdtr/FinInstnId /BICFI	BICFIIdentifier	Financial institution servicing an ac- count 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.

Message item	Data type/code	Utilisation
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/FinInstnld /BICFI	BICFIIdentifier	Financial institution that owes an amount of money to the (ultimate) financial institutional creditor.

Table 200 - FinancialInstitutionDirectDebit (pacs.010) – usage case Payment Order Message

Usage case example:
pacs.010_CLM_FinancialInstitutionDirectDebit_PaymentOrderMessage_Example.xml

14.6 Reference data (reda)

14.6.1 PartyQuery (reda.015)

14.6.1.1 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 \(reda.017\)](#) [▶ 501] containing the requested information is returned.

14.6.1.2 Schema

Outline of the schema

The *PartyQuery* 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 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>

14.6.1.3 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

Message item	Data type/code	Utilisation
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 201 - 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.

14.6.2 PartyReport (reda.017)

14.6.2.1 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 \(reda.015\)](#) [▶ 499] message.

14.6.2.2 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>

14.6.2.3 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-	ISODate	Opening date for the party

Message item	Data type/code	Utilisation
ment/PtyRpt/RptOrErr/PtyRpt/PtyOrErr/ SysPty/OpngDt		
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
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 202 - 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 203 - 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.

14.6.3 CashAccountAuditTrailQuery (reda.039)

14.6.3.1 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 \(reda.040\)](#) [▶ 506] containing the requested information is returned.

14.6.3.2 Schema

Outline of the schema

The *CashAccountAuditTrailQuery* 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 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>

14.6.3.3 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 204 - 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.

14.6.4 CashAccountAuditTrailReport (reda.040)

14.6.4.1 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 \(reda.039\)](#) [▶ 504] message.

14.6.4.2 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>

14.6.4.3 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/CshAcctAudtTrIRpt/Aud tTr/FldNm	RestrictedFINXMax35Text	Field name
Old field value Docu- ment/CshAcctAudtTrIRpt/RptOrErr/Csh Ac- ctAudtTrIRpt/CshAcctAudtTrIRpt/Aud tTr/OdFldVal	RestrictedFINXMax350Text	Old field value
New field value Docu- ment/CshAcctAudtTrIRpt/RptOrErr/Csh Ac- ctAudtTrIRpt/CshAcctAudtTrIRpt/Aud tTr/NewFldVal	RestrictedFINXMax350Text	New field value
Timestamp	ISODateTime	Timestamp

Message item	Data type/code	Utilisation
Docu- ment/CshAcctAudtTrlRpt/RptOrErr/Csh Ac- ctAudtTrlRpt/CshAcctAudtTrlOrErr/Aud tTrl/OprTmStmp		
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/ApprvgUsr	RestrictedFINXMax256Text	Approving user
Account identification Docu- ment/CshAcctAudtTrlRpt/RptOrErr/Csh AcctAudtTrlRpt/CshAcctId/Id/Othr/Id	RestrictedFINX2Max34Text	Account identification

Table 205 - 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 206 - CashAccountAuditTrailReport (reda.040) – usage case Error

Usage case example: CashAccountAuditTrailReport_example.xml

In this example a CB participating with BIC “NCBAXXYAAA” 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 “USERTWOEYES” on 2018-01-03 at 17:59.

14.6.5 PartyAuditTrailQuery (reda.042)

14.6.5.1 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 \(reda.043\)](#) [▶ 510] containing the requested information is returned.

14.6.5.2 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>

14.6.5.3 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 207 - 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 “PAYBXYAAA” for which it is responsible.

14.6.6 PartyAuditTrailReport (reda.043)

14.6.6.1 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 \(reda.042\)](#) [▶ 509] message.

14.6.6.2 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>

14.6.6.3 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/PtyAudtT rlRpt/PtyAudtTrlOrErr/AudtTrl/FldNm	RestrictedFINXMax35Text	Field name
Old field value Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtT rlRpt/PtyAudtTrlOrErr/AudtTrl/OdFldVal	RestrictedFINXMax350Text	Old field value
New field value Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtT rlRpt/PtyAudtTrlOrErr/AudtTrl/NewFldV al	RestrictedFINXMax350Text	New field value
Timestamp Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtT rlRpt/PtyAudtTrlOrErr/AudtTrl/OprTmSt mp	ISODateTime	Timestamp
Instructing user Docu- ment/PtyAudtTrlRpt/RptOrErr/PtyAudtT rlRpt/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 208 - 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 209 - 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 “PAYBXXYYAAAA”.

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”.

14.6.7 CalendarQuery(reda.064)

14.6.7.1 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\(reda.065\)](#) [▶ 515] containing the requested information is returned.

14.6.7.2 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>

14.6.7.3 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 210 - CalendarQuery (reda.064) – usage case Calendar query

Usage case example: CalendarQuery_example.xml

14.6.8 CalendarReport(reda.065)

14.6.8.1 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\(reda.064\)](#) [514] message.

14.6.8.2 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>

14.6.8.3 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/SysI d/MktInfrstrctrId/Prtry	Max35Text	Service
Currency Docu- ment/CalRpt/RptOrErr/CalRpt/Svc/Sys Ccy	ActiveCurrencyCode	Currency of the service for which the calendar is returned.

Table 211 - 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 212 - CalendarReport (reda.065) – usage case Error

Usage case example: CalendarReport_example.xml

15 Specific messages for CBs

15.1 ReturnAccount (camt.004) - specific for CBs

Will be completed in v2.0.

15.1.1 Overview and scope of the message

15.1.2 Schema

15.1.3 The message in business context

15.2 ModifyTransaction (camt.007) - specific for CBs

Will be completed in v2.0.

15.2.1 Overview and scope of the message

15.2.2 Schema

15.2.3 The message in business context

15.3 Receipt (camt.025) - specific for CBs

Will be completed in v2.0.

15.3.1 Overview and scope of the message

15.3.2 Schema

15.3.3 The message in business context

15.4 ResolutionOfInvestigation (camt.029) - specific for CBs

Will be completed in v2.0.

15.4.1 Overview and scope of the message

15.4.2 Schema

15.4.3 The message in business context

15.5 FIToFIPaymentCancellationRequest (camt.056) - specific for CBs

Will be completed in v2.0.

15.5.1 Overview and scope of the message

15.5.2 Schema

15.5.3 The message in business context

15.6 camt.998 - AuthorizePenalty_RM

15.6.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage AuthorisePenaltyRM* message.

The *CashManagementProprietaryMessage* 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 *AuthorisePenaltyRM* message.

The *AuthorisePenaltyRM* message is sent from a CB to the CLM component. It is used to authorise or cancel a penalty incurred by an infringement of minimum reserve. Within CLM, the *AuthorisePenaltyRM* message has the following usages:

TBC – version 2

The *AuthorisePenaltyRM* message is sent as a result of processing by the CB.

15.6.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

Related

References a previously received message, from the same sender.

Previous

References a previously sent message, to the same receiver.

ProprietaryData

Type of the proprietary document enclosed and the actual *AuthorisePenaltyRM* message itself.

References/links

The CLM-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/CLM/camt.998.001.03_CLM_AuthorisePenaltyRM

Business rules applicable to the schema

For business rules applicable to *AuthorisePenaltyRM* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

15.6.3 The message in business context

Usage case: TBC

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
TBC	TBC	TBC

Table 213 - CashManagementProprietary MessageAuthorisePenalty (camt.998) – usage case TBC

Message example: `camt.998_CLM_CashManagementProprietaryMessageAuthorisePenaltyRM`
<TBC>_Example.xml

15.7 camt.998 - GetPenalty_RM

15.7.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage GetPenaltyRM* message.

The *CashManagementProprietaryMessage* 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 *GetPenaltyRM* message.

The *GetPenaltyRM* message is sent from a CB to the CLM component. It is used to request details of penalties incurred by an infringement of minimum reserve.

Within CLM, the *GetPenaltyRM* message has the following usages:

- Query Penalty (CB only)

In response to a *GetPenaltyRM* message, a [camt.998 - ReturnPenalty_RM](#) [▶ 523] message containing the requested information is returned.

15.7.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

ProprietaryData

Type of the proprietary document enclosed and the actual *GetPenaltyRM* message itself.

References/Links

The CLM-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/CLM/camt.998.001.03> CLM GetPenaltyRM

Business rules applicable to the schema

For business rules applicable to *GetPenaltyRM* please refer to the chapter [Index of business rules and error codes](#) [▶ 537].

15.7.3 The message in business context

Usage case: Query Penalty

In this usage case, a CB is requesting details of all penalties for CLM MCAs in its banking community, based upon the query criteria provided.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
/Document/PrtryMsg/MsgHdr/MsgId	Max35Text	Message identification
/Document/PrtryMsg/PrtryData/Tp	Max35Text	Always "GetPenalty"
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/PnltyMgmtId/Tp	BalanceTypeCode_EMIP1	Balance type code

/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData- ta/PnltyMgmtld/CtrPtyId/BIC	BICIdentifier	Counterparty BIC
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/MntncPrd/FrDt	ISODate	Maintenance period from
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/MntncPrd/ToDt	ISODate	Maintenance period to
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/PnltySts	PenaltyStatusCode_EMIP2	Penalty status

Table 214 - CashManagementProprietaryMessage GetPenaltyRM (camt.998) – usage case Query Penalty

Message **example:**
camt.998_CLM_CashManagementProprietaryMessageGetPenaltyRM_QueryPenalty_Example.xml

15.8 camt.998 - ReturnPenalty_RM

15.8.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage ReturnPenaltyRM* message.

The *CashManagementProprietaryMessage* 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 *ReturnPenaltyRM* message.

The *ReturnPenaltyRM* message is sent from the CLM component to a CB. It is used to inform the details of penalties incurred by an infringement of minimum reserve.

Within CLM, the *ReturnPenaltyRM* message has the following usages:

- Query Penalty (CB only) (Data or Error response)

The *ReturnPenaltyRM* is sent in response to a [camt.998 - GetPenalty_RM \[521\]](#) message, which requested the information.

15.8.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

Related

References a previously received message, from the same sender.

ProprietaryData

Type of the proprietary document enclosed and the actual *ReturnPenaltyRM* message itself.

References/Links

The CLM-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/CLM/camt.998.001.03_CLM_ReturnPenaltyRM

Business rules applicable to the schema

No business rules are applicable to a *ReturnPenaltyRM* message.

15.8.3 The message in business context

Usage case: Query Penalty (Data response)

In this usage case, the CB recipient of the message is being informed regarding details of all penalties for CLM MCAs in its banking community, based upon the query criteria provided.

Specific message content

Message item	Data type/code	Utilisation
/Document/PrtryMsg/MsgHdr/MsgId	Max35Text	Message ID
/Document/PrtryMsg/Rltd/Ref	Max35Text	Related reference
/Document/PrtryMsg/PrtryData/Tp	Max35Text	Always "ReturnPenalty"
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryDa- ta/BizRpt/PnltyRpt/PnltyMgmtId/Tp	BalanceTypeCode_EMIP3	Balance type
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryDa-	CountryCode	Country

Message item	Data type/code	Utilisation
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/PnltyMgmtld/Ctry		
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/PnltyMgmtld/CtrPtyld/BIC	BICIdentifier	Counterparty BIC
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/MntncPrd/FrDt	ISODate	Maintenance period from
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/MntncPrd/ToDt	ISODate	Maintenance period to
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/PnltySts	PenaltyStatusCode_EMIP2	Penalty status
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/PnltyTp	PenaltyTypeCode_EMIP1	Penalty type
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/FirstAmt	ActiveCurrencyAndAmount_EMIP5	First amount
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/FirstAmt/Ccy	ActiveCurrencyCode	First amount currency
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/SecondAmt	ActiveCurrencyAndAmount_EMIP5	Second amount
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/PnltyRpt/SecondAmt/Ccy	ActiveCurrencyCode	Second amount currency
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acctld/BIC	BICIdentifier	Account ID BIC
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/Amt	ActiveCurrencyAndAmount_EMIP5	Amount
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/Amt/Ccy	ActiveCurrencyCode	Currency

Message item	Data type/code	Utilisation
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/CdtDbtInd	CreditDebitCode_EMIP1	Debit/credit ind
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/Tp	BalanceTypeCode_EMIP3	Balance type
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/ValDt	ISODateTime	Value date

Table 215 - CashManagementProprietaryMessage ReturnPenaltyRM (camt.998) – usage case Query Penalty (Data response)

Message **example:**
camt.998_CLM_CashManagementProprietaryMessageReturnPenaltyRM_QueryPenaltyData_Example.xml

Usage case: Query Penalty (Error response)

In this usage case, the CB recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent Query Penalty (camt.998).

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
/Document/PrtryMsg/MsgHdr/MsgId	Max35Text	Message ID
/Document/PrtryMsg/Rltd/Ref	Max35Text	Related reference
/Document/PrtryMsg/PrtryData/Tp	Max35Text	Always "ReturnPenalty"
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/OprlErr/Err/Prtry	Max4AlphaNumericText_EMIP2	Error code
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/OprlErr/Dsc	Max140Text	Error description

Table 216 - CashManagementProprietaryMessage ReturnPenaltyRM (camt.998) – usage case Query Penalty (Error response)

Message **example:**
camt.998_CLM_CashManagementProprietaryMessageReturnPenaltyRM_QueryPenaltyError_Example.xml

15.9 camt.998 - InsertCompulsoryReserve

15.9.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage InsertCompulsoryReserve* message.

The *CashManagementProprietaryMessage* 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 *InsertCompulsoryReserve* message.

The *InsertCompulsoryReserve* message is sent from a CB to the CLM component. It is used to enter the minimum reserve values for credit institutions within its community.

Within CLM, the *InsertCompulsoryReserve* message has the following usages:

- TBC – version 2

The *InsertCompulsoryReserve* message is sent as a result of processing by the CB.

15.9.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

Related

References a previously received message, from the same sender.

Previous

References a previously sent message, to the same receiver.

ProprietaryData

Type of the proprietary document enclosed and the actual *InsertCompulsoryReserve* message itself.

References/Links

The CLM-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/CLM/camt.998.001.03_CLM_InsertCompulsoryReserve

Business rules applicable to the schema

For business rules applicable to *InsertCompulsoryReserve* please refer to the chapter [Index of business rules and error codes](#) [537].

15.9.3 The message in business context

Usage case: TBC

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
TBC	TBC	TBC

Table 217 - CashManagementProprietaryMessage InsertCompulsoryReserve – usage case TBC

Message example:
camt.998_CLM_CashManagementProprietaryMessageInsertCompulsoryReserve_<TBC>_Example.xml

15.10 camt.998 - GetCompulsoryReserve_RM

15.10.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage GetCompulsoryReserveRM* message.

The *CashManagementProprietaryMessage* 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 *GetCompulso-*

ryReserveRM message. The *GetCompulsoryReserveRM* message is sent from a CB to the CLM component. It is used to request details of minimum reserve values within its community.

Within CLM, the *GetCompulsoryReserveRM* message has the following usages:

- Query Minimum Reserve of a Banking Community (CB only)

In response to a *GetCompulsoryReserveRM* message, a [camt.998 - ReturnCompulsoryReserve_RM \[530\]](#) message containing the requested information is returned.

15.10.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

ProprietaryData

Type of the proprietary document enclosed and the actual *GetCompulsoryReserveRM* message itself.

References/Links

The CLM-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/CLM/camt.998.001.03_CLM_GetCompulsoryReserveRM

Business rules applicable to the schema

For business rules applicable to *GetCompulsoryReserveRM* please refer to the chapter [Index of business rules and error codes \[537\]](#).

15.10.3 The message in business context

Usage case: Query Minimum Reserve of a Banking Community

In this usage case, a CB is requesting information regarding the minimum reserve currently set for CLM MCAs in its banking community, based upon the query criteria provided.

Specific message requirements

All content must comply with the business rules for the message.

Message item	Data type/code	Utilisation
/Document/PrtryMsg/MsgHdr/MsgId	Max35Text	Message ID
/Document/PrtryMsg/PrtryData/Tp	Max35Text	Always "GetValueOfMinimumReserve"
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/MinRsrvMgmtId/Tp	BalanceTypeCode_EMIP1	Balance type
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/MntncPrd/FrDt	ISODate	Maintenance period from
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/MntncPrd/ToDt	ISODate	Maintenance period to

Table 218 - CashManagementProprietaryMessage GetCompulsoryReserveRM (camt.998) – usage case Query Minimum Reserve of a Banking Community

Message **example:**
camt.998_CLM_CashManagementProprietaryMessageGetCompulsoryReserveRM_QueryMinimumReserveOfABankingCommunity_Example.xml

15.11 camt.998 - ReturnCompulsoryReserve_RM

15.11.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage ReturnCompulsoryReserveRM* message.

The *CashManagementProprietaryMessage* 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 *ReturnCompulsoryReserveRM* message.

The *ReturnCompulsoryReserveRM* message is sent from the CLM component to a CB. It is used to inform the details of minimum reserve values in the CBs community.

Within CLM, the *ReturnCompulsoryReserveRM* message has the following usages:

- Query Minimum Reserve of a Banking Community (CB only) (Data or Error response)

The *ReturnCompulsoryReserveRM* message is sent in response to a [camt.998 - GetCompulsoryReserve_RM](#) [528] message, which requested the information.

15.11.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

Related

References a previously received message, from the same sender.

ProprietaryData

Type of the proprietary document enclosed and the actual *ReturnCompulsoryReserveRM* message itself.

References/Links

The CLM-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/CLM/camt.998.001.03_CLM_ReturnCompulsoryReserveRM

Business rules applicable to the schema

No business rules are applicable to a *ReturnCompulsoryReserveRM* message.

15.11.3 The message in business context

Usage case: Query Minimum Reserve of a Banking Community (Data response)

In this usage case, the CB recipient of the message is being informed regarding the minimum reserve currently set for CLM MCAs in its banking community, based upon the query criteria provided.

Specific message content

Message item	Data type/code	Utilisation
/Document/PrtryMsg/MsgHdr/MsgId	Max35Text	Message ID
/Document/PrtryMsg/Rltd/Ref	Max35text	Related ref
/Document/PrtryMsg/PrtryData/Tp	Max35text	Always "ReturnMinimumReserve"

Message item	Data type/code	Utilisation
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/AcctId/BIC	BICIdentifier	Account ID BIC
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/Amt	ActiveCurrencyAndAmount_EMIP7	Amount
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/Amt/Ccy	ActiveCurrencyCode	Currency
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/CdtDbtInd	CreditDebitCode_EMIP1	Debit/credit ind
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/Tp	BalanceTypeCode_EMIP1	Balance type
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/MntncPrd/FrDt	ISODate	Maintenance period from
/Document/PrtryMsg/PrtryData/Data/EMIPPrtryData/BizRpt/AcctRpt/Acct/MulBal/MntncPrd/ToDt	ISODate	Maintenance period to

Table 219 - CashManagementProprietaryMessage ReturnCompulsoryReserveRM (camt.998) – usage case Query Minimum Reserve of a Banking Community (Data response)

Message **example:**
camt.998_CLM_CashManagementProprietaryMessageReturnCompulsoryReserve_QueryMinimumReserveOfABankingCommunityData_Example.xml

Usage case: Query Minimum Reserve of a Banking Community (Error response)

In this usage case, the CB recipient of the message is being informed of an operational error which occurred while attempting to fulfil a previously sent Query Minimum Reserve of a Banking Community (camt.998).

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
/Document/PrtryMsg/MsgHdr/MsgId	Max35Text	Message ID
/Document/PrtryMsg/Rltd/Ref	Max35text	Related ref
/Document/PrtryMsg/PrtryData/Tp	Max35text	Always "ReturnMinimumReserve"
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/OprlErr/Err/Prtry	Max4AlphaNumericText_EMIP2	Error code
/Document/PrtryMsg/PrtryData/Data/E MIPPrtryData/OprlErr/Desc	Max140Text	Error description

Table 220 - CashManagementProprietaryMessage ReturnCompulsoryReserveRM (camt.998) – usage case Query Minimum Reserve of a Banking Community (Error response)

Message **example:**
camt.998_CLM_CashManagementProprietaryMessageReturnCompulsoryReserve_QueryMinimumReserveOfABankingCommunityError_Example.xml

15.12 camt.998 - SendPeriodicFlow_RM

15.12.1 Overview and scope of the message

This chapter illustrates the *CashManagementProprietaryMessage SendPeriodicFlowRM* message.

The *CashManagementProprietaryMessage* 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 *SendPeriodicFlowRM* message.

The *SendPeriodicFlowRM* message is sent from the CLM component to a CB. It is used to inform the details of interests to be paid for compulsory reserve and penalties for infringements.

Within CLM, the *SendPeriodicFlowRM* message has the following usages:

- Notify CBs on Minimum Reserve Fulfilment

The *SendPeriodicFlowRM* is sent as a result of configured processing by the CLM component.

15.12.2 Schema

Outline of the schema.

The *CashManagementProprietaryMessage* message is composed of the following message building blocks:

MessageHeader

Uniquely identifies the message. The message identification must be unique amongst all messages of the same name sent by the same party.

Type of the proprietary document enclosed and the actual *SendPeriodicFlowRM* message itself.

References/Links

The CLM-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/CLM/camt.998.001.03_CLM_SendPeriodicFlowRM

Business rules applicable to the schema

No business rules are applicable to a *ReturnSendPeriodicFlowRM* message.

15.12.3 The message in business context

Usage case: Notify CBs on Minimum Reserve Fulfilment

In this usage case, the CLM component is (periodically) informing a CB regarding:

- | the correct observance of minimum reserve by its banking community,
- | the amounts of interest becoming due as a result of minimum reserve activity,
- | a notification of possible penalties owing to breaches of minimum reserve.

Specific message content

Message item	Data type/code	Utilisation
/Document/PrtryMsg/MsgId/Ref	Max35text	Message ID
/Document/PrtryMsg/PrtryData/Tp	Max35text	Always "ReturnPeriodicInformation"
/Document/PrtryMsg/PrtryData/PrtryData/Ctry	CountryCode	Country
/Document/PrtryMsg/PrtryData/PrtryData	ISODate	Maintenance period from

Message item	Data type/code	Utilisation
ta/MntncPrd/FrDt		
/Document/PrtryMsg/PrtryData/PrtryData/MntncPrd/ToDt	ISODate	Maintenance period to
/Document/PrtryMsg/PrtryData/PrtryData/RptValDt	ISODate	Report value date
/Document/PrtryMsg/PrtryData/PrtryData/NbOfItms	Number	Number of items
/Document/PrtryMsg/PrtryData/PrtryData/TotNbOfItms	Number	Total number of items
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/CtrPtyId/BIC	BICIdentifier	Counterparty BIC
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/AcctRpt/Acct/MulBal/Amt	ActiveCurrencyAndAmount_EMIP6	Amount
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/AcctRpt/Acct/MulBal/Amt/Ccy	ActiveCurrencyCode	Currency
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/AcctRpt/Acct/MulBal/CdtDbtInd	CreditDebitCode_EMIP1	Debit/credit ind
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/AcctRpt/Acct/MulBal/Tp	BalanceTypeCode_EMIP3	Balance type
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/IntrstRpt/Amt	ActiveCurrencyAndAmount_EMIP6	Interest amount
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/IntrstRpt/Amt/Ccy	ActiveCurrencyCode	Interest currency
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/PnltyRpt/FirstAmt	ActiveCurrencyAndAmount_EMIP6	Penalty first amount
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/PnltyRpt/FirstAmt/Ccy	ActiveCurrencyCode	Penalty first amount currency
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/PnltyRpt/SecondAmt	ActiveCurrencyAndAmount_EMIP	Penalty second amount
/Document/PrtryMsg/PrtryData/PrtryData/FlowRpt/PnltyRpt/SecondAmt/Ccy	ActiveCurrencyCode	Penalty second amount currency

Table 221 - CashManagementProprietaryMessage SendPeriodicFlowRM (camt.998) – usage case Notify CBs on Minimum Reserve Fulfilment

Message	example:
camt.998_CLM_CashManagementProprietaryMessageSendPeriodicFlowRM_NotifyCBsOnMinimumReserveFulfilment	_Example.xml

Part IV - Appendixes

16 Index and digital signature

16.1 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

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	
						sending party.
	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 BizMsgldr and the same Business sender "From" at the same day will be detected as duplicate.	head.001	admi.007			A message with the same BizMsgldr 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
	Debtor account is blocked	pacs.009	pacs.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	
	Creditor account is blocked	pacs.009	pacs.002			Creditor 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
	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 //Dynamic 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.009	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:	pacs.009	pacs.002			The instruction was detected as a duplicate

Rule ID	Description	Inbound Message	Outbound Message	Code field	Reason Code	Error Text
Will be filled in the next iterations	Non-exhaustive					Non-exhaustive
	- instructing agent - message type - instructed agent - transaction ID - EndtoEnd ID - Settlementdate - Settlement amount					
	The business sender must be a Central Bank which is responsible for the CLM Account Holder	pacs.009	pacs.002			The business sender must be a Central Bank which is responsible for the CLM Account Holder
	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 or a CLM holiday the payment will be rejected	pacs.009	pacs.002			If the execution date is on weekend or a CLM 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.	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 T2S or TIPS are also not possible.
	The cut-off time for CLM has to be observed	pacs.009	pacs.002			The cut-off time for bank to bank payments was

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	
						not observed
	Instructed agent must be a BIC of a CLM Account Holder	pacs.009	pacs.002			Instructed agent is not a BIC of a CLM Account Holder
	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.
	SettlementTimeRequest: Local time (hh:mm) must be before the cut-off time for bank to bank payments.	pacs.009	pacs.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.	pacs.009	admi.007			If Till time and Reject Time are both mentioned so the message is be rejected.
	Payments with code ASTI or SBTI are not allowed in CLM	pacs.009	admi.007			Payments with code ASTI or SBTI are not allowed in CLM
	In case of connected payments the account to be debited must be a MCA with a credit line with responsible CB = instructing agent CB.	pacs.009	pacs.002			The connected payment is not allowed on this account
	If code word /CONPAY/ is used in Local Instrument/Proprietary, the code word has to be followed by an amount with maximum 15 characters including comma and 0 to 2 decimal places (not valid for internal	pacs.009	admi.007			If code word /CONPAY/ is used in Local Instrument/Proprietary, the code word has to be followed by an amount with maximum 15 char-

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 from SF). The comma is mandatory.					acters including comma and 0 to 2 decimal places. The comma is mandatory.
	Debtor account is blocked	pacs.010	pacs.002			Debtor account is blocked
	Creditor account is blocked	pacs.010	pacs.002			Creditor account is blocked
	The technical sender in combination to the business sender must be allowed to credit the account of the instructing agent.	pacs.010	pacs.002			The technical sender in 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

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 business sender must be a Central Bank	pacs.010	pacs.002			The business sender is not a Central Bank
	Payments can be sent for the current business day and up to 10 calendar days in advance.	pacs.010	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 or a CLM holiday the payment will be rejected	pacs.010	pacs.002			If the execution date is on weekend or a CLM 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 are 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 T2S or TIPS are not possible anyway.
	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
	Account to be credited must be an account of the responsible central bank or of a central bank with a mandate to debit the account	pacs.010	pacs.002			Account to be credited is neither an account of the responsible central bank nor of a central bank with a mandate to debit 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	
	Account to be debited must be a MCA of an account holder.	pacs.010	pacs.002			Account to be debited is not a MCA of an account holder.
	Currency must denominate the same currency as accounts indicated for posting.	pacs.010	pacs.002			Currency does not denominate same currency as accounts indicated for posting.
	SettlementTimeRequest: Local time (hh:mm) must be before the CLM cut-off time for payments	pacs.010	pacs.002			SettlementTimeRequest: Local time (hh:mm) is after the CLM cut-off time for payments
	If Till time and Reject Time are both mentioned the message will be rejected.	pacs.010	admi.007			If Till time and Reject Time are both mentioned so the message is be rejected.
	Direct Debits as connected payments are only allowed if sent by a CB.	pacs.010	pacs.002			Direct Debits as connected payments are only allowed if sent by a CB.
	In case of a direct debit as connected payments the account to be credited must be a MCA with a credit line with responsible CB = instructing agent CB.	pacs.010	pacs.002			In case of a direct debit as connected payments the account to be credited must be a MCA with a credit line with responsible CB = instructing agent CB.
	If code word /CONPAY/ is used in Local Instrument/Proprietary, the code word has to be followed by an	pacs.010	admi.007			If code word /CONPAY/ is used in Local Instrument/Proprietary, the

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	
	amount with maximum 15 characters including comma and 0 to 2 decimal places (not valid for internal payments from SF). The comma is mandatory.					code word has to be followed by an amount with maximum 15 characters including comma and 0 to 2 decimal places. The comma is mandatory.
	Debtor account is blocked	pacs.004	pacs.002			Debtor account is blocked
	Creditor account is blocked	pacs.004	pacs.002			Creditor account is blocked
	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.
	A transaction may only have one status value for any one DateTimeStamp.	camt.005	admi.007			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	admi.007			At least QueryType or TransactionCriteria must be present. Both can be present together.
	If QueryType is used without TransactionCriteria, the query refers to the last similar query GetTransaction.	camt.005	admi.007			If QueryType is used without TransactionCriteria, the query refers to the last similar query GetTransaction.
	At least QueryType or TransactionCriteria must be present. Both can be present together.	camt.005	admi.007			At least QueryType or TransactionCriteria must be present. Both can be

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	
						present together.
	The QueryName is provided by the system in the ReturnTransaction.	camt.005	---			The QueryName is provided by the system in the ReturnTransaction.
	If NewCriteria is used, at least SearchCriteria or ReturnCriteria must be present. Both can be present.	camt.005	admi.007			If NewCriteria is used, at least SearchCriteria or ReturnCriteria must be present. Both can be present.
	If ChargesInformation is present, then ReturnedInstructedAmount must be present.	camt.005	admi.007			If ChargesInformation is present, then ReturnedInstructedAmount must be present.
	If PaymentIdentification/LongBusinessIdentification is present, then InstructingAgent's BIC with 11 characters is mandatory.	camt.005	admi.007			If PaymentIdentification/LongBusinessIdentification is present, then InstructingAgent's BIC with 11 characters is mandatory.
	If PaymentIdentification/LongBusinessIdentification is present, then InstructedAgent's BIC with 11 characters is mandatory.	camt.005	admi.007			If PaymentIdentification/LongBusinessIdentification is present, then InstructedAgent's BIC with 11 characters is mandatory.
	IF PaymentInstructionSatusDateTime is present Then PendingStatus and PendingAndFinalStatus are not	camt.005	admi.007			IF PaymentInstructionSatusDateTime is present Then Pending

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.					ingStatus and PendingAndFinalStatus are not allowed.
	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.
	PaymentInstructionStatusDateTime can only be used maximum once per request. Thus if the element InstructionStatus is repeated, PaymentInstructionStatusDateTime can only apply to a single status.	camt.005	admi.007			PaymentInstructionStatusDateTime can only be used maximum once per request. Thus if the element InstructionStatus is repeated, PaymentInstructionStatusDateTime can only apply to a single status.
	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.
	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

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 EntryDate is mandatory.
	If PaymentToReturnCriteria is not provided, MemberIdentificationIndicator default value is "true"	camt.005	---			No Error
	If PaymentFromReturnCriteria is not provided, MemberIdentificationIndicator default value is "true"	camt.005	---			No Error
	If ChargesInformation is present, then ReturnedInstructedAmount must be present.	camt.005	admi.007			If ChargesInformation is present, then ReturnedInstructedAmount must be present.
	If PaymentInstructionStatusIndicator is not provided, default value is "true"	camt.005	---			No Error
	If PaymentInstructionStatusDateIndicator is not provided, default value is "true"	camt.005	---			No Error
	If PaymentInstructionStatusReasonIndicator is not provided, default value is "true"	camt.005	---			No Error
	If InstructedAmountIndicator is not provided, default value is "true"	camt.005	---			No Error
	If CreditDebitIndicator is not provided	camt.005	---			No Error

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
	ed, default value is "false"					
	If InterbankSettlementAmountIndicator is not provided, default value is "true"	camt.005	---			No Error
	If ProcessingValidityTimeIndicator is not provided, default value is "true"	camt.005	---			No Error
	If InstructionCopyIndicator is not provided, default value is "true"	camt.005	---			No Error
	If PaymentTypeIndicator is not provided, default value is "true"	camt.005	---			No Error
	If TransactionIdentificationIndicator is not provided, default value is "true"	camt.005	---			No Error
	If InterbankSettlementDateIndicator is not provided, default value is "true"	camt.005	---			No Error
	If E2EIdentificationIndicator is not provided, default value is "true"	camt.005	---			No Error
	If PaymentMethodIndicator is not provided, default value is "true"	camt.005	---			No Error
	If DebtorIndicator is not provided, default value is "true"	camt.005	---			No Error
	If DebtorAgentIndicator is not provided, default value is "true"	camt.005	---			No Error

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 IntermediaryIndicator is not provided, default value is "true"	camt.005	---			No Error
	If CreditorAgentIndicator is not provided, default value is "true"	camt.005	---			No Error
	If CreditorIndicator is not provided, default value is "true"	camt.005	---			No Error
	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/TransactionInformationAndStatus/CancellationStatusReasonInformation/Reason] must be present	camt.029	admi.007			Element for reason is missing
	Used only for Connected Payments. This element has to be used in combination with TransactionDetails/LocalInstrument and if CONPAY is used this field is mandatory, otherwise it is not allowed.	camt.054	admi.007			Used only for Connected Payments. This element has to be used in combination with TransactionDetails/LocalInstrument and if CONPAY is used this field is mandatory, otherwise it is 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	
	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
	In case the MCA account holder is not eligible, CLM rejects the liquidity transfer and sends a negative receipt (camt.025) to the account holder	camt.050	camt.025			The MCA account holder is not eligible.
	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
	The two cash accounts in the different types of liquidity credit transfers have to belong to the same participant or be within the same liquidity transfer group	camt.050	camt.025			The two cash accounts in the different types of liquidity credit transfers have to belong to the same participant or be within the same liquidity transfer group
	In CLM Creditor account must be a MCA, a TIPS dedicated cash account, a T2S dedicated cash ac-	camt.050	camt.025			The creditor account is not valid

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	
	count, or a RTGS dedicated cash account.					
	Currency must denominate the same currency as accounts indicated for posting.	camt.050	camt.025			Currency does not denominate the same currency as accounts indicated for posting.
	The debtor account must be valid	camt.050	camt.025			The debtor account is not valid
	Instruction ID: In inbound messages, only the value "NOT PROVIDED" is allowed	camt.050	camt.025			Instruction ID: In inbound messages, only the value "NOT PROVIDED" is allowed
	For the setup of a marginal lending on request a party needs to: - A CLM participant - Have a MCA in CLM Be eligible to the marginal lending facility - Have a marginal lending account in CLM	RDM: Setup of a marginal lending on request	admi.007			The party has not the right to setup a marginal lending on request

Table 222 - CLM 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	Create Cash Account

description	user function
linked to Cash Accounts that fall under their responsibility.	
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 Dedicated Cash Account, T2S Dedicated Transit Account or T2S Central Bank 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
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	Create Cash Account

description	user function
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.	
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 Dedicated Cash Account or a T2S central bank 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 Dedicated Cash Account, Ancillary System Guarantee funds account, RTGS sub account, RTGS Central bank 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 performed on such a conventional value in case of four eyes second step or processing of retrieved queued requests.	Create Cash Account
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	Create Liquidity Transfer Or-

description	user function
Payment Bank responsible for the account to be debited.	der
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
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.	Create Liquidity Transfer Order
When performing a Liquidity Transfer Order Create request, the number of decimals in	Create Liquidity Transfer Or-

description	user function
the value provided for Amount must be compliant with the number of decimals foreseen for the relevant currency.	der
When performing a Cash Account delete or restore request, the Party Type of the Requestor must be NCB or Payment Bank.	Delete Cash Account
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.	
Users belonging to Payment Banks can only delete or restore TIPS Credit Memorandum Balances linked to Cash Accounts that fall under their responsibility.	
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.	Delete Cash Account
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.	Delete Cash Account
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.	Delete Cash Account
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.	Delete Cash Account
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.	Delete Cash Account
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	Delete Cash Account

description	user function
Account to be restored must refer to existing Restriction Types whose Object Restriction Type is Cash Account.	
When performing a Cash Account restore request the Linked Account of the T2S Dedicated Cash Account, T2S Central Bank 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 be already assigned to an existing and active instance for the same Cash Account.	Delete Liquidity Transfer Order
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

description	user function
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 Central Bank Accounts and T2S Dedicated Cash Accounts.	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
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	Update Cash Account

description	user function
Account Restriction, the Valid From must be equal to or greater than the current timestamp.	
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
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	Update Cash Account

description	user function
and TIPS respectively must be open outside of the Transit Account validity period.	
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
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

description	user function
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] entity with Object Restriction Type 'Party'.	Create 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

description	user function
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 section, must be equal to the current business date.	Create Party
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

description	user function
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
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

description	user function
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) cannot have a past validity date.	Update Party
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	Update Party

description	user function
Party belonging to the same System Entity and having the same Parent BIC.	
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 Valid To of the relevant Restriction Type.	Update Party
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

description	user function
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 Central Bank, 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 BIC.	Update Party
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	Update Party

description	user function
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.	
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
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 Dedicated Cash Account; if the limit type is autocollateralisation, external guarantee or unsecured credit the Cash Account type must be T2S Dedicated Cash Account or T2S Central Bank 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

description	user function
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 Dedicated Cash Account.	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: A Service Operator user can update all data A NCB user can update only Limits for RTGS DCAs, T2S CMBs and TIPS CMBs belonging to its own System Entity A Payment Bank user can update only Limits for RTGS DCAs, non-primary T2S CMBs and TIPS CMBs linked to its own Cash Account	Update Limit
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 Limit

description	user function
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.	
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
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	Delete Direct Debit Mandate

description	user function
deleted Direct Debit Mandate with future Valid From or past Valid To.	
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 criteria are allowed: - Account Identification - Party BIC	Query Liquidity Transfer Order
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

description	user function
Content for element 'CreDtTm' must match 'MsgId/CreDtTm'	Query Cash Account/Create Cash Account
Content for element "Org/FullLgINm" 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
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

description	user function
- 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
<p>For the user query 'Party Reference Data Query (PYRD)' the following search criteria are allowed:</p> <ul style="list-style-type: none"> - BIC of the Party - Parent BIC of the Party (NCB BIC or CSD BIC) - Party Type - Opening Date - Closing Date 	Query Party
<p>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)':</p> <ul style="list-style-type: none"> - BIC of the Party - BIC of the CSD - BIC of the NCB - Party Type - Opening Date - Closing Date 	Query Party
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
<p>At least one of the following search criteria fields should be present, if the tag search criteria is specified for query Party Audit Trail:</p> <ul style="list-style-type: none"> - PartyId - DatePeriod 	Query Party Audit Trail
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 223 - 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 224 - 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 225 - ESMIG validation rules

16.2 Digital signature on business layer

Will be completed in v2.0.

16.3 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.

16.4 Use of XML and canonicalisation algorithm

Exclusive XML canonicalization⁵⁴ 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#>

16.5 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](#) [333] .

“The NRO⁵⁵ 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.”

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/AppSpfcInf/Sgntr/ds:Signature⁵⁶

Hence a signature will then be constructed as follows:

54 Exclusive XML Canonicalization <http://www.w3.org/TR/xml-exc-c14n/>

55 Non-repudiation of origin is intended to protect against the originator's false denial of having sent the message.

56 Due to the XAdES requirement the ds:keyinfo element inside the ds:signature is covered/protected by the signature.

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.

One reference (in yellow below) points to the keyinfo element of the signature itself. This is a XAdES ⁵⁷ requirement.

1) A message type 1 ⁵⁸ signature example is reported in the below picture:

```
<ds:Signature Id="_8aaee938-014d-489e-a385-b72155000474" xmlns:ds="http://www.w3.org/2000/09/xmlsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmlsig-more#rsa-sha256" />
    <ds:Reference URI="">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmlsig#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>GULJy22YxtDKe7yEvdYfJ/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/xmlenc#sha256" />
      <ds:DigestValue>8CepFq00h78WgVHh23B16RFQRWhdM6rjY+b0texasK=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>QzvbMDLi8Q1PnsfKz...HNgeW=</ds:SignatureValue>
  <ds:KeyInfo Id="_4eaf74f7-086b-410e-b214-45136a615bac">
    <ds:X509Data>
      <ds:X509Certificate>MIIEXTCCA8ag...IY5uKtO3IGZ3XUsw=</ds:X509Certificate>
    </ds:X509Data>
  </ds:KeyInfo>
</ds:Signature>
```

Reference to the whole document, less the signature

Reference to KeyInfo (a XAdES requirement)

Figure 84 - 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>
    <AppISpfcInf>
      <SysUsr>SystemUserX1</SysUsr>
      <Sgntr>...</Sgntr>
      <TtlNbOfDocs>1</TtlNbOfDocs>
    </AppISpfcInf>
    <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>
```

57 ETSI TS 101 903 V1.4.2 (2010-12) XML advanced electronic signatures

58 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>LQSkT1Mksb6iSiyqwCmAA5/ZKd9NkwI068Kukx9JP/U=</ds:DigestValue>
  </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>rLCX6pUzTEYGAHMNu/NczFwbXVgncgVsjmhCNnNsXjbU8CqJeytFM3XJFvPocqq
  TX2ZsPg+GAE89xFBb2xe7j8Z1mgTwEEuU3uvofKjN7Lo4ZnIaUQxPUBStY6cp7K+YtAwQ31bfq2a/mWPQb
  b0C5fUsCwrn/Nxf/6q6PpO+MiMwW0j4mgFnkqv3pFvhmFPPWC1AuReS/RMLjZrGYVSBiBgxkv71D7ijTb
  bbZJzWfwlHK0z7fdzIA10wUzi+9mst858kIEcVX7QhbBdK8PxB5vRGau1lbMIGlRHWEe9fgN6y15rSvpfR
  ODewUS1GU+LgV9SuL3g+GxpwhYT5+MJ/A==</ds:SignatureValue>
  <ds:KeyInfo Id="_f6fa91c7-ee9f-4702-8f08-820bd7a86ac2">
  <ds:X509Data>
```

```

<ds:X509Certificate>MIID0DCCArigAwIBAgIBBTANBgkqhkiG9w0BAQsFADBMMQswCQYDVQQGEWJGUj
EcMBoGA1UECgwTS2V5bmVjdGlzLU9wZW5UcnVzdDEfMB0GA1UEAwwWT3B1b1RydXN0IFRlc3QgQ0EgU0hB
MjAeFw0xMjExMTUwMDUzMzVaFw0xNDExMTUwMDUzMzVaMFgxCAJBgNVBAYTAk1UMQ8wDQYDVQQKDAZPIF
RFU1QXExjAQBGNVBAAsMCU9VIFRFU1QgMjESMBAGA1UECwwJT1UgVEVTVCAxMRAwDgYDVQQDDAduZXN0IEN0
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtNB/1lzF05cVqDIlzQJRszZ9TK7AhlnxnR2E
P1hRnP7GRnnksqyYMJECiL/4NnTEhftQe7AGSaWeX7x0sGHJGd72NwmFQazVjHyaT8XSxaxUoG4kc1F5Qa
DOvxxUAHTtM2qYNjppqFyKkTGbA5D7IqS36zTBYawCE40k9hU2/pvInG3jiKA60U4of9oqEQe4+hw2Ixn0
1mRmxPunKYozWVn3ggL/QQlH/yggkBdplG2qmIUm09cvyVdycABW+5R56NyR42xVRcb56rvI5Qcbnbsrvk
cbmslGdo/qnKvxcchXstt3TqGq+kZ1CIHDoJsF8ZDQKuIjXMEgsurt/OHQIDAQABo4GwMIGtMB0GA1UdDg
QWBRRsJehOf8/t06YtF04hEYcc1C0zoTAFBgNVHSMEGDAWgBRRcv9bAGffzbq1TCZ0MpE7ji+fpTARBglg
hkgBhvhCAQEEBAMCB4AwDgYDVR0PAAQH/BAQDAgBAMEgGA1UdHwRBMD8wPaA7oDmGN2h0dHA6Ly9wa210ZX
N0Lm9wZW50cnVzdC5jb20vT3B1b1RydXN0X1Rlc3RfQ0FfU0hBMTU5cmwwDQYJKoZIhvcNAQELBQADggEB
AGMAu3Yo2Z9Ff1FLX/DHvcw8T5otZlaYtJiHdYcEtvhjY24vcXJzwBuHbfopVu91XZFuxXjG12SSyKsK4s
RHfUVPQdryAMGzMUW+0gVfjupV54jr6vkaELq2t6oyE52CHqvvlHyLJz5CIW6jDEmAzGNJZ2wdRr4fu9z
M2lm4X5JITsZGxY/JH02f1155QJuVn7NSffx8PXR5IKYNZ+Z7kcZNTSL9zDwYXob5PUBv60fXMhWPJtngz
80I8NGqDVQIjtnbgcsSgdchRMV4J0Ub8UK7RAJpG4aR/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>
```

16.6 Message type 2: single ISO 20022 message

For message type 2) the requirement in UDFS chapter [Digital Signature managed within the business layer](#) [▶ 333] states ⁵⁹:

"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 ⁶⁰

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 ⁶¹ 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 ⁶². 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;

59 See also MUG (Message user guide) for BAH; <http://www.iso20022.org/bah.page>

60 Due to the XAdES requirement the ds:keyinfo element inside the ds:signature is covered/protected by the signature.

61 ISO20022 do not support and specify an ID attribute, that can be used to uniquely identify BAH and ISO message.

62 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 2⁶³ signature example (with application specific reference) is reported in the below picture:

<pre> <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>hEXN3t4XgQt2fkJE7WH4xgg/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>bcF4Ty77sjjGLXSd5Yb8QqJijbvy4RRbJkh8zPEFbco=</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> </pre>	<p>Reference to the BAH, less the signature</p> <p>Application specific Reference (to the message)</p> <p>Reference to KeyInfo (a XAdES requirement)</p>
---	--

Figure 85 - 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.

63 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>
  <BizMsgIdr>1SR0524SEC500101</BizMsgIdr>
  <MsgDefIdr>semt.013.001.02</MsgDefIdr>
  <CreDt>2012-05-24T14:25:11Z</CreDt>
  <Sgnt>...</Sgnt>
</AppHdr>
```

Reference to the BAH less the signature

Figure 86 - Reference to the BAH (AppHdr)

Reference to the message (e.g. semt.013):

```
<Document xmlns:swift="urn:swift:xsd:semt.013.001.02_T2S.xsd" xmlns="urn:swift:xsd:semt.013.001.02" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <IntraPosRvmt Instr>
    <TxId>ISRS01801ALM1</TxId>
    <SfkpgAcct>
      <Id>000370550</Id>
    </SfkpgAcct>
    <FinInstnId>
      <ISIN>FC0003620449</ISIN>
    </FinInstnId>
    <IntraPosDtIs>
      <StlmQty>
        <FaceAmt>6000</FaceAmt>
      </StlmQty>
      <StlmDt>
        <Dt>2012-09-28</Dt>
      </StlmDt>
      <BalFr>
        <Cd>AVAS</Cd>
      </BalFr>
      <BalTo>
        <Prtry>
          <Id>FFR1</Id>
          <Issr>T2S</Issr>
          <SchmeNm>RT</SchmeNm>
        </Prtry>
      </BalTo>
    </IntraPosDtIs>
  </IntraPosRvmt Instr>
</Document>
```

The application will provide the signature API with the relevant message.

Figure 87 - 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/xmenc#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/xmenc#sha256"/>
<ds:DigestValue>hWGkHPu5IMYxe4KFYyaMOWFYq0w2pi+BnYnYHEwm/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/xmenc#sha256"/>
<ds:DigestValue>10eHeNdJM1v177M0HzFsmP0IBMYvdPXVuRcR77hAgUg=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>HllitYLicuu5drRrzu5CFxk5GZ3LD00nEPCrXkfWiu54y0zA3P2r6AIE1cYIdue
Y8nioLEvcZcvKVS4zt6bbHv8RRaWmU+Jf13x4vTH5g8W6RY10LPErRbTncn9r3Nb/hxeBj6Rztv3vR+gW+
JY2ly3pkTIAb80JhQ9kcauarcwqG6MAWM3UjK31j796Ldi7ddvHohgW1qHXzdIdBfcONatYnIXZrw/77DU
nBecimz4yqJvCo1Sri1asC0LHFdbeudgBivJtQ/CD1/So9Mkrw6VNUXohv5L3i3J3fNI9gmM1oC/ZJGL1H
LfOsyJ7GokRsydpd1YWFQvNNhu1OupanRA==</ds:SignatureValue>
<ds:KeyInfo Id="_98742d60-2afc-4fa7-a731-828756ce47b1">
<ds:X509Data>

<ds:X509Certificate>MIID0CCArigAwIBAgIBBTANBgkqhkiG9w0BAQsFADBMMQswCQYDVQGEWJGUj
EcMBoGA1UECgwTS2V5bmVjdG1zLU9wZW5UcnVzdDEfMB0GA1UEAwwW3B1b1RydXN0IFRlc3QgQ0EgU0hB
MjAeFw0xMjExMTUwMDU3MzVaFw0xNDExMTUwMDU3MzVaMFgxCzAJBgNVBAYTAklUMQ8wDQYDVQKDAZPIF
RFU1QxEjAQBgnVBAsMCU9VIFRFU1QgMjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5MjE5
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtNB/1lzF05cVqDI1zQJRszZ9TK7Ah1nxxnR2E
P1hrnP7GRnksqyYMEJcIL/4NnTEhftQe7AGSaWeX7x0sGHJGd72NwmFQazVjHyaT8XSxaxUoG4kc1F5Qa
D0vvxUAHTtM2qYNjppfYkKTGBA5D7IqS36zTBYawCE40k9hU2/pvInG3jiKA60U4of9oqEQe4+hW2IxnN0
1mRmxPunkYoZWVn3ggL/QQ1H/yggkBDpLG2qmIU09cvyVdyABW+5R56NyR42xVRcb56rvI5Qcbbnsrvk
cbms1Gdo/qnKvxcThXstt3TqGq+kZ1CIHDoJsF8ZDQKuIjXMEgsurt/OHQIDAQABo4GwMIgtMB0GA1UdDg
QWBBRsJeh0f8/t06YtF04hEYcc1C0zoTafBgNVHSMEGDAWgBRRcv9bAGffzBq1TCZ0MPE7ji+fpTARBglg
hkgBhvCAQEABAMCB4AwDgYDVR0PAAQH/BAQDAgBAEgGA1UdHwRBMD8wPaA7oDmGN2h0dHA6Ly9wa210ZX
N0Lm9wZW50cnVzdC5jb20vT3B1b1RydXN0X1Rlc3RfQ0FFU0hBMT5jcmmwDQYJKoZIhvcNAQELBQADggEB
AGMAu3Yo2Z9Ff1FLX/DHVcw8T5otZ1aYtJiHdYcEtvhjY24vcXJzwBuHbFopVu91XZFuxXjG12SSyK4s
RHfUVPQdryAMGzMUW+OgjVFjupV54jr6vkaELq2t6oyE52CHqvVlHyLJz5CIW6jDEmAzGNJZ2wdRr4f9z
M21m4X5JITsZGxy/JH02f1155QJuVn7NSffx8PxRsIKYNZ+Z7kczNTSL9zDwYXob5PUBv60fXMHWPJtngz
80I8NGqDVQIjtnbgcsSgDchRMVY4J0ub8UK7RAJpG4aR/5RkAmk06DLHXJteXfmsKfLyDq3H8B+eHgFJJW
CeYmVnqk755EVNE=</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
</Sgntr>
</AppHdr>
```

16.7 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.

17 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
	<p>imum size 0 and maximum size 32 KB. The channel can be used for exchange of messages and files.</p>	
messages	<p>A message is a data structure containing a business application header and a payload.</p>	CLM/RTGS
monetary financial institution	<p>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.</p>	CLM
multi-addressee	<p>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.</p>	RTGS
network service	<p>A modus for connection, four possible options:</p> <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	<p>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</p>	CLM/RTGS
non repudiation of origin	<p>Protection against the originator's false denial of having sent the message</p>	CLM/RTGS
notification	<p>A confirmation of a change in the business status of a payment, liquidity transfer, other instruction, account floor or account ceiling</p>	CLM/RTGS
offsetting	<p>Offsetting in the RTGS aims at increasing the capacity of the system to settle payments,</p>	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

18 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 ⁶⁴	liquidity transfer
LTO ⁶⁵	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

64 Only used in figures

65 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