Real-time gross settlement

User detailed functional specifications

Author	4CB
Version	0.3
Date	16 July 2018

Table of contents

List of figures

List of tables

Introduction (completed)

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

Therefore, the UDFS document focuses on the provision of information to RTGS participants to design and build the interface of their business application with RTGS (A2A/U2A). The UDFS RTGS is available for the whole community in order to ensure the same level of knowledge for all RTGS participants, including central banks.

The document is divided into three parts:

- The first part provides a full description of all the RTGS features and the related accounts and application processes, functional details concerning access to RTGS and connectivity, dependencies and interactions with other services/components, operations and support features. The background information provided in chapter Overview of RTGS component [26] supports the understanding of the RTGS component with its applications and the interaction of the common components described in the following chapters. Afterwards it guides the reader through the RTGS core functionalities (i.e. participation and accounts). Moreover, it provides an overview of common components used by RTGS (e.g. common reference data management (CRDM), data warehouse). The contingency services are explained in chapter Contingency services [231] and central bank specific information is provided in chapter Additional information for central banks [233].
- The second part provides process descriptions, which allow RTGS participants to interact with RTGS via A2A as well as a functional overview of the U2A processes. This part aims at providing a comprehensive description of all processes being available in RTGS and which the user may instruct. Moreover, the related settlement processes are explained in detail. Furthermore the chapter Dialogue between CRDM and CRDM actors describes the dialogue between CRDM and participants via A2A. Subsequently, also the interaction with ESMIG is outlined in chapter Dialogues and processes between ESMIG and participant [258].
- The third part provides a detailed description of all XML messages RTGS participants may use to interact in A2A mode with RTGS. The descriptions of the messages include all required elements according to the schema. Wherever a message or its fields are referenced throughout the document, only the reference name is used.

This document has been submitted for market consultation on XX.XX.2018 and was finally published on the ECB website on XX.XX.2018.

Reader's Guide (completed)

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.

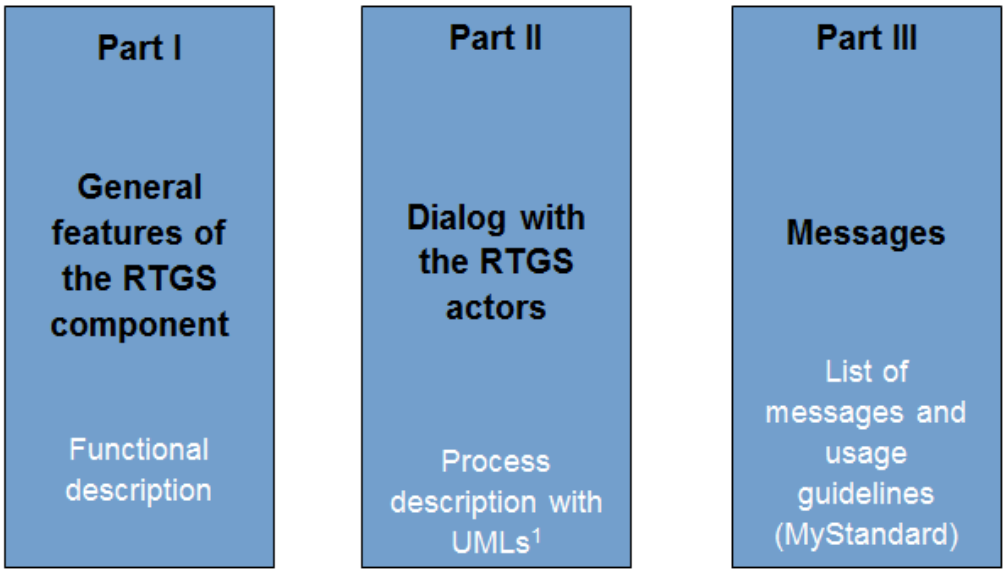


Figure 1 - Structure of the RTGS UDFS

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

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

- Overview of RTGS component [▸ 26] which summarizes the RTGS features and functionalities
- Access to RTGS [▸ 28], which focuses on how to connect to RTGS including authentication and authorisation processes and explains the envisaged usage of access rights depending on the respective roles;
- Parties and accounts [▸ 30], which provides a general description of the main reference data needed for RTGS and the accounts maintained in RTGS, specifying how they are used for the settlement of high value payments (e.g. which parties and RTGS participants are involved and how to set up accounts for different purposes including their usage);

- Contingency services [231], which informs how to act in case of a defined contingency situation.

Business oriented perspective

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

- Business day [50], where the business reader finds an overview of respective processes and schedules;
- Business and features description [51], which informs about the payments processing and settlement of payments and ancillary systems as well as the liquidity management and information management;
- Overview of used common components in RTGS component [182] completes the view on the message transfers used in RTGS;
- Payment instruction processing [235] to find a description of the processing of a payment instruction and useful information in order to understand the management of RTGS payment settlement
- Dialogue between CRDM and CRDM actors wherein query information may be of relevance;
- Index of business rules and error codes [420] includes the relevant codes to perform functional checks.

Technical oriented perspective

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

- Processes with RTGS components [235] respectively 11 “Dialogue between CRDM and CRDM actors”, where an overview of the possible A2A dialogue with RTGS is defined. Each sub-chapter of this chapter describes the flows within, to and from RTGS. The reader can focus on the functionality of note, analysing the procedures and main scenarios;
- III Catalogue of messages [259], where a detailed description of the content of a given XML message is provided;
- Index of status value and codes [422] with further details on the checks to be performed and codes used in the messages.

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

I General features of the RTGS component

1 Overview of RTGS component (completed)

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

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

- The RTGS component is multi-currency enabled, i.e. the settlement services support settlement in different currencies and according to their own calendars; none of the TARGET services offer conversion between currencies.
- The A2A communication between credit institutions and the RTGS component are based on the ISO 20022 compliant messages.

All interactions of the credit institutions with their central bank are not processed in the RTGS component but in the newly introduced central liquidity management (CLM) component.

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

RTGS makes use of the following Eurosystem services:

- The Eurosystem single market infrastructure gateway (ESMIG) provides the central authentication, authorisation and user management features. It is network provider agnostic and thus offers participants 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 application-to-application (A2A) mode (in store-and-forward and real-time communication protocol) and user-to-application (U2A) mode via GUI.
- The common reference data management (CRDM) component offers features that allow authorised users to set up, maintain and query all reference data that TARGET services share for their processing activities. CRDM ensures the consistency and integrity of all reference data, processing and relationships across services. Furthermore, it avoids duplication of reference data or redundant implementation

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

- The data warehouse (DWH) component provides the data from T2 (i.e. CLM and, RTGS) or T2S for historical, statistical and regulatory reporting. It offers predefined reports and templates for database queries. The data warehouse participants may access the data warehouse in A2A and U2A (via) mode and subscribe for respective reports and templates. The data of previous business days are available in data warehouse as of the next business day.
- The business day management offers the common scheduler and calendar for all services and components. A common scheduler 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 common calendar defines the days when a TARGET service or a common component is opened and follow 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 have to be defined in CRDM (e.g. to whom the invoice is addressed to, which main cash account 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 component collects all information, which is subject to legal archiving requirements such as all incoming and outgoing business transactions from and to participants as well as relevant reports such as account statements. The information from TARGET services and common components is stored in legal archiving in its original content and format after thirty calendar days and is accessible within its retention period of ten years.

2 Access to RTGS

2.1 Connectivity (U2A/A2A) (to be completed in iteration 4)

2.2 Authentication and authorisation process (to be completed in iteration 4)

2.3 Security (completed)

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

It means that the following security conditions are met:

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

2.3.1 Confidentiality (completed)

The confidentiality of data is ensured by the possibility to grant specific access rights for any given set of data, as detailed in section 1.2.3 “Access rights”. In conjunction with mechanisms of authentication and authorisation applied to all requests received by RTGS in both A2A and U2A mode, this guarantees that each RTGS actor’s data is treated confidentially and is not accessible to non-authorised actors.

2.3.2 Integrity (completed)

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

In U2A mode, RTGS offers users in addition the possibility to further ensure the data integrity via usage of a dual authorisation concept, the 4-eyes principle. In case this option is chosen for a specified set of RTGS

operations, a second independent verification and confirmation is required before an operation becomes active in RTGS. If, for example, a critical set of data should be modified and the person requesting the change is only allowed to do so under the 4-eyes principle, then a second person of the same party has to confirm the correctness of the request. Otherwise, the requested change is not implemented.

2.3.3 Availability (completed)

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

2.3.4 Monitoring (completed)

RTGS operational monitoring provides tools to the T2 operator for the detection in real-time of functional or operational problems. Technical monitoring allows for the detection of hardware and software problems via real-time monitoring of the technical components involved in the processing, including the network connections.

2.3.5 Auditability (completed)

RTGS provides an audit trail with which it is possible to reconstruct 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 messages (e.g. changes of access rights, alerts and exceptional events).

2.4 Graphical user interface (to be completed in iteration 4)

3 Parties and accounts (completed)

3.1 Parties and RTGS actors (completed)

Entities that interact with RTGS are generally known as RTGS actors. The RTGS participation model envisions different types of actors, with different roles and responsibilities, as outlined in chapter Concept of party in RTGS [▸ 31] . RTGS actors are defined as different entities in CRDM.

This chapter provides a detailed description of all the reference data CRDM stores and RTGS uses for all RTGS actors. More in detail, chapter Setup of RTGS actors [▸ 30] identifies the reference data related to the setup of actors for RTGS and it provides detailed information as to who is responsible for the setup of these reference data. Chapter Concept of party in RTGS [▸ 31] defines the concept of party in CRDM and the way this concept relates with the different types of legal entities that can interact with RTGS. Chapter Hierarchical party model [▸ 32] describes the so-called hierarchical party model, i.e. the organisational structure of parties in the CRDM repository. Chapters Party identification [▸ 32] and Reference data for parties in RTGS [▸ 33] illustrate the reference data required by RTGS for each actor, i.e. the way a party can be identified in RTGS and which attributes have to be stored for each actor.

3.1.1 Setup of RTGS actors (completed)

The setup of RTGS actors takes place in CRDM.

The T2 operator is responsible for setting up and maintaining party reference data for all central banks in RTGS. Central banks are responsible for setting up and maintaining party reference data for the parties of their national community.

The following table summarises, for each reference data object related to the setup of RTGS actors, the actor responsible for its configuration and it specifies which mode the actor can use for the configuration.

Table 1 - Setup of parties for RTGS

Reference data object	Responsible actor	Mode
Party (central bank)	T2 operator	U2A
Party (RTGS participant)	Central bank	U2A
Party (ancillary system)	Central bank	U2A
Banking group	Central bank	U2A

The RTGS actor “Authorised account user” will be described in iteration 4.

3.1.2 Concept of party in RTGS (completed)

Any RTGS actor, meaning any legal entity or organisation participating in and interacting with RTGS, is defined as an entity in the CRDM repository. Depending on their role in RTGS, RTGS actors may be defined as a party in CRDM. Each party belongs to one of the following party types:

- T2 operator
- central bank
- RTGS participant
- ancillary system

The **T2 operator** is the organisational entity that operates RTGS. He is responsible for the initial setup and day-to-day operations of RTGS and act as single point of contact for central banks in case of technical issues. They are responsible for monitoring the system and carrying out corrective actions in case of incidents or in the event of service/component unavailability. The T2 operator is also responsible for setting up and maintaining central banks reference data in the CRDM repository and, if required, they may operate RTGS functions on behalf of any RTGS actor, upon request of the respective central bank. They have full access to all live and all archived reference data and transactional data in RTGS.

Central banks are responsible for setting up and maintaining reference data in the CRDM repository for all RTGS actors belonging to their community. Central banks can also act as RTGS participants themselves (see below). In addition they can act on behalf of one of their RTGS participants in case of need.

In its central bank role, it may only own central bank accounts (see glossary for the definition of a central bank account); all other account types need to be owned under its RTGS participant role.

RTGS participants represent entities that own dedicated cash accounts (dedicated cash account) and/or sub accounts in RTGS and are identified by a BIC11. RTGS participants are responsible for their own liquidity management through their linked main cash account in CLM. They are responsible for setting up their own dedicated cash accounts, instructing payments and monitoring the liquidity usage. However, the creation and maintenance of the dedicated cash accounts is done by central banks.

Ancillary systems can be given the right to submit instructions via the dedicated cash account of a RTGS participant on its behalf or via a RTGS dedicated cash account sub-account dedicated to the ancillary system. In principle they shall not own a RTGS dedicated cash account. However, there may be exceptions in order to cover certain market conditions.

The role of **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. Banking group is not defined as a party, but as a banking group identifier that central banks can define.

Each legal entity may play different roles in RTGS. Any legal entity playing multiple business roles in RTGS results in the definition of multiple parties.

Conversely, a (non-central bank) legal entity owning two dedicated cash accounts within the books of a central bank would be defined as two different RTGS participants, each identified by a different BIC-11.

Similarly, a (non-central bank) legal entity holding two dedicated cash accounts within the books of two central banks would also be two separate RTGS participants, each identified by a different BIC-11.

3.1.3 Hierarchical party model (completed)

The party model of RTGS is based on a hierarchical three-level structure. The T2 operator is the only party on the top level of the hierarchy and is responsible for the setup of each party of the second level, i.e. each central bank in RTGS. Similarly, each party belonging to the second level (i.e. a central bank) is responsible for the setup of all parties of its community, represented by parties of the third level.

The hierarchical model also determines the reference data scope, i.e. the area of responsibility of each central bank and of the T2 operator. More into detail:

- The reference data scope of the T2 operator includes all reference data, i.e. all the reference data in scope of all central banks plus operational specific reference data;
- The reference data scope of a central bank includes its own reference data, plus the reference data of all its RTGS participants and ancillary systems;
- The reference data scope of a RTGS participant includes only its own reference data;
- The reference data scope of an ancillary system is restricted to the accounts and activities as authorised by the RTGS participant or it can be restricted to a sub-account only or it can also cover a technical liquidity account held by themselves or a central bank.

Each central bank and the T2 operator are responsible for their own reference data scopes, i.e. each of them is responsible for the input and maintenance of all information included in its reference data scope. The T2 operator may also act, upon request, on the reference data scope of a central bank and its parties of the third level.

3.1.4 Party identification (completed)

RTGS imposes a constraint in the assignment of BICs related to its parties, due to the fact that the settlement process must be able to determine the accounts to be debited and credited by a payment based on the BICs of the RTGS participant. This implies the need to ensure that any given BIC can only be assigned to one RTGS participant or ancillary system. Two different RTGS participants or ancillary systems must have assigned two different BICs.

For this reason, CRDM will prevent two different parties to be defined as RTGS participant or ancillary system if they are identified by the same 11-character BIC. Therefore, in order to allow a given legal entity to be defined as two different RTGS participants or ancillary systems (by the same central bank or by two different

central banks), the same legal entity must be defined in the CRDM repository as two parties identified by two different 11-character BICs.

3.1.5 Reference data for parties in RTGS (completed)

The following table gives an overview on the party reference data attributes in RTGS.

Table 2 - Party reference data attributes

Attribute	Description
Party identifier	It specifies the unique technical identifier of the party.
Party long name	It specifies the full name of the party.
Party short name	It specifies the short name of the party.
Jurisdiction	It specifies the country of jurisdiction for the party. This attribute shall be mandatory for a legal address. It shall be the same country as in the legal address, except for supranational institutions.
Street	It specifies the name of the street for the address.
House number	It specifies the house number for the address.
City	It specifies the name of the city for the address.
Postal code	It specifies the postal code for the address.
State or province	It specifies the state or province for the address. Its use shall depend on the country code of the address.
Country code	It specifies the two-character ISO country code (ISO3166-1) identifying the country code of the address.
Party BIC code	It specifies the BIC11 to uniquely identify the party in RTGS.
Parent BIC code	It specifies the BIC11 code of the parent responsible for the party. Where the party is a parent and there is no other party having responsibility over it, then parent BIC code is the same as the party BIC code.
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 type	Type of party. The exhaustive list of party types is as fol-

Attribute	Description
	lows: - T2 operator - central bank - RTGS participant - ancillary system
Party status	It specifies the business status (active,...) of a party for processing in the system.
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 monetary financial institution (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.
First activation date	It specifies the date of the first activation of the party.
Modification date	It specifies the activation date of the displayed party status.
Currency code	It specifies the national currency associated with a central bank.

The following table gives an overview on the party contact reference data attributes in RTGS.

Table 3 - Party contact reference data attributes

Attribute	Description
Contact name	It specifies the name of the contact for the party.
Contact position	It specifies the position or role of the contact for the party.
Valid from date	It specifies the date from which the party contact is valid.
Office telephone number	It specifies the office telephone number for the party contact.

Attribute	Description
Mobile number	It specifies the mobile number for the party contact.
Email address	It specifies the email address for the party contact.
Valid to date	It specifies the date until when the party contact is valid.

The following table gives an overview on the banking group reference data attributes in RTGS.

Table 4 - Banking group reference data attributes

Attribute	Description
Banking group identifier	It specifies the identifier of the banking group.
Banking group name	It specifies the name of the banking group.

3.2 Accounts structure and organisation (completed)

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

The T2 operator and central banks input and maintain in the CRDM repository the following categories of accounts, depending on their role:

- RTGS dedicated cash accounts
- sub-accounts
- dedicated transit accounts
- central bank accounts
- ancillary system guarantee funds accounts
- ancillary system technical accounts

Furthermore, RTGS participants may define:

- liquidity transfer groups
- direct debit mandate(s)
- main cash account linked to their dedicated cash accounts
- floor/ceiling information
- current limit(s)

- current reservation(s)
- standing liquidity transfer orders
- standing orders for reservation
- standing orders for limits
- notification message subscription
- report configuration

The following sections define the above mentioned reference data objects, whereas chapter Reference data for accounts in RTGS [▸ 40] provides a detailed description of the reference data required by RTGS for the same reference data objects.

3.2.1 Dedicated cash accounts in RTGS (completed)

The dedicated cash account (dedicated cash account) in RTGS is used for the settlement of real-time interbank and customer payments and transactions with ancillary systems. For credit institutions, it may either have a zero or a positive balance. Central banks dedicated cash accounts might however have a negative balance.

A RTGS actor may open several RTGS dedicated cash accounts but it shall ensure that each of these dedicated cash accounts is identified with a unique BIC11 (in addition to different account numbers).

It is up to central banks to create and maintain dedicated cash accounts for their RTGS participants.

3.2.2 Sub-accounts in RTGS (completed)

A sub-account is a technical account, belonging to a dedicated cash account, holding dedicated liquidity to allow the settlement of an ancillary system under ancillary system procedure “Settlement on dedicated liquidity accounts”. The sub-account is identified by an account number and directly linked to one and only one dedicated cash account, the latter being identified by a unique BIC11.

A sub-account may either have a zero or a positive balance. It is up to central banks to create and maintain sub-account for their RTGS participants.

3.2.3 Dedicated transit accounts (completed)

Dedicated transit accounts in RTGS are accounts that are owned by central banks which may have either zero or negative balance as they reflect any movement of liquidity from/to CLM. They are technical accounts involved in the liquidity transfer process and cannot be involved in the settlement of real-time interbank and customer payments and transactions with ancillary systems.

There is only one dedicated transit account per settlement currency in RTGS. The dedicated transit account for euro belongs to the European Central Bank.

It is up to the T2 operator to create and maintain the dedicated transit accounts for the central banks.

3.2.4 Central bank accounts (completed)

A central bank account in RTGS is a cash account owned by a central bank that is allowed to have negative balance. Specific requirements apply to non-euro area central banks.

A central bank account in RTGS is identified by a BIC11. Central banks have the possibility to open more than one central bank account, each one being identified by a unique BIC11.

It is up to the T2 operator to create and maintain the central bank accounts.

3.2.5 Ancillary system guarantee funds accounts (completed)

An ancillary system guarantee account is an account in RTGS for maintaining funds allocated to the settlement of balances of an ancillary system in case of failure of settlement bank(s) under ancillary system procedures “Standard multilateral settlement” and “Simultaneous multilateral settlement”.

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

It is up to central banks to create and maintain the ancillary system guarantee funds accounts.

3.2.6 Ancillary system technical accounts (completed)

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

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

It is up to central banks to create and maintain the ancillary system technical accounts.

3.2.7 Liquidity transfer groups (completed)

A liquidity transfer group refers to an optional grouping of dedicated cash accounts for the purpose of arranging intra-RTGS liquidity transfers between them. It is possible for an account to participate to one or multiple liquidity transfer groups.

The liquidity transfer group is identified by a specific ID.

It is up to central banks to create and maintain the liquidity transfer groups and define the dedicated cash account linked to each liquidity transfer group.

3.2.8 Direct debit mandate (completed)

For each RTGS participant CRDM manages the information about the direct debit(s) this participant has authorised and the related attributes (e.g. maximum amounts).

It is up to central banks to create and maintain the direct debit mandate(s) of a RTGS participant in CRDM.

3.2.9 Linked main cash account (completed)

In the event the floor or ceiling on a dedicated cash account is breached (after the settlement of a payment) and if the RTGS participant has opted for the automated liquidity transfer order generation, RTGS generates automatically an inter-service liquidity transfer order to pull cash from the linked main cash account (in the event the floor is breached) or push cash to the linked main cash account (in the event the ceiling is breached).

It is up to RTGS participants to create and maintain the linked main cash accounts in CRDM.

3.2.10 Floor/ceiling (completed)

For each RTGS dedicated cash account, a RTGS participant can define in CRDM a minimum (“floor”) and maximum (“ceiling”) amount that shall remain on the respective account. The RTGS participant can choose between the following behaviours that the system shall apply in the event the floor or ceiling on an account is breached (after the settlement of payments):

1. RTGS generates a notification that is sent to the RTGS participant informing about the floor/ceiling breach (upon which the RTGS participant can take action); or
2. RTGS generates automatically an inter-service liquidity transfer order to pull cash from the defined main cash account in CLM (in the event the floor is breached) or push cash to the defined main cash account in CLM (in the event the ceiling is breached).

It is up to RTGS participants to create and maintain the floor/ceiling information in CRDM.

3.2.11 Current limit (completed)

A limit is the maximum amount for normal payments that a direct RTGS participant is willing to pay to another specific participant/account per day (bilateral limit), or to all other participants/accounts (excluding those with whom a bilateral limit is defined) per day (multilateral limit). Limits are net values within the day. Payments out can exceed the limit where payments (that are credits) have also been received first. It is not pos-

sible to define a multilateral limit without any existing bilateral limit. With the usage of the current limit facility bilateral and/or multilateral limits can be set up and modified intra-day by RTGS participants. This information is defined at the level of the dedicated cash account and it is up to RTGS participants to set up and manage current limits in RTGS.

3.2.12 Current reservation (completed)

Liquidity can be reserved and modified intra-day by RTGS participants for the execution of special transactions with a certain priority class (urgent and high).

This information is defined at the level of the dedicated cash account and it is up to RTGS participants to set up and manage the current reservations in RTGS.

3.2.13 Standing liquidity transfer order (completed)

A standing liquidity transfer order is an instruction of a RTGS participant to transfer regularly a fixed amount of liquidity, upon a certain event, from a RTGS dedicated cash account to another account over a period with or without a predefined end date.

This information is defined at the level of the dedicated cash account and it is up to the RTGS participant to create and manage its standing liquidity transfer orders information in CRDM.

3.2.14 Standing order for reservation (completed)

A standing order for reservation is an instruction of a RTGS participant to set up an urgent/high reservation of a fixed amount for a business day on a dedicated cash account without a predefined end date.

This information is defined at the level of the dedicated cash account and it is up to the RTGS participant to create and manage its standing order for reservation information in CRDM.

3.2.15 Standing order for limit (completed)

A standing order for limit is an instruction of a RTGS participant to define bilateral and/or multilateral limits of a fixed amount within the RTGS on a regular basis. These limits are processed during the start of day procedure of the following business day.

This information is defined at the level of the dedicated cash account and it is up to the RTGS participant to create and manage its standing orders for limit in CRDM.

3.2.16 Notification message subscription (completed)

Message subscription shall allow a RTGS participant to elect another party to receive pre-defined messages either instead or in addition.

This information is defined at the level of the dedicated cash account and it is up to the RTGS participant to create and manage the notification message subscription in CRDM.

3.2.17 Report configuration (completed)

The RTGS participant can configure standard reports that RTGS shall create at certain times during a business day or at certain business day events. RTGS participants can specify in their report configuration, whether such report shall be sent to the recipient immediately in A2A mode or be stored for later querying in A2A mode or downloading via GUI. Such standard reports are available for later querying and downloading until the next report based on the same configuration is created.

Report configuration shall also allow a RTGS participant to elect another party to receive the report either instead or in addition.

This information is defined at the level of the dedicated cash account and it is up to the RTGS participant to create and manage the report configuration in CRDM.

3.2.18 Reference data for accounts in RTGS (completed)

[This chapter provides an overview of the attributes of the reference data objects previously described and does not give any indication on the structure of CRDM reference data tables.]

The following table shows an exhaustive list of account reference data attributes in RTGS.

Table 5 - Reference data attributes

Attribute	Description
Account number	It specifies the number of the account (unique across all services).
Account type	Type of account. The exhaustive list of account types in RTGS is as follows: - dedicated cash account - sub-account for ancillary system settlement - dedicated transit account - central bank account

Attribute	Description
	• ancillary system guarantee funds account • ancillary system technical account
Currency	It specifies the currency of the account.
Account owner	It specifies the BIC11 of the party owning the account (unique within RTGS).
Status	Blocking status for the account. Exhaustive list of possible values: • blocked for credit • blocked for debit • blocked for credit and debit • unblocked
Floor	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	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.
Maximum amount for direct debit per day	It specifies the maximum amount of direct debits which can be debited each day on the dedicated cash account.
Main cash account to be debited	It specifies the main cash account to be debited within the billing process.
Ancillary system used	It specifies the ancillary system used.
Ancillary system model used	It specifies the ancillary system model used.
Linked main cash account	It specifies the linked main cash account.
Minimum reserve party	It specifies the party for which this account is included for minimum reserve calculation.
Default flag	It indicates whether the account is the default choice of the party.

Attribute	Description
Account monitoring group identifier	It specifies the unique technical identifier of an account monitoring group.
Opening date	Opening date of the account.
Closing date	Closing date of the account.

Each dedicated cash account is linked to one and only one RTGS participant (i.e. the account owner); similarly, each dedicated transit account is linked to one and only one central bank (the European Central Bank for the Euro transit account, the relevant central bank for any other settlement currency).

Moreover, one or many sub accounts may be linked to one and only one dedicated cash account.

Furthermore, each dedicated cash account may be linked to one or many liquidity transfer groups and to one or many account monitoring groups.

The following table shows an exhaustive list of liquidity transfer group reference data attributes in RTGS.

Table 6 - Liquidity transfer group reference data attributes

Attribute	Description
Liquidity transfer group identifier	It specifies the unique technical identifier of the liquidity transfer group.
Liquidity transfer group name	It specifies the name of the liquidity transfer group.
Account(s)	It specifies the account(s) belonging to the liquidity transfer group.

The following table shows an exhaustive list of direct debit reference data attributes in RTGS.

Table 7 - Direct debit reference data attributes

Attribute	Description
Direct debit identifier	It specifies the unique technical identifier of the direct debit mandate.
Account number	It specifies the account on which the direct debits are authorised.
Payee party identifier	It specifies the party from whom payment requests were authorised under this mandate and to whom the corresponding payments are made.

Attribute	Description
Payee reference	The reference provided by the payee party to be included in the payment details for recognition of the payment.
Maximum amount (counterpart)	It specifies the maximum amount the authorised issuer is able to direct debit during the single business day.
Maximum amount per payment	It specifies the maximum amount the authorised issuer is able to debit directly in a single direct debit.
Valid from date	It specifies the date from which the direct debit instruction is valid.
Valid to date	It specifies the date until which the direct debit instruction is valid.

The following table shows an exhaustive list of the standing liquidity transfer order reference data attributes in RTGS.

Table 8 - Standing liquidity transfer order reference data attributes

Attribute	Description
Standing liquidity transfer order identifier	It specifies the unique technical identifier of the standing liquidity transfer order.
Transfer Type	It specifies the type of the liquidity transfer. The exhaustive list of transfer type options in RTGS is as follows: - inter-service liquidity transfer from RTGS dedicated cash account to main cash account - intra-service liquidity transfer to another RTGS dedicated cash account - inter-service liquidity transfer to dedicated cash account in another settlement service
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.
Currency	It specifies the currency of the amount to be debited with the liquidity transfer.
RTGS dedicated cash account to be debited	It specifies the dedicated cash account to be debited in

Attribute	Description
	RTGS.
Account to be credited	It specifies the account (dedicated cash account and/or main cash account) to be credited.
Trigger event	It specifies the event type that will trigger 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.

The following table shows an exhaustive list of the standing order for reservation reference data attributes in RTGS.

Table 9 - Standing order for reservation reference data attributes

Attribute	Description
Standing order for reservation identifier	It specifies the unique technical identifier of the standing order for reservation.
Priority type	It specifies the type of priority. The exhaustive list of priority class options is as follows: - urgent - high
Reservation amount	It specifies the amount of the required reservation.
Account	It specifies the account number of the dedicated cash account for which the reservations are made.
Valid from date	It specifies the date from which the standing order for reservation is valid.
Valid to date	It specifies the date until which the standing order for reservation is valid.

The following table shows an exhaustive list of the standing order for limit reference data attributes in RTGS.

Table 10 - Standing order for limit reference data attributes

Attribute	Description
Standing order for limit identifier	It specifies the unique technical identifier of the standing order for limit.
Account number	It specifies the number of the respective RTGS dedicated cash account on which the bilateral/multilateral limit(s) are defined.
Limit type	It specifies the type of limit. The exhaustive list of limit type options is as follows: - bilateral - multilateral
Defined limit	It specifies the value of the limit(s) defined at the level of the dedicated cash account.
Counterparty BIC	It specifies the BIC of the RTGS participant for which normal payments are restricted by the bilateral limit.
Valid from date	It specifies the date from which the standing order for limit is valid.
Valid to date	It specifies the date until which the standing order for limit is valid.

The following table shows an exhaustive list of the message subscription reference data attributes in RTGS.

Table 11 - Message subscription reference data attributes

Attribute	Description
Message subscription identifier	It specifies the unique technical identifier of the message subscription.
Message identifier	It specifies the identifier of the message subscribed to by the RTGS participant.
Account	It specifies the account number of the dedicated cash account for which the message has been subscribed.
Recipient	It specifies the identifier of the party subscribing to the message for the account.
Alternative recipient identifier	It specifies the identifier of the party nominated to receive

Attribute	Description
	the message either instead of or in addition to the recipient.
Additional copy	This flag indicates that the recipient will still receive the message in addition to the nominated alternative recipient.
Business case	It specifies the business case for which a message has to be sent.
Subscription valid from	It specifies the date from which the subscription is valid.
Subscription valid to	It specifies the date until which the subscription is valid.

The following table shows an exhaustive list of the report configuration reference data attributes in RTGS.

Table 12 - Report configuration reference data attributes

Attribute	Description
Report configuration identifier	It specifies the unique technical identifier of the report configuration.
Report identifier	It specifies the configured report for the account.
Account	It specifies the account number of the dedicated cash account for which the report has been configured.
Recipient	It specifies the identifier of the party configuring the report for the account.
Parameters for report	It specifies whether the relevant report are received in full or delta mode, and whether in push or pull mode.
Scheduled time	It specifies the scheduled time when the report is provided. Either scheduled time or scheduled event must be specified, but not both.
Scheduled event	It specifies the event that shall trigger the report to be produced. Either scheduled time or scheduled event must be specified, but not both.
Configuration valid from	It specifies the date from which the configuration is valid.
Configuration valid to	It specifies the date until which the configuration is valid.

3.3 Shared reference data (completed)

3.3.1 RTGS directory (completed)

Directories provide information on all participants that are reachable for payments via a Eurosystem market infrastructure. There is a dedicated directory for all participants in RTGS. This RTGS directory stores all the needed routing information in order to support the routing of payments in RTGS. The structure of the RTGS directory is described later in this chapter. A party can also request that its BIC is not published in the directory. In such a case, its counterparts can make payments to the account linked to this BIC only if the party has previously provided the BIC to them.

RTGS actors may receive the RTGS directory in two ways:

- Push mode: RTGS sends after the end-of-day processing the full version or the delta version of the RTGS directory to all RTGS actors who created for this an appropriate report configuration; or
- Pull mode: at any time during the service hours of CRDM, a RTGS actor may download either the full version or the delta version of the RTGS directory from a CRDM web-page.

The structure of the RTGS directory will be described in iteration 4.

3.3.2 RTGS calendar (completed)

The RTGS calendar specifies the calendar days when RTGS is open and follows the defined business day schedule. Different calendars per currency are set up to operate different closing days.

3.3.3 RTGS scheduled events (completed)

The RTGS scheduled events specifies the scheduled events that will automatically trigger a specified process within the RTGS component.

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

Table 13 - Attributes of the RTGS scheduled events

Attribute	Description
Scheduled event identifier	It specifies the unique technical identifier of a scheduled event.
Process identifier	It specifies the unique technical identifier of a business process.
Scheduled event status	It indicates whether the scheduled event has occurred and

Attribute	Description
	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.

3.3.4 RTGS currency (completed)

The RTGS currency specifies the available settlement currencies in RTGS.

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

Table 14 - Attributes of the RTGS currency

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.

3.4 Interaction with CRDM (completed)

CRDM provides features that allow duly authorised users to set up, update, delete and query all reference data that are shared by multiple services/components (e.g. CLM, RTGS...) for their processing activities. The access to CRDM is possible in U2A mode (for all functions) and in A2A mode (for a subset of functions) via ESMIG. In order to ensure a timely and consistent propagation of common reference data to the relevant components, CRDM implements a publish-subscribe feature allowing each component to receive all the common reference data (and their changes) they require for their processing.

In a nutshell:

- CRDM publishes all changes (in push mode) of common reference data (e.g. creations of new objects, updates of already existing objects).
- Other subscriber components get those changes too and apply them to their local reference data management component, according to their needs.
- Other detailed information can be found in chapter 6.1.

As far as RTGS is concerned, all reference data setup and maintenance operations are performed in the CRDM while changes on local data are performed in RTGS directly. The reference data are then propagated from the CRDM to RTGS asynchronously on a daily basis. However, the immediate update of specific reference data (e.g. blocking of RTGS dedicated cash account) will be done directly in RTGS and will not be propagated from CRDM.

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

The set of reference data that RTGS receives on business day T+1 includes all the active data of the mentioned business date. If an item, propagated on date T, contains a validity-date in the future (e.g. T+2), RTGS acquires it during the daily propagation but the item will be available in RTGS only when the validity date is reached.

The following diagram shows a conceptual overview of the interactions between CRDM and RTGS.

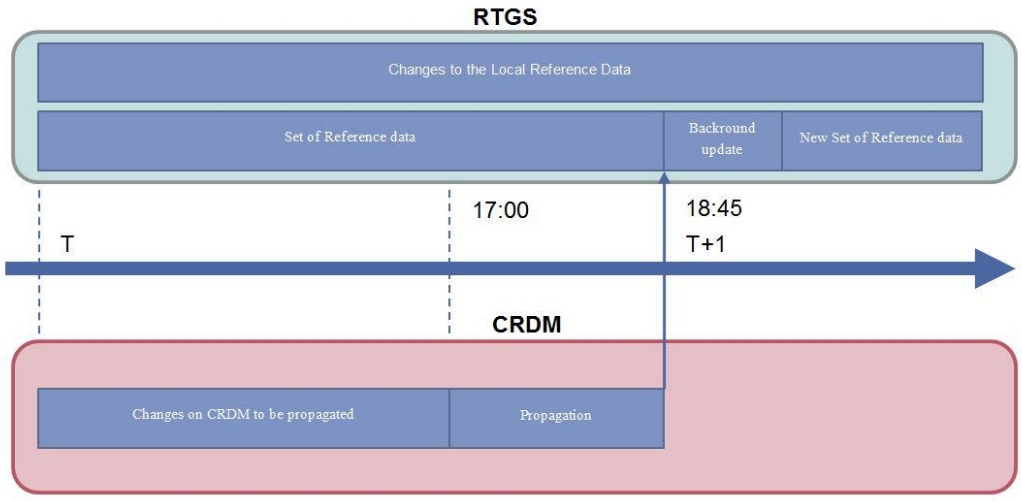


Figure 2 - Interaction between CRDM and RTGS

4 Business day (to be completed in iteration 4)

5 Business and features description

5.1 Payment types

5.1.1 Overview (completed)

The RTGS component enables the settlement of real-time inter-bank payments, customer payments and liquidity transfers as well as the settlement of ancillary system related payment instructions. The term payment order encompasses payments as well as liquidity transfers.

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

Table 15 - Overview of payments in the RTGS component

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

Besides payments, also liquidity transfers (LiquidityCreditTransfer (camt.050) [▶ 336]) are processed in the RTGS component. Different users can initiate liquidity transfers. Further details on liquidity management can be found in chapter Liquidity management [▶ 140].

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

- the owner of the account to be debited
- the owner of the account to be credited (in case of direct debits)
- a third party (e.g. in case of mandated payments sent by a central bank or an ancillary system)
- a central bank acting on behalf of a credit institution
- an ancillary system using interbank payments

Depending on the message subscription, an RTGS participant may receive certain notifications. The relevant message type used by the RTGS component may also depend on who has sent the payment order to the RTGS component (i.e. in case the account holder is different from the actor submitting the payment order).

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

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

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

5.1.2 Comparison of different payment types (to be completed in iteration 4)

5.1.3 Definition of execution time (completed)

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

- payments with an “earliest debit time indicator”
- payments with a “latest debit time indicator”

The following table describes payments with a set execution time.

Table 16 - 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)	▪ Option a: payments to be executed up to a certain time (message element: RejectTime) ▪ Option b: payments which should be executed up to certain time (only warning indicator) (message element: TillTime)
Effect	▪ The payment is stored until the indicated time. ▪ At the earliest debit time, the payment runs through the entry disposition.	▪ Setting a latest debit time only means a special identification via the U2A or A2A query. ▪ In the entry disposition, the payment is treated like any other payment of the same priority.
Management	If the payment cannot be settled at the earliest debit time, it is queued till cut-off time for payment type is reached (or the payment is revoked).	If the payment cannot be settled until the indicated debit time, ▪ Option a: the payment is rejected and a failure notification is sent. ▪ Option b: the payment remains in the queue until the cut-off for the respective payment type is reached (or the payment is revoked).

In case a payment with a “latest debit time indicator” is not executed 15 minutes prior to the defined time, an automatic notification in the GUI is triggered. The notification is directly displayed on top of all screens of the participant whose account is debited. Further details are provided in the RTGS User Handbook.

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

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

The defined execution time of a payment can be changed if the payment is not executed yet. Further details on the effect of changing the settlement time can be found in the chapter Amendment of payments [▸ 76].

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

fact that the original “earliest debit time indicator” had been reached and it was already tried to settle this payment.

5.1.4 Warehouse functionality (completed)

Basics

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

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

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

Rules

The validation of warehoused payments is a layered approach:

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

No checks are made by the RTGS component in the time between.

Processing on settlement day

On the indicated settlement day with the start of the day trade phase the warehoused payments are processed by the RTGS component. These payments are processed with an entry timestamp identical to the start of day time settlement phase and on top of the queue of incoming payments which have the same priority. They are immediately settled if enough liquidity is available (normal processing of payments in the entry disposition, see chapter Entry disposition [▸ 87]). Otherwise they are queued until the settlement attempt is successful (see chapter Dissolution of the payment queue [▸ 95]).

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

Information and control functions

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

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

5.1.5 Backup payments (completed)

In case an RTGS participant's own technical application has an outage, this participant might not be in a position to send payments to or receive payments from the RTGS component.

Such breakdown on the side of the RTGS participant may result in

1. pay-in obligations in other systems like CLS cannot be met
2. liquidity being built up on the affected RTGS participant's dedicated cash account in case other RTGS participants submitted or continue to submit payment orders in favour of the affected participant

In order to give the affected participant a possibility to reduce the business impact of the outage, the RTGS component offers a functionality to generate payments by using the so-called backup payment functionality.

In general, this functionality is blocked and it can only be used, once the responsible central bank has authorised the affected RTGS participant upon his request to use this functionality.

There are two categories of backup payments available:

1. backup contingency payments to CLS/EURO1
2. backup liquidity redistribution payments to other RTGS participants

If need be, the central bank responsible for the affected RTGS participant can act on behalf. Further details are provided in the RTGS User Handbook.

5.1.5.1 Backup contingency payments (completed)

Objective

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

Rules for CLS payments

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

Table 17 - CLS backup payments

Payment priority	Urgent
Generation	via the GUI
Message type	pacs.009
Sender of this message	RTGS
Receiver of this message	CLS
Fields for input via GUI	
Fields predefined (cannot be changed)	
Tag in the payment message	
Tag in the statement message	
Tag in the A2A payment queue	
Tag in the U2A payment queue	Backup payment

Rules for backup contingency payments to EURO1 collateral account

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

Table 18 - EURO1 collateral account backup payments

Payment priority	High
Generation	via the GUI
Message type	pacs.009
Sender of this message	RTGS
Receiver of this message	EBA (for collateral account)
Fields for input via GUI	
Fields predefined (cannot be changed)	
Tag in the payment message	

Tag in the statement message	
Tag in the A2A payment queue	
Tag in the U2A payment queue	Backup payment

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

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

Table 19 - EURO1 liquidity bridge backup payment

Payment priority	High
Generation	via the GUI
Message type	pacs.009
Sender of this message	RTGS
Receiver of this message	EBA (for pre-settlement account)
Fields for input via GUI	
Fields predefined (cannot be changed)	
Tag in the payment message	
Tag in the statement message	
Tag in the A2A payment queue	
Tag in the U2A payment queue	Backup payment

5.1.5.2 Backup liquidity redistribution payments (completed)

Objective

Backup liquidity redistribution payments are intended to redistribute excess liquidity accumulated on the RTGS dedicated cash account of the affected RTGS participant. It aims at mitigating the possibility of a shortage of liquidity within the RTGS component.

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

Rules for backup liquidity redistribution payments

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

Table 20 - Backup liquidity redistribution payments

Redistributing liquidity payments can be transferred to...	RTGS participants
Payment priority	High
Generation	via the GUI
Message type	pacs.009
Sender of this message	RTGS DN
Receiver of this message	According to the routing configuration of the instructed agent
Fields for input via GUI	
Fields predefined (cannot be changed)	
Tag in the payment message	
Tag in the statement message	
Tag in the A2A payment queue	
Tag in the U2A payment queue	Backup payment

5.1.5.3 Rules for backup payments

5.1.5.3.1 Generation (completed)

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

Table 21 - General procedure for generating backup payments

Step	Action
1	Information to the central bank responsible for the affected RTGS participant. Result: The central bank activates the backup functionality in the GUI for the RTGS participant concerned.
2	GUI users from the affected RTGS participant have to re-login to the GUI before being able to open the backup functionality. Generation of backup contingency and backup liquidity redistribution payments in the GUI by users from the affected RTGS participant or by the central bank acting on behalf of affected RTGS participant.

Further information on the GUI interactions can be found in the RTGS User Handbook.

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

- the generation of backup payments must first be activated by the central bank responsible for the RTGS participant facing technical problems (i.e. affected RTGS participant),
- the number of people authorised to generate these payments, can be kept small (separate role in the GUI),
- the “four eyes“ principle (different people responsible for initial recording and release) is obligatory,
- as far as possible, backup payments are generated automatically in the RTGS component.

5.1.5.3.2 Notification of affected participant (sender) (completed)

On request, the affected RTGS participant as sender of a backup contingency or backup liquidity redistribution payment receives a notification (BankToCustomerDebitCreditNotification (camt.054) [▶ 349]). Such notification includes the code word BACP. The debit notification reaches it as soon as its connection is operational again.

5.1.5.3.3 Notification to the receiver (completed)

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

5.1.5.3.4 Subsequent delivery of single payments (completed)

Basic principles

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

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

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

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

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

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

Transmission of unprocessed payments with new settlement date

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

Transmission of unprocessed payments with original settlement date

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

- The affected RTGS participant must request the temporary lifting of the settlement date check to the central bank which switches off the settlement date check for the current business day.

- If more than the current business day is required for dealing with the unprocessed payments with old settlement date, the lifting of the settlement date check for any consecutive business day has to be requested separately at the beginning of the concerned day trade phases.
- Once having completed the sending of payments with original (past) settlement date, the affected RTGS participant should inform the central bank in order to reactivate the settlement date check with immediate effect.

Note: Not all counterpart RTGS participants may be in the position to process payments with a settlement date in the past.

Account statement sorted by settlement date

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

5.1.6 Payment priorities (completed)

In general, all payment orders submitted to the RTGS component are settled immediately, provided that sufficient liquidity is available on the RTGS dedicated cash account of the RTGS participant and other relevant conditions (e.g. limits) are met.

To settle payment orders in the RTGS component considering their individual urgency, they can be submitted by the sender using one of the following priorities:

- urgent
- high
- normal

All priorities have specific characteristics.

Some of the priorities can only be used by certain users. Within a priority no further priorisation is possible (no sub-priorities). That means “urgent payments” are settled following the principles of entry disposition and execution of offsetting payments (see chapter Settlement of payments in the entry disposition [▸ 89]).

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

The priority class “urgent” is only available for

- ancillary systems;
- CLS pay-ins sent by a RTGS participant;

- inter-service liquidity transfers generated in the RTGS component to transfer liquidity to another service/component;
- automated inter-service liquidity transfers generated in CLM to “pull” liquidity from the RTGS component.

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

Further details on changing the priority of a payment can be found in chapter Comprehensive queue management [▶ 91].

5.2 Payments processing and settlement of payments

5.2.1 Overview (completed)

The aim of the process is to allow an RTGS participant to initiate a customer or an inter-bank payment to another RTGS participant. A customer or inter-bank payment can be submitted to and received from the RTGS component by

- the owner of the account to be debited
- the owner of the account to be credited (in case of direct debits)
- a third party (e.g. in case of mandated payments sent by a central bank or an ancillary system)
- a central bank acting on behalf of a credit institution

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

The so-called multi addressee access implies that an entity is authorised to submit and receive payments directly without having an own RTGS dedicated cash account.

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

Table 22 - Features to be used for different payment messages

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

The RTGS dedicated cash account to be debited and credited are not necessarily linked to the BICs mentioned in the business application header. They have to be taken from the respective payment order (pacs or camt). After simultaneous booking on the RTGS dedicated cash accounts, the payment/the liquidity transfer is final and irrevocable.

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

5.2.2 Concept of payment submitters (to be completed in iteration 4)

5.2.3 Flow of payment related messages (completed)

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

Note: In order to ease the readability pacs.009COV is not mentioned separately, but included in the description for pacs.009.

Case 1: payment credit message with positive validation and settlement

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

Message flow

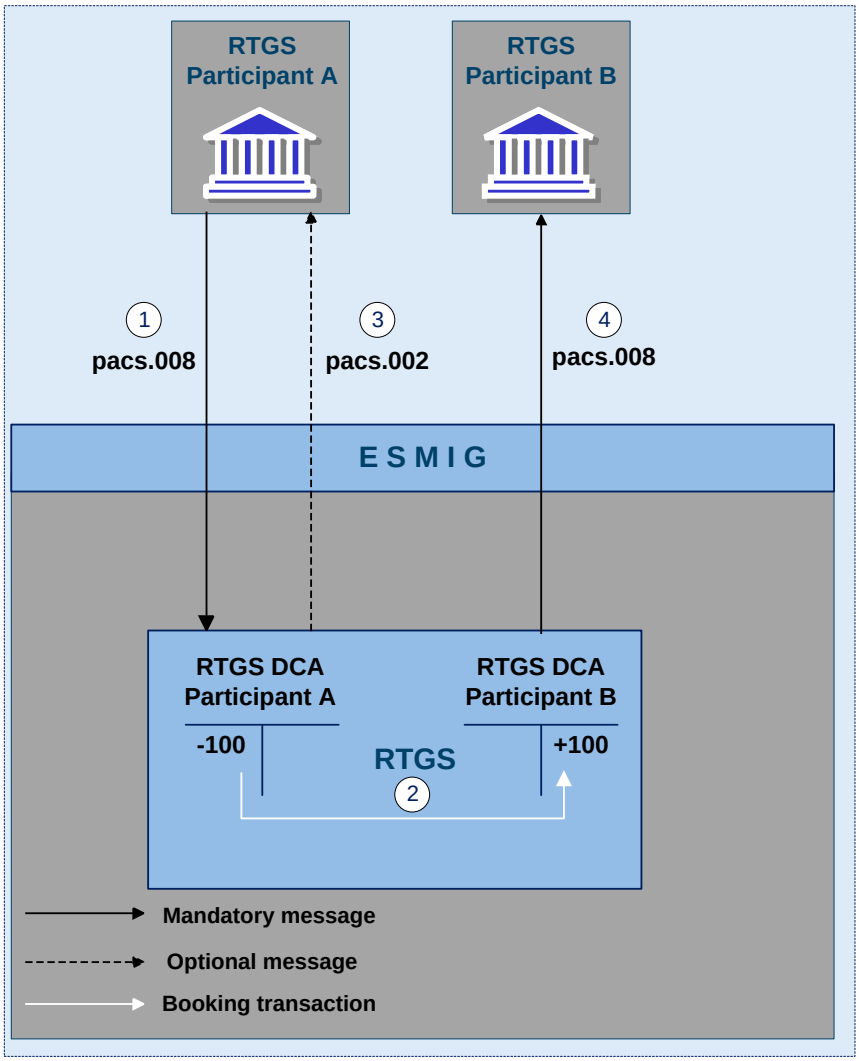


Figure 3 - pacs.008 – CustomerCreditTransfer

Process description

Table 23 - Payment messaging on the basis of pacs.008

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

Used messages

- CustomerCreditTransfer (pacs.008) [▸ 373]
- PaymentStatusReport (pacs.002) [▸ 365]

Message flow

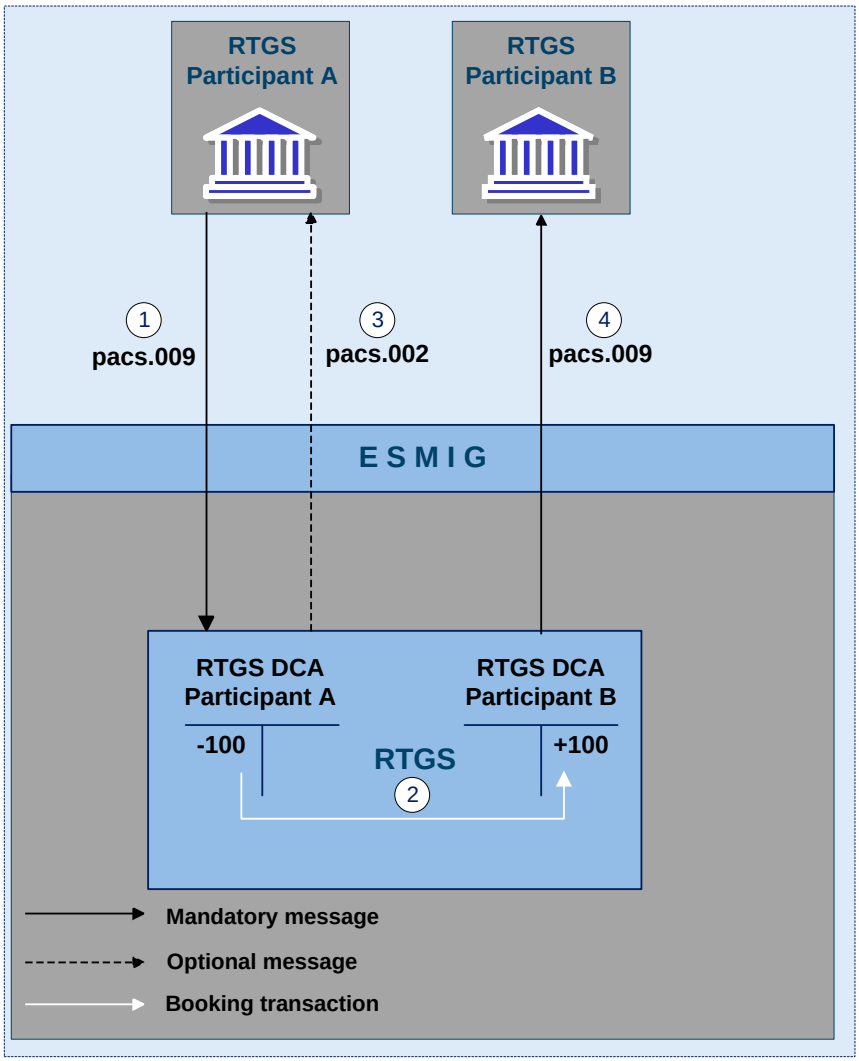


Figure 4 - pacs.009 - FinancialInstitutionCreditTransfer

Process description

Table 24 - Payment messaging on the basis of pacs.009

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

Used messages

- FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 383]
- PaymentStatusReport (pacs.002) [▸ 365]

Case 2: payment return message with positive validation and settlement

The following payment flow illustrates the payment messaging on basis of a pacs.004 and with regard to the RTGS component.

Message flow

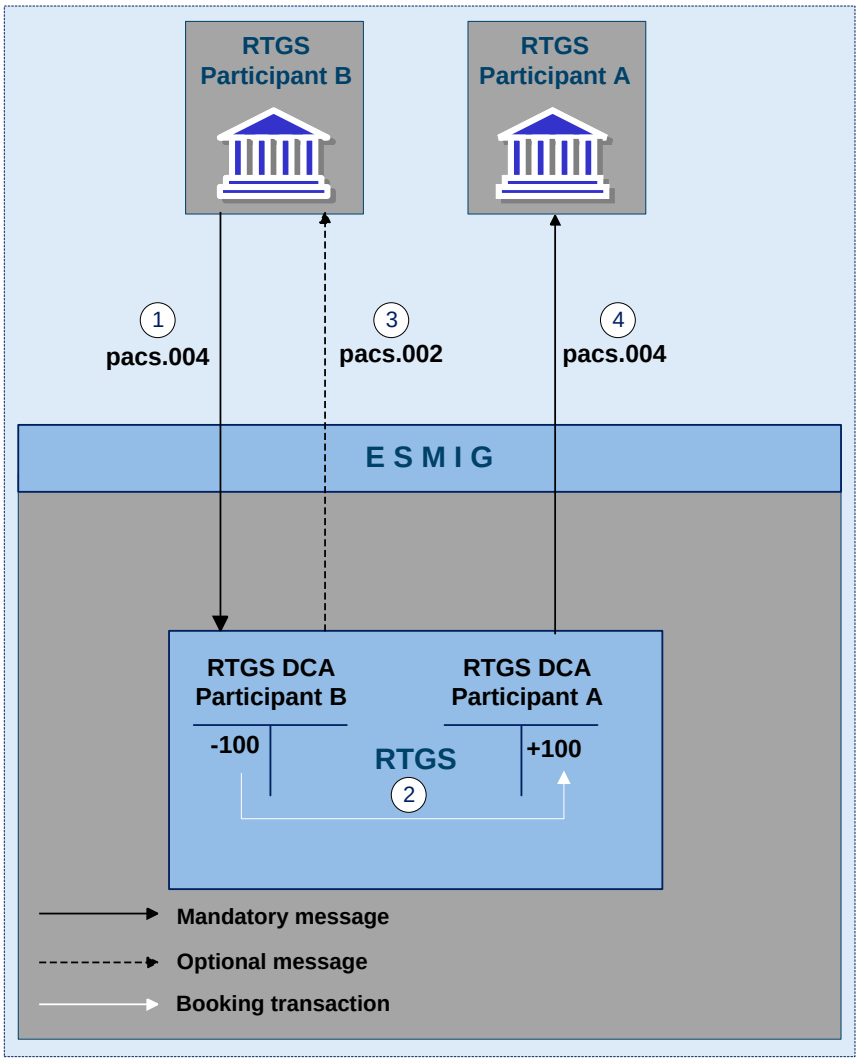


Figure 5 - pacs.004 - PaymentReturn

Process description

Table 25 - Payment messaging on the basis of pacs.004

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

Used messages

- PaymentReturn (pacs.004) [▸ 369]
- PaymentStatusReport (pacs.002) [▸ 365]

Case 3: payment debit message with positive validation and settlement

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

Message flow

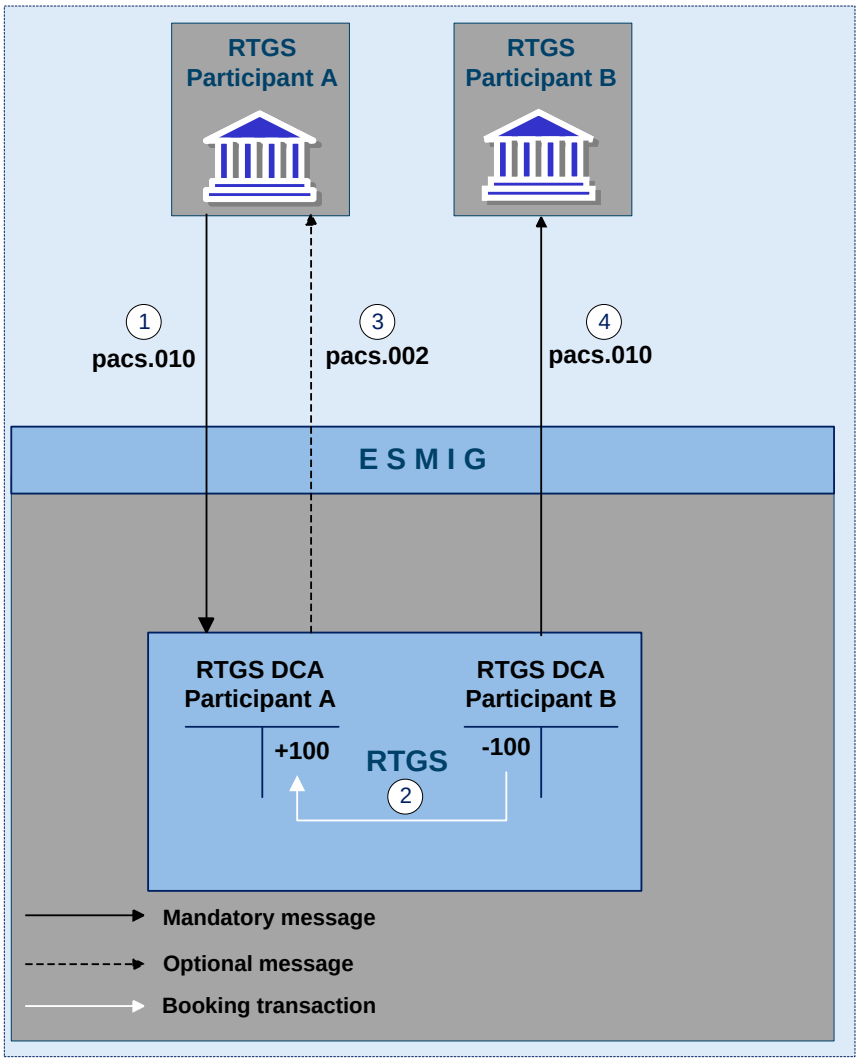


Figure 6 - pacs.010 - FinancialInstitutionDirectDebit

Process description

Table 26 - Payment messaging on the basis of pacs.010

Step	Processing in/between	Description
1	RTGS participant A via ESMIG to the RTGS component	RTGS participant A sends a pacs.010 via ESMIG to the RTGS component.
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS dedicated cash accounts of RTGS participants A and B
3	RTGS component via ESMIG to RTGS participant A	Creation and forwarding of pacs.002 by the RTGS component (optional) via ESMIG to RTGS participant A
4	RTGS component via ESMIG to RTGS participant B	Creation and forwarding of pacs.010 by the RTGS component via ESMIG to RTGS participant B (mandatory)

Used messages

- FinancialInstitutionDirectDebit (pacs.010) [▸ 402]
- PaymentStatusReport (pacs.002) [▸ 365]

5.2.4 Rejection of payments (completed)

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

In case

- the technical validation in the RTGS component fails, the RTGS component creates and forwards a notification (ReceiptAcknowledgement (admi.007) [▸ 276]) to the submitter of the payment order.
- the business validation in the RTGS component fails, the RTGS component creates and forwards a rejection notification (PaymentStatusReport (pacs.002) [▸ 365]) to the submitter of the payment order. The pacs.002 refers to the original instruction by means of references and a set of elements from the original instruction.

Note: The sending of a negative notifications is mandatory and not subject to message subscription.

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

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

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

Further information on the relevant business rules and the respective error codes are listed in chapter Index of business rules and error codes [420].

5.2.4.1 Technical validations (completed)

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

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

Message flow

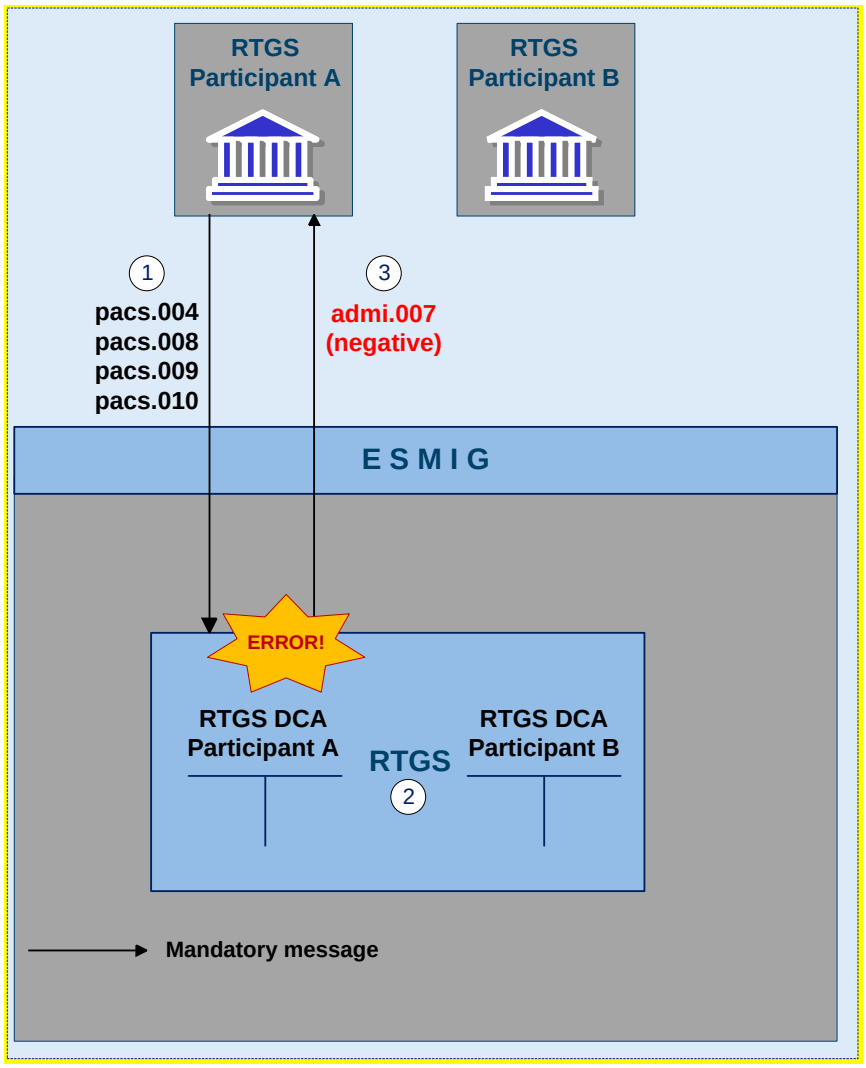


Figure 7 - pacs.008/009/010/004 technical validation error

Process description

Table 27 - Technical validation failure

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

Used messages

- CustomerCreditTransfer (pacs.008) [373]
- FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [383]
- FinancialInstitutionDirectDebit (pacs.010) [402]
- PaymentReturn (pacs.004) [369]
- ReceiptAcknowledgement (admi.007) [276]

5.2.4.2 Business validations (completed)

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

Note: The RTGS component performs various checks during the business validation and does not stop after the first negative validation result, but goes on with the business validation as there could be further negative valid action results in the subsequent checks.

Message flow

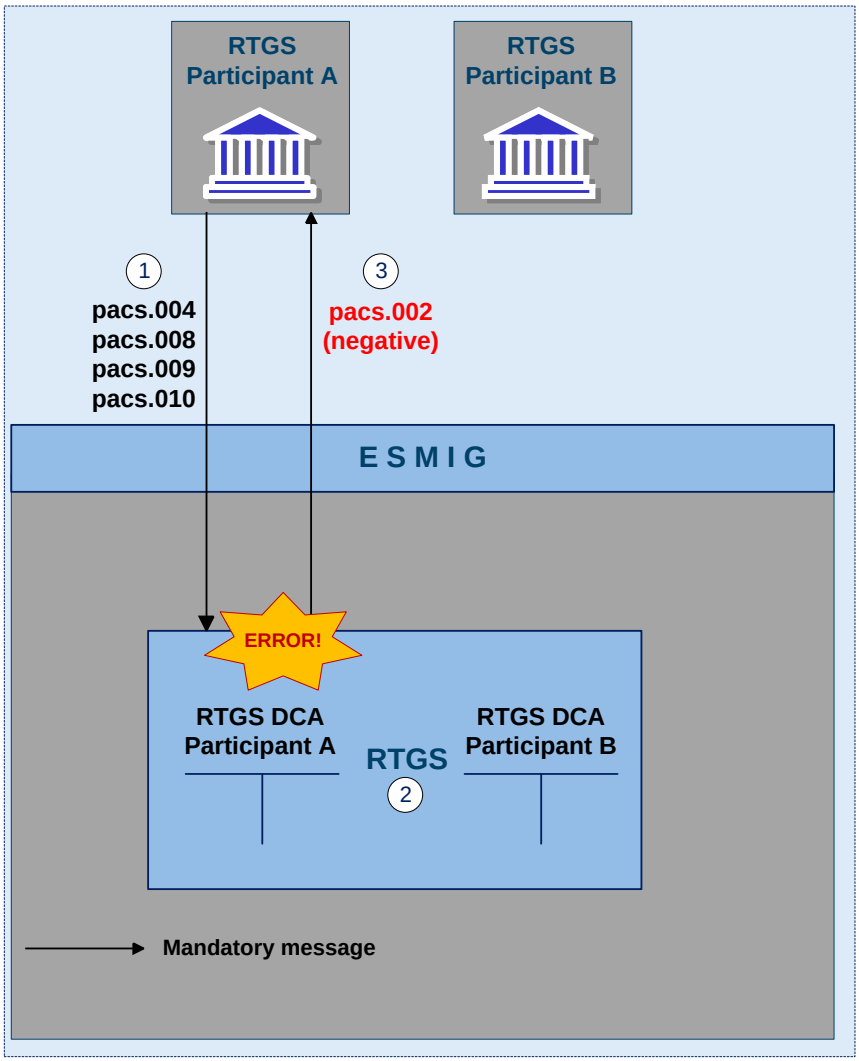


Figure 8 - pacs.008/009/010/004 business validation error

Process description

Table 28 - Business validation failure

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

Used messages

- CustomerCreditTransfer (pacs.008) [▸ 373]
- FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 383]
- FinancialInstitutionDirectDebit (pacs.010) [▸ 402]
- PaymentReturn (pacs.004) [▸ 369]
- PaymentStatusReport (pacs.002) [▸ 365]

5.2.5 Amendment of payments (completed)

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

Various control options are offered.

Table 29 - Options for changing the parameters of payments

Action	RTGS participant
Change priority	RTGS participant to be debited
Re-ordering (increase / decrease)	RTGS participant to be debited
Change of set execution time (if defined before sending to the RTGS component)	RTGS participant sending the payment

These features are necessary to enable RTGS participants to react on changed liquidity conditions during the day. The consequences for the settlement of the affected payments can be found in chapter Comprehensive queue management [▸ 91].

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

The following rules apply in principle:

- Interventions must be made via the business interface of the RTGS component in U2A and A2A. A description of individual U2A processes can be found in the RTGS User Handbook.
- Several payment orders together can be modified at the same time.
- The business interface shows receipt and execution or non-execution of a modified order.

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

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

Message flow

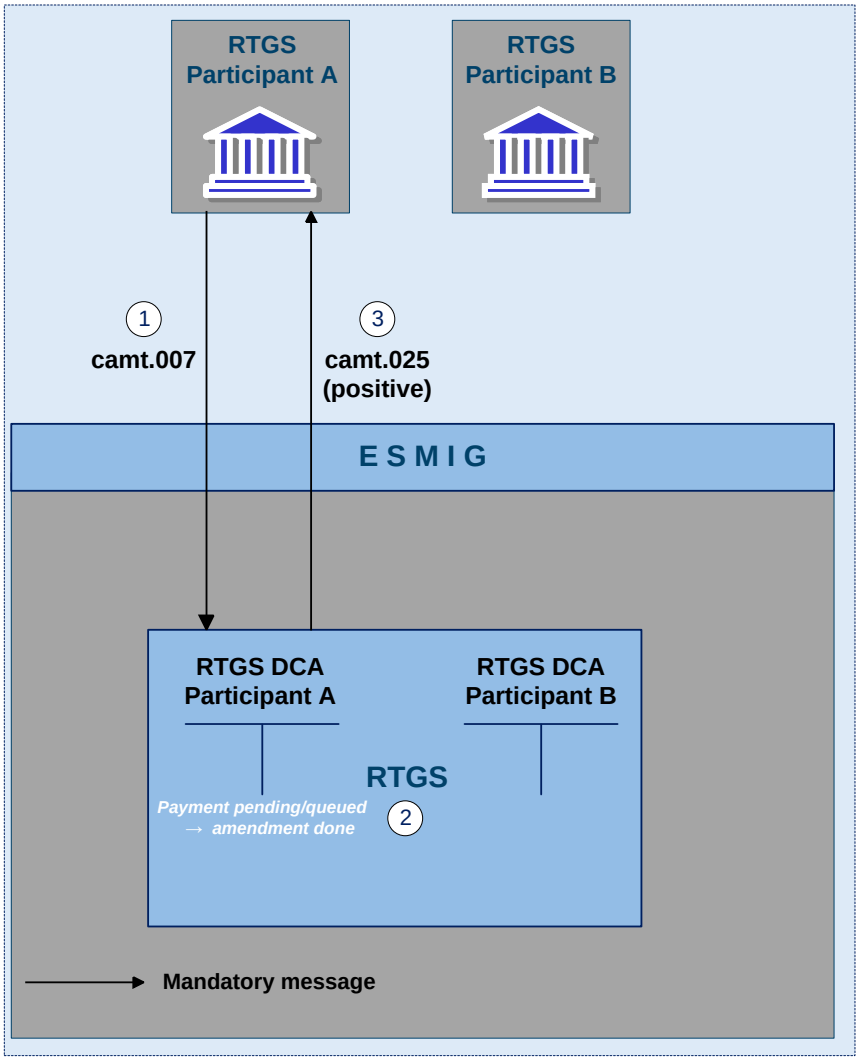


Figure 9 - camt.007 amendment of payment (positive)

Process description

Table 30 - Amendment of payments

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

Used messages

- ModifyTransaction (camt.007) [▸ 299]
- Receipt (camt.025) [▸ 318]

Case 1: changing priority

The following options for changing the priority exist:

- from normal to high
- from high to normal

It is not possible to change an urgent priority.

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

The modified payment

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

Table 31 - Effects of changing the priority

Action	Effect
Change of the first queued high payment into a normal payment	▪ If no urgent payment is queued immediate attempt to settle the remaining high payments following the FIFO-principle. ▪ If urgent payments are queued no immediate attempt to settle any high payments.
Change of a normal payment into a high payment	▪ If the payment changed from normal to high moves to the top of the queued high payments and no urgent payments are queued, immediate attempt to settle high payments following the FIFO-principle. ▪ Otherwise, no immediate attempt to settle high payments.

Case 2: re-ordering the queued payments

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

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

The re-ordering can be done at any time during the day time settlement phase. The updated payment can be checked by querying the payment queue.

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

Table 32 - Effects of re-ordering the queued payments

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

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

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

However, it is not possible for an RTGS participant to re-order queued liquidity transfers. The only liquidity transfers in the RTGS component which can be queued are automated inter-service liquidity transfers from CLM due to pending central bank operations in case only partial settlement was possible. In case an automated inter-service liquidity transfer from CLM due to pending central bank operations is queued, it always stays at the top of the urgent queue and no re-ordering is possible. Further details can be found in chapter Liquidity transfer [▸ 141].

Case 3: changing the execution time

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

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

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

Table 33 - Effects of changing the execution time

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

5.2.6 Revocation of payments (completed)

Revocation of a queued payment

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

A revocation of a payment is only possible as long as the payment is not settled on the RTGS dedicated cash account. It is also possible to revoke warehoused payments. A successful processing of the PaymentCancellationRequest results in the revocation of the payment (see case 1). As soon as the payment is finally settled, the RTGS component forwards the PaymentCancellationRequest for further processing to the receiving RTGS participant (see case 2).

The revocation can be done in the RTGS component in U2A or A2A. A description of individual U2A processes can be found in the RTGS User Handbook.

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

- pacs.008
- pacs.009/pacs.009COV
- pacs.010

For each payment submitted a dedicated PaymentCancellationRequest (FIToFIPaymentCancellationRequest (camt.056) [▸ 355]) needs to be sent. In case of a direct debit, the RTGS participant to be credited can send the PaymentCancellationRequest.

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

Case 1: Successful revocation of a queued payment

Message flow

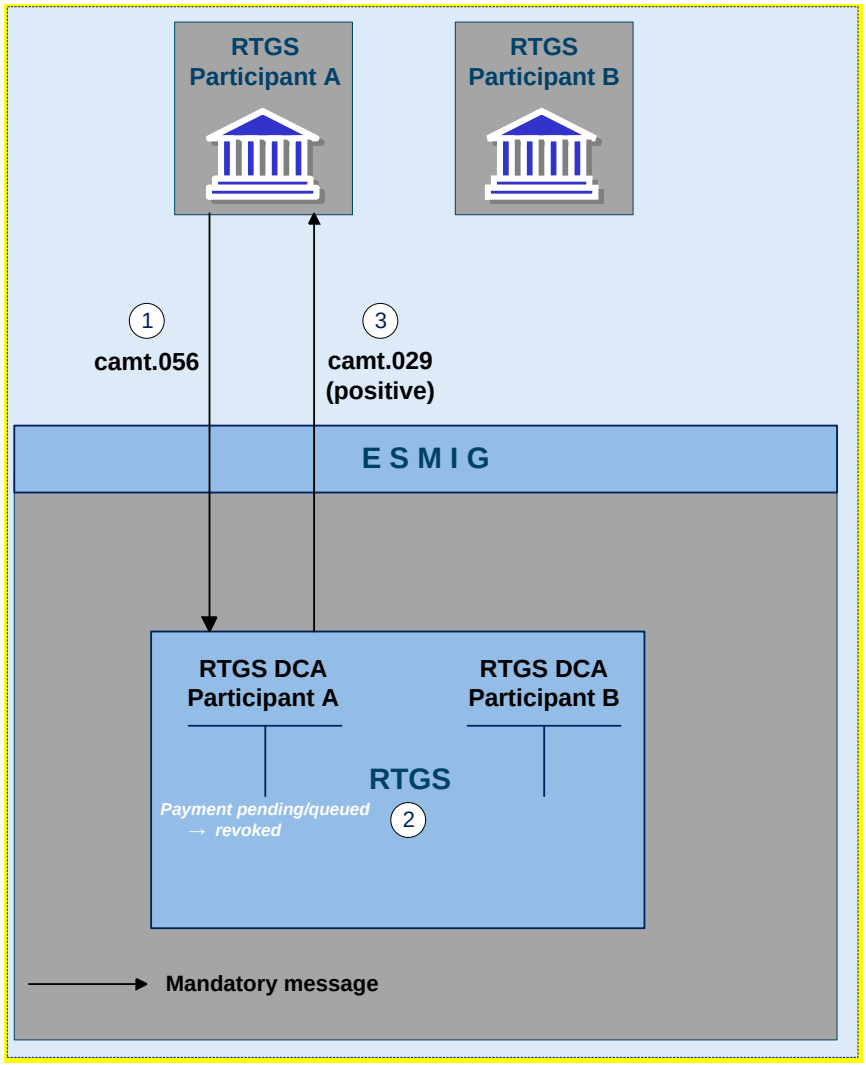


Figure 10 - camt.056 revocation of payment (positive)

Process description

Table 34 - Successful revocation of a queued payment

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

Used messages

- FITOFIPaymentCancellationRequest (camt.056) [▸ 355]
- ResolutionOfInvestigation (camt.029) [▸ 320]

Case 2: Cancellation request for already settled payments

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

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

In case the RTGS participant B sends

- a negative reply, this negative reply is forwarded to the RTGS participant A who sent the PaymentCancellationRequest;
- a payment return message, this payment order triggers the booking on the RTGS dedicated cash accounts involved and which – after successful settlement - is sent to the RTGS participant A.

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

In case the PaymentCancellationRequest is sent to revoke a pacs.010 which is already in a final status, the RTGS does not forward the PaymentCancellationRequest to the receiving RTGS participant B. In such case, the RTGS participant A is informed about the final status of the payment and the fact that the revocation is not possible.

Message flow

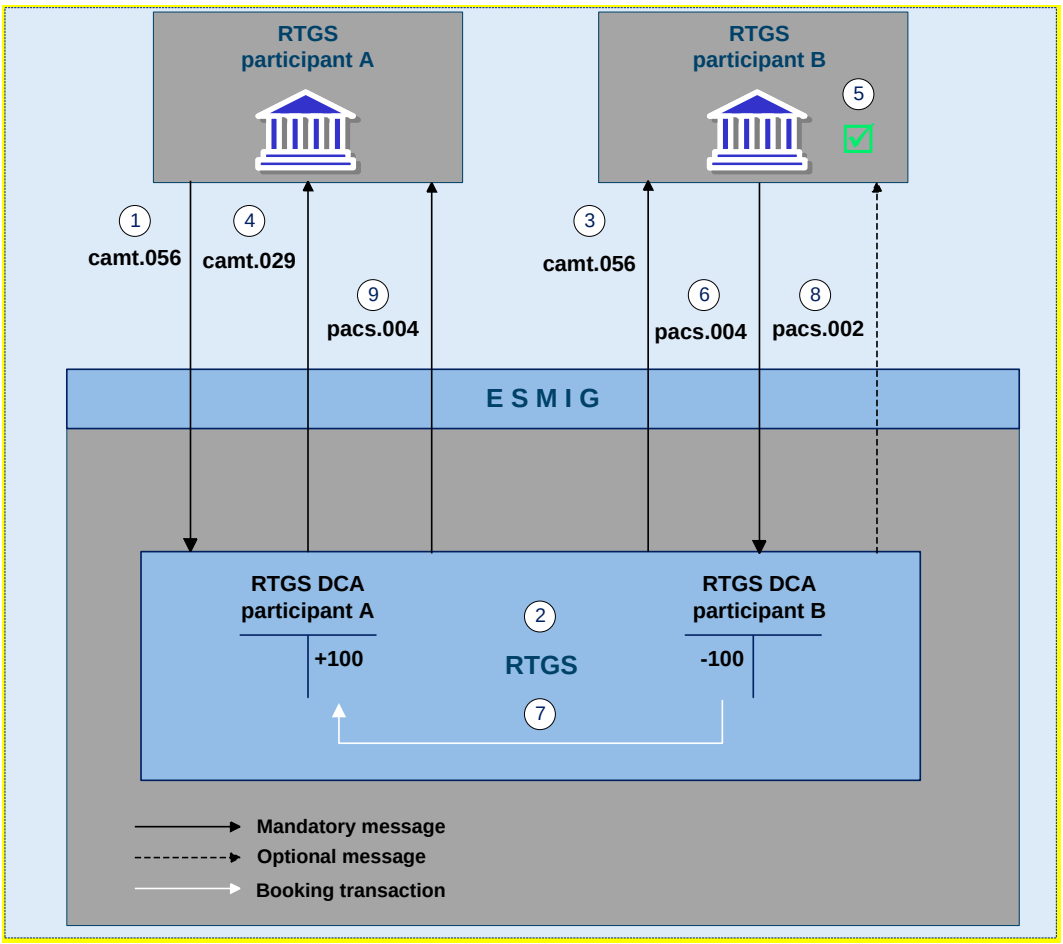


Figure 11 - camt.056 FIToFIPaymentCancellationRequest/camt.029 ResolutionOfInvestigation - positive case

Process description

Table 35 - Cancellation request for already settled payments – positive case

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

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

Used messages

- FIToFIPaymentCancellationRequest (camt.056) [▸ 355]
- ResolutionOfInvestigation (camt.029) [▸ 320]
- PaymentStatusReport (pacs.002) [▸ 365]
- PaymentReturn (pacs.004) [▸ 369]

Message flow

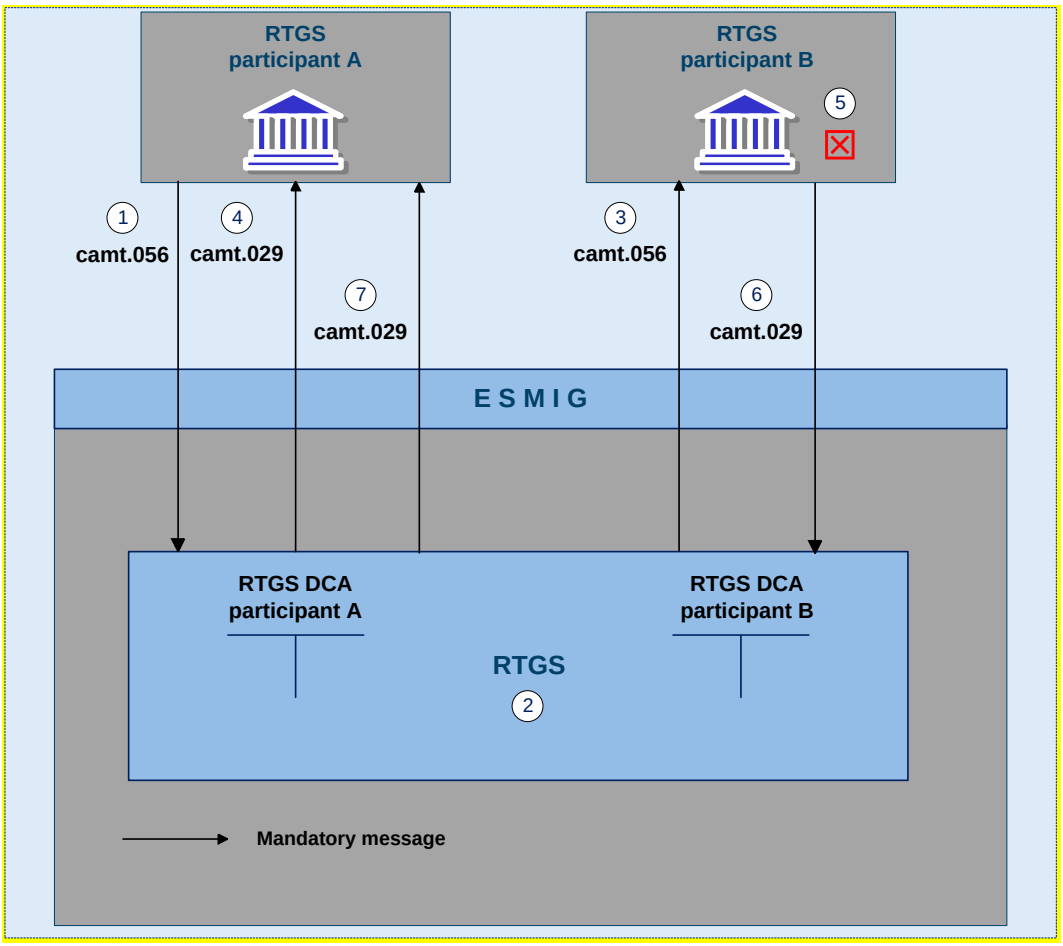


Figure 12 - camt.056 FIToFIPaymentCancellationRequest / camt.029 ResolutionOfInvestigation - negative case

Process description

Table 36 - Cancellation request for already settled payments – negative case

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

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

Used messages

- FIToFIPaymentCancellationRequest (camt.056) [▸ 355]
- ResolutionOfInvestigation (camt.029) [▸ 320]

5.2.7 Processing of payments

5.2.7.1 Entry disposition

5.2.7.1.1 General remarks (completed)

Basics

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

These features may inter alia

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

Moreover, liquidity management tools for RTGS participants allow for achieving a flexible and need-based control of payment flows, thereby limiting possible liquidity risks. The features are implemented in the RTGS component on a flexible and optional basis. This is to allow each RTGS participant to meet its individual needs, i.e. each RTGS participant can individually decide whether to use certain tools or not.

Objective for settlement of payment orders

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

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

Influencing factors

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

- balance on the RTGS dedicated cash account
- defined limits
- used priority
- order of submitted payment orders
- opposing payments and synchronisation of submitted payments
- Defined execution time

Basic principles

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

- Every payment order should be marked as “normal“, “high “ or “urgent“. If no priority class is selected, payments will be handled as normal payments.
- Attempt to settle single or group of payment orders immediately after their submission, with the exception of payment orders with a defined earliest debit time indicator (FromTime). In case a FromTime is defined, these payment orders are included in the settlement process from the time indicated as earliest debit time.
- Offsetting payments are used to save liquidity (bilateral optimisation mechanism).
- Payment orders to be settled are simultaneously booked on the RTGS dedicated cash account linked to the submitter (debit: camt.050 and pacs.004/008/009/009COV; credit: pacs.010) and the counterparty RTGS dedicated cash account in the RTGS component (credit: camt.050 and pacs.004/008/009/009COV; debit: pacs.010).
- Only payment orders which are not yet executed (i.e. queued) may be revoked.
- Queuing of payment orders which cannot be settled immediately, according to their priority in different queues (urgent queue, high queue, normal queue).
- In case of an automated inter-service liquidity transfer stemming from CLM due to pending central bank opertations which was only partially executed in the RTGS component, an inter-service liquidity transfer with the remaining amount is placed on top of the urgent queue in the RTGS component. This is the only scenario in which inter-service liquidity transfers are queued in the RTGS component.

Note: This automated inter-service liquidity transfer which aims at transferring liquidity from the RTGS component to CLM will be put on top of the urgent queue.

- Continuous attempt to settle payment orders in the queues.
- The entry disposition and the optimisation procedures for queues can run at the same time.

5.2.7.1.2 Settlement of payments in the entry disposition (completed)

For urgent payment orders the FIFO-principle applies.

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

For high payments the FIFO-principle applies, too.

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

Normal payment orders are processed according to the "FIFO by-passing" principle. In order to save as much liquidity as possible, the FIFO-principle would not be the optimal one; i.e. normal payment orders submitted may be executed even if other normal payment orders are still in the queue (provided that the balance on the RTGS dedicated cash account is sufficient).

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

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

Table 37 - Payment orders taken into account in the entry disposition

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

Unsuccessful entry disposition

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

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

Detailed sequence of settlement checks

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

If the urgent and high queue are **not** empty, a bilateral offsetting check with potential liquidity increase takes place. This offsetting check is only successful if offsetting payment orders from the RTGS dedicated cash account to be credited are available and the RTGS dedicated cash account to be debited with the payment order afterwards has an increased liquidity position. If offsetting payment orders exist, it is checked if the submitted payment order fulfils the other settlement criteria (i.e. bilateral/multilateral Limits [▸ 159] and liquidity reservations not breached). If no such offsetting payment orders exist, the payment order is put in the queue.

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

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

If the other settlement criteria (i.e. bilateral/multilateral limit and liquidity reservations not breached) are fulfilled, then the operation(s) is (are) settled on the RTGS dedicated cash account (i.e. debit as well as credit booking on the respective RTGS dedicated cash accounts take place). If the other settlement criteria are not fulfilled, then the payment order(s) is (are) put in the queue until sufficient liquidity is available and the other settlement criteria are fulfilled (details on the dissolution of the queues are given in chapter Dissolution of the payment queue [▸ 95]).

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

Note: In case of direct debits, the RTGS participant sending the payment order expects a liquidity increase on its RTGS dedicated cash account and the RTGS dedicated cash account of the receiver is debited.

Rejection during end-of-day processing

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

5.2.7.2 Comprehensive queue management (completed)

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

As long as a payment is not settled, the RTGS participant has the ability to change the relevant parameters of the payment. Further details on amending payment orders can be found in chapter Amendment of payments [▸ 76].

Note: Depending on the configuration chosen by the RTGS participant, in case of pending urgent or high payments an inter-service liquidity transfers might be triggered in order to transfer liquidity from the linked MCA to the RTGS dedicated cash account. Further details on such inter-service liquidity transfers can be found in chapter Liquidity transfer [▸ 141].

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

Table 38 - Control options for comprehensive queue management

Action	RTGS participant
Change priority Exception 1: It is not possible to change the priority of urgent payment orders Exception 2: In case of pacs.010 the receiver (i.e. the debtor) has the ability to change the priority	RTGS participant to be debited
Re-ordering (increase / decrease) Exception: in case of pacs.010 the receiver (i.e. the debtor) has the ability to re-order)	RTGS participant to be debited
Change of set execution time (if defined before sending to the RTGS component)	RTGS participant sending the payment
Revocation (Revocation of payments [81])	RTGS participant sending the payment

These control options enable an RTGS participant to react on changed liquidity conditions during the day. It is possible to modify a single payment order or several payment orders at the same time. In case it is not possible to execute a modification the RTGS participant is notified accordingly. Amendments are possible in A2A as well as in U2A via the GUI.

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

Changing the priority of a payment order

Table 39 - Possibilities for changing priorities

Priority of a payment order		
Urgent	High	Normal
	→	←

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

In case of such change, the payment

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

- is processed according to the rules of the (new) priority.

Table 40 - Effect of changed priority

Action	Effect
Change of the first queued high payment into a normal payment	- If no urgent payment order is queued immediate attempt to settle the remaining high payments following the FIFO principle - If urgent payments are queued no immediate attempt to settle any high payment
Change of a normal payment into a high payment	- If the payment changed from normal to high moves to the top of the queued high payments and no urgent payment orders are queued, immediate attempt to settle high payments following the FIFO principle - Otherwise, no immediate attempt to settle urgent payments

Re-ordering of queued payment orders

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

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

Table 41 - Effect of changing the order of queued payment orders

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

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

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

Changing the defined execution time

In principle, RTGS participants can submit payments with a defined execution time. It is possible to include an earliest debit time indicator and/or a latest debit time indicator (see chapter Definition of execution time [▸ 52]).
In case a submitted payment includes an earliest debit time indicator and/or a latest debit time indicator it is possible to change the earliest debit time indicator and/or the latest debit time indicator via A2A or U2A. Such a change has no impact on the payment processing, but on the queue management as the time indication only support the queue management of the RTGS participant.

Table 42 - Effect of changing the execution time

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

Revocation of a queued payment order

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

Details on the revocation via A2A using a PaymentCancellationRequest (camt.056) can be found in chapter Revocation of payments [▸ 81].

5.2.7.3 Dissolution of the payment queue

5.2.7.3.1 Settlement of queued urgent/high payments (completed)

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

Table 43 - Possible events for queue resolution

Events	by ...
Liquidity increase	▪ incoming settled payment (i.e. credits) ▪ incoming settled intra–service liquidity transfers ▪ incoming inter-service liquidity transfer from other services/components (i.e. credits)
Intervention on queue level	▪ If the payment order on the top of the urgent/high queue is changed (change of order, change of priority, revocation)

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

Continuously resolving of the queue

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

Optimisation for the processing on sub-accounts

For optimisation of the processing of urgent ancillary system payment instructions on the sub-accounts of settlement banks a special algorithm is used. It can be seen as an exception of the below described algorithms for the settlement of queued normal payments. Further details on the settlement of ancillary system payment instructions can be found in chapter Settlement of ancillary systems [▸ 103].

5.2.7.3.2 Settlement of queued normal payments (completed)

Principles

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

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

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

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

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

Sequence of algorithms

During the business day the algorithms run sequentially,

- while there is no pending simultaneous multilateral settlement of an ancillary system (see chapter Simultaneous multilateral settlement [▸ 112]):
  - first algorithm “partial optimisation” then algorithm “multiple optimization”...
  - if algorithm “partial optimisation” succeeds then two algorithm schedule options are in place, i.e. either algorithm “multiple optimization” runs always after algorithm “partial optimisation” or algorithm “partial optimisation” runs again.
  - changes of the algorithm schedule lie within the sole responsibility of the operator in order to be able to react in a flexible way to changed liquidity conditions.
- while there is a pending simultaneous multilateral settlement of an ancillary system:
  - algorithm “partial optimisation with ancillary system”

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

Consequences of a running algorithm

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

Algorithm: “Partial optimisation”

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

Table 44 - Main characteristics of algorithm “partial optimisation”

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

For the retaining of transactions the following rules apply.

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

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

Algorithm: “Multiple optimisation”

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

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

Part 1

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

Table 45 - Main characteristics of algorithm "multiple optimisation" – Part 1

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

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

Part 2

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

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

Table 46 - Main characteristics of algorithm "multiple optimisation" – Part 2

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

Algorithm: Partial optimisation with ancillary system

Algorithm “partial optimisation with ancillary system” is developed to support the simultaneous multilateral settlement of ancillary system (see chapter Simultaneous multilateral settlement [▸ 112]). It ensures an efficient and fast processing of the related ancillary system payment instructions. In order to smoothen the settlement and to reduce the overall liquidity needed, other “urgent“ payments as well as “high“ and “normal“ ones are also included.

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

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

Table 47 - Main characteristics of algorithm “partial optimisation with ancillary system”

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

Inclusion of all pending payments:

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

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

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

Ordering of ancillary system payment instructions in the queue

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

- by their priority (urgent, high, normal);

- within the priority following
  - the time they have entered the RTGS component (FIFO principle);
  - their earliest debit time - if defined (exception 1);
  - the time of the start of the settlement period (exception 2 - only for ancillary system payment instructions (see chapter Settlement of ancillary systems [▸ 103]).

Several ancillary system involved in one running algorithm "partial optimisation with ancillary system"

In the same run of algorithm "partial optimisation with ancillary system" several ancillary system using ancillary system settlement procedure B (see chapter Simultaneous multilateral settlement [▸ 112]) is included if they intend to settle at the same time.

Settlement process in detail

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

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

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

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

Algorithm "partial optimisation with ancillary system" ends

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

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

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

Sequence of the various algorithms

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

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

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

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

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

5.2.7.3.3 Algorithm: “Optimisation on sub-accounts” (completed)

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

This algorithm aims at resolving ancillary system payment instructions using dedicated liquidity on sub-accounts. The algorithm only checks sub-accounts instead of RTGS dedicated cash accounts and only covered ancillary system payment instructions are settled. In case of uncovered ancillary system payment instructions, these ancillary system payment instructions are put back in the queue of the single sub-account. It runs only once a time until the next start by the RTGS component.

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

Table 48 - Main characteristics of algorithm "optimisation on sub-accounts"

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

5.2.7.4 Treatment of backup payments in the settlement process (completed)

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

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

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

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

If backup payments are in the queue for urgent (in case of CLS backup contingency payments) or high (in case of other backup contingency and backup liquidity redistribution payments) payments, they are treated in the RTGS component as any other payment order. As a consequence, revocation (see chapter Revocation of payments [▸ 81]) as well as queue management (see chapter Comprehensive queue management [▸ 91]) is possible.

5.3 Settlement of ancillary systems (partially completed)

5.3.1 Overview (partially completed)

To settle ancillary system related payment instruction in central bank money the needed functionalities are offered in the RTGS component. These allow the ancillary systems to have i) a broader accessibility of participants and ii) a broad range of streamlined functionalities.

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

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

Types of ancillary systems are:

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

Settlement procedures

The settlement of ancillary system payment instructions takes place in different settlement procedures. The table below is a breakdown of the settlement procedures. Details of the procedures can be found in the following chapters.

Table 49 - Settlement procedures

Procedure	Description
Standard multilateral settlement	Ancillary system sends simultaneously debits and credits. All debits have to be booked before credits are settled.
Simultaneous multilateral settlement	Ancillary system sends simultaneously debits and credits to the RTGS component. All debits and credits are simultaneously checked for settlement. If this check is passed all debits and credits are booked simultaneously.
Settlement on dedicated liquidity accounts (real-time)	Settlement bank can dedicate a liquidity amount to settle balances coming from a specific ancillary system. The dedication is achieved by setting aside the needed liquidity on the dedicated liquidity account. Such a settlement procedure can be used for mandatory procedure only.
Settlement on dedicated liquidity accounts (interfaced)	Settlement bank can dedicate a liquidity amount to settle balances coming from a specific ancillary system. The dedication is achieved by setting aside the needed liquidity on a specific sub-account. Such a settlement procedure can be used for mandatory and optional procedure.

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

Account types for ancillary systems

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

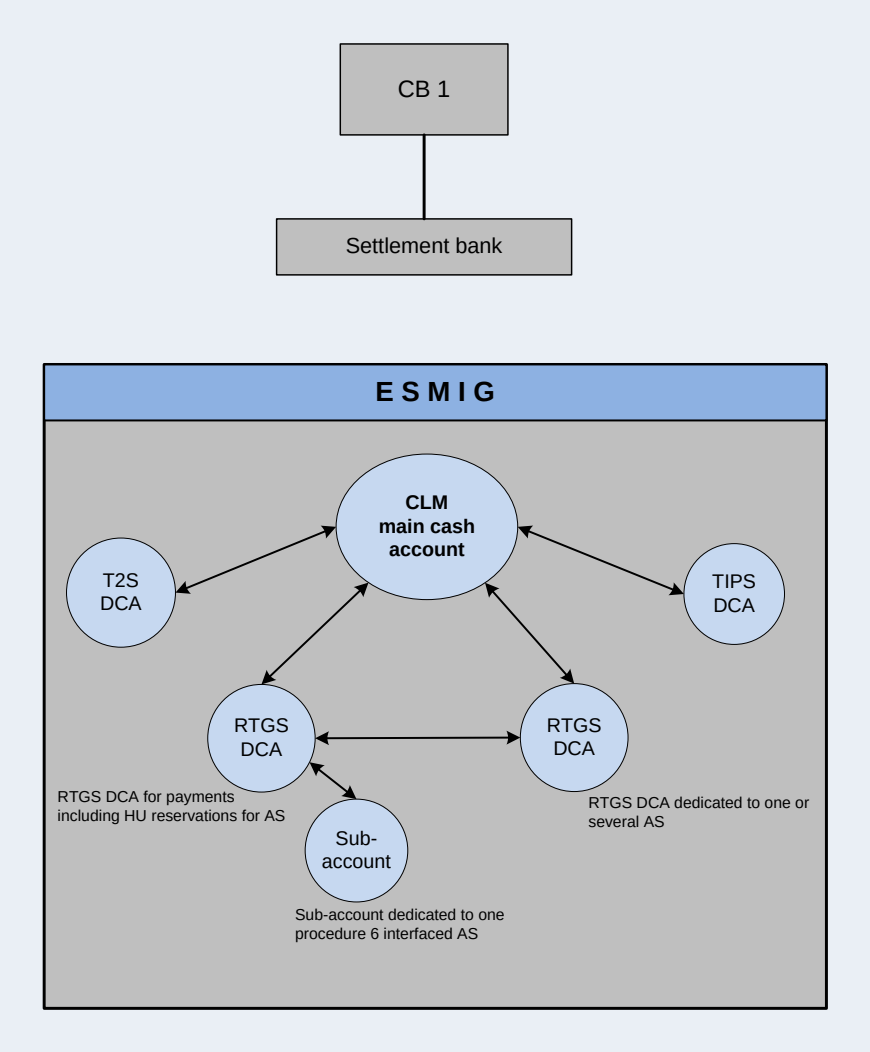


Figure 13 - Generic account constellation for an ancillary system participant

Besides the dedicated cash accounts for securities (i.e. T2S dedicated cash accounts) and instant payments settlement in central bank money (i.e. TIPS dedicated cash account) , the settlement bank in the example above has an RTGS dedicated cash account for high value payments (with a reserved amount for urgent ancillary system payment instructions) and two further accounts for ancillary system payment instructions: one account for ancillary system procedure "Settlement on dedicated liquidity accounts (interfaced)" as a sub-account of the RTGS dedicated cash account for high value payments and the second account (for other ancillary systems) as an RTGS dedicated cash account dedicated to one or several ancillary systems.

Table 50 - Account types and their ownership

Account type	Account holder	Description	Procedure
RTGS dedicated cash account	Settlement bank	Used as an RTGS dedicated cash account for the settlement of ancillary system	Can be used in all procedures except for the settlement on dedicated liquidity account (interfaced)
RTGS dedicated cash account dedicated to ancillary system	Settlement bank	Used as an RTGS dedicated cash account specifically for the settlement of one or several ancillary system	Can be used in all procedures except for the settlement on dedicated liquidity account (interfaced)
Sub-account	Settlement bank	Used to set aside liquidity for exclusive settlement of a specific ancillary system and needs to be mapped to the RTGS dedicated cash account	Settlement on dedicated liquidity account (interfaced) only

Account type	Account holder	Description	Procedure
Dedicated liquidity account	Ancillary system or the central bank of the ancillary system	Used to transfer liquidity from a settlement bank's RTGS dedicated cash account to the dedicated liquidity account. The ancillary system reflects this liquidity to the settlements bank's account held within the ancillary system.	Settlement on dedicated liquidity accounts (real-time) only
Guarantee funds account	Guarantor, central bank or the ancillary system	Used in case the optional guarantee mechanism has to be activated by an ancillary system or the central bank on its behalf. The same guarantee account can be used for both procedures; it is also possible to use two different ones	Standard multilateral settlement and simultaneous multilateral settlement
Technical account	Ancillary system or the central bank of the ancillary system	Used as intermediary account for the collection of debits and credits resulting from the settlement of ancillary system payment instructions	**Optional** - individual payment instructions sent by ancillary system **Mandatory** - standard multilateral settlement - simultaneous multilateral settlement - Settlement on dedicated liquidity account (interfaced) In case multilateral settlement is used (both, standard and simultaneous) a specific technical account is needed for each of both procedures.

Liquidity used for settlement of ancillary system transactions

The necessary liquidity used for settlement originates from different accounts. Sources of liquidity and liquidity transfer types are described in chapter Dedication of liquidity for ancillary system settlement [▸ 166].

Monitoring of ancillary system settlement

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

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

- number and amount of ancillary system payment instructions related to ancillary system settlement
- transactions queued because of lack of liquidity
- uncovered transactions shortly before a settlement period ends
- rejected, revoked or reversed ancillary system payment instructions and
- booked ancillary system payment instructions

5.3.2 Standard multilateral settlement (partially completed)

Basics

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

The RTGS component will be responsible to settle first all debits and, only once all debits are successfully executed, to execute also all credits at once. The identification of debit or credit payment instructions is made on the basis of the ancillary system technical account. Whenever a Settlement Bank's RTGS dedicated cash account is debited and the ancillary system technical account is credited, the transaction is considered as a debit whilst debiting the ancillary system technical account and crediting the Settlement Bank's RTGS dedicated cash account will be considered a credit. In turn, for the standard multilateral settlement the usage of the ancillary system technical account is mandatory. Due to the peculiarities of the settlement, i.e. in order to ensure that after the settlement of debits the needed amount is present on the technical account and not used for other purposes in the framework ancillary system payments processing, a dedicated ancillary system technical account for standard multilateral settlement is to be used. Additionally, the sum of all debits must be equal to the sum of credits within one message.

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

Optional connected mechanisms

The standard multilateral settlement may include optional connected mechanisms:

- information period
- settlement period (“till”)
- guarantee fund mechanism

For further details on the usage and functionalities offered by the optional connected mechanisms please refer to chapter Optional connected mechanisms [▸ 133]

Used messages

- Proprietary messages (ASTransferInitiation, ASInitiationStatus)
- BankToCustomerDebitCreditNotification (camt.054) [▸ 349]
- Receipt (camt.025) [▸ 318]
- ReceiptAcknowledgement (admi.007) [▸ 276]

Process description

The standard multilateral settlement consists of the following steps

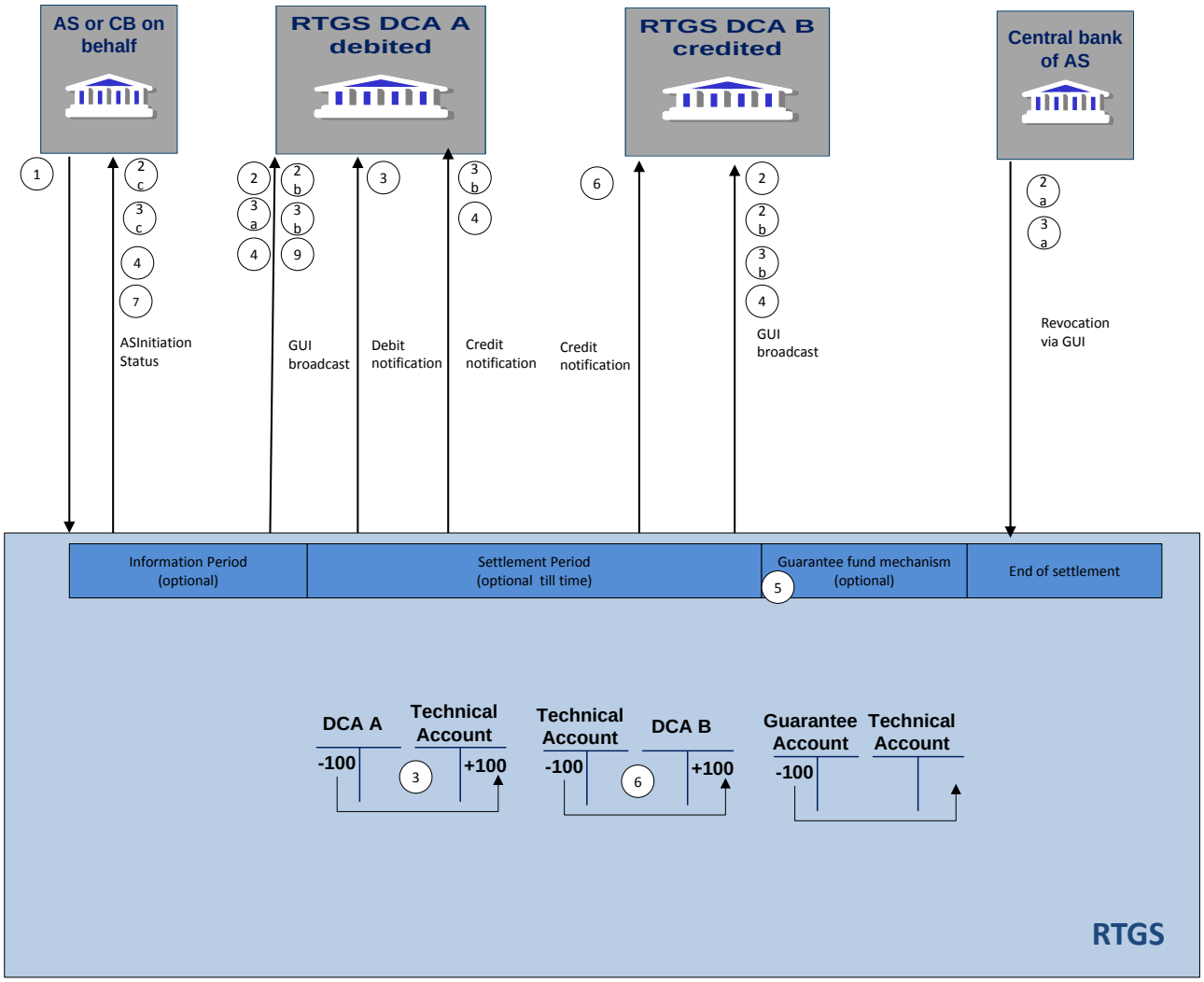


Figure 14 - Flow standard multilateral settlement

Table 51 - Process flow for standard multilateral settlement

Phase	Step	Processing in/between	Description
Initiation	1	Ancillary system via ESMIG to RTGS	The ancillary system (or the relevant Central Bank on its behalf) sends a message (ASTransferInitiation) with all multilateral balances to be debited and credited on the settlement banks' RTGS dedicated cash accounts
Information period	2	RTGS	If the optional "Information period" is used, the involved settlement banks receive via GUI the broadcast notification on the start of the information period. If no settlement bank disagrees (the suitable communication means has to be agreed within the contractual relationship with the ancillary system) during the information period the processing will continue
	2a	RTGS	If a settlement bank disagrees, no settlement is triggered. The relevant central bank will revoke the full batch via GUI.

Phase	Step	Processing in/between	Description
	2b	RTGS via ESMIG to settlement banks	After disagreement all involved settlement banks are informed via GUI broadcast about failure of settlement due to disagreement
	2c	RTGS via ESMIG to ancillary system	The ancillary system is informed about the settlement failure due to disagreement via ASInitiationStatus message. Processing stops
Settlement of debit positions	3	RTGS	Debits are processed for settlement. Once they have all been settled, the credits will be processed immediately after. The settlement takes place with debiting the related settlement banks' RTGS dedicated cash accounts and crediting the ancillary system technical account. Each debit ancillary system payment instruction is checked against the liquidity available in the related settlement banks' RTGS dedicated cash accounts. If the liquidity covers the needed amount, the ancillary system payment instruction is booked. The settlement banks receive a debit notification after successful execution of their debit. If liquidity is not sufficient the ancillary system transaction is posted in queue.
	3a	RTGS	The settlement banks are informed about queuing by a broadcast message delivered in U2A mode. Immediately after putting the group of debit transactions in the queue the optimisation process starts (settlement algorithms). Pending ancillary system payment instructions are settled by resolving the queue. The Central Bank of the ancillary system is allowed to revoke the batch of ancillary system payment instructions as long as they are not final.
	3b	RTGS via ESMIG to settlement banks	A broadcast is sent to all the involved settlement banks informing about the settlement failure due to revocation. Already settled ancillary system payment instructions will be reversed and a credit notification is sent to the previously debited settlement banks if subscribed.
	3c	RTGS via ESMIG to ancillary system	The ancillary system is informed about the settlement failure due to revocation via ASInitiationStatus message.

Phase	Step	Processing in/between	Description
	4	RTGS	If the ancillary system (or the relevant central bank on its behalf) has indicated a “Settlement period” time, RTGS - if related ancillary system payment instructions are still pending - continuously checks whether the time limit is reached. If the time limit is exceeded, and guarantee fund mechanism is not set up, the settlement fails and the whole batch of ancillary system payment instructions is rejected. Consequently RTGS will trigger the reversing procedure. The ancillary system technical account has to be debited and the settlement banks’ RTGS dedicated cash accounts credited (only for those ancillary system payment instructions which were settled during the interrupted settlement cycle). The Ancillary System is notified on the Settlement Failure with ASInitiationStatus, the settlement banks receive a broadcast informing on the failed settlement
	5	RTGS	If the time limit is exceeded and the guarantee fund mechanism is set up, it can be activated according to the agreed procedures. For details such as the involved messages and notifications please refer to chapter Optional connected mechanisms [▸ 133]
Settlement of credit positions	6	RTGS	RTGS processes all credits. The settlement banks are informed via a credit notification on an optional basis.
End of settlement	7	RTGS via ESMIG to ancillary system	After all ancillary system payment instructions have been booked the ancillary system (or the relevant central bank on its behalf) receives a notification (ASInitiationStatus), confirming the settlement of the entire set of ancillary system payment instructions .

At each step throughout the process information for settlement banks and ancillary systems is available, please refer to chapter Overview [▸ 103].

5.3.3 Simultaneous multilateral settlement (partially completed)

Basics

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

The RTGS component will be responsible settle all debits and credits received in such a set of ancillary system payment instructions simultaneously. The identification of debit or credit ancillary system payment instructions is made on the basis of the ancillary system technical account. Whenever a settlement bank’s

RTGS dedicated cash account is debited and the ancillary system technical account is credited, the ancillary system payment instruction is considered as a debit whilst debiting the ancillary system technical account and crediting the settlement bank's RTGS dedicated cash account will be considered a credit. Additionally the sum of all debits must be equal to the sum of credits within one message. The usage of the ancillary system technical is thus mandatory (i.e. each ancillary system payment instruction will have to present the ancillary system technical account on either debit or credit side).

In order to achieve the simultaneous execution of debits and credits, the simultaneous multilateral settlement benefits from the usage of a dedicated settlement algorithm (please refer to chapter Dissolution of the payment queue [▸ 95]). During the optimization algorithm, RTGS checks that there is sufficient liquidity to settle all debit and credit ancillary system payment instructions of an ancillary system simultaneously ("All or nothing"). If this check is successfully passed, all debit and credit ancillary system payment instructions are booked simultaneously. If the check fails, all linked ancillary system payment instructions remain in the queue and the partial optimization with ancillary system optimisation algorithm is triggered again.

In order to limit the negative impact of failed settlement, the ancillary system can make use of the guarantee fund mechanism. Due to the above mentioned optimisation, prior to the optional running of the guarantee fund mechanism it is needed to single out the failed payment orders. This is achieved by transforming all ancillary system payment instructions from simultaneous multilateral settlement into standard multilateral settlement and executing those debits covered by the needed liquidity. This fallback implies also that the ancillary system technical account used for simultaneous multilateral settlement is solely used for simultaneous multilateral settlement but not by any other procedure in the framework of ancillary system processing.

Optional connected mechanisms

The standard multilateral settlement may include optional connected mechanisms:

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

For further details on the usage and functionalities offered by the optional connected mechanisms please refer to chapter Optional connected mechanisms [▸ 133].

Used messages

- Proprietary messages (ASTransferInitiation, ASInitiationStatus)
- BankToCustomerDebitCreditNotification (camt.054) [▸ 349]
- Receipt (camt.025) [▸ 318]
- ReceiptAcknowledgement (admi.007) [▸ 276]

Process description

The simultaneous multilateral settlement consists of the following steps

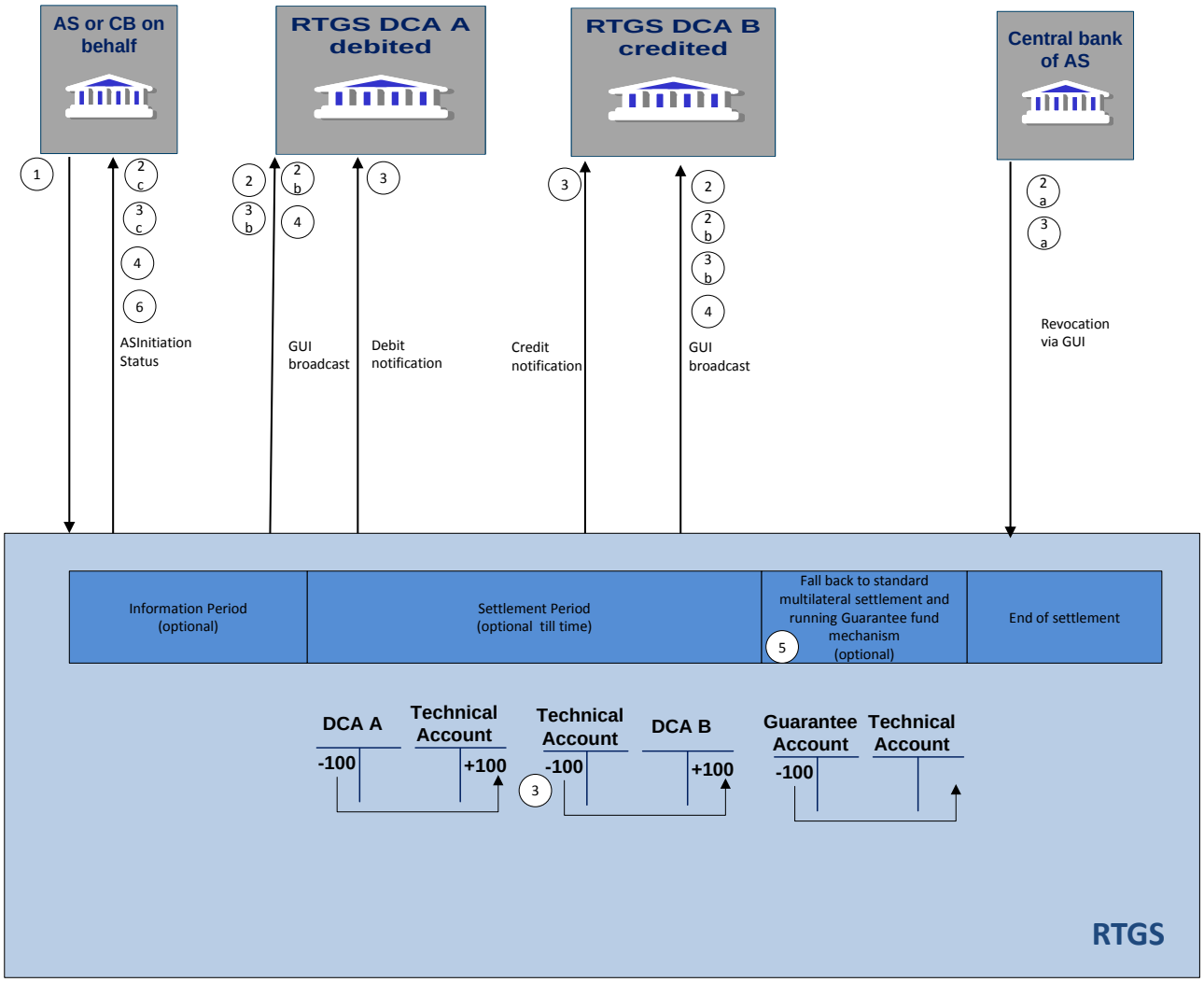


Figure 15 - Flow simultaneous multilateral settlement

Table 52 - Process flow for standard multilateral settlement

Phase	Step	Processing in/between	Description
Initiation	1	Ancillary system via ESMIG to RTGS	The ancillary system (or the relevant central bank on its behalf) sends a message (ASTransferInitiation) with all multilateral balances to be debited and credited on the settlement banks accounts.
Information period	2	RTGS	If the optional “Information period” is used, the involved settlement banks receive via GUI the broadcast notification on the start of the information period. If no settlement banks disagrees (the suitable communication means has to be agreed within the contractual relationship with the ancillary system) during the information period the processing will continue.

Phase	Step	Processing in/between	Description
	2a	RTGS	If a settlement bank disagrees, no settlement is triggered. The relevant central bank will revoke the full set of ancillary system payment instructions via GUI.
	2b	RTGS via ESMIG to settlement banks	After disagreement all involved settlement banks are informed via GUI broadcast about failure of settlement due to disagreement.
	2c	RTGS via ESMIG to ancillary system	The ancillary system is informed about the settlement failure due to disagreement via ASInitiationStatus message. Processing stops.
Settlement	3	RTGS	In case no revocation due to disagreement applies, debits and credits are processed simultaneously for settlement using the optimisation algorithm. RTGS checks that there is sufficient liquidity to settle all debit and credit ancillary system payment instructions of an ancillary system simultaneously. If this check is successfully passed, all debit and credit ancillary system payment instructions are booked simultaneously. If the check fails, all linked ancillary system payment instructions remain in the queue and the partial optimisation with ancillary system algorithm is triggered again. Via GUI it is possible to single out the RTGS dedicated cash accounts not having enough liquidity.
	3a	RTGS	The central bank of the ancillary system is allowed to revoke the batch of ancillary system payment instructions as long as they are not final.
	3b	RTGS via ESMIG to settlement banks	A broadcast is sent to all the involved settlement banks informing about the settlement failure due to revocation.
	3c	RTGS via ESMIG to ancillary system	The ancillary system is informed about the settlement failure due to revocation via ASInitiationStatus message.
	4	RTGS	If the ancillary system (or the relevant Central Bank on its behalf) has indicated a “Settlement period (“till”)”, RTGS - if related payments are still unsettled - continuously checks whether the time limit is reached. If the time limit is exceeded, and guarantee fund mechanism is not set up, the settlement fails and the whole batch of instructions is rejected. The ancillary system is notified on the settlement failure with ASInitiationStatus, the settlement banks receive a GUI broadcast informing on the failed settlement.

Phase	Step	Processing in/between	Description
	5	RTGS	If the time limit is exceeded and the guarantee fund mechanism is set up, it can be activated according to the agreed procedures. For details such as the involved messages and notifications please refer to chapter Optional connected mechanisms [133]. In order to identify the ancillary system payment instructions not covered by the needed liquidity, all ancillary system payment instructions will be transferred into standard multilateral settlement and a single settlement attempt will be made (i.e. first all debits are executed, please refer to chapter Standard multilateral settlement [108]. Only afterwards the guarantee fund mechanism is started. In such a scenario, it has to be kept in mind that the execution of debits and credits will not be simultaneously anymore. This behavior also implies that, in case the guarantee mechanism ends unsuccessfully (i.e. an error within the ancillary systems guarantee procedures), a reversal of the already booked debits is to be executed.
End of settlement	6	RTGS via ESMIG to ancillary system	After all ancillary system payment instructions have been booked the ancillary system (or the relevant central bank on its behalf) receives a notification (ASInitiationStatus), confirming the settlement of the entire set of ancillary system payment instructions.

At each step throughout the process information for settlement banks and ancillary systems is available, please refer to chapter Overview [103].

5.3.4 Settlement on dedicated liquidity accounts (partially completed)

Basics

Ancillary systems which run settlement procedures based on the confidence of liquidity “fixed” amount can benefit from a pre-funding function that allows settlement banks to set aside the needed liquidity in one or more separate sub-accounts dedicated to a specific ancillary system (interface) or the dedicated liquidity account of a specific ancillary system (real-time).

The settlement on dedicated liquidity accounts (interfaced and real-time) must therefore be used to settle balances for an amount equal to or lower than the set aside liquidity.

Settlement on dedicated liquidity is a very convenient functionality to provide liquidity in batch-mode either for the interfaced settlement procedure or for the real-time settlement procedure. During the procedures, the liquidity adjustment mechanisms are the only opportunity to supply liquidity.

A settlement bank can use this settlement procedure with reference to several ancillary systems.

Accounting

Following accounts are suitable for the described procedures:

Table 53 - Accounting

Account type	Account holder	Description	Procedure
Sub-account	Settlement bank	Used to set aside liquidity for exclusive settlement of a specific ancillary system and needs to be mapped to the RTGS dedicated cash account.	Settlement on dedicated liquidity account (interfaced)
Technical account	Ancillary system or the central bank of the ancillary system	Used as intermediary account for the collection of debits and credits resulting from the settlement of ancillary system transactions.	Settlement on dedicated liquidity account (interfaced)
Dedicated liquidity account	Ancillary system or the central bank of the ancillary system	Used to transfer liquidity from a settlement bank's RTGS dedicated cash account to the dedicated liquidity account. The ancillary system reflects this liquidity to the settlements bank's account held within the ancillary system.	Settlement on dedicated liquidity accounts (real-time)

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

The settlement banks participating in ancillary systems using procedure "settlement on dedicated liquidity account (interfaced)" have to open one sub-account per ancillary system.

Procedures and cycles

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

Within a procedure several cycles for settlement can run consecutively. Before a cycle is started, a certain period of time for liquidity transfers between the RTGS and sub-accounts or between the RTGS and dedicated liquidity account shall be foreseen. The closing of the mandatory procedure (interfaced only) and the opening/closing of the optional procedure as well as the opening and closing of cycles can be done via messages or GUI screen (depending on the business day phase).

Set aside liquidity

To set aside liquidity for the settlement different options are offered by the RTGS. Please refer to chapter Liquidity management features [▸ 152].

5.3.4.1 Settlement on dedicated liquidity accounts (interfaced) (partially completed)

As mentioned above, the settlement on dedicated liquidity accounts (interfaced) is based on ancillary systems payment instructions initiated by ancillary systems between the settlement banks' sub-accounts and the technical account held by the ancillary systems. For credits only, also the RTGS dedicated cash account of the settlement bank can be addressed.

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

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

During the whole process, the ancillary system will be notified on the amounts available on the subaccounts. This happens whenever the liquidity on subaccounts changes (by standing orders or current orders) or by providing the result of the settlement instructed by the ancillary systems (i.e. ASInitiationStatus). Thus, the ancillary system is always in a position to know the liquidity set aside for their settlement. Settlement shall only be started once the liquidity needed is available on the sub-accounts. In turn, during the settlement cy-

cle only on an exceptional basis (i.e. an error on ancillary system side) ancillary system payment instructions should be pending on sub-accounts due to missing liquidity.

Liquidity provision

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

- setting up standing orders in reference data (to be executed with each start of procedure, for mandatory and optional settlement procedure different standing orders can be stored. Standing orders set up in reference data will only become effective with the next business day.
- sending camt.050 LiquidityCreditTransfer messages (current order liquidity transfer)
- using the dedicated RTGS GUI liquidity transfer screens (current order liquidity transfer)
- the ancillary system sending ASTransferInitiation messages debiting the settlement banks RTGS dedicated cash account and crediting the same settlement bank's sub-account (current order liquidity transfer)

Liquidity transfers will be executed in the following way:

- Standing order liquidity transfers are executed with each start of procedure (different amounts for each of the two procedures can be specified).
- Current order liquidity transfers will be executed during an open procedure (mandatory or optional settlement procedure). They will be executed with immediate effect during an open procedure with no cycle running. In the opposite case, where a cycle is running, the liquidity transfer will be stored and executed only once the cycle has closed.

Effects on liquidity transfers in case of missing liquidity

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

Table 54 - Amounts taken into account

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

Mandatory procedure

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

dedicated to the ancillary system, i.e. the balances present on the sub-accounts will be retransferred to the linked RTGS dedicated cash account.

Optional procedure

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

Overview on the settlement process

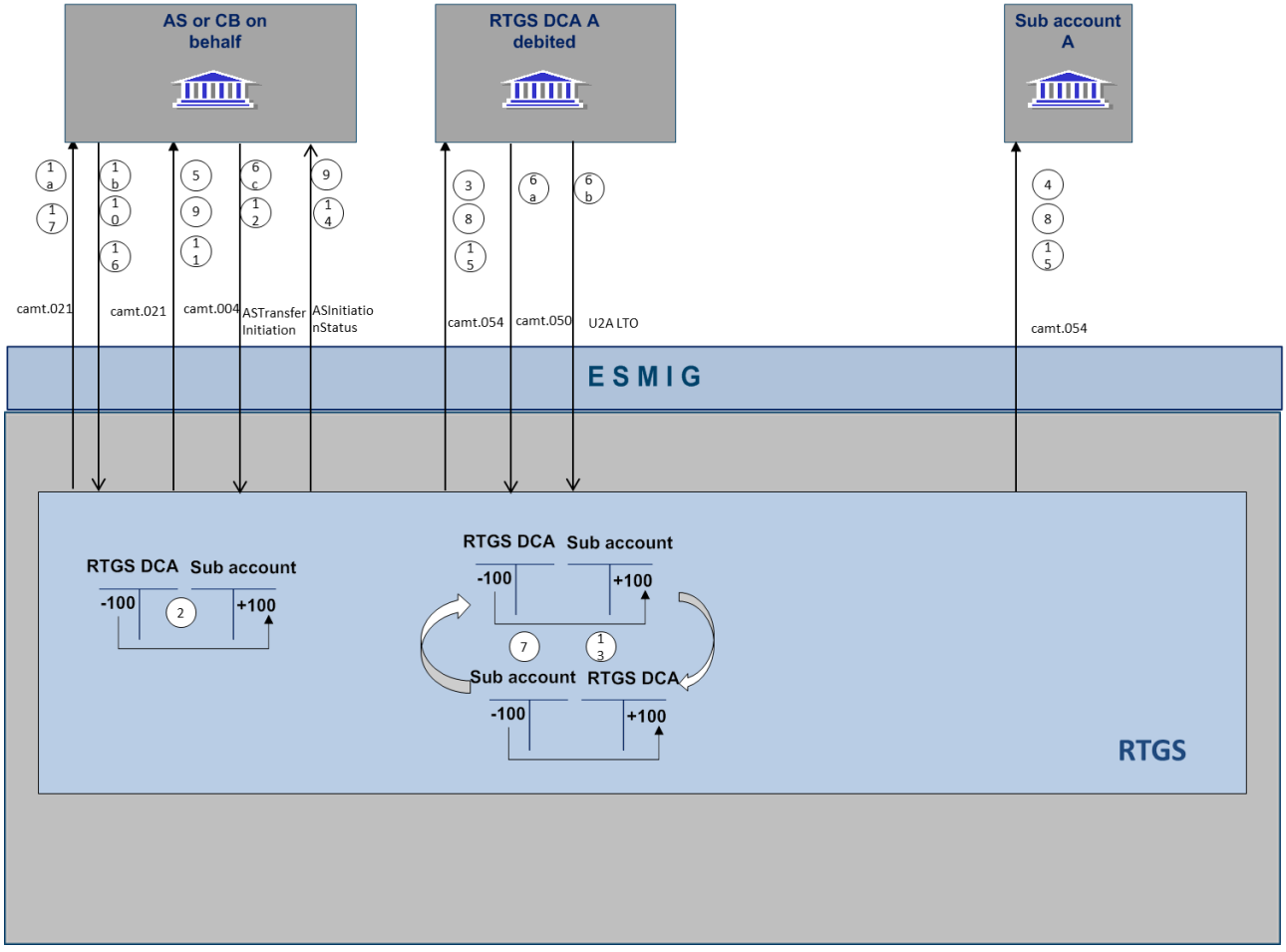


Table 55 - Start of procedure and liquidity provision

Phase	Step	Processing with	Description
Start of procedure	1a (mandatory procedure)	RTGS via ESMIG to ancillary system	Start of procedure message for mandatory procedure is automatically initiated by the RTGS component at 19:30h of new business day. The ancillary system will be notified on the event (camt.021-ReturnGeneralBusinessInformation).
	1b (optional procedure)	Ancillary system via ESMIG to RTGS	The ancillary system (or central bank on behalf) sends a message (camt.021–ReturnGeneralBusinessInformation) indicating the start of the optional procedure. The closure of the mandatory procedure prior to this is mandatory. The procedure can also be opened using U2A GUI screen.
Standing orders execution	2	RTGS	The start of procedure triggers the execution of existing standing orders debiting the settlement banks' RTGS dedicated cash accounts and crediting the pertaining sub-accounts. (please refer to table "amounts taken into account")
	3	RTGS via ESMIG to settlement banks	On an optional basis, the settlement banks are notified on the debited amount on the RTGS dedicated cash account (camt.054 Debit notification)
	4	RTGS via ESMIG to settlement banks	On an optional basis, the settlement banks are notified on the credited amount on the sub-account (camt.054 Credit notification)
	5	RTGS via ESMIG to ancillary system	The ancillary system is notified on credit of the sub-account for the amounts actually booked (camt.004-ReturnAccount)
Liquidity adjustment	6a	Settlement Banks via ESMIG to RTGS	Settlement banks can adjust (increase or decrease) the liquidity on the subaccounts by using current order liquidity transfers (camt.050).

Phase	Step	Processing with	Description
	6b	Settlement Banks via ESMIG to RTGS	Settlement banks can adjust (increase or decrease) the liquidity on the subaccounts by using current order liquidity transfers (dedicated U2A GUI screens).
	6c	Ancillary system via ESMIG to RTGS	The ancillary system can take over the responsibility to manage the liquidity on the sub-account (e.g. due to standing orders or current orders stored within the ancillary system) by sending the liquidity transfer order (increase or decrease) via ASTransferInitiation to RTGS
	7	RTGS	The liquidity transfers are processed between the RTGS dedicated cash accounts and sub-accounts.
	8	RTGS via ESMIG to settlement banks	The settlement banks are informed on an optional basis with camt.054 on the debits/credits executed on their RTGS dedicated cash accounts and sub-accounts.
	9	RTGS via ESMIG to ancillary system	Notified to the ancillary system: –with camt.004 ReturnAccount when the settlement bank has issued the current order. –with ASInitiationStatus when the ancillary system has issued the current order
Start of cycle	10	Ancillary system via ESMIG to RTGS	In order to block the liquidity set aside on the sub accounts, the ancillary system can open a settlement cycle using camt.021-ReturnGeneralBusinessInformation message (or via dedicated U2A GUI screen).

Phase	Step	Processing with	Description
Blocking of liquidity	11	RTGS via ESMIG to ancillary system	Once the cycle is started, the liquidity on the sub-accounts is blocked and can only be increased as long as the cycle is open. Any liquidity transfer order leading to a liquidity decrease on the sub-account will be stored and executed only once the cycle has been closed. The ancillary system is notified on the liquidity blocked on all subaccounts with camt.004-ReturnAccount message. For any liquidity increase executed during the cycle the ancillary system is notified for each booking also via camt.004-ReturnAccount
Settlement	12	Ancillary system (or central bank on behalf) via ESMIG to RTGS	The ancillary system instructs the settlement transactions with ASTransferInitiation.
	13	RTGS	Settlement takes place debiting the sub-accounts and crediting technical accounts and afterwards debiting the technical account crediting the sub-accounts (crediting can take place directly on the RTGS dedicated cash account if indicated by the ancillary system). In case (due to error on ancillary system side) one or more transactions are not covered by the needed liquidity, the transactions remain queued on the sub-account. At the end of the cycle all transactions debiting the same sub-account with insufficient liquidity for their settlement are rejected even if only one transaction cannot be settled. The settlement can avail itself of the optimization process (i.e. settlement algorithm, please refer to %%[please add link to 5.2.7.3])

Phase	Step	Processing with	Description
	14	RTGS via ESMIG to ancillary system (or central bank on behalf)	After the end of the settlement the ancillary system will receive one file as confirmation. The file will contain a list of the credits and debits settled (ASInitiationStatus). If some transactions are not settled until the end of cycle, the ASInitiationStatus will be sent at the end of the cycle with the individual status of each transaction.
	15	RTGS via ESMIG to settlement banks	On an optional basis settlement banks receive camt.054 notifications for the debits and credits on the sub-accounts
End of cycle	16	Ancillary system (or central bank on behalf) via ESMIG to RTGS	Ancillary system (or central bank on behalf) sends an XML “end of cycle“ message to RTGS (camt.021-ReturnGeneralBusinessInformation (optional in U2A via GUI)
	17	RTGS via ESMIG to ancillary system (or central bank on behalf)	The remaining liquidity on the sub-accounts is released and the ancillary system is notified with camt.021-ReturnGeneralBusinessInformation. Stored liquidity transfers for liquidity decrease will now be executed and informed via camt.054 to settlement banks and camt.004-ReturnAccount to the ancillary system. A new Liquidity adjustment phase is now available. The ancillary system can also start a new cycle.
End of procedure	18	Ancillary system (or central bank on behalf) via ESMIG to RTGS	Ancillary system (or central bank on behalf) can send an end of procedure“ message (camt.021-ReturnGeneralBusinessInformation) or using the U2A GUI functionality to close the procedure.
	19	RTGS	The remaining liquidity on sub-accounts is transferred back to the settlement banks’ RTGS dedicated cash accounts

Phase	Step	Processing with	Description
	20	RTGS via ESMIG to ancillary system (or central bank on behalf)	The ancillary system is informed via camt.004-ReturnAccount on the back transfer of liquidity.
	21	RTGS via ESMIG to settlement banks	On an optional basis the settlement banks receive camt.054 notifications on the back transfer of liquidity.

5.3.4.2 Settlement on dedicated liquidity accounts (real-time) (partially completed)

As mentioned above, the settlement on dedicated liquidity accounts (real-time) is based on ancillary system payment instructions initiated by ancillary systems between the settlement banks' RTGS dedicated cash account and the ancillary system dedicated liquidity account. The ancillary system reflects this liquidity to the settlement bank's account held within the ancillary system.

For Settlement on dedicated liquidity accounts (real-time) the settlement phase is an internal process of the ancillary system and therefore no details are provided here.

During the whole process, the ancillary system will be notified on the amounts available on the dedicated liquidity account. This happens whenever the liquidity on this account changes (by standing orders or current orders) or by providing the result of the settlement instructed by the ancillary system (i.e. ASInitiationStatus). Thus, the ancillary system is always in a position to know the liquidity set aside for their settlement. Settlement shall only be started once the liquidity needed is available on the dedicated liquidity account. In turn, during the settlement cycle only on an exceptional basis (i.e. an error on ancillary system side) transactions should be pending due to missing liquidity.

When the procedure is closed at 18:00h the dedicated liquidity account not necessarily has to have a zero-balance.

Liquidity provision

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

- setting up standing orders in reference data (to be executed with each start of procedure. Standing orders set up in reference data will only become effective with the next business day.
- sending camt.050 LiquidityCreditTransfer messages (current order liquidity transfer)
- using the dedicated RTGS GUI liquidity transfer screens (current order liquidity transfer)
- the ancillary system sending ASTransferInitiation messages debiting the settlement banks RTGS dedicated cash account and crediting the dedicated liquidity account (current order liquidity transfer)

Liquidity transfers will be executed in the following way:

- Standing order liquidity transfers are executed with each start of procedure (different amounts for each of the two procedures can be specified).
- Current order liquidity transfers will be executed during an open procedure (mandatory or optional settlement procedure). They will be executed with immediate effect during an open procedure with no cycle running. In the opposite case, where a cycle is running, the liquidity transfer will be stored and executed only once the cycle has closed.

Effects on liquidity transfers in case of missing liquidity

The amounts taken into account for the execution of the different types of liquidity transfers are depicted below.

Table 56 - Amounts taken into account

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

Mandatory procedure

The mandatory procedure is opened by the RTGS component on the new business day (19:30h) in an automated way for all ancillary systems using settlement on dedicated liquidity accounts. This procedure cannot be reopened by the ancillary system (or its central bank on behalf). For the settlement on dedicated li-

quidity accounts real-time the mandatory procedure cannot be closed by the ancillary system (or its central bank on behalf).

Overview on the settlement process

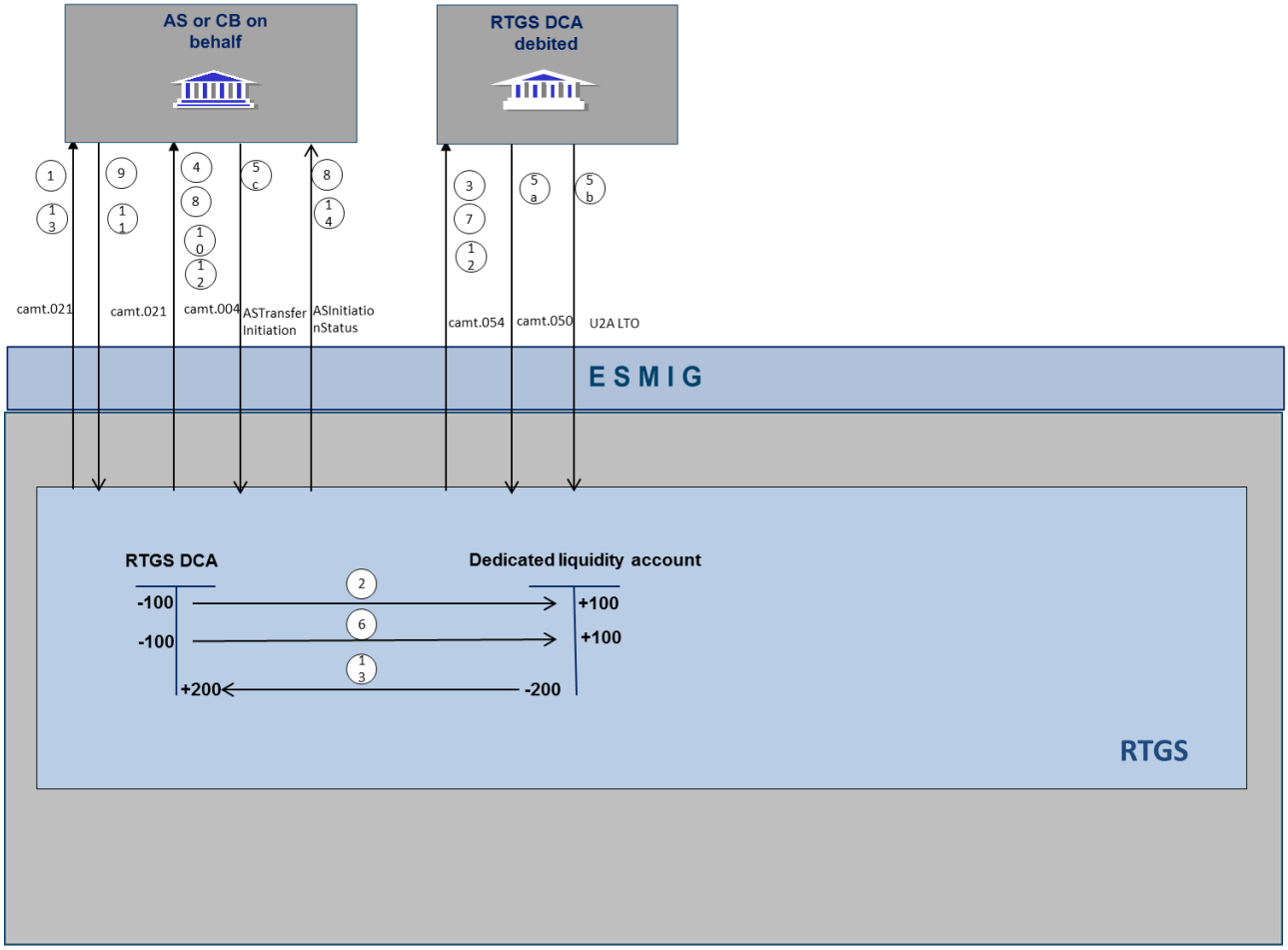


Table 57 - Start of procedure and liquidity provision

Phase	Step	Processing with	Description
Start of procedure	1	RTGS via ESMIG to ancillary system	Start of procedure message for mandatory procedure is automatically initiated by the RTGS component at 19:30h on the new business day. The ancillary system will be notified on the event (camt.021 ReturnGeneralBusinessInformation).
Standing orders execution	2	RTGS	The start of procedure triggers the execution of existing standing orders debiting the settlement bank's RTGS dedicated cash accounts and crediting the dedicated liquidity account. (please refer to table "amounts taken into account")
	3	RTGS via ESMIG to settlement banks	On an optional basis, the settlement banks are notified on the debited amount on the RTGS dedicated cash account (camt.054 Debit notification)
	4	RTGS via ESMIG to ancillary system	The ancillary system is notified on credit of the dedicated liquidity account for the amounts actually booked (camt.004 ReturnAccount)
Liquidity adjustment	5a	Settlement banks via ESMIG to RTGS	Settlement banks can adjust (increase) the liquidity on the dedicated liquidity account by using current order liquidity transfers (camt.050).
	5b	Settlement banks via ESMIG to RTGS	Settlement banks can adjust (increase) the liquidity on the dedicated liquidity account by using current order liquidity transfers (dedicated U2A GUI screens).

Phase	Step	Processing with	Description
	5c	Ancillary system via ESMIG to RTGS	The ancillary system can take over the responsibility to manage the liquidity on the dedicated liquidity account due to current orders stored within the ancillary system by sending the liquidity transfer order (increase or decrease) via ASTransferInitiation to RTGS. The ancillary system cannot set standing orders on behalf of its settlement bank, to provide such a functionality the ancillary system has to store and manage its own procedure outside the RTGS any potential standing orders and send them at the appropriate time as current orders.
	6	RTGS	The liquidity transfers are processed between the RTGS dedicated cash accounts and the dedicated liquidity account.
	7	RTGS via ESMIG to settlement banks	The settlement banks are informed on an optional basis with camt.054 on the debits executed on their RTGS dedicated cash accounts.
	8	RTGS via ESMIG to ancillary system	Notified to the ancillary system: - with camt.004 ReturnAccount when the settlement bank has issued the current order. - with ASInitiationStatus when the ancillary system has issued the current order
Start of cycle	9	Ancillary system via ESMIG to RTGS	Ancillary system sends a “start of cycle messages” to RTGS (camt.021 ReturnGeneralBusinessInformation (optional in U2A via GUI). The incoming liquidity transfers will not be any longer immediately executed.

Phase	Step	Processing with	Description
	10	RTGS via ESMIG to ancillary system	The ancillary system is notified about the amount actually credited with a camt.004 ReturnAccount
End of cycle	11	Ancillary system (or central bank on behalf) via ESMIG to RTGS	ancillary system sends an XML “end of cycle“ message to RTGS (camt.021 ReturnGeneralBusinessInformation (optional in U2A via GUI)
	12	RTGS via ESMIG to ancillary system and settlement banks	Possible current orders received during the cycle will now be executed and informed via camt.054 to settlement banks and camt.004 ReturnAccount to the ancillary system.
Liquidity release	13	RTGS via ESMIG to ancillary system	The RTGS releases the remaining liquidity and notifies the ancillary system with a camt.021 ReturnGeneralBusinessInformation

Cross-ancillary system settlement

Basics

In addition to the above described procedures for the settlement of ancillary system, there is also the possibility to send payment instructions on a cross-ancillary system basis. As a precondition to use this feature a bilateral agreement between the sending and receiving ancillary system needs to be in place. This agreement has to be input in the reference data by the relevant central banks on behalf of the ancillary systems. The functionality of cross-ancillary system settlement is independent from the procedure the sending and receiving ancillary system are using (i.e. interface vs. real-time). Such transfers are always instructed by the ancillary system (or its central bank on behalf) as a single payment via an ASTransferInitiation. A prerequisite for the settlement of such transaction is that the payment is sent during an open cycle of the sending ancillary system and an open procedure of the receiving ancillary system (regardless if mandatory or optional procedure is open). In such case the settlement is executed immediately whatever is the status of the cycle of the receiving ancillary system. Reverse transactions are not allowed.

Real-time ancillary system to interface ancillary system

With this payment instruction the dedicated liquidity account of the sending ancillary system on behalf of a settlement bank is debited in order to credit the sub-account of one of the settlement banks of the receiving ancillary system. The receiving ancillary system is notified with an ASTransferNotice on the incoming liquidity to the sub-account including the information of the resulting balance. The receiving ancillary system has the

possibility to use this credit immediately. The sending ancillary system or its central bank on behalf is notified with an ASInitiationStatus on the outcome of the request. If the request was sent by the central bank on behalf the ancillary system is notified with a camt.004-ReturnAccount on successful settlement. On an optional basis the settlement bank of the receiving ancillary system is notified with a camt.054 Credit notification.

Real-time ancillary system to real-time ancillary system

With this payment instruction the dedicated liquidity account of the sending ancillary system on behalf of a settlement bank is debited in order to credit the dedicated liquidity account of the receiving ancillary system in favour of one of the settlement banks. The receiving ancillary system is notified with an ASTransferNotice on the incoming liquidity to the sub-account including the information of the resulting balance. The receiving ancillary system has the possibility to use this credit immediately. The sending ancillary system or its central bank on behalf is notified with an ASInitiationStatus on the outcome of the request. If the request was sent by the central bank on behalf the ancillary system is notified with a camt.004-ReturnAccount on successful settlement. On an optional basis the settlement bank of the receiving ancillary system is notified with a camt.054 Credit notification.

Interface ancillary system to interface ancillary system

With this payment instruction the sub-account of a settlement-bank of the sending ancillary system is debited in order to credit the sub-account of one of the settlement banks of the receiving ancillary system. The receiving ancillary system is notified with an ASTransferNotice on the incoming liquidity to the sub-account including the information of the resulting balance. The receiving ancillary system has the possibility to use this credit immediately. The sending ancillary system or its central bank on behalf is notified with an ASInitiationStatus on the outcome of the request. If the request was sent by the central bank on behalf the ancillary system is notified with a camt.004-ReturnAccount on successful settlement. On an optional basis the settlement bank of the receiving ancillary system is notified with a camt.054 Credit notification.

Interface ancillary system to real-time ancillary system

With this payment instruction the sub-account of a settlement-bank of the sending ancillary system is debited in order to credit the dedicated liquidity account of the receiving ancillary system in favour of one of the settlement banks. The receiving ancillary system is notified with an ASTransferNotice on the incoming liquidity to the sub-account including the information of the resulting balance. The receiving ancillary system has the possibility to use this credit immediately. In case the liquidity on the sub-account is insufficient the transaction is rejected. The sending ancillary system or its central bank on behalf is notified with an ASInitiationStatus on the outcome of the request. If the request was sent by the central bank on behalf the ancillary system is notified with a camt.004-ReturnAccount on successful settlement. On an optional basis the settlement bank of the receiving ancillary system is notified with a camt.054 Credit notification.

5.3.5 Optional connected mechanisms (partially completed)

General aspects

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

- information period
- scheduled time (“from”)
- settlement period (“till”)
- guarantee fund mechanism

Depending on the procedure and the pertaining messages used to instruct the ancillary system payment instructions, the ancillary system (or central bank on behalf) has to fill in a specific field provided in the group header (ASTransferInitiation) to make use of these mechanisms (this is not valid for the guarantee fund mechanism as this relies on reference data).

Only the ancillary system (or the central bank acting on its behalf) is entitled to insert these parameters in the message. Once a message is sent the parameters can be updated in U2A mode:

- by the ancillary system for optional mechanism “Settlement period” before the inserted “till”- time has been expired
- by the settlement bank only for optional mechanism “Scheduled time” before the inserted “from”- time has been reached

“Information period” and “guarantee fund mechanism” parameters can be updated neither by the ancillary systems nor by the settlement banks nor by central bank.

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

Table 58 - Usability of optional connected mechanism per ancillary system processing procedure

Settlement Procedure	Information Period	Time indicator	Settlement Period ("till")	Guarantee fund mechanism
Standard multilateral settlement	X		X	x
Simultaneous multilateral settlement	X		X	X
Settlement on dedicated liquidity accounts (interfaced)				
Settlement on dedicated liquidity accounts (real-time)				

Information period

The information period option allows settlement banks a more efficient liquidity management giving the possibility of knowing in advance the liquidity needed for the settlement of a specific ancillary system payment order. This optional connected mechanism is usable for

- standard multilateral settlement and
- simultaneous multilateral settlement

The information period option can be used by indicating a specific time (within the operational hours for ancillary system processing) or duration (the calculated end time as well has to be within the operational hours for ancillary system processing) within an ASTransferIntiation message. The usage of this option will lead to

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

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

Process flow

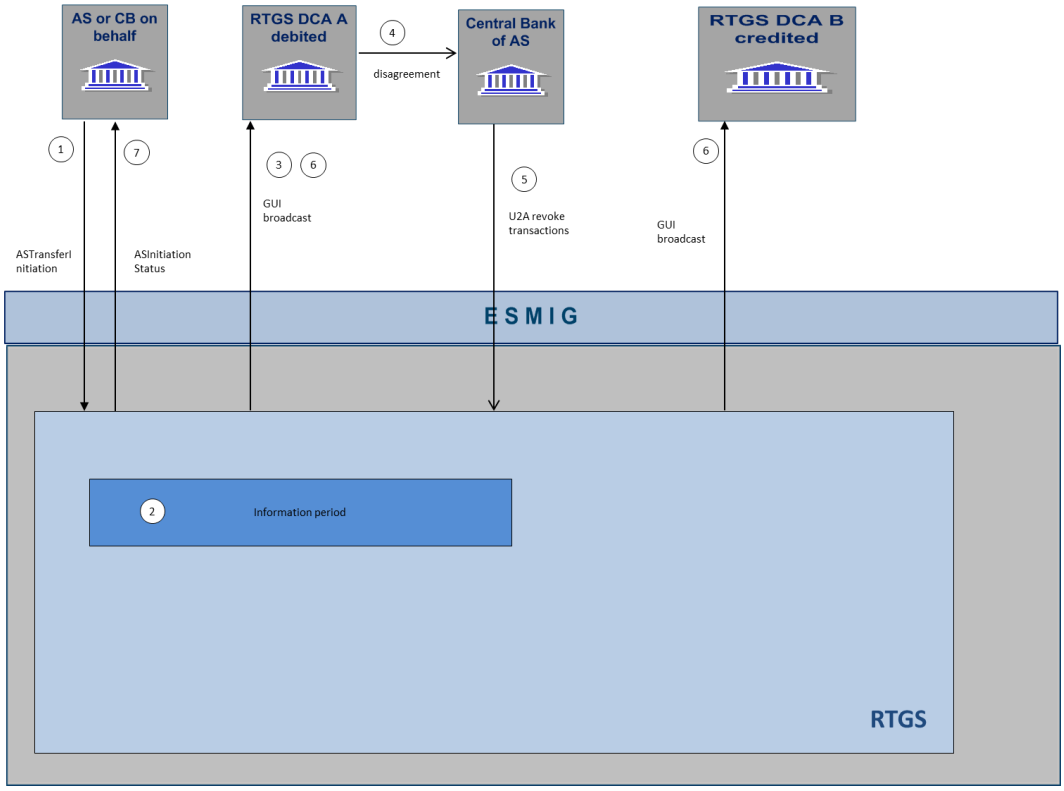


Table 59 - Process flow information period with disagreement

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

Action	Step	Interaction	Description
	4	Settlement bank to central bank of the ancillary system	In case one or several settlement banks disagree on the amount of the ancillary system payment orders present in the pertaining set of ancillary system payment instructions, it may contact the central bank of the ancillary system. The procedure on how, if and when such disagreement is to be used is to set up based on contractual agreements within the ancillary system and it's community. Also the way the settlement bank contacts the central bank (directly or indirectly via the pertaining ancillary system) is out of scope of the RTGS component.
	5	Central bank via ESMIG to RTGS	The central bank revokes via GUI the disagreed set of ancillary system payment instructions, leading to a rejection of all ancillary system payment instructions and settlement is not triggered. The information period and all processing of the involved ancillary system payment instructions is stopped.
Notification in case of disagreement	6	RTGS via ESMIG to all settlement banks	All settlement banks are informed via broadcast on the rejection of the transactions due to disagreement.
	7	RTGS via ESMIG to ancillary system	The ancillary system is informed via ASInitiaionStatus message on the rejection due to disagreement
End of information period	8	RTGS	In case no disagreement was expressed during the information period, the indicated end of the Information Period will mark the start of the settlement period.

Settlement period ("till")

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

Similar to Information Period option, the Settlement Period ("till") option has to be indicated per ASTransferInitiation in the GroupHeader of the message and is then valid for the whole set of ancillary system payment instructions.

The ancillary system (or its responsible central bank on behalf), according to rules established at national level, can modify the end of the settlement period ("Change settlement period" in U2A mode) before it is expired.

Ancillary systems are expected to use the settlement period option to avoid the extension of the arranged settlement timeframe. This option helps the ancillary system to control the execution time of their ancillary system payment instructions but also helps the settlement banks to have a better control of their liquidity.

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

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

Time indicator

Ancillary systems sending payment orders (pacs.009) in the framework of settlement procedures real-time settlement and bilateral settlement can take benefits from using the time indicator options available for payment orders (for the effects please refer also to chapter Amendment of payments [▸ 76]).

- FromTime specifies the time only after which a payment order can be submitted to settlement
- TillTime specifies the time when the party expects the payment order to be settled. 15 minutes before TillTime, a warning notification will be triggered if the payment order has not been settled by that time. When the TillTime is reached and the payment order is not yet settled, then the payment order shall not be rejected and it may still be submitted for settlement beyond this time. If TillTime is specified, then RejectTime cannot be specified.
- RejectTime specifies the latest time for a payment order to be submitted to settlement. 15 minutes before RejectTime, a warning notification will be triggered if the payment order has not been settled by that time. As soon as the RejectTime is reached and if the payment order has not been settled, the payment order will be rejected and a settlement failure notification will be sent out.

In fact, even though the effects on settlement of the underlying transactions are similar to the ones of the options Information Period and Settlement Period ("till"), there are differences.

- No GUI broadcast is sent to settlement banks after reception of payment orders with From Time
- No GUI broadcast is sent after reaching the From Time and payment order is queued
- There is no disagreement possible before reaching the From Time. Being payment orders, the standard functionalities for revoking payment orders prior to their final execution apply.
- Time indicators are to be presented in each payment order, there is no way to indicate it once for the whole set of transactions (i.e. if payment orders are bundled into a file, the execution times have to be set individually for each and every payment order included).

Guarantee fund mechanism

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

This optional mechanism can be used only:

- in relation to a multilateral (both standard and simultaneous) procedure and
- together with “Settlement period” time option

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

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

Process description

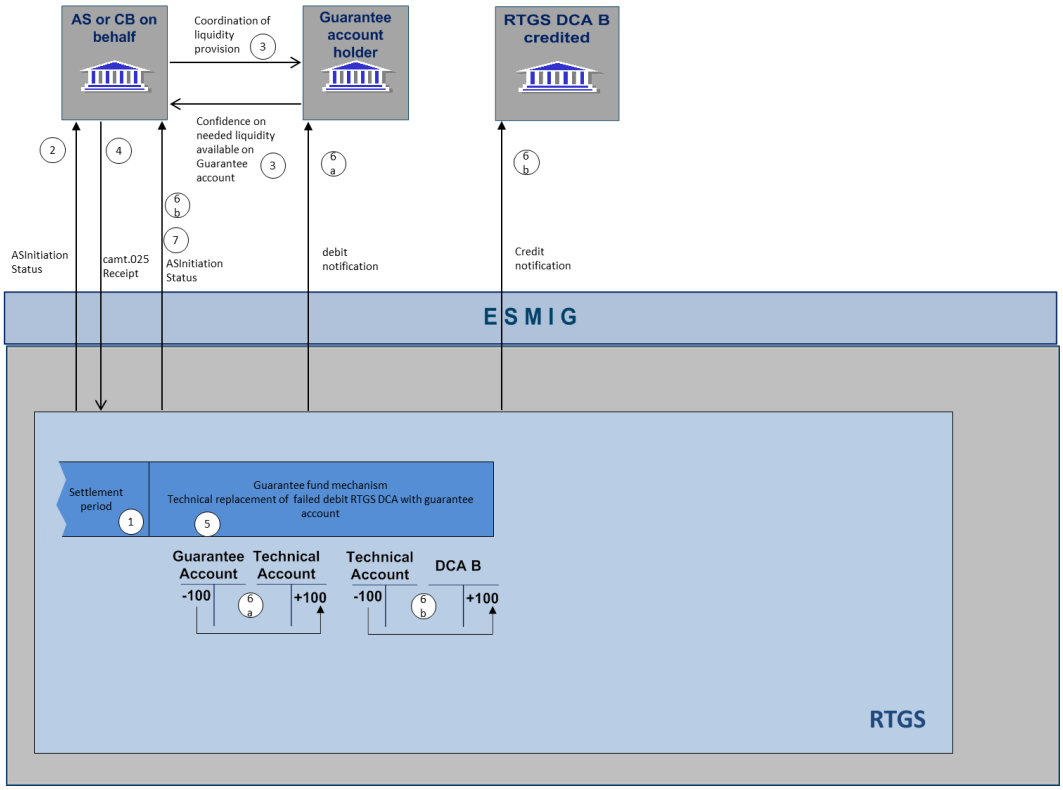


Table 60 - Process description

Phase	Step	Description
Settlement failure	1	If the “settlement period (“till”) time is indicated and the settlement of either standard or simultaneous multilateral settlement is not yet achieved when the allotted time is exceeded the settlement fails. In case of simultaneous multilateral settlement, prior to the start of the guarantee fund mechanism a transformation of all ancillary system payment instructions to standard multilateral settlement is performed and debit ancillary system payment instructions covered by needed liquidity are executed. Please refer to chapter Simultaneous multilateral settlement [112] for the flow of messages related to this scenario, it will not be depicted here).
Guarantee fund mechanism	2	If the guarantee fund mechanism has been set up (reference data), the ancillary system is notified on settlement failure with an ASInitiationStatus message containing the request to confirm the use of the guarantee fund mechanism by using the “Decision Indicator” flag within this message.
	3	Depending on the guarantee schema either the collection of the needed liquidity has been granted in advance by the ancillary system and its community (i.e. prefunding) or the ancillary system has to co-ordinate the liquidity collection making it available on the specific “guarantee account”. This can be done, depending on the specific set up of the guarantee account by using liquidity transfers (camt.050) or payment orders (pacs.009/pacs.010). Note: The notifications to the guarantee account holder then depends on the messages used. In the graphic only the prefunding is depicted. In any case, before the guarantee fund mechanism starts the ancillary system has to assure the needed liquidity is provided on the guarantee account.
	4	The ancillary system sends an XML message (camt.025 Receipt) to give either a positive or a negative confirmation in order to proceed or not with a new settlement phase against the guarantee account.
New settlement phase	5	If the ancillary system confirms the actual use of the guarantee fund mechanism RTGS re-enters the transactions for which the liquidity was missing in order to be settled on the guarantee account by substituting the failed debtor’s RTGS dedicated cash account with the guarantee account.
	6a	In case of sufficient liquidity the settlement of the debit from guarantee account to the ancillary system technical account will be executed. Depending on the message subscription also the guarantee account holder is notified with a camt.054 debit notification.

Phase	Step	Description
	6b	After successfully debiting the guarantee account, all credit bookings from the ancillary system technical account to the RTGS dedicated cash accounts of the settlement banks will be executed. The ancillary system is notified about the ending of the whole settlement procedure. On an optional basis, the settlement banks of the creditor side are notified with a camt.054 credit notification.
	7	If the ancillary system sends a negative confirmation or there is a lack of liquidity on the guarantee account the “reversing procedure” is initiated in order to transfer back the already executed debits from the ancillary system technical account to the RTGS dedicated cash accounts of the settlement banks. All involved settlement banks are notified with a GUI broadcast on failed settlement. The ancillary system will receive an ASInitiationStatus informing on the failed settlement.

5.4 Liquidity management

5.4.1 Available liquidity (completed)

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

In principle, the positive balance on the RTGS dedicated cash account is available to settle payment orders and payment instructions from ancillary systems on the RTGS dedicated cash account of an RTGS participant. The credit line – if available - is managed on the main cash account in CLM.

Depending on the priority of a payment order (see chapter Payment priorities [▸ 61]) and the liquidity management features used by the RTGS participant, the actual liquidity available for settlement of this payment order might be less than the balance on the RTGS dedicated cash account (see table below).

Table 61 - Effect of reservations on the available liquidity

Effect	Urgent payment order	High payment order	Normal payment order
Available liquidity	Balance on the RTGS dedicated cash account	Balance on the RTGS dedicated cash account ./. Urgent reserve	Balance on the RTGS dedicated cash account ./. Urgent reserve ./. High reserve

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

As central bank accounts in the RTGS component can have a negative balance, the available liquidity for central banks is not limited.

5.4.2 Liquidity transfer (completed)

5.4.2.1 Overview (completed)

In general, liquidity transfers debiting a RTGS dedicated cash account are initiated by the RTGS participant (either in A2A or U2A). In order to instruct the transfer of cash from one cash account to another cash account via A2A, the Liquidity Transfer Order Message (LiquidityCreditTransfer (camt.050) [▶ 336]) is used. Liquidity transfers are not classified as payments (i.e. pacs) but are cash management instructions using camt messages. In order to transfer liquidity to the ancillary system dedicated liquidity account (real-time) in A2A, the settlement banks can also use the SBTransferInitiation message.

Further details on the initiation of liquidity transfers via U2A are provided in the RTGS User Handbook.

In general, liquidity transfer orders can be used to transfer liquidity

- between two cash accounts within the RTGS component, i.e. RTGS dedicated cash account, sub-account for ancillary system, ancillary system technical account (intra-service)
- from an RTGS dedicated cash account to a CLM main cash account - or vice versa (inter-service)
- from an RTGS dedicated cash account to a dedicated cash account of the TIPS or T2S service - or vice versa (inter-service)

A liquidity transfer can be executed **within** the RTGS component only if

- all involved RTGS dedicated cash accounts belong to the same liquidity transfer group or
- a Central Bank account is involved or
- the accounts belong to the same party or
- it is a liquidity transfer between an RTGS dedicated cash account and the sub-accounts linked to this RTGS dedicated cash account or
- it is a liquidity transfer between an RTGS dedicated cash account and an ancillary system technical account.

In general liquidity transfers are never queued in the RTGS component, they are either

- settled immediately (fully or partially) or
- rejected.

Only under certain conditions an automatically generated liquidity transfer can be pending. This is only the case if a CLM MCA has insufficient liquidity for settling a central bank operation AND there is not sufficient liquidity on the RTGS dedicated cash account to settle this automated inter-service liquidity transfer. In such scenario any incoming liquidity (up to the required amount) on the RTGS dedicated cash account will be transferred stepwise to the MCA until the original amount of the automated inter-service liquidity transfer due to pending central bank operations (i.e. the amount needed to settle the pending central bank operations in CLM) is completely settled.

Note: Whenever such automated inter-service liquidity transfer is pending, it gets the top priority and therefore in principle does not allow the settlement of any other payment order.

Once a liquidity transfer is booked on the RTGS dedicated cash account, this booking is irrevocable and unconditional.

The following types of liquidity transfers can be initiated in the RTGS component.

Table 62 - Liquidity transfer types

Liquidity transfer type	Description
Immediate liquidity transfer order	Immediate transfer of a certain amount of liquidity after initiation of the payment order by the RTGS dedicated cash account holder or an authorised third party.
Event-based liquidity transfer order	Transfer of liquidity non-recurring at a pre-defined event. Specification of events by the RTGS component. Definition of amount and event by the RTGS dedicated cash account holder.
Standing liquidity transfer order	Transfer of liquidity (a certain amount) regularly at a certain event. Definition by the RTGS dedicated cash account holder.

For immediate liquidity transfer orders the process will be initiated by either the RTGS participant itself or by another authorised actor on the RTGS participant's behalf by sending the respective liquidity transfer to the RTGS component. For event-based and standing liquidity transfer orders the process will be initiated by the RTGS itself whenever the respective event to trigger the liquidity transfer order is reached. The RTGS component will then process the liquidity transfer.

If the content of the immediate liquidity transfer order is either invalid or would result in checks to fail, it is rejected and a rejection notification is sent to the sender (depending on the channel a message in A2A mode or an error message on the screen in U2A mode). If the content of the liquidity transfer order is valid and certain checks have been passed, the RTGS component will try to transfer (part of) the liquidity amount requested to the relevant cash account referred to in the liquidity transfer order. Where the intra-RTGS liquidity transfer (partially) succeeds, RTGS will transfer (part of) the amount requested and RTGS will send a (partially) transfer success notification to the participant/ancillary system involved (where the participant opted for it).

In case of partial execution of a liquidity transfer order, the respective debit notification sent to the account owner of the debited RTGS dedicated cash account will contain the amount actually settled (which might differ from the instructed amount).

5.4.2.2 Initiation of liquidity transfers (completed)

Liquidity transfers in the RTGS component in A2A using a camt.050 message are initiated by either

- the RTGS participant itself
- by another authorised actor (e.g. an ancillary system or another credit institution) or
- by the RTGS component itself, based on information provided by CRDM.

Liquidity transfer orders can have the following type:

- immediate liquidity transfer via A2A or U2A or
- standing liquidity transfer order or
- event-based liquidity transfer order.

As regards the execution of liquidity transfers which can be initiated in the RTGS component the following principles apply.

Table 63 - Execution of liquidity transfers

Liquidity transfer type	Initiator	Settlement
Immediate liquidity transfer	RTGS participant	Only full settlement
	Ancillary system (on behalf)	Partial settlement possible; In case of partial settlement no further settlement attempt will be performed.
	Central bank (on behalf)	Only full settlement
Event-based liquidity transfer	Pre-configured by RTGS participant	Partial settlement possible
Standing order liquidity transfer	Pre-configured by RTGS participant	Partial settlement possible – in case several standing order liquidity transfers are triggered with the same event a pro rata execution applies. In case of partial settlement, no further settlement attempt will be performed.

5.4.2.3 Liquidity transfer process

5.4.2.3.1 Liquidity transfer between two dedicated cash accounts of the RTGS component (completed)

Important preconditions

1. Both involved accounts exist and are active.
2. Respective access rights have been granted to the sender.
3. A liquidity transfer group was set up by the responsible central bank.

Message flow

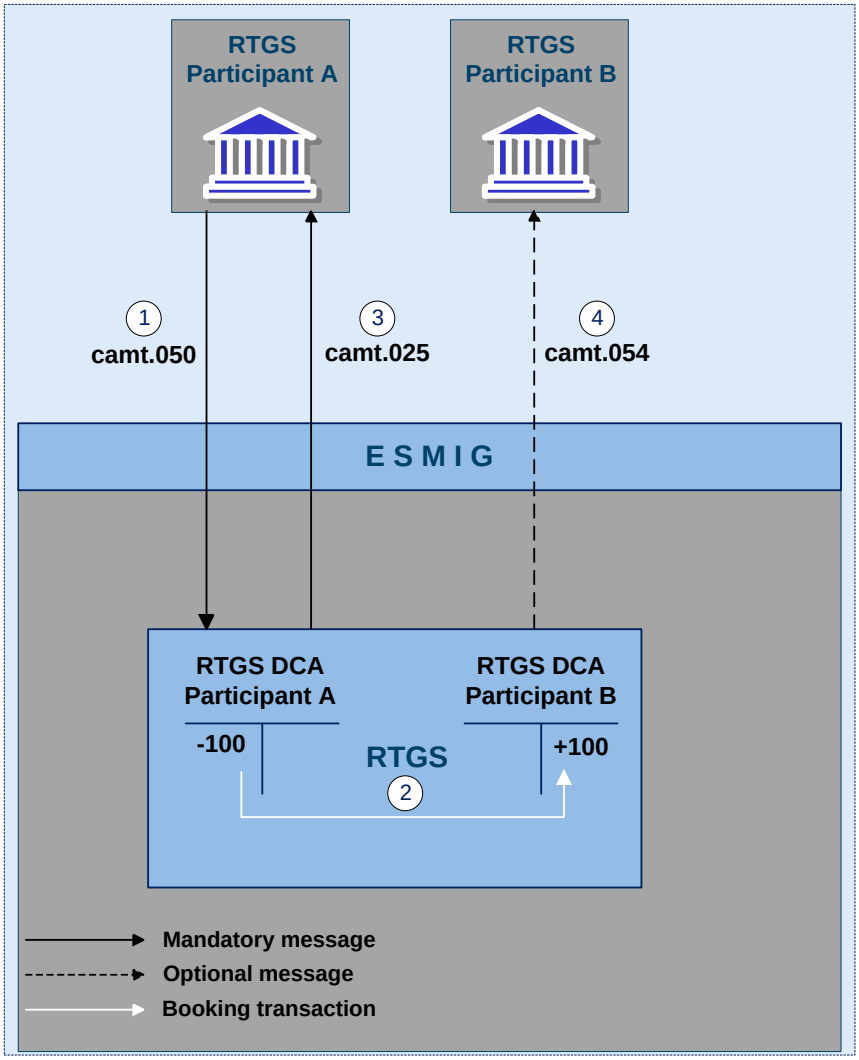


Figure 16 - Liquidity transfer order between two RTGS dedicated cash accounts in the RTGS component

Process description

The liquidity transfer between two RTGS dedicated cash accounts consists of the following process steps:

Table 64 - Process description

Step	Processing in/between	Description
1	RTGS participant A via ESMIG to the RTGS component	The RTGS participant A sends a camt.050 via ESMIG to the RTGS component.
2	RTGS component	RTGS message check and validation in the RTGS component are positive. Simultaneous booking on the RTGS dedicated cash accounts of RTGS participants A and B
3	RTGS component via ESMIG to RTGS participant A	Creation and forwarding of camt.025 (mandatory) to participant A
4	RTGS via ESMIG to RTGS participant B	Creation an forwarding of camt.054 (optional) to participant B

Used messages

- LiquidityCreditTransfer (camt.050) [▸ 336]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 349]
- Receipt (camt.025) [▸ 318]

In addition to the classical intra-service liquidity transfer between two RTGS dedicated cash accounts, the following business cases are also considered to be intra-RTGS liquidity transfers:

- Liquidity transfer from an RTGS dedicated cash account to a linked sub-account dedicated to an ancillary system using the “interfaced” ancillary system procedure (and vice versa).
- Liquidity transfer from an RTGS participant’s RTGS dedicated cash account to the dedicated liquidity account related to an ancillary system using ancillary system procedure “real-time” (and vice-versa).
- Liquidity transfer from one RTGS dedicated cash account to another RTGS dedicated cash account of the same party.

5.4.2.3.2 Liquidity transfer from dedicated cash account of the RTGS component to CLM main cash account (completed)

Preconditions

1. Both RTGS dedicated cash account and main cash account exist and are active.
2. Respective access rights have been granted to the sender.

Message flow

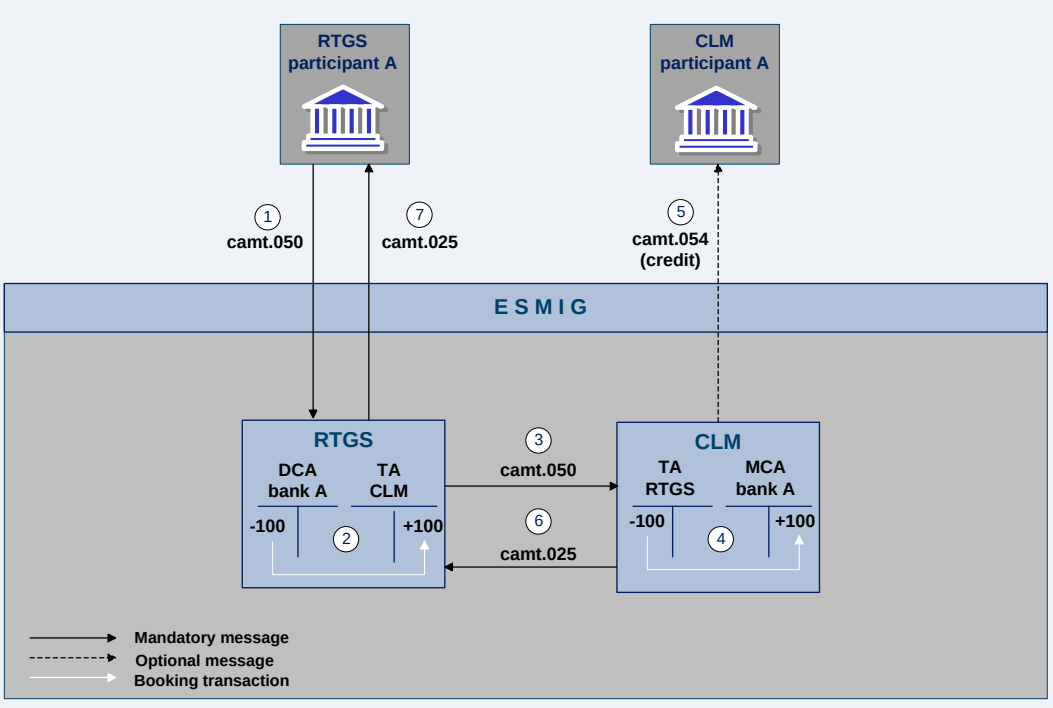


Figure 17 - Liquidity transfer from a RTGS dedicated cash account to a CLM main cash account

Process description

The liquidity transfer order between an RTGS dedicated cash account and a main cash account in CLM consists of the following process steps:

Table 65 - Process description

Step	Processing in/between	Description
1	RTGS participant A via ESMIG to the RTGS component	A camt.050 is sent from RTGS participant A to the RTGS component
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS dedicated cash account of RTGS participant A and the CLM Transit Account
3	RTGS component to CLM component	Internal processing
4	CLM	Simultaneous booking on the RTGS Transit Account and the main cash account of CLM participant A (can be owned by a different party)

Step	Processing in/between	Description
5	CLM via ESMIG to the CLM participant A	A camt.054 (credit) is sent by the CLM component via ESMIG to the CLM participant A (optional)
6	CLM to RTGS component	Internal processing
7	RTGS component via ESMIG to RTGS participant A	Creation and forwarding of a camt.025 to RTGS participant A (mandatory)

Used messages

- LiquidityCreditTransfer (camt.050) [▸ 336]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 349]
- Receipt (camt.025) [▸ 318]

5.4.2.3.3 Liquidity transfer from dedicated cash account of the RTGS component to a dedicated cash account in different settlement services (completed)

Preconditions

1. Both RTGS dedicated cash account and dedicated cash account in other service exist.
2. Respective access rights have been granted to the sender.

Message flow

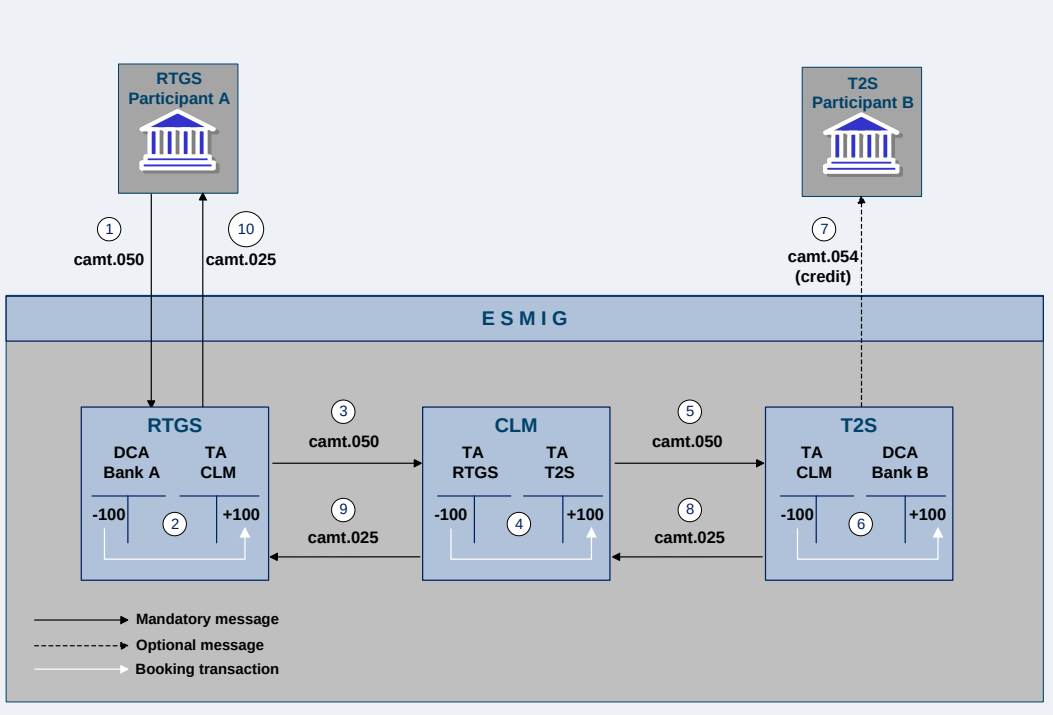


Figure 18 - Liquidity transfer from an RTGS dedicated cash account to a dedicated cash account in the T2S service

Note: The detailed functionality of CLM and T2S is out of scope of this UDFS.

Process description

The liquidity transfer from the RTGS dedicated cash account to a dedicated cash account of a different service (T2S as example) consists of the following process steps:

Table 66 - Process description

Step	Processing in/between	Description
1	RTGS participant A via ESMIG to the RTGS component	A camt.050 is sent from the RTGS participant A to the RTGS component via ESMIG.
2	RTGS component	Message check and validation in the RTGS component positive Simultaneous booking on the RTGS dedicated cash account of RTGS participant A and the CLM transit account
3	RTGS component to CLM	Internal processing
4	CLM	Simultaneous booking on the RTGS Transit Account and the T2S Transit Account
5	CLM to T2S service	A camt.050 is forwarded to the T2S service

Step	Processing in/between	Description
6	T2S service	Simultaneous booking on the CLM transit account and the dedicated cash account of T2S participant B
7	T2S service via ESMIG to the T2S participant B	A camt.054 (credit) is sent by the T2S service via ESMIG to the T2S participant B (optional).
8	T2S service to the CLM component	A camt.025 is forwarded to the CLM component
9	CLM to the RTGS component	Internal processing
10	RTGS component via ESMIG to the RTGS participant A	Creation and forwarding of a camt.025 to RTGS participant A generated by the RTGS component (mandatory)

Used messages

- LiquidityCreditTransfer (camt.050) [▸ 336]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 349]
- Receipt (camt.025) [▸ 318]

5.4.2.3.4 Liquidity transfer from dedicated cash account in different settlement service to a dedicated cash account of the RTGS component (completed)

Important preconditions

1. Both RTGS dedicated cash account and dedicated cash account in other service exist.
2. Respective privileges have been granted to the sender.

Message flow

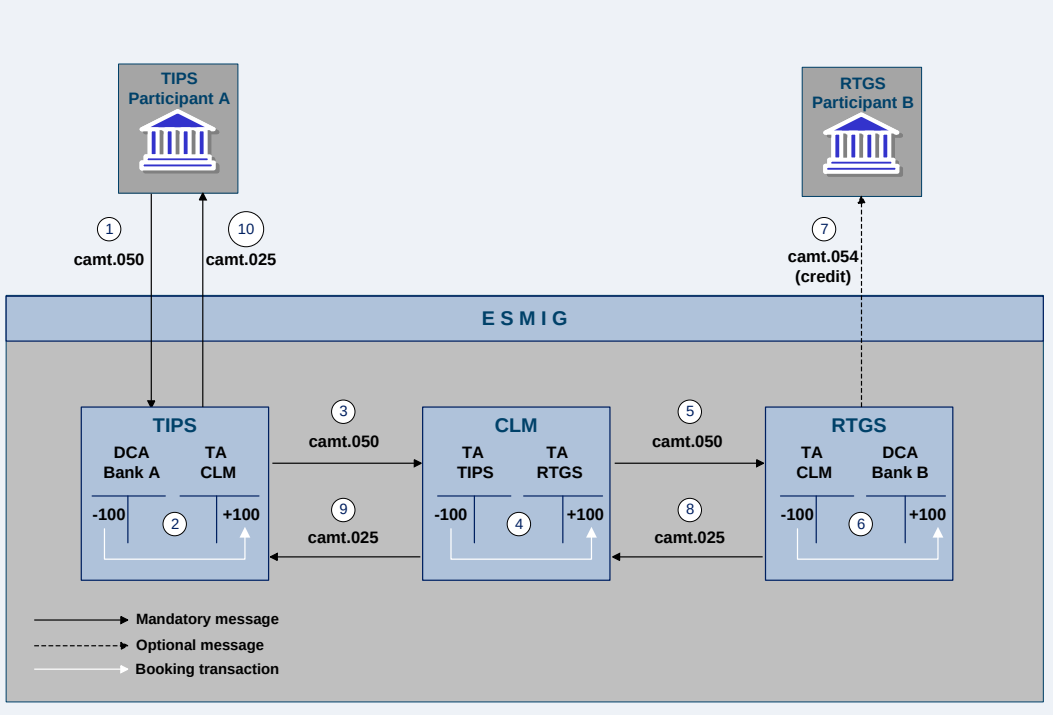


Figure 19 - Liquidity transfer from dedicated cash account of the TIPS service to an RTGS dedicated cash account

Note: The detailed functionality of TIPS and CLM are out of scope of this UDFS.

Process description

The liquidity transfer from a different service (TIPS in this example) to the RTGS dedicated cash account consists of the following process steps:

Table 67 - Process description

Step	Processing in/between	Description
1	TIPS participant via ESMIG to TIPS	A camt.050 is sent from the TIPS participant A to TIPS via ESMIG
2	TIPS	Message check and validation in TIPS service positive Simultaneous booking on the TIPS dedicated cash account of TIPS participant A and the CLM transit account
3	TIPS to CLM	A camt.050 is forwarded to CLM
4	CLM	Simultaneous booking on the TIPS transit account and the RTGS transit account
5	CLM to RTGS component	A camt.050 is forwarded to the RTGS component

Step	Processing in/between	Description
6	RTGS component	Simultaneous booking on the CLM transit account and the RTGS dedicated cash account of RTGS participant B
7	RTGS component via ESMIG to the RTGS participant B	A camt.054 (credit) is sent by the RTGS component via ESMIG to the RTGS participant B
8	RTGS component to the CLM	Internal processing
9	CLM to TIPS	A camt.025 is forwarded to TIPS
10	TIPS via ESMIG to the TIPS participant A	Creation and forwarding of a camt.025 to TIPS participant A generated by TIPS (mandatory)

Used messages

- LiquidityCreditTransfer (camt.050) [▸ 336]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 349]
- Receipt (camt.025) [▸ 318]

5.4.2.4 Rejection of liquidity transfer orders (completed)

Liquidity transfer orders sent to the RTGS component have to pass several validations before the liquidity is booked on the accounts and effectively transferred. Validations performed include technical checks, format checks as well as business validations.

For different reasons, a liquidity transfer order can be rejected and a notification with the appropriate error code for rejection is returned to the sender. The validations are distinguished into technical validations and business validations.

In case the technical validation is not successful, a ReceiptAcknowledgement (admi.007) is sent to the sender indicating which error occurred.Rejections of camt.050 messages sent in A2A mode due to the business validations result in a receipt message (camt.025) being sent to the sending actor including the respective error code(s) according to chapter Index of business rules and error codes [▸ 420].

Note: The sending of a negative notifications is mandatory and not subject to message subscription.

In case of liquidity transfer orders initiated via U2A the failed validations are shown directly in the GUI.

5.4.2.4.1 Business validations (completed)

The validations described below will be performed in one step in order to capture all the possible breaches. Therefore, the checks must not stop after the first breach occurring as there could be further breaches in the

subsequent checks. If the validation failed overall, a rejection notification with appropriate reason codes for all breaches which occurred must be sent to the sender. The comprehensive list of business rules and error codes can be found in chapter Index of business rules and error codes [▸ 420].

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

- liquidity transfer group check
- duplicate check
- process specific authorisation checks
- settlement date check
- field and reference data checks
- direct debit check
- check of backup payments
- mandated payment check
- account checks

5.4.3 Liquidity management features

5.4.3.1 Reservation

5.4.3.1.1 Overview (completed)

The RTGS component offers two different types of reservation:

- urgent - With the usage of the urgent reservation facility, liquidity can be reserved for the execution of urgent payment orders.
- high - With the usage of the high reservation facility, liquidity can be reserved for the execution of urgent and high payment orders.

The RTGS participant decides which payment order should have access to the reserved liquidity by determining the appropriate priority.

Reservation can be effected by RTGS participants or other actors that have the appropriate access rights using U2A or A2A. Further details on the U2A functionality can be found in the RTGS User Handbook.

In case of e.g. technical problems faced by an RTGS participant, the responsible central bank can act on behalf of this RTGS participant.

In general, RTGS participants have the possibility to

- create or to modify reservations with immediate effect during the current business day as a one-time reservation in the RTGS component. This includes
  - establishing a specific amount during the current day with immediate effect as a one-time reservation.
  - "resetting" to zero the liquidity reserved for the current business day only with immediate effect.
  - changing the amount on demand during the day with immediate effect.
- create, modify or delete a standing order reservation in CRDM valid from the following business day(s) (i.e. valid as of the next business day until next change).

In case the available liquidity on the RTGS dedicated cash account is lower than the amount to be reserved, the part which can be reserved will be reserved and the remaining part of the reservation will be queued (i.e. the pending value) and the RTGS component will process it in an event-oriented manner. Consequently, in case of incoming credits, the RTGS component will decrease the pending value and increase the respective reservation accordingly.

Standing order reservation

Standing order reservations are created and managed in CRDM. The definition of standing order reservations is only possible for RTGS dedicated cash accounts and not for sub-accounts.

The amount defined in the standing order for reservation will be valid at the start-of-day 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).

It is possible to have a standing order for the two types of reservations at the same time. Consequently, the RTGS participant can have an urgent reserve and a high reserve in parallel. At the start-of-day, reservations are set according to the standing orders and up to the available balance on the RTGS dedicated cash account.

One-time reservation with immediate effect

One-time reservation are created and managed directly in the RTGS component. The definition of such reservations is only possible for RTGS dedicated cash accounts and not for sub-accounts.

As outlined above it is possible to create a reservation for the current business day only. Moreover, it is possible to modify an existing reservation and to "reset to zero" the amount of the reservation with immediate effect for the current business day only. Owing to the asynchronous processing in the RTGS component, incoming liquidity might be blocked and used by a parallel booking process before the attempt to increase the reservation has been performed.

Upon receipt of

- end of day notification,
- a reservation revocation or

- a new reservation order,

the RTGS component stops processing the original reservation order, i.e. the new reservation replaces the pending one.

5.4.3.1.2 Liquidity reservation and management process (completed)

Reservation process – one time reservation with immediate effect

The following message flows illustrate the reservation creation (ModifyReservation (camt.048) [331]), the amendment (ModifyReservation (camt.048) [331]) and the “reset to zero” (DeleteReservation (camt.049) [335]) in the RTGS component.

In case an RTGS participant wants to query a reservation, this can be done in A2A (see chapter Query management for RTGS, CRDM, scheduler and billing [178]) as well as in U2A.

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

Process description

Table 68 - Create one-time liquidity reservation with immediate effect

Step	Processing in/between	Description
1	RTGS participant via ESMIG to the RTGS component	RTGS participant A sends a camt.048 via ESMIG to the RTGS component
2	RTGS component	RTGS message check and validation. In case of a negative technical business validation an admi.007 is sent. In case of negative business validation a camt.025 is sent. In case of a successful validation, execution of the one-time reservation request.
3	RTGS component via ESMIG to RTGS participant A	In case of (partial) execution of the reservation a camt.025 is created and sent via ESMIG to RTGS participant A. Note: In case of an immediate reservation sent by an actor different from the account owner, also the sender of the camt.048 receives a camt.025.
4	RTGS component	The remaining reservation request (i.e. the pending value) will be queued and processed in an event-oriented way. In case of an increase of the available liquidity an asynchronous resolving process attempts to process the pending reservation order. New reservation requests related to the RTGS dedicated cash account replace already pending reservation requests. **Note**: Even if the increase of available liquidity is not sufficient for the complete processing, the pending reservation will be processed partly (the pending reservation is decreased and the existing reservation is increased).

Used messages

- ModifyReservation (camt.048) [▸ 331]
- Receipt (camt.025) [▸ 318]
- ReceiptAcknowledgement (admi.007) [▸ 276]

Process description

Table 69 - Modify one-time liquidity reservations with immediate effect

Step	Processing in/between	Description
1	RTGS participant via ESMIG to RTGS component	RTGS participant A sends a camt.048 via ESMIG to the RTGS component in order to modify the reservation with immediate effect
2	RTGS component	RTGS message check and validation. In case of a negative technical business validation an admi.007 is sent. In case of negative business validation a camt.025 is sent. In case of a successful validation, execution of modification request.
3	RTGS component via ESMIG to RTGS participant A	In case of execution of the modification of the reservation a camt.025 is created and sent via ESMIG to RTGS participant A. Note: In case of the increase of the reservation was sent by an entity different from the account owner, also the sender of the camt.048 receives a camt.025.
4	RTGS component	The remaining reservation request (i.e. increase) will be queued and processed in an event oriented way. In case of an increase of the available liquidity an asynchronous resolving process attempts to process the pending reservation order. New reservation requests related to the RTGS dedicated cash account replace pending reservation requests. **Note**: - Even if the increase of available liquidity is not sufficient for the complete processing the pending reservation will be processed partly (the pending reservation is decreased and the existing reservation is increased). - In case the RTGS participant sends a deletion, this deletion is taken into account and replaces the pending reservation request.

Used messages

- ModifyReservation (camt.048) [▸ 331]
- Receipt (camt.025) [▸ 318]
- ReceiptAcknowledgement (admi.007) [▸ 276]

“Resetting to zero”

Note: Owing to the fact that the messages used are the same for one-time reservation with immediate effect and standing order reservation, in principle the message flow applies for both cases.

Process description

Table 70 - “Resetting to zero” of a reservation

Step	Processing in/between	Description
1	RTGS participant via ESMIG to RTGS component	RTGS participant A sends a camt.049 via ESMIG to the RTGS component in order to reset the reservation to zero.
2	RTGS component	RTGS message check and validation. In case of a negative technical validation, an error message (admi.007) is sent. In case of negative business validation a camt.025 is sent. In case of a successful validation the reservation will be set to zero for the current business day and a camt.025 is sent to the RTGS participant A. Note: In case the resetting to zero was sent by an actor different from the account owner, also the sender receives a camt.025.

Used messages

- DeleteReservation (camt.049) [▸ 335]
- Receipt (camt.025) [▸ 318]
- ReceiptAcknowledgement (admi.007) [▸ 276]

5.4.3.1.3 Effect of liquidity reservation (completed)

The following tables explain the effect of the reservation functionality for the processing of credit transfers in the RTGS component and provide a numeric example:

Table 71 - Effect of reservations for payment procession

Effect	Urgent payment	High payment	Normal payment
Available liquidity for settlement of payment orders	Balance on RTGS dedicated cash account	Balance on RTGS dedicated cash account minus urgent reserve	Balance on RTGS dedicated cash account minus urgent reserve minus high reserve
Effect of outgoing payments	• Reduction of balance on RTGS dedicated cash account • Reduction of urgent reserve • If the urgent reserve is not sufficient, the liquidity will be used as follows: – Available liquidity for normal payments. – Reduction of the high reserve.	• Reduction of balance on RTGS dedicated cash account • Reduction of high reserve • If the high reserve is not sufficient, the available liquidity for normal payments will be used.	• Reduction of balance on RTGS dedicated cash account
Effect of incoming payments	Increase of balance on RTGS dedicated cash account	Increase of balance on RTGS dedicated cash account	Increase of balance on RTGS dedicated cash account

Note: Directs debit effect the liquidity reservation the other way round.

Table 72 - Usage of urgent and high reserve – numeric example

Activity	Balance on RTGS dedicated cash account	Urgent reserve	High reserve	Available liquidity for normal payments
Start	1,000	100	200	700
Settlement of ancillary system = 50 (debit)	950 ⇩	50 ⇩	200 ⇔	700 ⇔
Submitting high payment to bank B = 200	750 ⇩	50 ⇔	0 ⇩	700 ⇔
Submitting normal payment to bank C = 20	730 ⇩	50 ⇔	0 ⇔	680 ⇩

Activity	Balance on RTGS dedicated cash account	Urgent reserve	High reserve	Available liquidity for normal payments
Settlement of ancillary system = 100 (credit)	830 ⇧	50 ⇔	0 ⇔	780 ⇧
Incoming high payment from bank B = 50	880 ⇧	50 ⇔	0 ⇔	830 ⇧
Incoming normal payment from bank C = 30	910 ⇧	50 ⇔	0 ⇔	860 ⇧
Set a new high reservation with immediate effect = 500	910 ⇔	50 ⇔	500 ⇧	360 ⇩
Settlement of urgent payment in favour of central bank = 450 (debit)	460 ⇩	0 ⇩	460 ⇩	0 ⇩

5.4.3.2 Limits

5.4.3.2.1 Overview (completed)

In general, limits determine the amount of liquidity an RTGS participant is willing to accept as liquidity outflow for settling credit transfers with priority normal which are to be debited on his RTGS dedicated cash account.

The following types of limits can be used in the RTGS component:

- bilateral limit
- multilateral limit

The limits are debit limits and not credit limits, i.e. they define the amount an RTGS participant is willing to pay

- to another RTGS participant in case of a bilateral limit or
- to all other RTGS participants towards which no bilateral limit has been defined

without receiving any incoming payments (i.e. incoming credit transfers) first.

Limits can be defined and managed by RTGS participants or other actors that have the appropriate access rights using U2A or A2A. Further details on the U2A functionality can be found in the RTGS User Handbook.

Limits are set up at account level, i.e. a bilateral/multilateral limit applies for payments processed on one specific RTGS dedicated cash account only.

At the start-of-day, limits are set according to the standing orders (so called defined limit) and are updated throughout the business day after each relevant credit and debit (so called free limit position). As a consequence, a normal payment will only be settled if it does not cause a breach of the free limit position. In case no limit is defined, the RTGS participant's liquidity available for the respective priority is available for each payment.

In general, RTGS participants have the possibility to

- modify limits with immediate effect during the day trade settlement phase in the RTGS component. The modification of limits with immediate effect includes the increase, the decrease and the reduction to zero. If a limit is set to zero, it is not possible to increase it again on the same business day.
- create, modify or delete a defined limit in CRDM valid from the following business day(s) (i.e. valid as of the next business day until next change).

The limitation process consists of the following elements:

- definition of bilateral limits towards selected RTGS participants with account type "RTGS dedicated cash account"
- definition of a multilateral limit towards all RTGS participants towards which no bilateral limit is defined

Objectives for the use of limits

The setting of the limits enables the RTGS participant

- to ensure an early submission of normal payments with full control of the liquidity outflow at the same time
- to avoid free-riding on the liquidity of one RTGS participant by another RTGS participant
- to synchronise the payment flow with other RTGS participants and to promote its early submission

5.4.3.2.1.1 Bilateral limits (completed)

Bilateral position

The bilateral position from RTGS participant A towards RTGS participant B is defined as the sum of payments received from RTGS participant B (i.e. credits for RTGS participant A) minus the sum of payments made to RTGS participant B (debits for RTGS participant A). This means if the result is negative, the bilateral limit will be utilised with this amount.

Effect of bilateral limit

With the bilateral limit, the RTGS participant restricts the use of liquidity when submitting payments for another RTGS participant. Direct debits effect the bilateral position just the other way round as in case of direct debits outgoing payments are credits and incoming payments are debits.

Once a defined bilateral limit has been created in CRDM and is taken into account during the start-of-day for the current business day, the defined limit can be changed directly in RTGS with immediate effect throughout the business day.

5.4.3.2.1.2 Multilateral limits (completed)

Multilateral position

The multilateral position from RTGS participant A is defined as the sum of payments (credits for RTGS participant A) received from all RTGS participants towards which no bilateral limit has been defined, minus the sum of payments (debits for RTGS participant A) made to these RTGS participants. This means if the result is negative, the multilateral limit is utilised with this amount.

Effect of multilateral limit

With the multilateral limit, the RTGS participant restricts the use of liquidity, when submitting payments for any other RTGS participant for which a bilateral limit has not been set.

Direct debits effect the multilateral position just the other way round because outgoing payments are credits and incoming payments are debits.

5.4.3.2.1.3 Rules for definition of limits (completed)

The creation of standing order limits is done in CRDM and the definition is done per RTGS dedicated cash account.

Changes and “resetting to zero” of bilateral and multilateral limits with immediate effect for the current business day are done in the RTGS component directly.

The following general rules apply:

- The minimum amount of a limit is one million.
- It is not possible to define a bilateral limit vis-à-vis central banks. For central bank accounts is not possible to define limits.
- A bilateral or multilateral limit with an amount of zero is a limit which is considered as “not defined”.
- A multilateral limit can be defined if at least one bilateral limit exists.

- Incoming urgent and high payments (i.e. credits) from an RTGS participant towards which a bilateral/multilateral limit is defined, increase the free limit position.

In order to take into account a defined limit (bilateral or multilateral) for the settlement of payments, the defined limit needs to be defined before the end of the previous business day. This means that an amount above zero has to be defined at the latest before the end of the previous business day.

Once a defined multilateral limit has been created in CRDM and is taken into account during the start-of-day for the current business day, the defined limit can be changed directly in RTGS with immediate effect throughout the business day.

5.4.3.2.2 Process for the definition and management of limits (completed)

The creation, the amendment and the deletion of a defined limit (i.e. the standing order for limits) is managed in CRDM.

The following message flow illustrates the amendment (camt.011) or “resetting to zero” (camt.012) with immediate effect for the current business day in the RTGS component.

Case: limit amendment/deletion message with positive validation

Process description

Table 73 - Limit management – positive validation

Step	Processing in/between	Description
1	RTGS participant via ESMIG to the RTGS component	RTGS participant A sends a camt.011 / camt.012 via ESMIG to the RTGS component
2	RTGS component	RTGS message check and validation positive Execution of reservation request (amendment or deletion)
3	RTGS via ESMIG to RTGS participant A	Creation and forwarding of camt.025 by the RTGS component via ESMIG to RTGS participant A

Used messages

- ModifyLimit (camt.011) [▸ 307]
- DeleteLimit (camt.012) [▸ 308]
- Receipt (camt.025) [▸ 318]

Case: limit amendment/deletion message with negative validation

Process description

Table 74 - Limit management – negative validation

Step	Processing in/between	Description
1	RTGS participant via ESMIG to the RTGS component	RTGS participant A sends a camt.011 / camt.012 via ESMIG to the RTGS component
2	RTGS component	RTGS message check and validation negative
3	RTGS component via ESMIG to RTGS participant A	Creation and forwarding of camt.025 by the RTGS component via ESMIG to RTGS participant A (mandatory)

Used messages

- ModifyLimit (camt.011) [▸ 307]
- DeleteLimit (camt.012) [▸ 308]
- Receipt (camt.025) [▸ 318]

Initiator of limit setting and changing

Limits are exclusively set by RTGS participants. Only in the case of a technical problem on the RTGS participant's side, the responsible central bank can be authorised to adjust the amount of a limit with effect for the next algorithm.

5.4.3.2.3 Effect of limits (completed)

General effect of limits

The following table explains the effects of limits on the processing and subsequent settlement of payments.

Table 75 - Effects of limits

Normal payment	
Available liquidity for settlement of normal payments	Balance on RTGS dedicated cash account minus urgent reserve minus high reserve
Effect of outgoing payments (i.e. debits on the RTGS dedicated cash account*)	▪ Reduction of balance on RTGS dedicated cash account ▪ Reduction of bilateral or multilateral position (Payments will be queued, if the amount of the normal payment is higher than the Free Limit Position)
Effect of incoming payments (i.e. credits on the RTGS dedicated cash account [1])	▪ Increase of balance on RTGS dedicated cash account ▪ Increase of the Free Limit Position

Bilateral limit

The processing of normal payments in case RTGS participant A has set a bilateral limit for RTGS participant B is illustrated in the following simplified example

1 Direct debits effect the bilateral/multilateral position just the other way round because outgoing payments are credits and incoming payments are debits.

Table 76 - Processing in case of bilateral limit

Bilateral relation	Bilateral limit set	Submitted normal payments	Explanation
RTGS participant A vis-à-vis RTGS participant B	3 million EUR	10 million EUR	Up to a maximum of 3 million EUR of RTGS participant A's liquidity will be used to settle normal payments between RTGS participant A and RTGS participant B.
RTGS participant B vis-à-vis RTGS participant A	Not relevant in this example	6 million EUR	If RTGS participant A has sufficient liquidity available, a maximum of 9 million EUR from RTGS participant A and 6 million EUR from RTGS participant B can be settled. 1 million EUR from bank A cannot be settled and are queued until - additional payments (high/normal) from RTGS participant B will be settled or - RTGS participant A increases the bilateral limit to an amount of 4 million EUR or sets the bilateral limit to zero. Otherwise the normal payments will not be settled and will be rejected by the end of the day.

Multilateral limit

The processing of normal payments in the case of bank A has set a multilateral limit is illustrated in a following simplified example (bank A has not defined bilateral limits vis-à-vis those banks).

Table 77 - Processing in case of multilateral limits

Multilateral relation	Multilateral limit set	Submitted normal payments	Explanation
RTGS participant A vis-à-vis RTGS participants C, D, E, ...	2 million EUR	20 million EUR	Up to a maximum of 2 million EUR of RTGS participant A's liquidity will be used to settle payments between RTGS participant A and RTGS participants C, D, E, ...
RTGS participants C, D, E, ... vis-à-vis RTGS participant A	Not relevant in this example	15 million EUR	If RTGS participant A has sufficient liquidity available, a maximum of 17 million EUR from RTGS participant A and 15 million EUR from RTGS participants C, D, E, ... can be settled. 3 million EUR from RTGS participant A cannot be settled and are queued until - additional payments (high/normal) of RTGS participants C, D, E, ... will be settled or - RTGS participant A increases the multilateral limit to an amount of 5 million EUR or sets the limits to zero. Otherwise the normal payments will not be settled and rejected by the end of the day.

5.4.3.3 Dedication of liquidity for ancillary system settlement (completed)

For the settlement of ancillary systems the RTGS participant can "set aside" liquidity for this purpose only.

Depending on the settlement procedure the ancillary system is using, the liquidity needs to be provided on different accounts:

- sub-account for the procedure "settlement on dedicated liquidity accounts (interfaced)"(account owner = RTGS participant)
- dedicated liquidity account for procedure "settlement on dedicated liquidity accounts (real-time)"(account owner = ancillary system or its central bank)

Moreover, the RTGS participant can open a dedicated RTGS dedicated cash account (account owner = RTGS participant) which is used for ancillary system settlement only.

To transfer liquidity to the RTGS participant's sub-account or to the dedicated liquidity account, the following possibilities can be used:

- Setting up standing orders in CRDM. These will become effective with the next business day.

- Current order liquidity transfers using camt.050 LiquidityCreditTransfer messages or via dedicated RTGS GUI liquidity transfer screens
- Current order liquidity transfers initiated by the ancillary system using ASTransferInitiation messages debiting the settlement banks RTGS dedicated cash account and crediting the settlement bank's sub-account (procedure interface) or the dedicated liquidity account (procedure real-time)

Standing orders are executed with each start of a procedure (mandatory and optional). Different amounts for both procedures can be specified. Further details can be found in chapter Settlement of ancillary systems [▸ 103]. Current order liquidity transfers will be executed with immediate effect during an open procedure with no cycle running. In the opposite case, where a cycle is running, the liquidity transfer will be stored and executed only once the cycle has closed.

In case the available liquidity on the RTGS dedicated cash account is not sufficient, the following shall apply:

- if the total sum of all standing orders of a settlement bank is larger than the liquidity on its RTGS dedicated cash account, all standing orders will be reduced in a pro-rata mode, i.e. the existing liquidity is divided by the total sum of standing orders and the resulting factor will be used to reduce each standing order of this participant (mandatory procedure). In optional procedure the standing order will be rejected
- a current order initiated by the settlement bank will be rejected (mandatory and optional procedure)
- a current order initiated by the ancillary system (or central bank on behalf) will be partially settled up to the available liquidity on the RTGS dedicated cash account (mandatory and optional procedure)

5.4.3.4 Floor/ceiling

5.4.3.4.1 Definition of floor/ceiling threshold (completed)

The RTGS component can generate a floor/ceiling notification related to an RTGS dedicated cash account in case a floor/ceiling threshold has been defined in advance. In case such threshold has been defined, the sending of a floor/ceiling notification will be triggered by the RTGS component after the successful settlement of a payment or ancillary system related payment instruction whenever the amount on the RTGS dedicated cash account undercuts the floor amount or exceeds the ceiling amount.

Since this functionality is optional, it is up to the holder of the RTGS dedicated cash account (i.e. the RTGS participant) to define a floor/ceiling threshold in CRDM.

The holder of the RTGS dedicated cash account can define a minimum ("floor") or maximum ("ceiling") amount for its RTGS dedicated cash account(s). The RTGS participant has the option to choose what shall be done by the RTGS component once the balance is below the defined floor or above the defined ceiling amount.

Two options are available:

1. the RTGS component generates a notification to be sent to the RTGS participant as the owner of the RTGS dedicated cash account informing about the floor/ceiling breach (upon which the RTGS participant can take action); or
2. the RTGS component automatically generates an inter-service liquidity transfer to pull liquidity from the linked main cash account in case the floor is breached on the RTGS dedicated cash account or the RTGS component pushes liquidity to the linked main cash account in case the ceiling threshold was reached. When using this functionality, the RTGS dedicated cash account holder needs to define also a target floor amount and a target ceiling amount for its RTGS dedicated cash account.

The floor / ceiling functionality itself will only be triggered after the settlement of a payment or a payment instruction stemming from the settlement of ancillary systems. It is not triggered for liquidity transfers.

5.4.3.4.2 Breach of floor/ceiling threshold – notification (completed)

If the RTGS participant choses the first option, the RTGS component generates and sends out a notification with the information that the account balance is below the floor or that the account balance is above the ceiling respectively

- in U2A (please refer to the respective part of the RTGS User Handbook) or
- in A2A mode (ReturnAccount (camt.004) [▸ 282], Floor and ceiling processing [▸ 242])

The notification will be sent every time, the threshold is undercut (floor) or exceeded (ceiling). However, the RTGS component does not send the notification if, after trespassing the threshold, the balance of the RTGS dedicated cash account remains consistently below the floor or above the ceiling threshold defined.

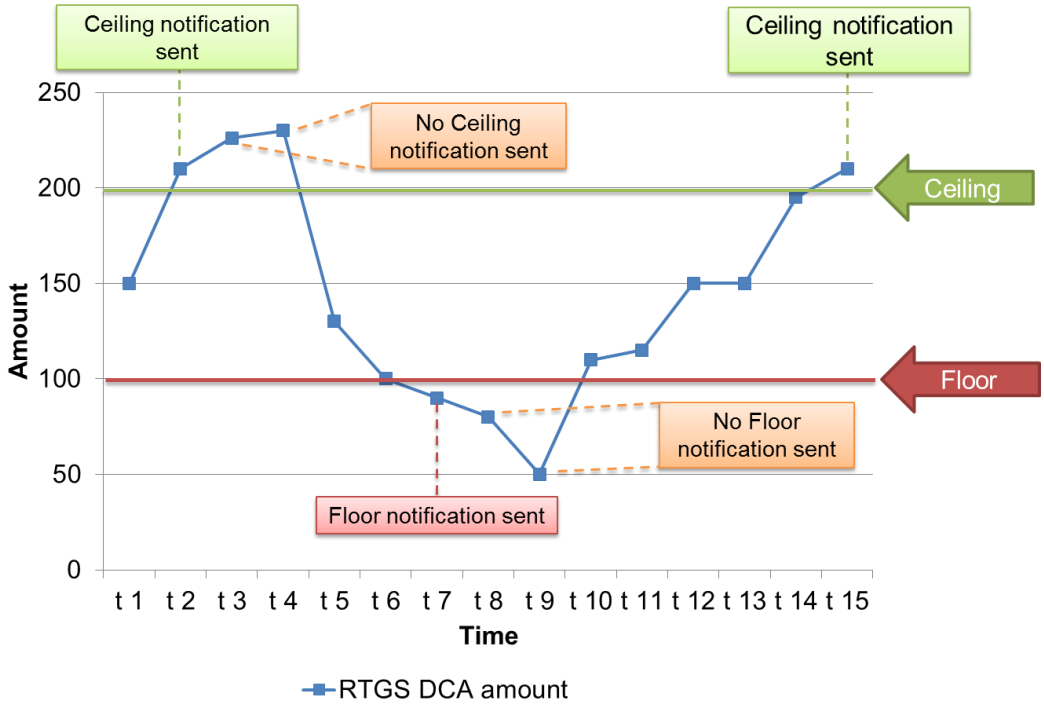


Figure 20 - Breach of floor/ceiling threshold - notification

5.4.3.4.3 Breach of floor/ceiling threshold - automatic liquidity transfer (completed)

If the RTGS participant choses the second option, the RTGS component creates and releases an inter-service liquidity transfer.

- In case of a breach of the floor threshold the needed amount will be pulled from the main cash account and credited on the RTGS dedicated cash account.
- The used main cash account will be the one linked to the RTGS dedicated cash account as defined in CRDM.
- The amount to be transferred is the difference between the current balance on the RTGS dedicated cash account and the predefined target amount. The target floor amount could be different, but will in any case be equal or above the floor amount.
- In case of a breach of the ceiling threshold the amount will be pushed to the main cash account in CLM where it will be credited and the RTGS dedicated cash account will be debited.
- The used RTGS dedicated cash account will be the same as for the floor threshold meaning it will be the one linked to the main cash account as defined in CRDM.

- The amount to be transferred to the main cash account is the difference between the current balance and the predefined target ceiling amount. The target ceiling amount could be different but will be below the ceiling amount.
- The target amount for ceiling will be a different one as the target amount of the floor threshold.

After the successful execution of the inter-service liquidity transfer, the amount on the RTGS dedicated cash account will be again within the boundaries of the floor or ceiling amount.

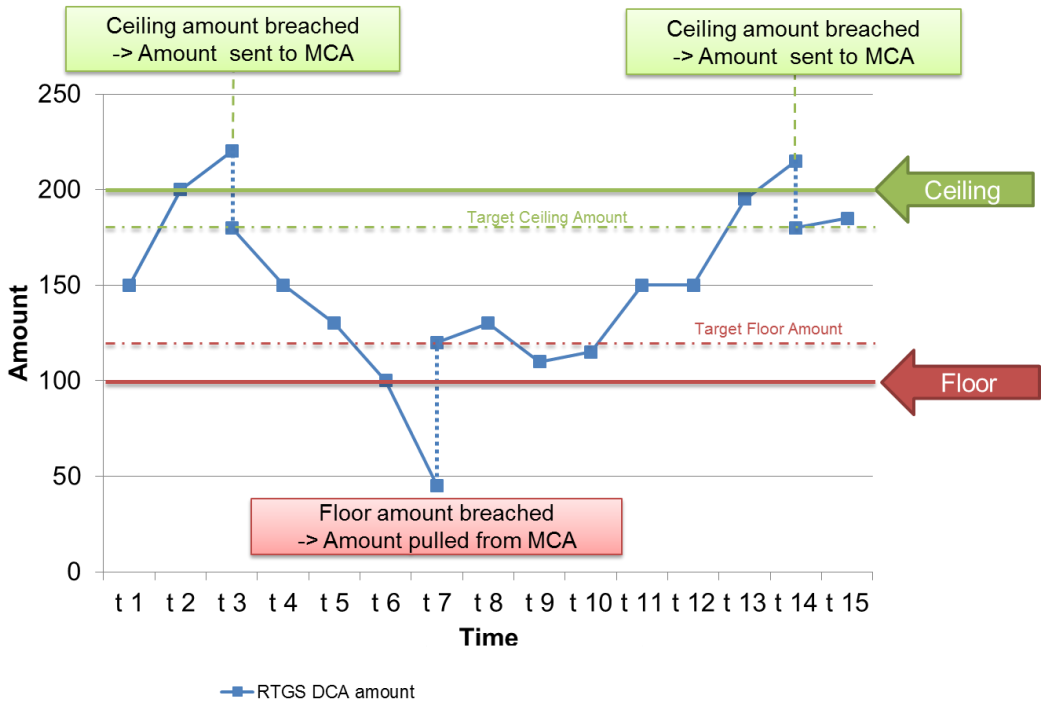


Figure 21 - Breach of floor/ceiling threshold – automatic liquidity transfer

5.5 Information management for RTGS

5.5.1 RTGS status management (completed)

5.5.1.1 Concept (completed)

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

5.5.1.2 Overview (completed)

The status management process manages the status updates of the different instructions (e.g. payment, liquidity transfers, amendment instructions) existing in RTGS in order to communicate relevant status updates via status advice messages to the RTGS actors throughout the lifecycle of the instruction. Some status notifications are mandatory, others are provided on optional basis. Status information on push basis is only available in A2A mode. Respective status advice messages are pushed via store-n-forward network service. For exceptional business cases notifications in U2A are foreseen.

The status management handling also provides the reason codes to be sent to RTGS actors in case of negative result of an RTGS component process (e.g. to determine the reason why an instruction is unsuccessfully validated or settled).

The status of an instruction is indicated through a value, which is subject to change through the lifecycle of the instruction. This value provides RTGS actors with information about the situation of the instruction with respect to a given RTGS process at a certain point in time.

Since each instruction in the RTGS can be submitted to several processes, each instruction in RTGS has several status. However, each of these status has one single value at a certain moment in time that indicates the instruction's situation at the considered moment. Depending on its instruction type, an instruction is submitted to different processes in RTGS. Consequently, the status featuring each instruction depend on the considered instruction type.

The following sections provide:

- the generic principles for the communication of status and reason codes to RTGS actors
- the list of status featuring each instruction type as well as the possible values for each of these status

Reason codes are not exhaustively detailed below but are provided in chapter Index of status value and codes [▸ 422].

5.5.1.3 Status management process (completed)

Communication of status and reason codes to RTGS actors

RTGS actors can query during the day the status values and reason codes of their instructions (e.g. payments, liquidity transfers, tasks, reference data updates).

The status can be classified into two different types, common to all types of instructions:

- "Intermediate status". In general an instruction will have more than one status in its lifetime. If the status of an instruction is not a final status type, then the instruction is still under process in RTGS. With each step in the process of the instruction the status will change until a final status is reached. Further status updates are communicated to the RTGS actor if reached.

- "Final status". This is the last status of an instruction (i.e. the status that an instruction has when processing for that instruction ends). At a point in time, any instruction in RTGS reaches a final status, all respective processes are completed.

For some specific status updates, the status management process informs the RTGS actor of the status change by means of the sending of status advice messages (according to their message subscription configuration).

Status and status values in RTGS

The detailed status concept will be provided in iteration 4.

5.5.2 RTGS report generation (completed)

5.5.2.1 Concept (completed)

RTGS provides the possibility to periodically create the predefined report "Statement of account". RTGS triggers the generation of the "Statement of account" report based on the reference data configuration. It is only foreseen at business event "End of day". The report is not created intraday. Depending on the direct RTGS participant's preferences the report is either sent out directly after creation or stored for later retrieval.

Table 78 - Report "Statement of accounts"

Report name	ISO message	ISO code
Statement of accounts	BankToCustomerStatement	BankToCustomerStatement (camt.053) [343]

The respective business process is described in chapter Receive report [250].

5.5.2.2 Overview (completed)

The report "Statement of account" includes information on one single RTGS dedicated cash account of a direct RTGS participant. It is not possible to receive one combined "Statement of account" for more than one RTGS dedicated cash account. Furthermore it does not include information from other component, i.e. there is no report including combined information of CLM and RTGS.

The report provides information about all items that have been booked on the RTGS dedicated cash account and balance information of the current business day.

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

The configuration of a report is independent from the message subscription for notifications, i.e. no message subscription reference data is needed in case the report should be sent (push mode).

5.5.2.3 Report generation process (completed)

Preconditions for report creation

In order to avoid unnecessary processing and storage RTGS does not create reports automatically. So, to initiate the creation of a report, the report receiver has to configure the report in advance. The configuration of the report has to be done via the graphical user interface for the reference data, which is described in the RTGS User Handbook.

This configuration is stored as reference data and is valid until the report receiver decides that the report has not to be created anymore or until the “valid to” date stored within the report configuration is reached.

Moment of data extraction

The creation of a “Statement of account” report is always triggered at the end of day of the RTGS component after finalization of booking processes [business event “EOD”]. A new report configuration can be set-up at the earliest for the next business day. The possible validity limitations have to be specified when the report is configured for the first time. The respective component only creates those reports, for which the underlying report configurations is valid at the current business day.

Availability of the report in RTGS

A generated report is available for download until it is replaced by a new (next) version of it, i.e. a report that is created at the end of day of the current business day replaces the report that was created at the end of day of the previous business day. The replaced report is no longer available for download in RTGS. In A2A mode RTGS pushes the specific report, provided that the push preference for the report is stored in for the respective recipient reference data (i.e. report configuration). The message is sent out based on the routing information stored for the direct RTGS participant. Otherwise the report is just stored after generation and can be downloaded in pull mode via U2A. Additionally a resend request allows the actor to initiate a re-delivery of the last report, which was pushed before.

CRDM parameter synthesis

The following parameters are created and updated by the CRDM actor (see Table 85 - Report configuration [▸ 190]) for the setup of a report.

Table 79 - CRDM parameter synthesis

Parameter	Mandatory/ optional	Possible values	Hint
Report type	Mandatory	Statement of accounts	
Concerned account	Mandatory	RTGS dedicated cash account	
Possible recipient of a report	Mandatory	Direct RTGS participant	
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.

Concerned account

Each report provides information on a certain scope of data. The data scope is indicated by the RTGS dedicated cash account for which it is configured. The concerned account has to be specified, when the report is configured for the first time. It is necessary to store one configuration per RTGS dedicated cash account 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 (e.g. co-manager)

A created report can be received by one or several receivers. Each direct RTGS participant can decide, if it wishes to receive a report directly after its creation or if it wants to query it ad-hoc via U2A.

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.

If a recipient does not wish to receive a report directly after its creation but to request it afterwards, this RTGS behaviour has to be stored in the reference data configuration of the report as well. Furthermore this recipient is stored as recipient of a report.

As a general principle the recipient(s) of a report can be different from the concerned account owner, but has to be configured in the same system entity. For information about the setup of report configuration for specif-

ic concerned account owners and recipients of a report, please see RTGS UHB chapters related to report configuration setup.

It is allowed to request a resending of the currently available statement of account. Hereafter, the respective cases are described:

Case: resend request with positive validation and re-delivery

A resend request allows delivering the report message once more to the same technical address as used for the initial report delivery.

Message flow

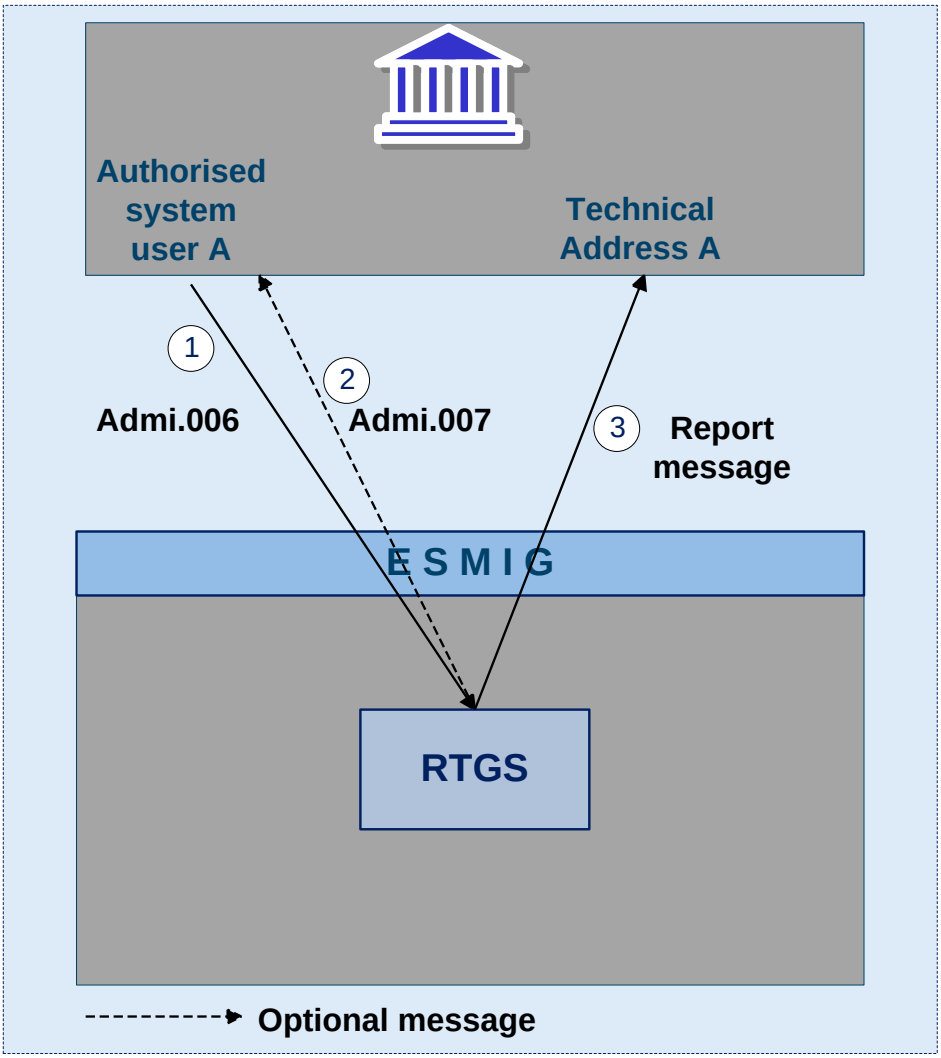


Figure 22 - Resend request with positive validation and re-delivery

Process description

Step	Processing in/between	Description
1	Direct RTGS participant via ESMIG to RTGS	An authorised system user of a direct RTGS participant A sends a admi.006 via ESMIG to RTGS
2	RTGS	RTGS message check and validation positive
3	RTGS via ESMIG to direct RTGS participant	Admi.007 including positive validation result via ESMIG to direct RTGS participant A generated by RTGS (optional)
4	RTGS via ESMIG to direct RTGS participant	Re-delivery of report message camt.053 to the original technical address (mandatory)

Used messages

- BankToCustomerStatement (camt.053) [▸ 343]
- ResendRequest (admi.006) [▸ 273]
- ReceiptAcknowledgement (admi.007) [▸ 276]

Case: resend request with negative validation

Message flow

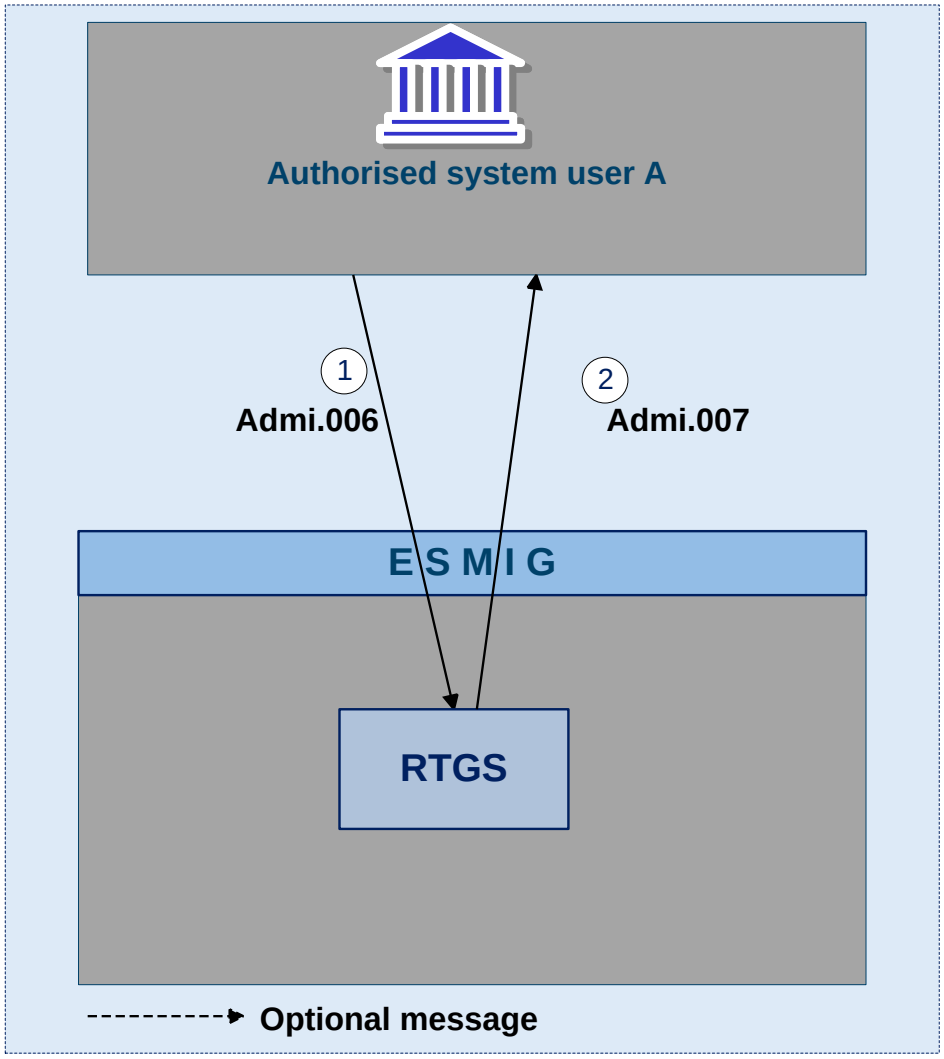


Figure 23 - Resend request with negative validation

Process description

Step	Processing in/between	Description
1	Direct RTGS participant via ESMIG to RTGS	An authorised system user of a direct RTGS participant A sends a admi.006 via ESMIG to RTGS
2	RTGS	RTGS message check and validation negative
3	RTGS via ESMIG to direct RTGS participant	Admi.007 including negative validation result via ESMIG to direct RTGS participant A generated by RTGS (mandatory)

Used messages

- ResendRequest (admi.006) [▸ 273]
- ReceiptAcknowledgement (admi.007) [▸ 276]

5.5.3 Query management for RTGS, CRDM, scheduler and billing (completed)

5.5.3.1 Concept for RTGS, CRDM, scheduler and billing (completed)

Queries are provided by RTGS, CRDM, scheduler and billing to the submitting actoras a means of satisfying his information needs on demand. The submitting actor can obtain information on different business items by submitting query requests to the mentioned components. These are answered on the basis of the latest data available.

For requests on RTGS only 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 inchapter Execute query [▸ 246].

5.5.3.2 Overview for RTGS, CRDM, scheduler and billing (completed)

RTGS, CRDM, Scheduler and Billing provide 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 actors of the respective service/component.

They can send query requests to service/components in A2A mode or in U2A mode. Generally, all these query requests are processed in real-time. Exceptions occur during the maintenance window. During the maintenance window query management does not service any requests. In case ESMIG is available and the network interface is not closed, an A2A query request during business service maintenance window will be handled by using timeout management. In case the network interface is closed, the network service provider informs the authorised submitting actor about the closure of the real-time channel.

5.5.3.3 Query management process for RTGS, CRDM, scheduler and billing (completed)

Initiating queries for RTGS, CRDM, scheduler and billing

In order to obtain the desired information the submitting actor needs to submit a query request to a component. For the communication with components in A2A mode all query and response messages are set up as XML messages compliant with the ISO20022 standard. For the communication with components in U2A mode a graphical user interface based on a standard browser application is provided.

In general an authorised submitting actor can send each query request in A2A mode as well as in U2A mode. However, there are some queries which are only accessible via U2A mode. Query availability in the respective communication mode is shown in the table below. Query request and return criteria are described in detail in RTGS User Handbook for U2A mode and in chapter 11 with link to MyStandards for A2A mode.

Table 80 - Initiating queries for RTGS, CRDM, scheduler and billing

Related component	Query type	Initiation via GUI (U2A mode)	Initiation via XML message (A2A mode)
RTGS	Account balance query	X	X
RTGS	Account statement query	X	-
RTGS	Audit trail for RTGS query	X	X
RTGS	Broadcast 2uery	X	-
RTGS	Current limits query	X	X
RTGS	Current reservations query	X	X
RTGS	Message query	X	-
RTGS	Payment query	X	X
CRDM	Ancillary system reference data query	X	X
CRDM	Ancillary system settlement bank reference data query	X	X
CRDM	Audit trail for CRDM query	X	X
CRDM	Calendar query	X	X
CRDM	Central bank query	X	-
CRDM	Direct debit mandate query	X	X
CRDM	Directory Query	X	X
CRDM	Error code query	X	X
CRDM	Event query	X	X
CRDM	Liquidity transfer group query	X	-
CRDM	Message subscription query	X	-
CRDM	Participant reference data query	X	X
CRDM	Party reference data query	X	X

Related component	Query type	Initiation via GUI (U2A mode)	Initiation via XML message (A2A mode)
CRDM	Role query	X	-
CRDM	RTGS cash account reference data query	X	X
CRDM	Sub account reference data query	X	X
CRDM	Standing order limits query	X	X
CRDM	Standing order liquidity transfer query	X	X
CRDM	Standing order reservations query	X	X
CRDM	User query	X	-
Scheduler	System time query	X	X
Billing	VAT query	X	-
Billing	Invoice query	X	-

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

Preconditions for successful processing of queries

The relevant component validates the plausibility of search criteria that were specified by the submitting actor. In addition, the relevant component ensures that the submitting actor of the query is allowed to initiate the query and to retrieve the requested information by checking, whether the submitting actor possesses all necessary privileges granted in advance (taking into account the validity dates) and ensuring the data scope.

Providing data for queries

If all checks performed by respective component were successful, it extracts the requested business information from the production data. The submitting actor receives the latest available data.

If one or more plausibility or authorisation checks performed by respective component fail, the submitting actor receives a response indicating the error that has occurred which is specified using the respective error code.

Retrieving the query response

In case the extraction of the query data is successful, the respective component sends a query response containing the requested business information back to the requesting actor. In case the extraction of the que-

ry data returns a zero result, the submitting actor receives appropriate information. If a retrieval of the query result fails, then an error response is provided to the submitting actor.

If the submitting actor has sent the query via U2A mode, the response is given to the same submitting actor in U2A mode. The U2A dialogue is described more in detail in the RTGS UHB.

If the submitting actor has sent the query via A2A mode, the response is given to the same component user in A2A mode. The respective component does not allow the routing of the query response to a dedicated technical address.

Parameter synthesis

No specific configuration from the submitting actor is needed.

6 Overview of used common components in RTGS component

6.1 CRDM features (completed)

6.1.1 Concept (completed)

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 (completed)

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 central banks, 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 central banks. 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.

CRDM common component executes immediately all reference data maintenance instructions. The related reference data changes become effective in the relevant TARGET service(s), 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(s).

All common reference data objects can be created and maintained in U2A mode, whereas only a sub-set of them can be maintained also through the 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 actors can specify the applicable mode for the functions and the common reference data objects they manage (see chapter Access rights [▸ 183]).

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 (completed)

This section provides information on access rights management in the CRDM. More into detail, chapter Access rights concepts [▸ 183] presents some basic concepts (e.g. user, privilege, role and data scope) related to access rights management. On this basis, chapter Access rights configuration [▸ 199] illustrates all the available options for the configuration of access rights. Finally, chapter Access rights configuration process [▸ 207] 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.

6.1.3.1 Access rights concepts (completed)

This chapter presents the main concepts related to access rights management in the CRDM.

6.1.3.1.1 User function (completed)

Data migration tool files, XML messages and GUI functions are the atomic elements users can trigger through the data migration tool 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 (completed)

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
- table other

Table 81 - Access rights management

Privilege	User function	Data scope
Administer party [2]	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 central banks) 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 central banks).
Create user	User – new	Users within own system entity (for central banks) or own party (for payment banks).
Create user certificate distinguish name link	User certificate DN link – new	Links within own system entity (for central banks) or for own users (for payment banks).
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 central banks) 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 central banks).
Delete user	User – delete/restore	Users within own system entity (for central banks) or own party (for payment banks).
Delete user certificate distinguish name link	User certificate DN link – delete/restore	Links within own system entity (for central banks) or for own users (for payment banks).

2 This privilege enables a user to act as party administrator for their own party.

Privilege	User function	Data scope
Grant privilege	Grant privilege	Privileges granted to parties, roles and users within own system entity (for central banks) or to own users (for payment banks)
Grant/revoke role	Grant/revoke role	Roles granted to parties and users within own system entity (for central banks) or to 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 central banks) 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 central banks)
Update User	User – edit	Users within own system entity (for central banks) or own party (for payment banks).

Table 82 - Party data management

Privilege	User function	Data scope
Create banking group	Banking group – new	Banking groups within own system entity (for central banks)
Create monetary financial institution	Monetary financial institution – new	Monetary financial institutions within own system entity (for central banks)
Create party	Party – new	Parties within own system entity (for central banks)
Create party-service link	Party-service link - new	Links within own system entity (for central banks)

Privilege	User function	Data scope
Create technical address network service link	Technical address network service link - new	Links within own system entity (for central banks)
Delete banking group	Banking group – delete/restore	Banking groups within own system entity (for central banks)
Delete monetary financial institution	Monetary financial institution – delete/restore	Monetary financial institutions within own system entity (for central banks)
Delete party	Party – delete/restore	Parties within own system entity (for central banks) excluding own party
Delete party-service link	Party-service link - delete/restore	Links within own system entity (for central banks)
Delete technical address networks service link	Technical address network service link - delete/restore	Links within own system entity (for central banks)
Update banking group	Banking group – edit	Banking groups within own system entity (for central banks)
Update monetary financial institution	Monetary financial institution – edit	Monetary financial institutions within own system entity (for central banks)
Update party	Party – edit	Parties within own system entity (for central banks)
Update party-service link	Party-service link - edit	Links within own system entity (for central banks)

Table 83 - Cash account data management

Privilege	User function	Data scope
Create account monitoring group	Account monitoring group – new	Account monitoring groups within own system entity (for central bank)
Create authorised account user	Authorised account user - new	Links within own system entity (for central bank) or for own cash accounts (for payment bank).
Create cash account	Cash account – new	Cash accounts within own system entity (for central bank) or CMBs linked to cash accounts owned by own party (for payment bank)

Privilege	User function	Data scope
Create direct debit mandate	Direct debit mandate - new	Direct debit mandates on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Create limit	Limit – new	Limits on CMBs defined on cash accounts within own system entity (for central bank) 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 central bank) or owned by own party (for payment bank)
Create liquidity transfer order group	Liquidity transfer order group – new	Liquidity transfer order groups containing liquidity transfer orders on cash accounts within own system entity (for central bank) 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 central bank) 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 central bank) 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 central bank)
Delete authorised account user	Authorised account user - delete/restore	Links within own system entity (for central bank) or for own cash accounts (for payment bank).
Delete cash Account	Cash account – delete/restore	Cash accounts within own system entity (for central bank) or CMBs linked to cash accounts owned by own party (for payment bank)

Privilege	User function	Data scope
Delete direct debit mandate	Direct debit mandate – delete/restore	Direct debit mandates on cash accounts within own system entity (for central bank) 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 central bank) 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 central bank) or owned by own party (for payment bank)
Delete liquidity transfer order group	Liquidity transfer order group – delete/restore	Liquidity transfer order groups containing liquidity transfer orders on cash accounts within own system entity (for central bank) 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 central bank) 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 central bank) or owned by own party (for payment bank)
Update account monitoring group	Account monitoring group – edit	Account monitoring groups within own system entity (for central bank)
Update authorised account user	Authorised account user - edit	Links within own system entity (for central bank) or for own cash accounts (for payment bank).
Update cash account	Cash account – edit	Cash accounts within own system entity (for central bans) or CMBs linked to cash accounts owned by own party (for payment bank)

Privilege	User function	Data scope
Update direct debit mandate	Direct debit mandate – edit	Direct debit mandates on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Update limit	Limit – edit	Limits on CMBs defined on cash accounts within own system entity (for central bank) 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 central bank) or owned by own party (for payment bank)
Update liquidity transfer order group	Liquidity transfer order group – edit	Liquidity transfer order groups containing liquidity transfer orders on cash accounts within own system entity (for central bank) 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 central bank) 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 central bank) or owned by own party (for payment bank)

Table 84 - Message subscription configuration

Privilege	User function	Data scope
Create message subscription rule	Message subscription rule – new	Message subscription rules within own system entity (for central banks) 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 central banks) or for own party (for payment banks)

Privilege	User function	Data scope
Delete message subscription rule	Message subscription rule – delete/restore	Message subscription rules within own system entity (for central banks) 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 central banks) or for own party (for payment banks)
Update message subscription rule	Message subscription rule – edit	Message subscription rules within own system entity (for central banks) 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 central banks) or for own party (for payment banks)

Table 85 - Report configuration

Privilege	User function	Data scope
Create report configuration	Report configuration – new	Report configurations within own system entity (for central banks) or for own party (for payment banks)
Delete report configuration	Report configuration – delete/restore	Report configurations within own system entity (for central banks) or for own party (for payment banks)
Update report configuration	Report configuration – edit	Report Configurations within own system entity (for central banks) or for own party (for payment banks)

Table 86 - Reference data queries

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 central banks) or for own cash accounts (for payment banks).
Banking group query	Banking group – list	Any banking group
BIC query	BIC query	Any BIC

Privilege	User function	Data scope
Cash account audit trail query	Revisions - selection criteria + list	Data within own system entity (for central bank) or linked to own party (for payment bank)
Cash account list query	Cash account list query	Cash accounts within own system entity (for central bank) 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 central bank) 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 central banks) or linked to own party (for payment banks)
Data changes of a business object list query	n/a	Data within own system entity (for central banks) 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 central bank) 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 central bank) 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 central banks) 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 central banks) or to own users (for payment banks)

Privilege	User function	Data scope
Granted system privileges list query	Grant/revoke system privileges list query	Privileges granted to parties, roles and users within own system entity (for central banks) or to own users (for payment banks)
Limit query	Limit query	Limits on CMB defined on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Liquidity transfer order details query	Liquidity transfer order – details	Liquidity transfer orders on sash accounts within own system entity (for central bank) 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 central bank) or owned by own party (for payment bank)
Liquidity transfer order group query	Liquidity transfer order group – list	Liquidity transfer order groups within own system entity (for central bank) or containing cash accounts owned by own party (for 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 central banks) 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 central banks) or for own party (for payment banks)

Privilege	User function	Data scope
Message subscription rule set list query	Message subscription rule set list query	Message subscriptions within own system entity (for central banks) or for own party (for payment banks)
Monetary financial institution query	Monetary financial institution – list	Any monetary financial institution
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 central bank) or linked to own party (for payment bank)
Party list query	Party list query	Parties within own system entity (for central bank) or own party (for payment bank)
Party reference data query	Party reference data query	Parties within own system entity (for central bank) or own party (for payment bank)
Party-service link list query	Party-service link list query	Links within own system entity (for central banks) or linked to own party (for payment banks)
Party-service link query	Party-service link query	Links within own system entity (for central banks) or linked to own party (for payment banks)
Privilege query	Privilege – selection criteria + list	Any privilege
Queued data changes query	Queued data changes – select + list	Data within own system entity (for central banks) or linked to own party (for payment banks)
Report configuration details query	Report configuration details query	Report configurations within own system entity (for central banks) or for own carty (for payment banks)
Report configuration list query	Report configuration list query	Report configurations within own system entity (for central banks) or for own party (for payment banks)
Residual static data audit trail query	Static data audit trail query	Data within own system entity (for central banks) or linked to own party (for payment banks)

Privilege	User function	Data scope
Role list query	Role list query	Roles created or granted to parties and users within own system entity (for central banks) 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 central bank) 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 central bank) 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 central bank) 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 central bank) or owned by own party (for payment bank)
System entity query	System entities – select + list	Own system entity (for central banks)
System user link query	System user link query	Links within own system entity (for central banks) 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 central banks) or linked to own party (for payment banks)

Table 87 - TIPS functions

Privilege	User function	Data scope
Adjust CMB limit	Adjust CMB limit	Data within own system entity (for central bank) 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 central bank) 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 central bank) 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 central bank) or linked to own party (for payment bank)
Modify CMB blocking status	Block/unblock CMB	Data within own system entity (for central bank) 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 central bank) 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 central bank) or for which own party is set as authorised user (for payment bank)

Table 88 - Other

Privilege	User function	Data scope
Data migration tool access	n/a	n/a

See chapter Configuration of privileges [▸ 199] for information on the configuration of privileges.

6.1.3.1.3 Role (completed)

A role is a set of privileges. See chapter Configuration of roles [▶ 206] for information on the configuration of roles.

6.1.3.1.4 User (completed)

A user is an individual or application that interacts with CRDM triggering the available CRDM user functions. See chapter Configuration of users [▶ 199] for information on the configuration of users.

6.1.3.1.5 Common reference data objects and the hierarchical party model (completed)

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 central banks are level 2 parties, all payment banks are level 3 parties [3]. All the other reference data objects are linked to a party. For example:

- a cash account is linked to its central bank or payment bank.

3 Participation types may be further detailed with information specific to each individual service, if the service foresees this possibility.

- a restriction type is linked to the operator.

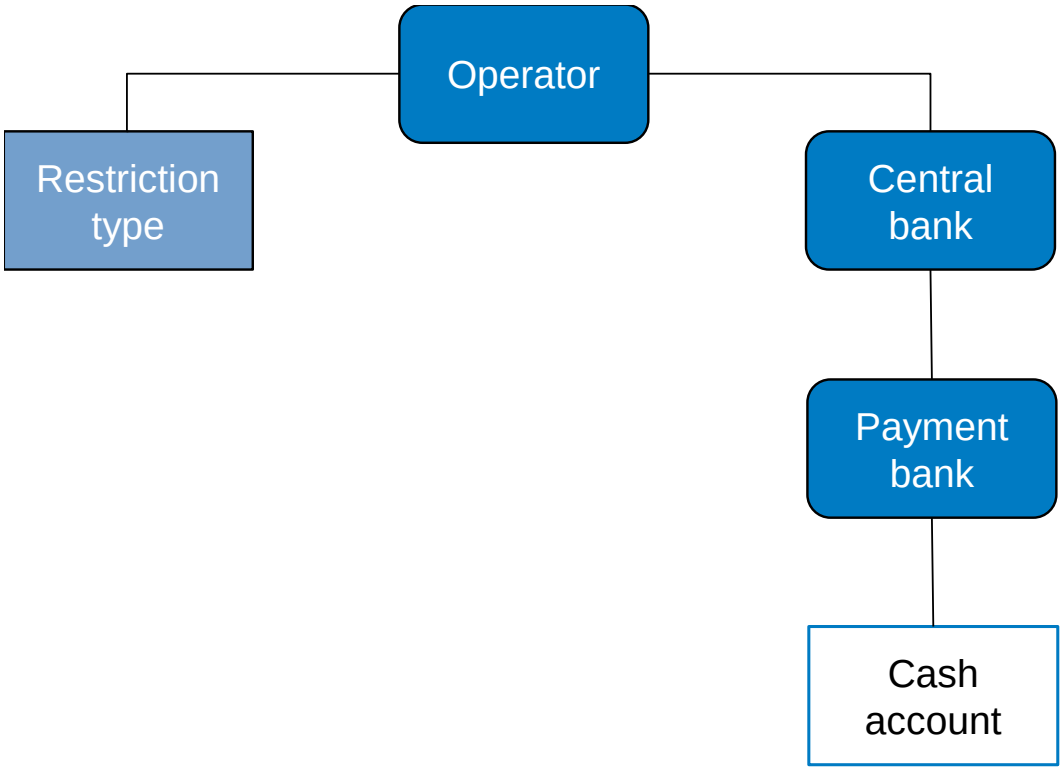


Figure 24 - Common reference data objects and the hierarchical party model

6.1.3.1.6 Data scope (completed)

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 central banks have visibility on all reference data objects belonging to the same system entity [4];
- users of the payment banks have visibility on reference data objects that are (directly or indirectly) linked to the same party.

The following example describes the concept of data scope [5].

4 A system entity in CRDM corresponds to a partition of data equating to the scope of a central bank or of the operator. For example, the system entity of a central bank includes all the data related to its payment banks.

5 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 central bank and to the operator respectively, are granted with the same privilege to query cash accounts:

Table 89 - User privileges (data scope)

User	Privilege
X	Cash account reference data query
Y	Cash account reference data query
Z	Cash account reference data query

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

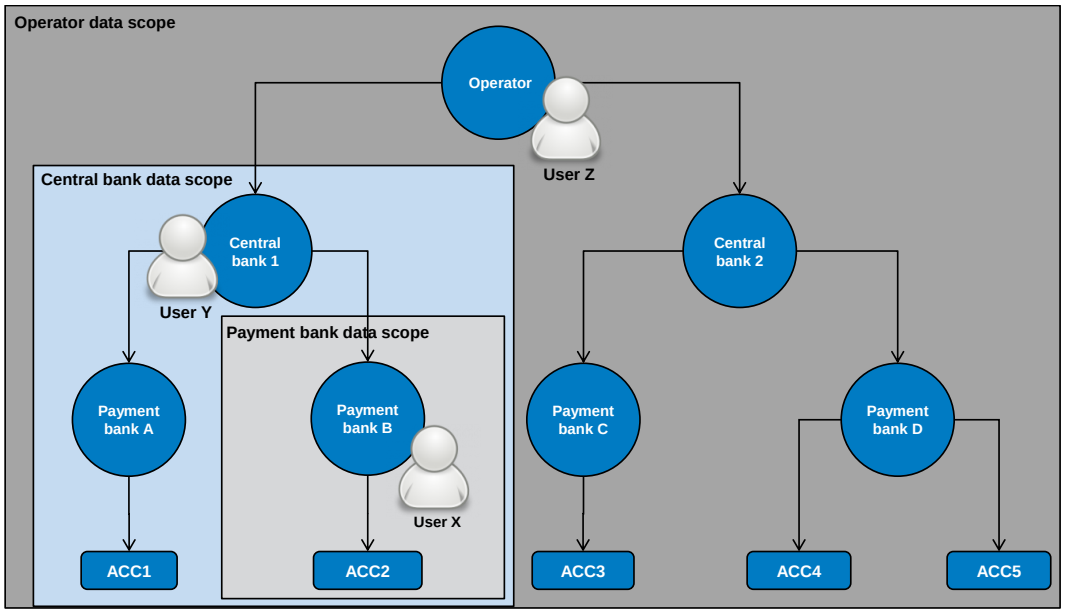


Figure 25 - Data scopes

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:

- 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.
- User Y of central bank 1 gets a data scope including cash accounts ACC1 and ACC2, as these accounts belong to payment banks of central bank 1. User Y cannot query any other cash account in CRDM, i.e. any cash account falling under the data scope of any other central bank.
- 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 (completed)

This section 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 (completed)

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 CRDM operator system administrator creates a new system administrator for a central bank
- when a central bank 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 [▸ 197]). 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 (completed)

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. [6]

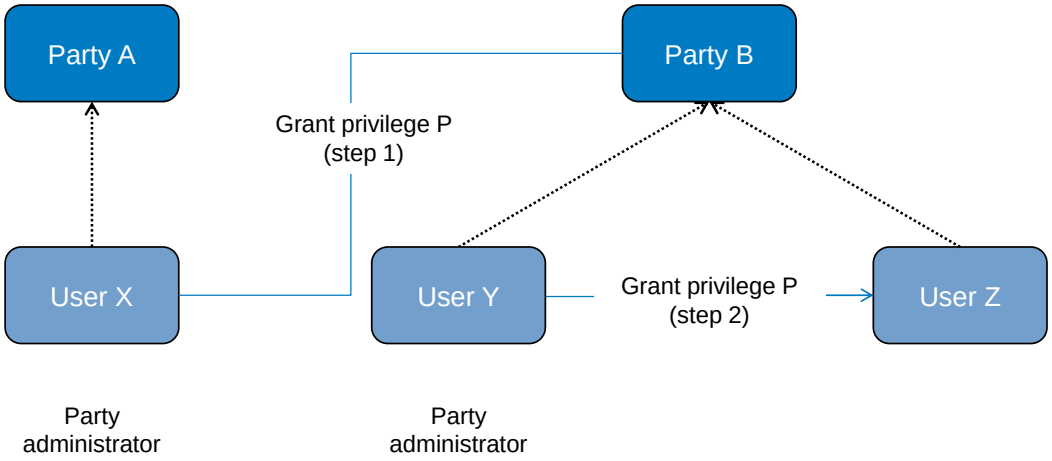


Figure 26 - Access rights configuration steps

The two configuration steps are as follows:

- User X, as a party administrator of party A, grants privilege P to party B. From this moment on, privilege P becomes available to the party administrator Y of party B.
- User Y, as a party administrator of party B, grants privilege P to user Z. From this moment on, user Z can trigger the user functions linked to privilege P.

At Party level, access rights are propagated following the hierarchical party model, i.e. the operator propagates access rights to central banks which in turn propagate them to their payment banks. If necessary, the operator can act on behalf of a central bank 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 [▸ 206].

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.

6 Party A may be the operator or any other party which was previously granted privilege P.

Table 90 - Privilege assignment options

Option	Description
Deny	This option specifies whether the associated user function is allowed (deny is false) or explicitly denied (deny is true).
Administration	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). 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).
Four-eyes	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. This option is relevant only when the deny option is set to false and it is always not relevant for privileges related to queries.

Example - assignment of privileges to roles

The following table shows some examples of assignment of privileges to roles:

Table 91 - 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

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 [7];
- 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

7 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:

Table 92 - 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

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 UX can create cash accounts according to the two-eyes principle (as the privilege is assigned without four-eyes);
- with admin, i.e. user UY 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).

Similarly, row 2 stipulates that the privilege to create cash accounts is assigned to user U_Y:

- with deny, i.e. user UY cannot create cash accounts;
- with admin, i.e. user UY 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 UX can create cash accounts, but without having the possibility to grant the same privilege to any other user. Vice versa, user UY 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:

Table 93 - Assignment of privileges to parties

Privilege	Party	Deny	Admin	Four-eyes
Cash account reference data query	Payment bank A	False	True	False

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 central banks can ensure that all the privileges revoked from one of their parties are revoked also from all the users of the same party:

Table 94 - Cascade process when revoking privileges

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	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. 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. 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 central bank 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).
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.

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.

6.1.3.2.3 Configuration of roles (completed)

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 central banks 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:

Table 95 - Cascade process when revoking roles

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	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. 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. In order to ensure that the relevant role is revoked also from the users of the payment bank, the party administrator of the central bank 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. 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 central bank 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 11 – cascade process when revoking privileges).

6.1.3.3 Access rights configuration process (completed)

As described in chapter Configuration of privileges [▸ 199], 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 central bank (identified as party A in the diagram).

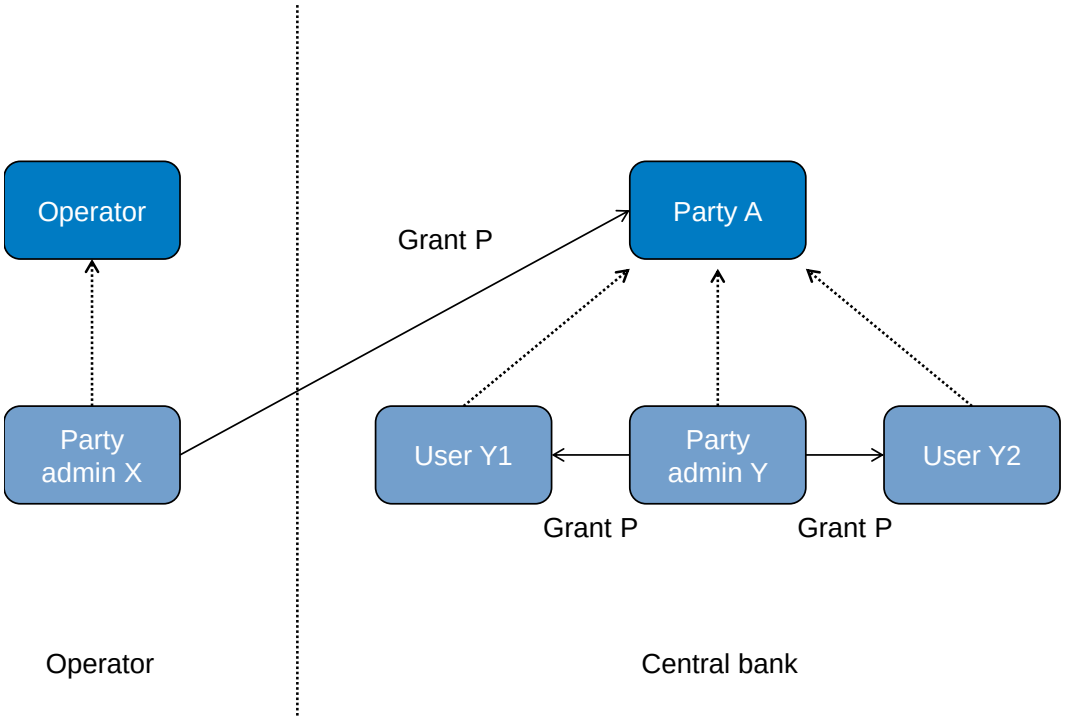


Figure 27 - Access rights configuration process (A)

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 central bank 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 central bank (party A in the diagram).

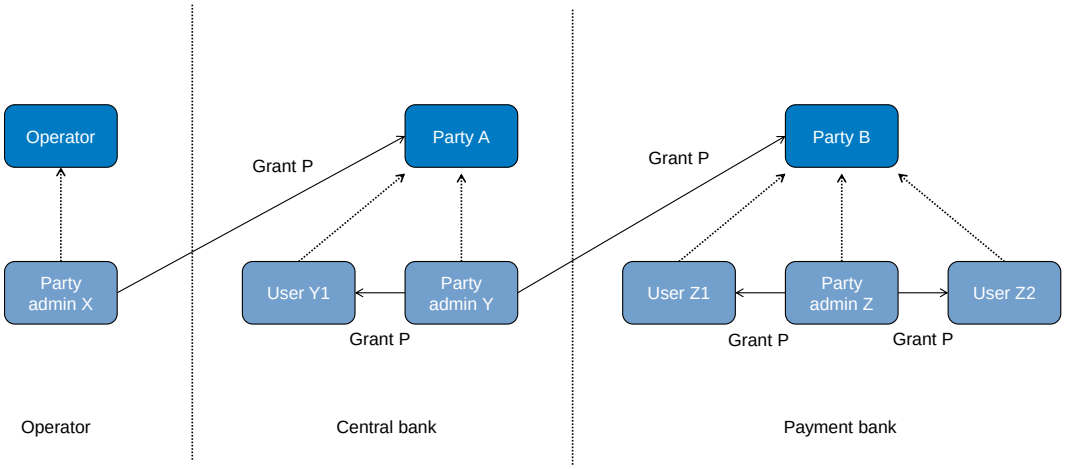


Figure 28 - Access rights configuration process (B)

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 central bank);
- 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 illustrates 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 [▸ 199] , 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 [▸ 206].

6.1.3.3.1 Configuration of access rights at party level (completed)

This task consists in 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 central banks.

The following diagram shows an example in which the party administrator of the operator grants to all the central banks the same set of roles and privileges. This set includes all the privileges needed by the central banks and all the privileges needed by the payment banks.

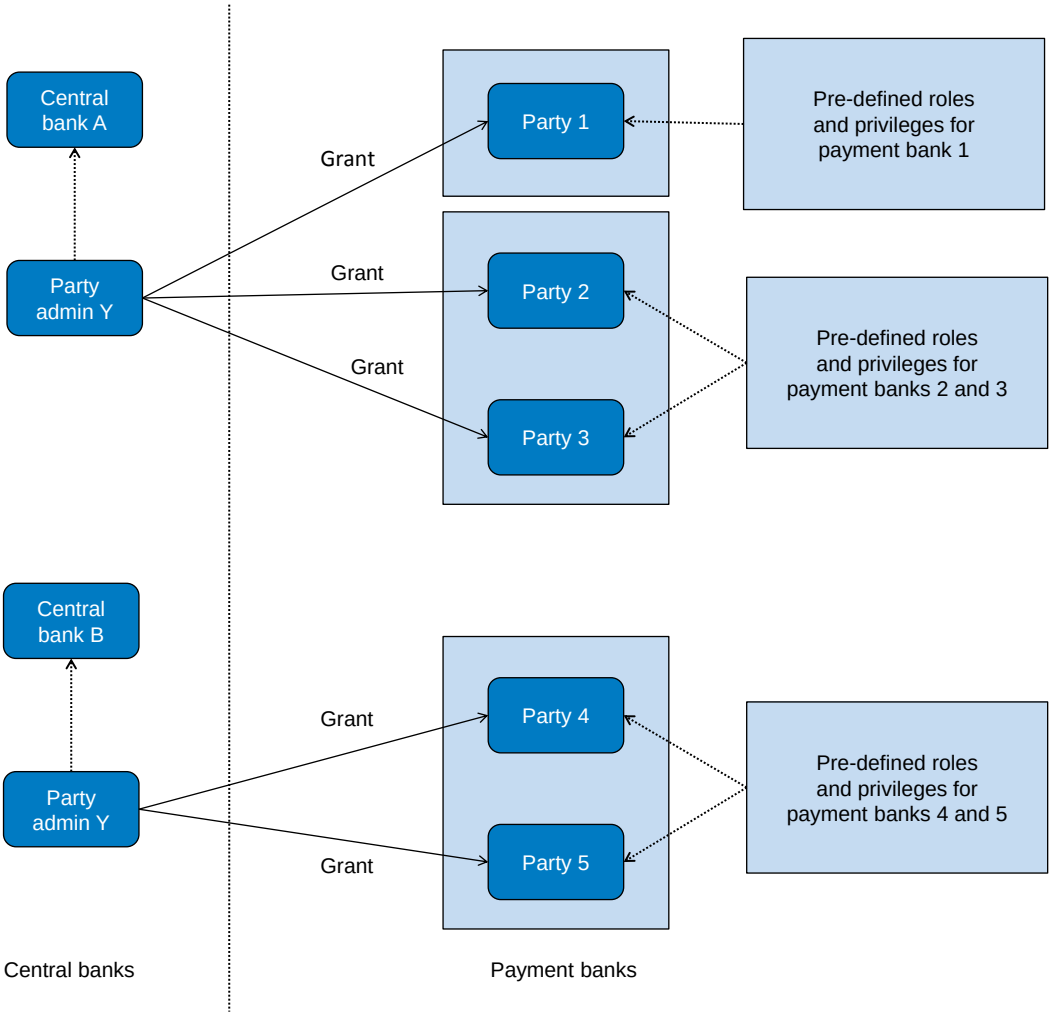


Figure 29 - Example - configuration of access rights at party level by the operator

A party administrator of each central bank assigns the relevant set of roles [8] and privileges to all its payment banks. In this example the party administrator of a central bank 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 central bank 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.

8 New roles can only be created and maintained by the operator and central bank parties. Payment banks can only grant/revoke roles that have previously been granted to them by their central banks.

6.1.3.3.2 Configuration of access rights at user level (completed)

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.

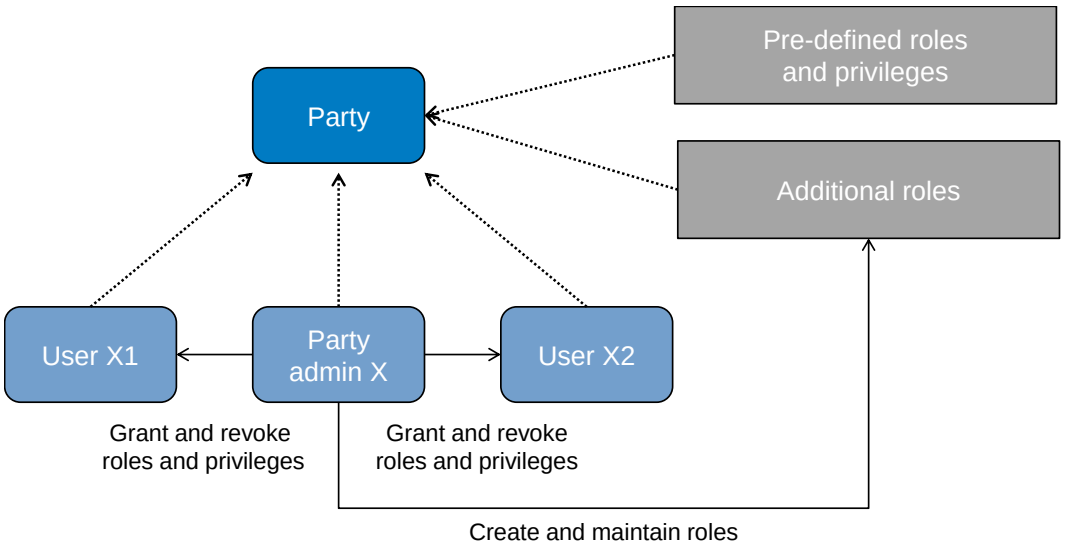


Figure 30 - Configuration of access rights at user level

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 [9] additional roles, besides the ones previously granted at party level [10]
- 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 (to be completed in iteration 4)

To be provided in a future version.

9 New roles can only be created and maintained by the operator and central bank parties. Payment Banks can only grant/revoke roles that have previously been granted to them by their central banks.

10 These additional roles can only be granted with available privileges, i.e. privileges previously granted at party level.

6.1.5 Instructing scenarios (to be completed in iteration 4)

6.1.6 Reference data maintenance process (completed)

6.1.6.1 Reference data objects (completed)

Duly authorised actors 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:

Table 96 - Common reference data objects

Area	Object	Responsible CRDM actors [11] [12]
Party	Party	Operator, central bank
	Party service link	Operator, central bank
	Banking group	Central bank
	Monetary financial institution	Central bank
Cash account	Cash account	Central bank
	Limit	Payment bank
	Authorised account user	Payment bank
	Account monitoring Group	Central bank
	Standing liquidity transfer order	Payment bank
	Liquidity transfer group	Payment bank
	Direct debit mandate	Payment bank
	Standing order for reservation	Payment bank
	Floor/ceiling	Payment bank

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

12 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 central banks (and of payment banks, upon request of the relevant central bank) and for the central banks to act on-behalf of their payment banks, under well-defined contingency scenarios.

Area	Object	Responsible CRDM actors [11] [12]
Access rights management	User	All
	Role	Operator, central bank
	Privilege	Operator
	Certificate DN	All
	User-certificate DN link	All
	Role user [13]	All
	Role party [14]	Operator, central bank
	Grantee privilege [15]	Operator, central bank, payment bank
Message subscription configuration	Message subscription rule	Central bank, payment bank
	Message subscription rule set	Central bank, payment bank
Network configuration	DN BIC routing	Payment bank
	Network service	Operator
	Technical address network service link	Operator, central bank
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

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

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

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

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 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

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.

6.1.6.2 Reference data maintenance types (completed)

CRDM allows a duly authorised actor 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 Reference data archiving and purging [▸ 220]).
- restore [16]: it reactivates a previously deleted common reference data object, i.e. it updates the approval 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 section Access rights [▸ 183] 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:

Table 97 - Management of reference data objects in DMT

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

16 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.

Table 98 - Management of reference data objects in A2A mode

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

6.1.6.3 Validity of reference data objects (completed)

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:

Table 99 - Common reference data objects with unlimited validity period

Area	Object
Party	Banking group
	Monetary financial institution
Cash account	Account monitoring group
	Liquidity transfer group

Area	Object
Access rights management	User Role Privilege Certificate DN User-Certificate DN link Role user link Role party link Privilege role link
Network configuration	Network service Technical address network service link
Configuration parameters	Country Currency Currency service link System entity Service Currency service link

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 Eurosystem market infrastructure service(s) 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 Eurosystem market infrastructure service(s), 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 actor.

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:

Table 100 - Common reference data objects with limited validity period [17]

Area	Object	Creation	Update	Deletion
Party	Party	Validity date may take the value of the current date.	May take effect on the current date [18].	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.
	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.

17 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 in 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.

18 This is not applicable to the party code, which cannot be updated if it is currently active.

Area	Object	Creation	Update	Deletion
	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.

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 Eurosystem market infrastructure 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.

6.1.6.4 Reference data archiving and purging (completed)

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.

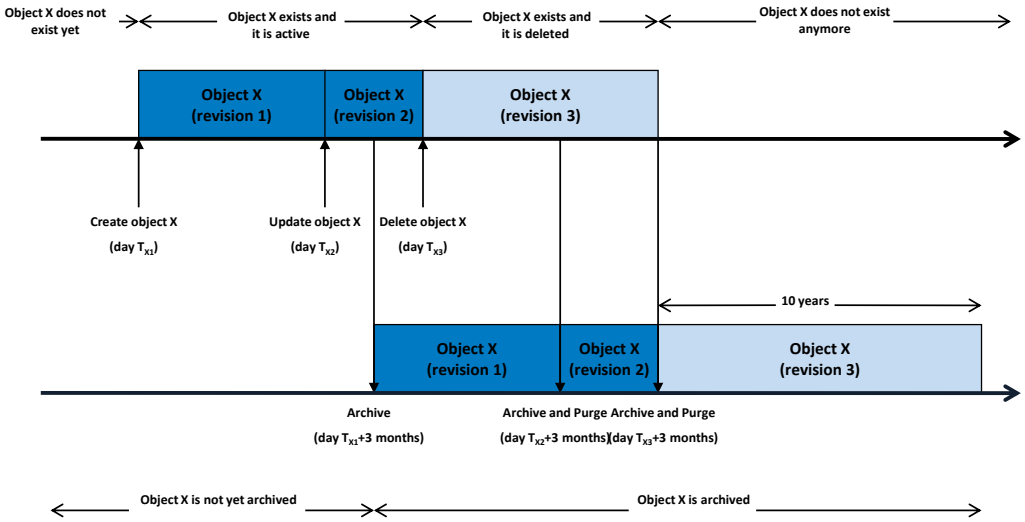


Figure 31 - Example - archiving and purging after deletion of a common reference data object

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. During business day T_{X2} (with $T_{X2}<T_{X1}$ + 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+} 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 $T_{X3}<T_{X2}$ + 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+} 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+} 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.

6.1.6.5 Lifecycle of reference data objects (completed)

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:

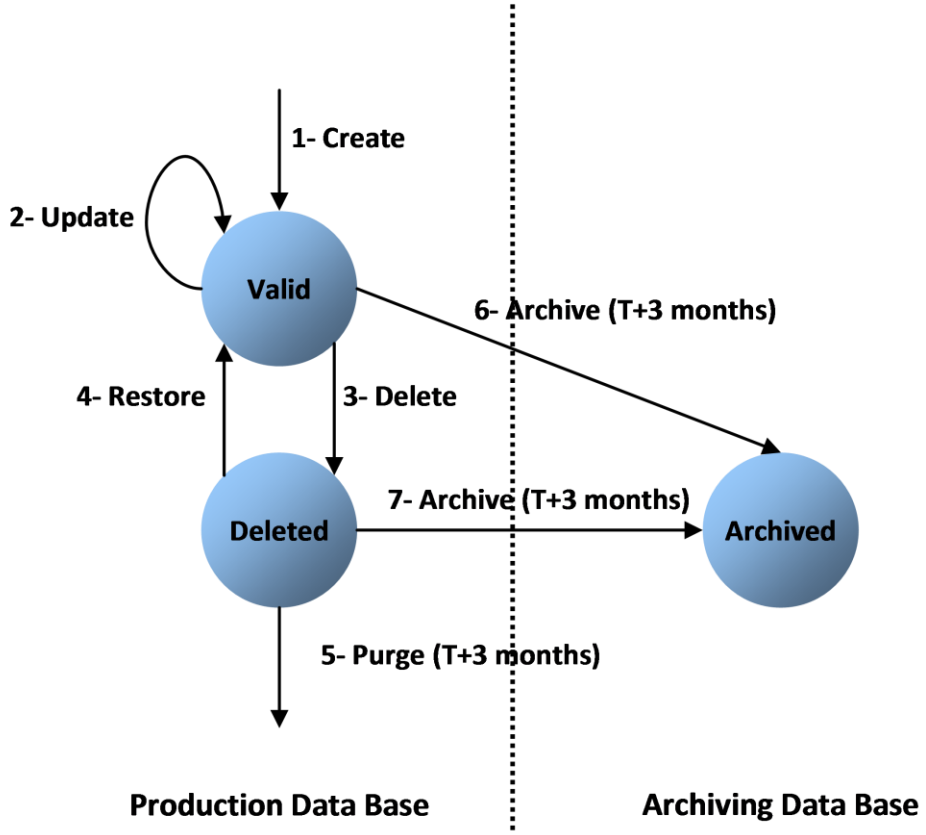


Figure 32 - Lifecycle of common reference data objects with unlimited validity period

When a duly authorised actor 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 ob-

ject, 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

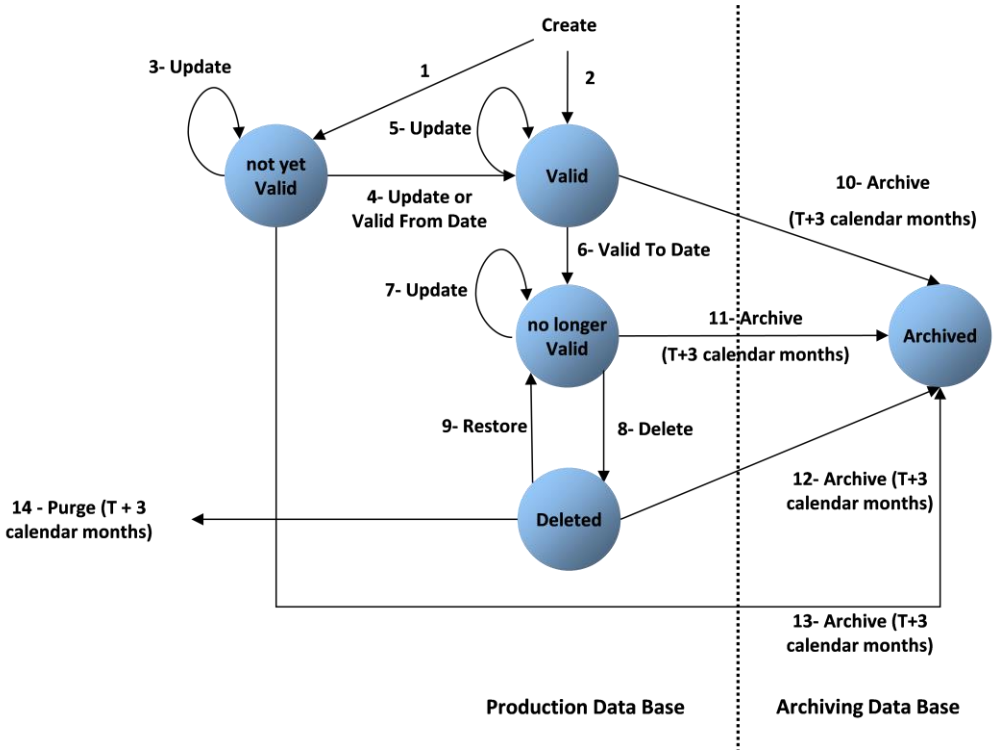


Figure 33 - Lifecycle of common reference data objects with limited validity period

When a duly authorised user submits to CRDM a reference data maintenance instruction to create a common reference data object with limited validity period, CRDM processes it and, in case of successful processing, it creates the relevant object. This object is either valid or not yet valid, depending on the starting date of its validity period, and it exists in the production data base only (transitions 1 and 2).

From this moment on, a duly authorised user may submit to the CRDM one or many reference data maintenance instructions to update the common reference data object. If the object is valid, then it remains valid, regardless of the result of CRDM processing, i.e. whether the reference data maintenance instruction is successfully or unsuccessfully processed (transition 5). If the object is not yet valid, two sub-cases are possible:

- If the reference data maintenance instruction also updates the starting date of the validity period to the current business date and it is successfully processed, then the common reference data object becomes valid (transition 4).
- In all other cases, whether the reference data maintenance instruction is successfully or unsuccessfully processed, the common reference data object remains not yet valid (transition 3).

A common reference data object becomes valid from the starting business date of the validity period (transition 4).

A common reference data object is valid until the end of day of the final date of the validity period (transition 6).

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 Common Reference Data Management service 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).

6.1.6.6 Reference data propagation (completed)

CRDM allows users to configure reference data to be used in the local reference data management of other TARGET services (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 preset 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 component or back-office application, 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.

Table 101 - CRDM data segregation per service/component

Area	Object	Service(s)/component	Segregation principles
Party	Party	CLM, RTGS, T2S, TIPS	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	None	Only relevant for CRDM; defines the availability of party data for a given service.
	Banking group	CLM	All data is available in CLM.
	Monetary financial institution	CLM	All data is available in CLM.
Cash account	Cash account	CLM, RTGS, T2S, TIPS	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.

Area	Object	Service(s)/component	Segregation principles
	Authorised account user	TIPS	All data is available in TIPS.
	Account monitoring group	CLM	All data is available in CLM.
	Standing liquidity transfer order	CLM, RTGS, T2S	Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Liquidity transfer group	CLM, RTGS	Data is available in different services depending on the cash account type attribute of the cash accounts it refers to.
	Limit	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Direct debit mandate	CLM, RTGS	Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Standing order for limit	RTGS	All data is available in RTGS.
	Standing order for reserva-tion	CLM, RTGS	Data is available in different services depending on the cash account type attribute of the cash accounts it refers to.
	Floor/ceiling	CLM, RTGS	Data is available in different services depending on the cash account type attribute of the cash account it refers to.

Area	Object	Service(s)/component	Segregation principles
Access rights management	User	CLM, RTGS, T2S	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	CLM, RTGS, T2S, TIPS	All data is available in T2S. Data containing privileges related to CLM, RTGS or TIPS is available in that service.
	Privilege	T2S	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 single service which defines the service that contains the user function activated by the privilege.
	Certificate DN	CLM, RTGS, T2S, TIPS	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	CLM, RTGS, T2S, TIPS	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.

Area	Object	Service(s)/component	Segregation principles
	Role user	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the privileges contained in the role refer to.
	Role party	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the privileges contained in the role refer to.
	Grantee privilege	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the privilege refers to.
Message subscription configuration	Message subscription rule set	CLM, RTGS, T2S, TIPS	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	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the underlying reference data objects the rule refers to.
Network configuration	Network service	CLM, RTGS, T2S, TIPS	Data is available in different Services based on an attribute that defines a direct reference to a single Service.
	Technical address network service link	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the related network service refers to.
	DN BIC routing	TIPS	All data is available in TIPS.
Report configuration	Report configuration	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the specific type of report being subscribed.

Area	Object	Service(s)/component	Segregation principles
Restriction type management	Restriction type	RTGS, T2S, TIPS	Data is available in different services based on an attribute that defines a direct reference to a single service.
Billing configuration	Service item	None	Only relevant for CRDM and Billing.
Configuration parameters	Country	CLM, RTGS, T2S, TIPS	All data is available in all services.
	Currency	CLM, RTGS, T2S, TIPS	All data is available in all services.
	Currency service link	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the link refers to.
	System entity	CLM, RTGS, T2S, TIPS	All data is available in all services.
	BIC directory	CLM, RTGS, T2S, TIPS	All data is available in all services.
	Service	None	Only relevant for CRDM.

6.2 Data warehouse (to be completed in Version 2.0)

6.2.1 Introduction (to be completed in Version 2.0)

6.2.2 Scope of the data warehouse (to be completed in Version 2.0)

6.2.3 Access (to be completed in Version 2.0)

6.2.3.1 Connectivity (to be completed in Version 2.0)

6.2.3.2 Authentication and authorisation (to be completed in Version 2.0)

6.2.4 User roles and access rights (to be completed in Version 2.0)

6.2.4.1 Overview (to be completed in Version 2.0)

6.2.4.2 User rights (to be completed in Version 2.0)

6.2.4.3 User profiles (to be completed in Version 2.0)

6.2.5 Data warehouse queries and reports (to be completed in Version 2.0)

6.2.5.1 Overview (to be completed in Version 2.0)

6.2.5.2 Types of queries and reports (to be completed in Version 2.0)

6.2.5.3 Predefined queries and reports (to be completed in Version 2.0)

6.3 Billing (to be completed in Version 2.0)

6.4 Legal archiving (to be completed in Version 2.0)

7 Contingency services (to be completed in Version 2.0)

8 Operations and support (to be completed in Version 2.0)

8.1 Business application configuration (to be completed in Version 2.0)

8.2 Calendar management (to be completed in Version 2.0)

8.3 Business day management (to be completed in Version 2.0)

8.4 Business and operations monitoring (to be completed in Version 2.0)

8.5 Archiving management (to be completed in Version 2.0)

8.6 Trouble management (to be completed in Version 2.0)

9 Additional information for central banks (to be completed in iteration 4)

9.1 Role of central banks in the RTGS component (to be completed in iteration 4)

9.2 Reference data for central banks (to be completed in iteration 4)

9.2.1 Specific data for central banks (to be completed in iteration 4)

9.2.2 Setup of RTGS related reference data (to be completed in iteration 4)

9.3 Settlement of payments - specific functions for central banks (to be completed in iteration 4)

9.4 End-of-day procedures (to be completed in iteration 4)

9.5 Query management - central bank specific functions for central banks (to be completed in iteration 4)

9.6 Data warehouse - specific functions for central banks (to be completed in Version 2.0)

9.7 Billing - specific functions for central banks (to be completed in Version 2.0)

9.8 Contingency - specific functions for central banks (to be completed in Version 2.0)

II Dialogue with the RTGS participant

10 Processes with RTGS components

10.1 Interface processing - send file (to be completed in iteration 4)

10.2 Local reference data management - maintain local reference data object (to be completed in iteration 4)

10.3 Payment instruction processing

10.3.1 Send payment order (completed)

This process starts

- when the submitting actor sends one of the following messages via ESMIG to the RTGS component:

Table 102 - Messages sent by the submitting actor to RTGS component

Message	Message name
PaymentReturn (pacs.004) [369]	PaymentReturn
CustomerCreditTransfer (pacs.008) [373]	CustomerCreditTransfer
FinancialInstitution-CreditTransfer (GEN and COV) (pacs.009) [383]	FinancialInstitutionCreditTransfer
FinancialInstitu-tionDirectDebit (pacs.010) [402]	FinancialInstitutionDirectDebit
LiquidityCreditTransfer (camt.050) [336]	LiquidityCreditTransfer

- when the RTGS component receives a message from the file splitting process (refer to interface process "Interface processing - send file [235]").

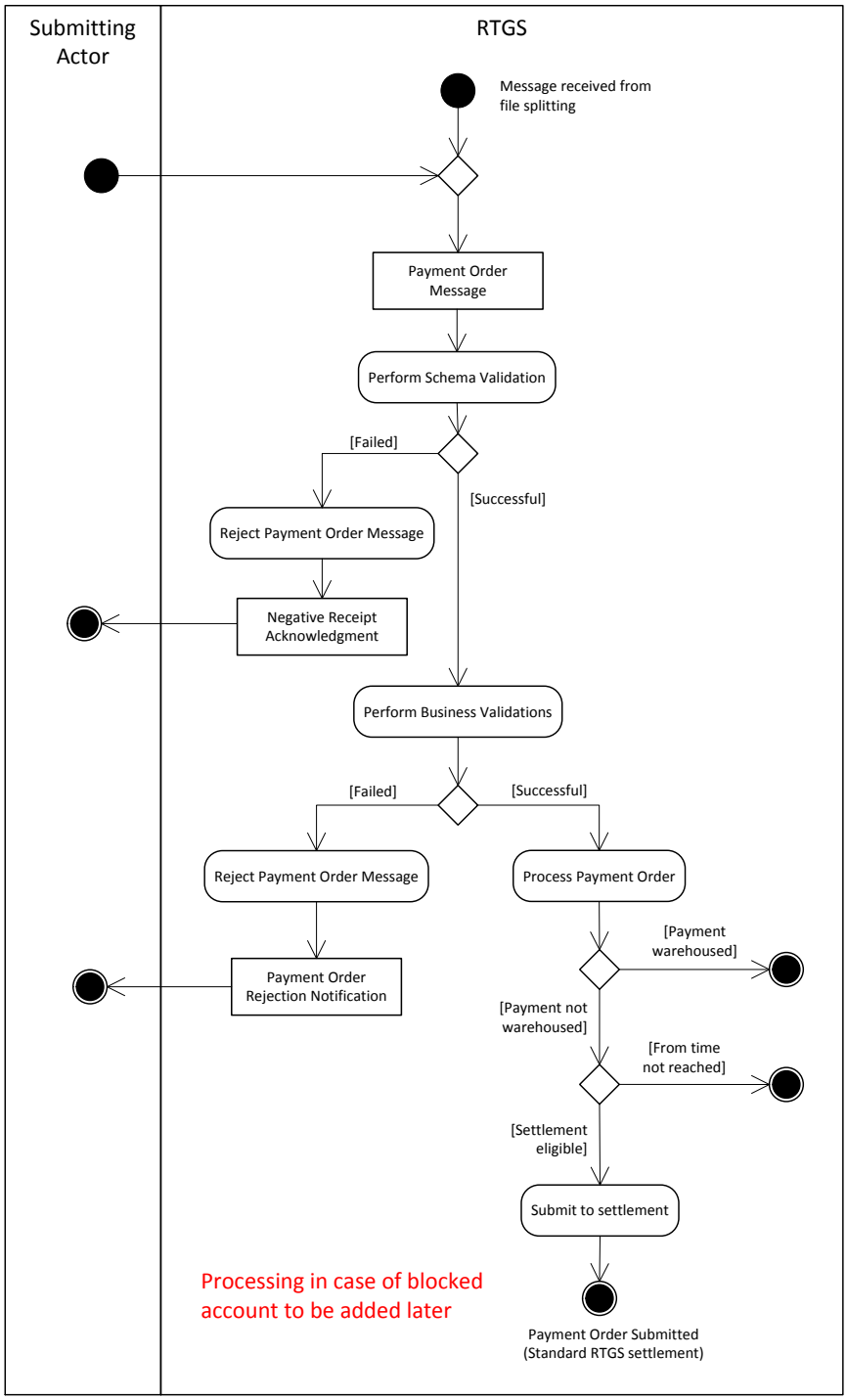


Figure 34 - Send RTGS payment order

Schema validation:

In the first step, the RTGS component performs the schema validation of the payment order message.

- **[Failed]** In case the schema validation fails, the RTGS component rejects the payment order message and the submitting actor receives a "*Negative Receipt Acknowledgement*" ReceiptAcknowledgement (admi.007) [▸ 276].

Note: RTGS identifies all possible schema validation errors and does not stop the schema validation after the first error is found.

- **[Successful]** In case of a successful schema validation, the RTGS component continues with the business validation.

Business validation:

In the second step, RTGS performs the business validation with possible outcomes being:

- **[Failed]** In case the business validation fails, the RTGS component rejects the payment order message and the submitting actor receives a "*Payment Order Rejection Notification*" PaymentStatusReport (pacs.002) [▸ 365].

  Note: The RTGS component continues with all possible business validations even after the business validation identifies one or more errors. It does not stop after identifying the first business validation error. Consequently, the rejection notification includes all relevant error codes.

- **[Successful]** In case the business validation is successful, RTGS continues with the processing of the payment order.

  As part of this processing step, the RTGS component determines

  – whether the payment order is a warehoused payment;

  – whether the defined "FromTime" when specified in the payment has not been reached;

  – whether the payment order is directly eligible for the settlement.

The processing submits the payment order directly to the Standard RTGS settlement [▸ 238] process when it is directly eligible for settlement.

10.3.2 Revoke/cancel payment order (to be completed in iteration 4)

10.3.3 Amend payment order (to be completed in iteration 4)

10.3.4 Modify ASI payment order (to be completed in iteration 4)

10.3.5 Execute standing order (to be completed in iteration 4)

10.3.6 Reservation management (to be completed in iteration 4)

10.3.7 Limit management (to be completed in iteration 4)

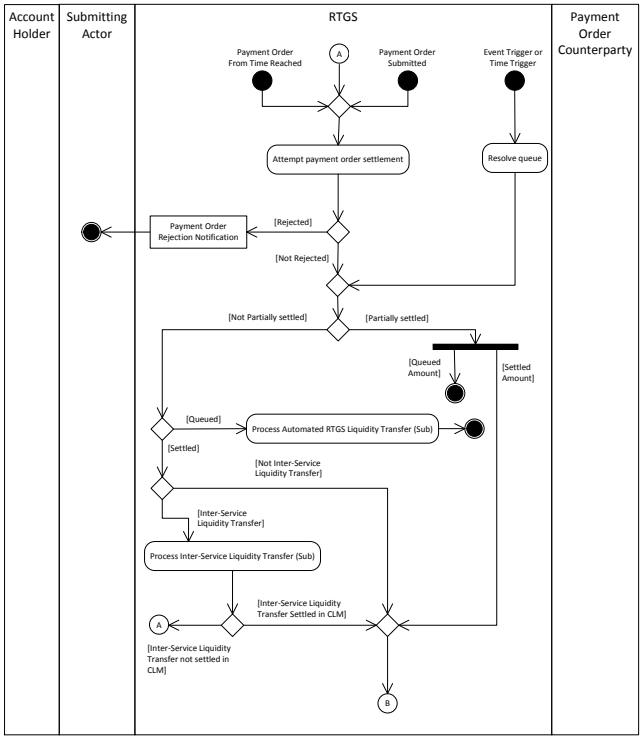
10.3.8 Reject pending payment instructions at end of day (to be completed in iteration 4)

10.3.9 Settle RTGS payment order

10.3.9.1 Standard RTGS settlement (completed)

The process "*attempt payment order settlement*" starts

- after receiving a successfully validated payment order [Payment Order Submitted],
- in case of an inter-service liquidity transfer initiated in the RTGS component could not be successfully booked in the other service and the amount needs to be credited back to the RTGS dedicated cash account [A] and
- for a successfully validated payment order that specifies "FromTime" and the "FromTime" has been reached [Payment Order From Time Reached].



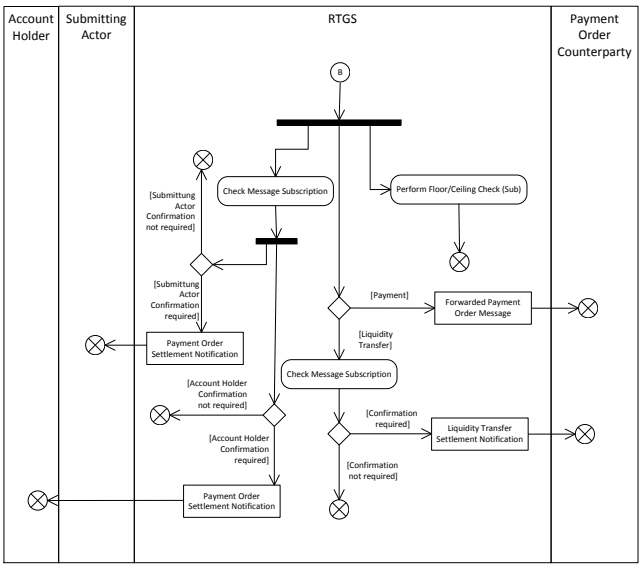


Figure 35 - Standard RTGS settlement

In the first step, the process "*attempt payment order settlement*" tries to settle the submitted payment order, resulting in one of the following outcomes:

- **[Rejected]** In case settlement of the liquidity transfer is not possible due to insufficient liquidity, the process rejects the liquidity transfer and sends a "Payment Order Rejection Notification" Receipt (camt.025) [▶ 318] to the submitter of the original incoming camt.050.

Note: This is not valid for automated inter-service liquidity transfers from CLM due to pending central bank operations.

- **[Not Rejected]**
  - The payments settle or queue.
  - The automated inter-service liquidity transfers from CLM due to pending central bank operations could be settled, partially settled or queued.
  - The liquidity transfers sent by the account holder settle.
  - The liquidity transfers sent by a submitting actor not being the account holder settle or partially settle.

In the second step

- for all accepted (not rejected) payment orders
- as well as for all queued payments forwarded to the process "*resolve queue*" in case of an event or time trigger

the result of the process can be

- **[Partially Settled]** The only scenarios, in which a liquidity transfer is partially settled are
  - an automated inter-service liquidity transfer from CLM due to pending central bank operations and insufficient liquidity on the RTGS dedicated cash account.
  - those transmitted by a submitting actor not being the account holder.

  Note: Payments are never partially settled in the RTGS component.

  For the partially settled amount the same messages are sent to the involved parties as for fully settled liquidity transfers.
- **[Queued]** Payments which cannot settle are queued. As a consequence the sub-process "*automated RTGS liquidity transfer*" is triggered.

  Liquidity transfers are queued when it is an automated inter-service liquidity transfer from CLM due to pending central bank operations which cannot settle in the RTGS component. In case of partial settlement of these liquidity transfers, the remaining part is queued by the RTGS component.

  Note: In case of a new automated inter-service liquidity transfer from CLM due to pending central bank operations the RTGS component uses a "cancel and replace logic", i.e. the already pending automated inter-service liquidity transfer is cancelled and the new one is taken into account for further processing.
- **[Settled]** After successful settlement the *payment order counterparty* receives in case of
  - payments, one of the following messages:

Table 103 - Message sent after settlement

Message	Message name
PaymentReturn (pacs.004) [369]	PaymentReturn
CustomerCreditTransfer (pacs.008) [373]	CustomerCreditTransfer
FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [383]	FinancialInstitutionCreditTransfer /FinancialInstitutionCreditTransferCOV
FinancialInstitutionDirectDebit (pacs.010) [402]	FinancialInstitutionDirectDebit

– liquidity transfers:

a "*Liquidity Transfer Settlement Notification*" BankToCustomerDebitCreditNotification (camt.054) [349] provided that a respective message subscription configuration has been set up in advance.

Note: RTGS treats inter-service liquidity transfers that another service/component initiates as any other intra-service liquidity transfer.

- the *submitting actor* receives in case of

– intra-service liquidity transfers initiated via camt.050:

a "*Payment Order Settlement Notification*" Receipt (camt.025) [318] provided that a respective message subscription configuration has been set up in advance;

– inter-service liquidity transfers initiated via camt.050 in RTGS:

a "*Payment Order Settlement Notification*" Receipt (camt.025) [318] only after sucessful settlement in the other service or component provided that a respective message subscription configuration has been set up in advance.

– payments:

a "*Payment Order Settlement Notification*" PaymentStatusReport (pacs.002) [365] provided that a respective message subscription configuration has been set up in advance.

- the *account holder* receives the following messages provided that the submitting actor and the account holder differ in case of

– intra-service liquidity transfers initiated via camt.050 :

a "*Payment Order Settlement Notification*" BankToCustomerDebitCreditNotification (camt.054) [349] provided that a respective message subscription configuration has been set up in advance.

– inter-service liquidity transfers initiated via camt.050 in RTGS:

a "*Payment Order Settlement Notification*" BankToCustomerDebitCreditNotification (camt.054) [▶ 349] only after successful settlement in the other service provided that a respective message subscription configuration has been set up in advance.

– payments:

a "*Payment Order Settlement Notification*" BankToCustomerDebitCreditNotification (camt.054) [▶ 349] provided that a respective message subscription configuration has been set up in advance.

10.3.9.1.1 Floor and ceiling processing (completed)

This process starts after settlement of a payment (i.e. pacs.004/pacs.008/pacs.009/pacs.009COV/pacs.010) or an ancillary system payment instruction on the RTGS dedicated cash account.

Note: The settlement of liquidity transfers on RTGS dedicated cash accounts trigger no floor/ceiling processing.

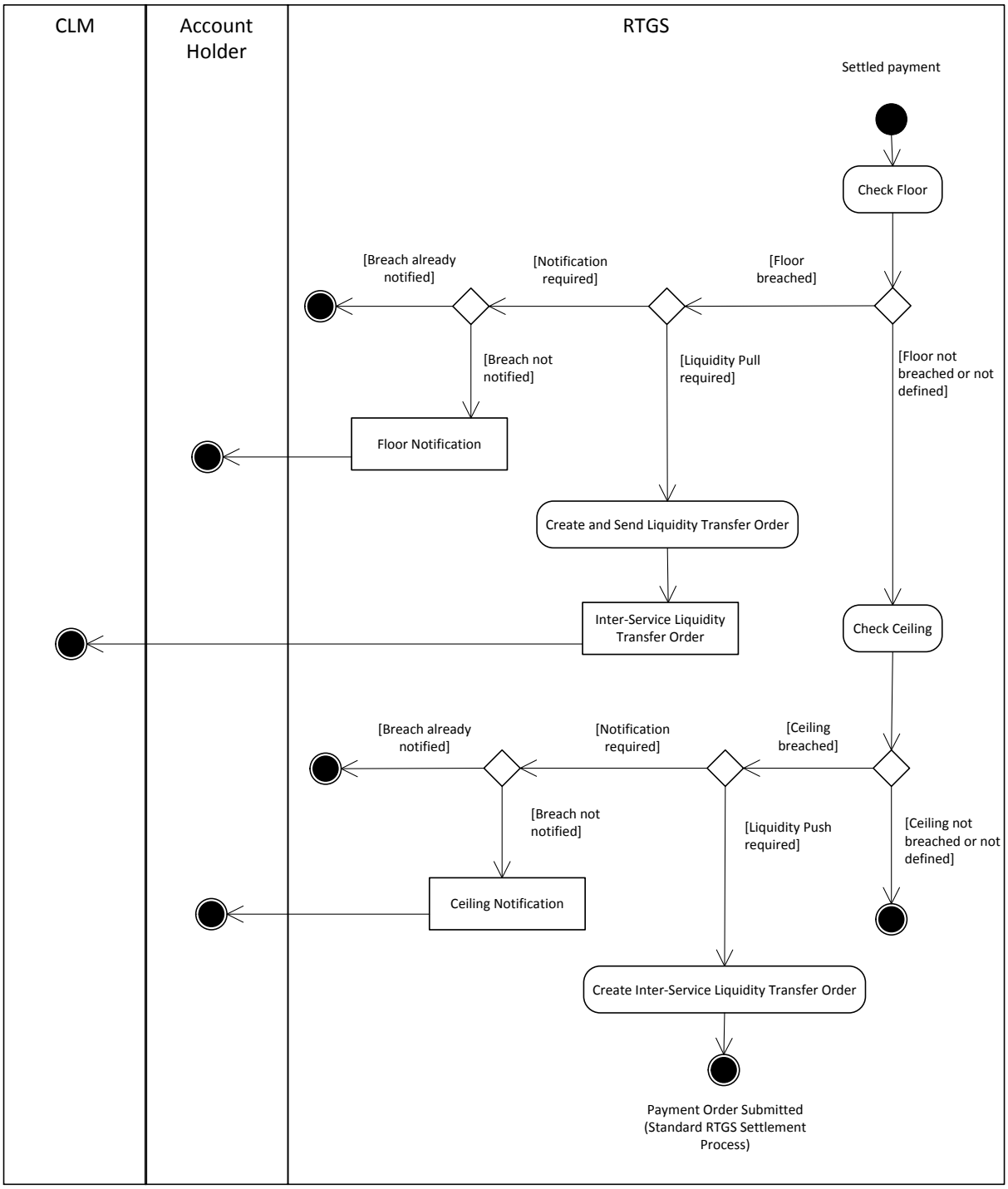


Figure 36 - Floor and ceiling processing

Floor processing:

In case

– of a breach of a previously defined floor,

– the configuration to receive a floor notification has been set up in advance and

– no prior notification of the breach to the account holder,

the RTGS dedicated cash account holder receives a "*Floor Notification*" ReturnAccount (camt.004) [▸ 282]

- 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 main cash account has been set up in advance

RTGS sends to the CLM component an inter-service liquidity transfer order, LiquidityCreditTransfer (camt.050) [▸ 336], in order to pull liquidity up to the targeted floor amount.

Ceiling processing:

- In case
  - of a breach of a previously defined ceiling,
  - the configuration to receive a ceiling notification has been set up in advance and
  - no prior notification of the breach to the account holder,

the RTGS dedicated cash account holder receives a "*Ceiling Notification*" ReturnAccount (camt.004) [▸ 282].

- 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 main cash account has been set up in advance

RTGS sends to the CLM component an inter-service liquidity transfer order as LiquidityCreditTransfer (camt.050) [▸ 336] in order to push liquidity to reach the predefined target ceiling amount on the RTGS dedicated cash account in the RTGS component.

10.3.9.1.2 Automated liquidity transfer (completed)

This optional process starts when a payment with priority urgent or high does not settle and, therefore, is queued. In addition, it is necessary that the RTGS participant has defined in advance that in such case liquidity shall be pulled from the linked main cash account.

Note: This functionality can be used independently from the definition of a floor/ceiling. Details on the inter-service liquidity transfers due to a floor/ceiling configuration can be found in chapter Floor and ceiling processing [▸ 242].

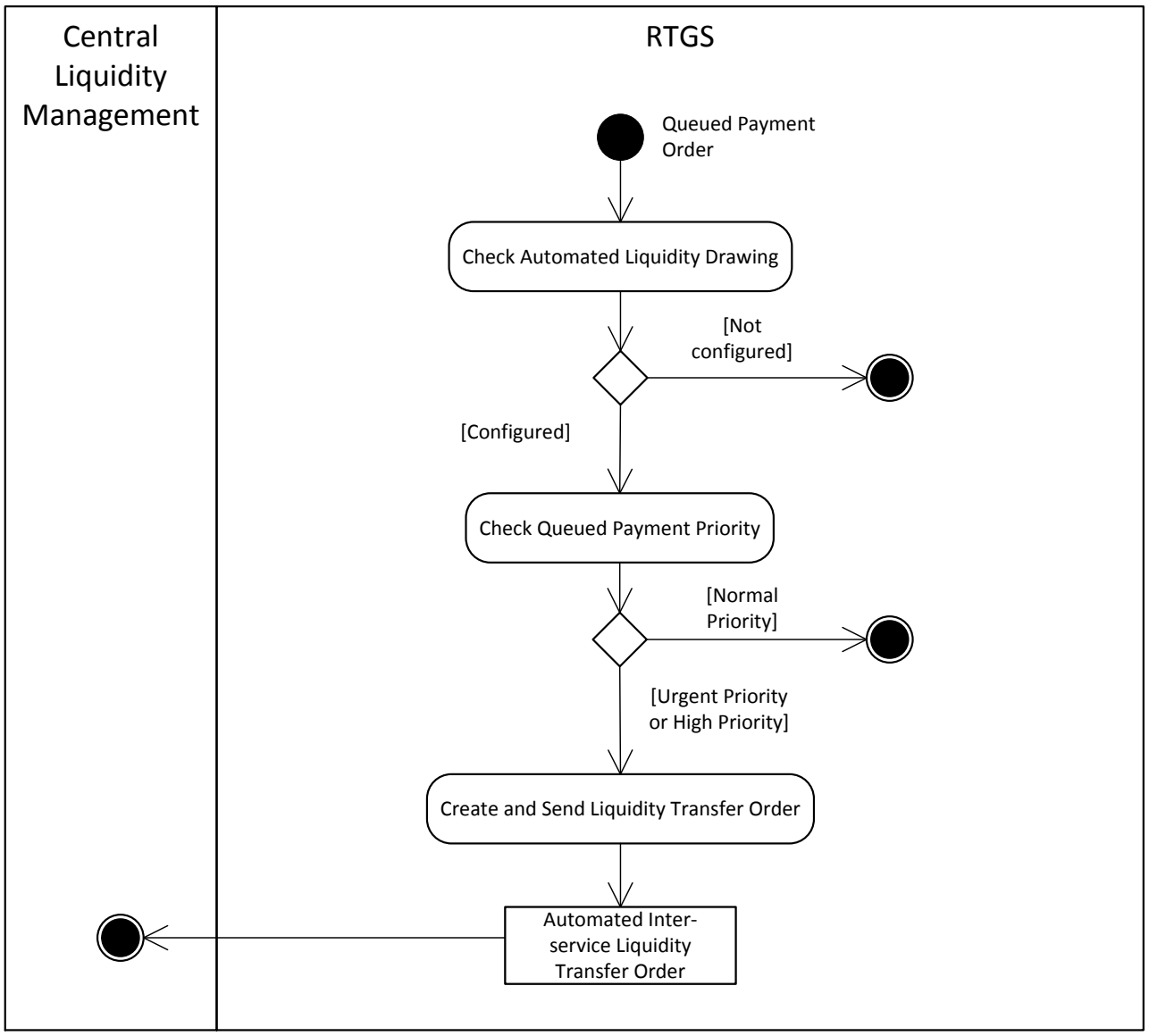


Figure 37 - Process automated RTGS liquidity transfer order

The RTGS component automatically creates a new inter-service liquidity transfer order and sends a LiquidityCreditTransfer (camt.050) [▸ 336] to CLM to pull the liquidity needed from CLM in order to settle the queued payment order in the RTGS component.

Note: There is no earmarking and in case new payments with a higher priority than the queued payment order are submitted, the liquidity might be used to settle payment orders with a higher priority.

10.3.9.2 Till/reject time check (to be completed in iteration 4)

10.3.9.3 Ancillary system interface 4 settlement (to be completed in iteration 4)

10.3.9.4 Ancillary system interface 5 settlement (to be completed in iteration 4)

10.3.9.5 Ancillary system interface 6 real-time settlement (to be completed in iteration 4)

10.3.9.6 Ancillary system interface 6 integrated settlement (to be completed in iteration 4)

10.3.9.7 Blocking of account / participant (to be completed in iteration 4)

10.3.10 Revalidate warehoused payments at start of day (to be completed in iteration 4)

10.4 Information services (completed)

10.4.1 Execute query (completed)

This is a general process description for executing a query, 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 RTGS, CRDM, scheduler and billing [▸ 178] describes the respective business scope.

The following activity diagram provides respective processes in the context of the RTGS component:

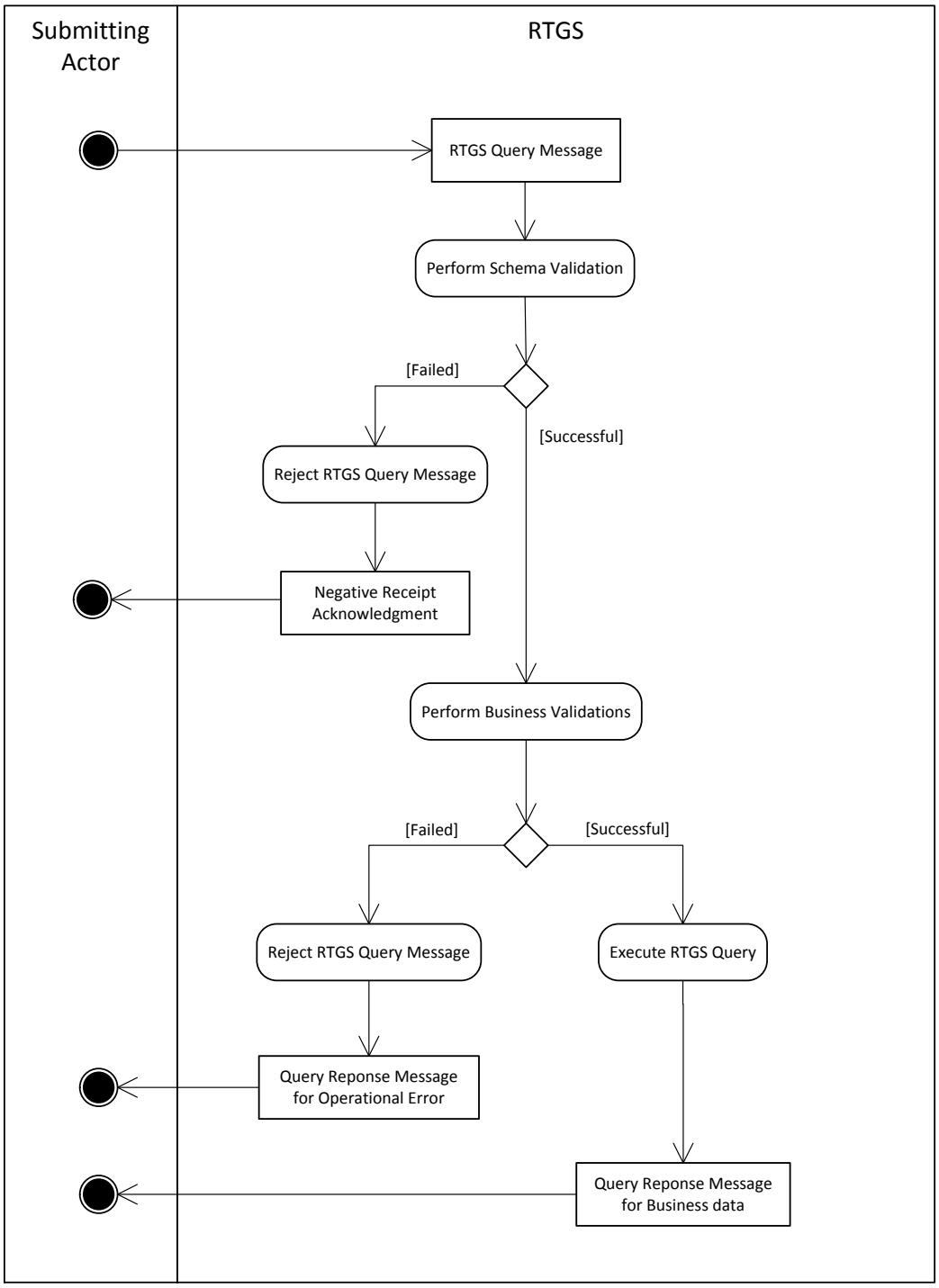


Figure 38 - RTGS send query

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 sends an admi.007 message [Negative Receipt Acknowledgment] to the submitting actor including all information regarding the reasons for failed validation.
- **[successful]** The process triggers the business validation.

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] to the submitting actor.
- **[successful]** The process “Execute RTGS query” starts. It extracts the required business data and 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.

Table 104 - A2A messages for query processing

Related component	Query type	Query request message	Query response message for operational error	Query response message for business data
RTGS	Account Balance Query	GetAccount (camt.003) [280]	ReturnAccount (camt.004) [282]	ReturnAccount (camt.004) [282]
RTGS	Audit Trail for RTGS Query	GetAudit		ReturnAudit
RTGS	Current Limits Query	GetLimit (camt.009) [302]	ReturnLimit (camt.010) [304]	ReturnLimit (camt.010) [304]
RTGS	Current Reservations Query	GetReservation (camt.046) [325]	ReturnReservation (camt.047) [327]	ReturnReservation (camt.047) [327]
RTGS	Payment Query	GetTransaction (camt.005) [287]	ReturnTransaction (camt.006) [292]	ReturnTransaction (camt.006) [292]
RTGS	Settlement Information Query	camt.998 GetSettlementInformation	camt.998 ReturnSettlementInformation	camt.998 ReturnSettlementInformation
CRDM	Ancillary System Reference Data Query	PartyQuery (reda.015) [417]	PartyReport (reda.017) [417]	PartyReport (reda.017) [417]
CRDM	Ancillary system Settlement Bank Reference Data Query	PartyQuery (reda.015) [417]	PartyReport (reda.017) [417]	PartyReport (reda.017) [417]
CRDM	Audit Trail for CRDM Query	GetAudit	ReturnAudit	ReturnAudit

Related component	Query type	Query request message	Query response message for operational error	Query response message for business data
CRDM	Calendar Query	GetCalendar	ReturnCalendar	ReturnCalendar
CRDM	Direct Debit Mandate Query	GetDirectDebit	ReturnDirectDebit	ReturnDirectDebit
CRDM	Directory Query	GetDirectory	ReturnDirectory	ReturnDirectory
CRDM	Error Code Query	GetErrorCode	ReturnErrorCode	ReturnErrorCode
CRDM	Event Query	GetBusinessDayInformation (camt.018) [309]	ReturnBusinessDayInformation (camt.019) [311]	ReturnBusinessDayInformation (camt.019) [311]
CRDM	Participant Reference Data Query	PartyQuery (reda.015) [417]	PartyReport (reda.017) [417]	PartyReport (reda.017) [417]
CRDM	Party Reference Data Query	PartyQuery (reda.015) [417]	PartyReport (reda.017) [417]	PartyReport (reda.017) [417]
CRDM	RTGS Cash Account Reference Data Query	AccountQuery (acmt.025) [271]	AccountReport (acmt.026) [272]	AccountReport (acmt.026) [272]
CRDM	Sub Account Reference Data Query	AccountQuery (acmt.025) [271]	AccountReport (acmt.026) [272]	AccountReport (acmt.026) [272]
CRDM	Standing Order Limits Query	GetLimit (camt.009) [302]	ReturnLimit (camt.010) [304]	ReturnLimit (camt.010) [304]
CRDM	Standing Order Liquidity Transfer Query	GetStandingOrder (camt.069) [359]	ReturnStandingOrder (camt.070) [359]	ReturnStandingOrder (camt.070) [359]
CRDM	Standing Order Reservations Query	GetReservation (camt.046) [325]	ReturnReservation (camt.047) [327]	ReturnReservation (camt.047) [327]
Scheduler	System Time Query	GetBusinessDayInformation (camt.018) [309]	ReturnBusinessDayInformation (camt.019) [311]	ReturnBusinessDayInformation (camt.019) [311]

10.4.2 Receive report (completed)

This is a general description of the RTGS process “Receive report”. RTGS uses reports to periodically provided RTGS actors with a defined set of data according to their data scope and access rights.

The chapter RTGS report generation [▶ 172] describes the respective business scope.

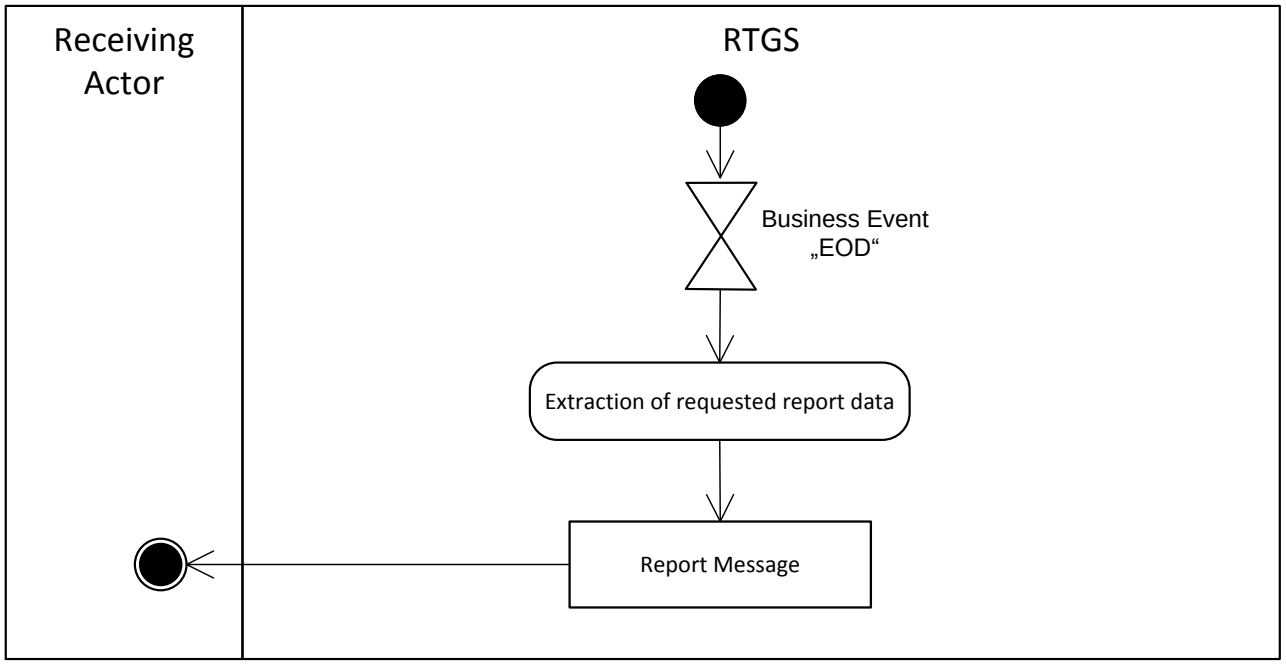


Figure 39 - RTGS receive report

The defined business event end of day [EOD] triggers the process “Extraction of requested report data”. It uses the report configuration in order to provide all necessary reports on the basis of the configured RTGS dedicated cash account. The RTGS component creates the report, including making the necessary calculations on raw data for aggregated values and storing them for further processing. RTGS sends the [Report message] via ESMIG to the receiving actor when a report configuration for the report is set up.

Report name	ISO message	ISO code
Statement of accounts	BankToCustomerStatement	camt.053

11 Dialogues and processes

11.1 Dialogues and processes between CRDM and CRDM actor (completed)

11.1.1 A2A Common reference data maintenance and query process (completed)

11.1.1.1 Reference data maintenance process (completed)

The common reference data maintenance 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 (completed)

The shared generic message flow is as follows:

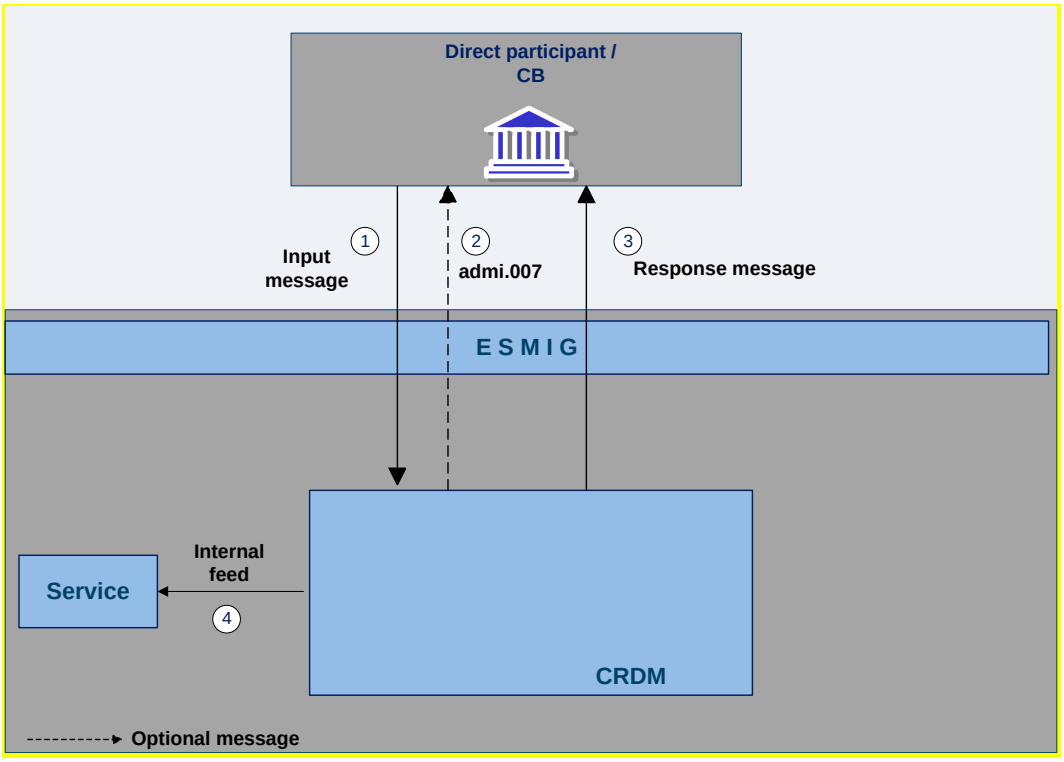


Figure 40 - Common reference data maintenance process

Table 105 - Common reference data maintenance process

Step	Activity
1	The authorised actor (participant, responsible central bank 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.

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.

Table 106 - Common reference data maintenance messages

Business scenario	Input message	Response message
Create standing order	camt.024	camt.025
Modify standing order	camt.024	camt.025
Delete standing order	camt.071	camt.025
Modify limit	camt.011	camt.025
Delete limit	camt.012	camt.025
Modify standing order for reservation	camt.048	camt.025
Delete standing order for reservation	camt.049	camt.025
Create cash account	acmt.007	acmt.010
Create cash account	acmt.007	acmt.011
Delete cash account	acmt.019	acmt.010
Delete cash account	acmt.019	acmt.011
Modify cash account	acmt.015	acmt.010
Modify cash account	acmt.015	acmt.011
Create party	reda.014	reda.016
Modify party	reda.022	reda.016
Delete party	reda.031	reda.016

11.1.1.2 Common reference data query (completed)

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 (completed)

The shared generic message flow is as follows:

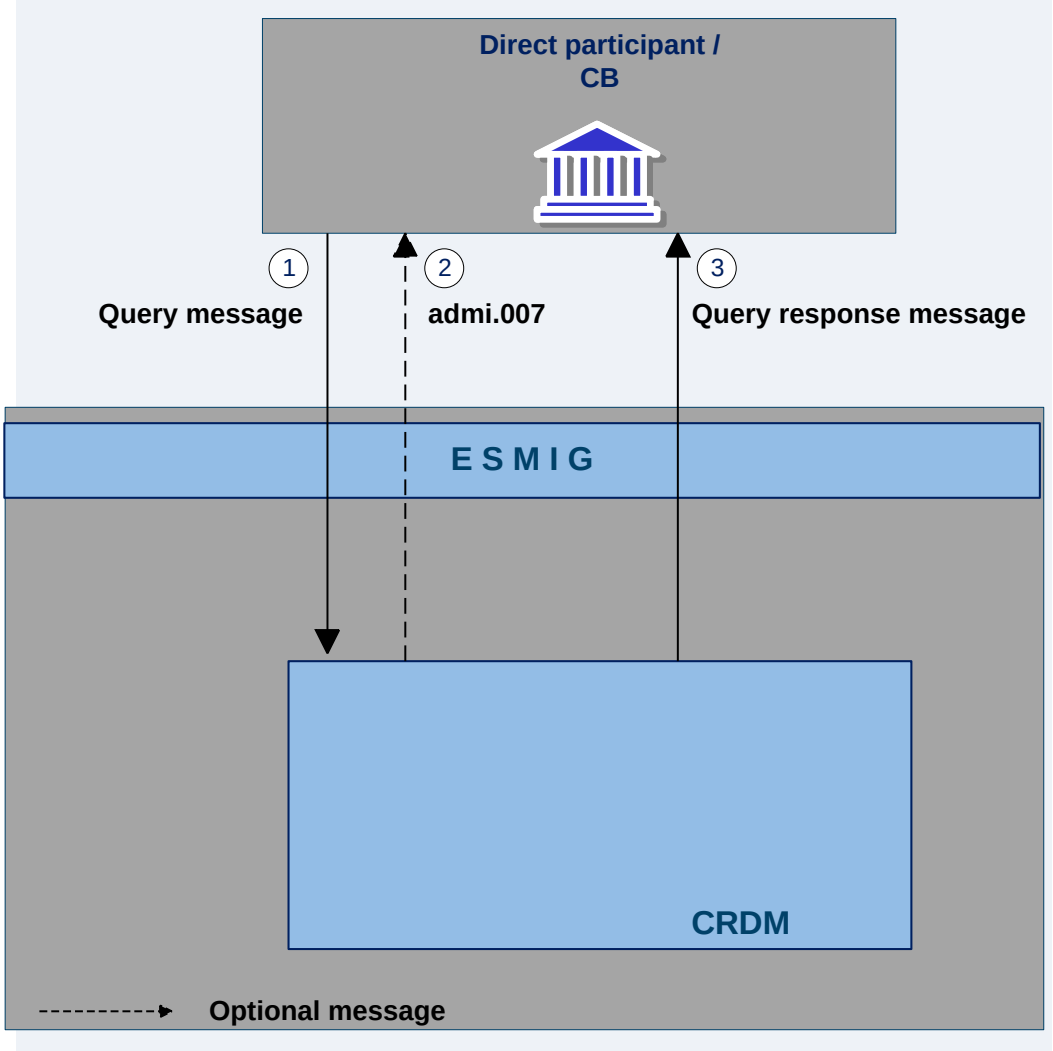


Figure 41 - Common reference data query process

Table 107 - 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, that is retrieved records or business error found during the validation.

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.

Table 108 - Common reference data query messages

CRDM entity	Query messages	Query response message
Standing order	camt.069	camt.070
Account	acmt.025	acmt.026
Account audit trail	reda.039	reda.040
Party	reda.015	reda.017
Party audit trail	reda.042	reda.043

11.1.2 Data migration tool file upload (completed)

11.1.2.1 Introduction (completed)

This use case covers the standard situation of a central bank or payment bank CRDM actor loading reference data into common CRDM common component. 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.

11.1.2.2 Activity diagram (completed)

The following diagram details all the processing steps of the DMT file upload use case.

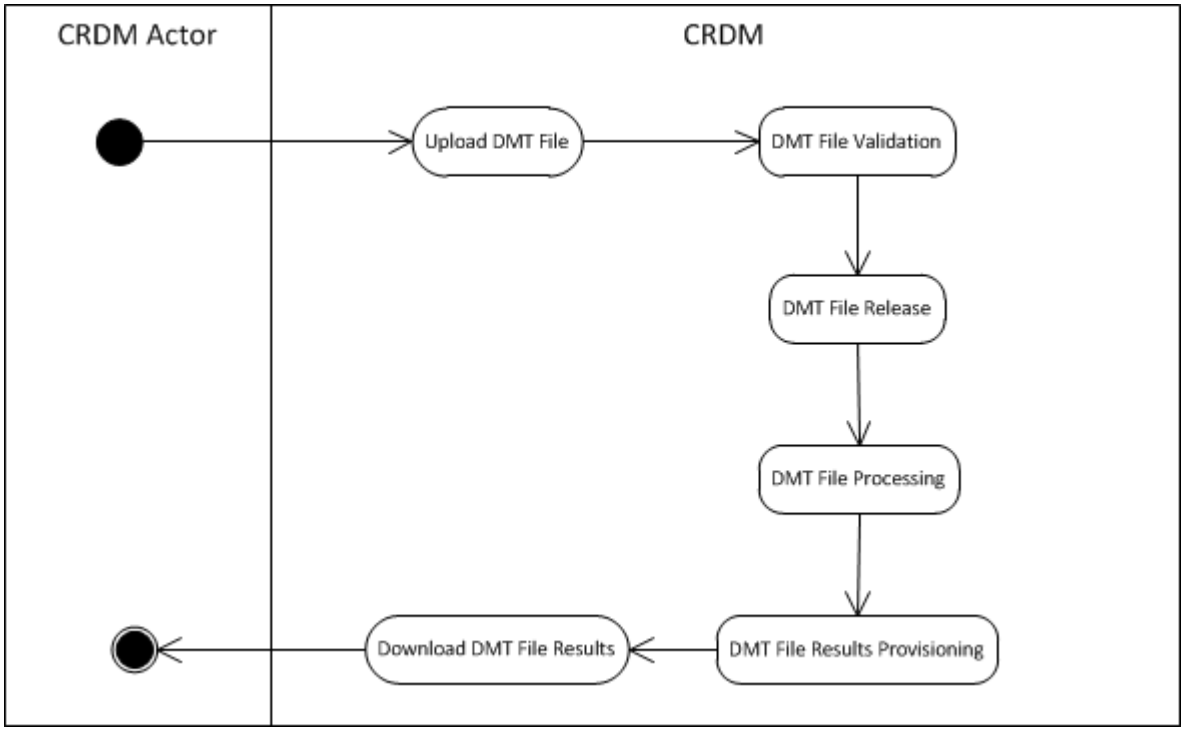


Figure 42 - DMT file upload process

11.1.2.2.1 Upload DMT file (completed)

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.

11.1.2.2.2 DMT file validation (completed)

CRDM performs a technical validation on the uploaded file to ensure that the technical constraints are respected.

11.1.2.2.3 DMT file release (completed)

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.

11.1.2.2.4 DMT file processing (completed)

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.

11.1.2.2.5 DMT file results provisioning (completed)

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.

11.1.2.2.6 Download DMT file results (completed)

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 contained in chapter Catalogue of messages.

Table 109 - DMT files 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

11.2 Dialogues and processes between ESMIG and participant (to be completed in iteration 4)

11.3 Dialogues and processes with data warehouse (to be completed in iteration 4)

11.4 Dialogues and processes with billing (to be completed in iteration 4)

III Catalogue of messages

12 Messages – introduction (completed)

The current messages are based on ISO 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.

13 Messages - general information

13.1 Message validation (completed)

13.1.1 Structure of ISO 20022 messages (completed)

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 version [19] 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.

The below example provides an indication of the overall structure of ISO 20022 messages.

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlIns:xsi=http://www.w3.org/2001/XMLSchema-Instance
xmlIns="urn:iso:20022:tech:xsd:camt.033.001.03">
    <camt.033.001.03>
        <Assgnmt>
            <Id>ABCDEFGHIJKLMNOPQRST123456789012345</Id>
            <Assgnr>CORPBE22</Assgnr>
            <Assgne>CHASUS33</Assgne>
            <CreDtTm>2002-07-21T08:35:30</CreDtTm>
        </Assgnmt>
        <Case>
            <Id>Case001</Id>
            <Cretr>CORPUK33</Cretr>
            <ReopCaseIndctn>true</ReopCaseIndctn>
        </Case>
    </camt.033.001.03>
</Document>
```

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.

---

19 A "variant" is a restricted version of a global message which fits the needs of a particular community while remaining in strict compliance with the original ISO 20022 message. For example, optional items can be removed or made mandatory, choices can be removed to keep no or fewer options, internal code lists can be reduced to the subset of codes that is actually used, size of text fields can be reduced, etc.
A "version" helps to cater for the evolution of message requirements and for the correction of possible problems and errors of a message. Upon the publication of a new message version a message switches from one way of being used to a new way of being used. Each message (variant) usually has one current version which is the most recent one. The former and the current version coexist for a certain while in order to ease the migration.
Example: Within the ReturnAccount message camt.004.001.01 the number 001 reflects the variant of the message in use whereas the number 01 reflects the current version of the message variant in use.

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. 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 20022 groups data types into standardised representation classes. These representation classes provide a set of possible data which can be inserted into the concerned message element.

For example, the message element “Bank Identifier” can be assigned to the representation class “BICIdentifier” or message element “Text” can be assigned to the representation class “Max35Text”.

Choice components allow the user of the message to choose between several possibilities. The message user may only choose one possible option in the instance.

Another term which specifies the partitioning within a message instance is the message item. Such a message item can be either a message building block or a message element. Message items which occur as XML tags within the message instance can appear at any level of nesting in the message.

A message building block is a message item which is specific to the concerned message (i.e. the user cannot find it in the ISO 20022 Data Dictionary). Within the corresponding schema file of the message the building block must be defined as an immediate child of the message. This is not to be confused with reusable groupings of one or more message elements, known as message components (i.e. that the user can find in the ISO 20022 Data Dictionary).

13.1.2 RTGS-specific schema customisation (completed)

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 were customised to adapt them to the specificities applicable in the context of RTGS.

The customisation of the schema files used in RTGS followed a particular approach which combines the needs of the RTGS actors to have a coherent logic across the messages and the need within RTGS to have a usable and efficient schema definition. RTGS derived this approach from the following customisation principles:

- customised RTGS schema files are compliant with the initial ISO 20022 schema files;
- when possible, RTGS customisation drops all the message elements with no direct connection to the user requirements of RTGS;
- when possible, RTGS customisation restricts element types to the RTGS-specific usage;
- RTGS customisation defines the necessary content of mandatory fields which cannot be pruned (i.e. “removed”) from the ISO schema files;
- RTGS customisation restricts the list of possible code values to the sole codes allowed in RTGS;
- RTGS customisation sets the length of the values to the length applicable in RTGS;
- RTGS customisation sets the occurrence of message elements to the occurrence applicable in RTGS;
- RTGS customisation makes optional message elements mandatory if their usage in RTGS is always compulsory;
- RTGS customisation restricts the allowed characters to those used in RTGS with a pattern;
- RTGS customisation restricts numeric fields applicable to RTGS (e.g. for amounts).

Based on the chosen approach four scenarios apply to the customisation for RTGS purposes:

1. a (part of a) message only contains elements which are supported by RTGS and there is hence no need for any pruning;
2. RTGS does not need a certain element but it cannot be pruned in the message because of a particular customer need;
3. neither RTGS nor RTGS actors need a certain element and therefore it is pruned;
4. neither RTGS nor its users need a certain element but as mandatory element in the ISO schema file it cannot be pruned and may be filled with a dummy value in RTGS.

For the scenarios 1, 3 and 4, RTGS only allows message elements according to the customised schema file. RTGS rejects any inbound message containing message elements which are not part of the RTGS customised schema file. Message elements under the scope of scenario 4 are not subject to further processing in RTGS. RTGS actors can hence fill these fields either with dummy values or real data (inserting real data does not lead to any processing, either).

For scenario 2 an alternative procedure applies. If message elements are present in the message and in the RTGS customised schema file although the message element is per se dispensable, RTGS nevertheless processes the message. For these message elements only schema validations are applicable. RTGS does not validate these elements against its business rules.

However, for all messages, RTGS prunes elements which are not within the general scope of its functionalities.

RTGS rejects messages during schema validation in cases where actors:

- Use elements in the message which are not present in the RTGS customised schema file;
- Use values in allowed elements but do not respect the restrictions of these values foreseen in the RTGS customised schema.

For RTGS outbound messages the logic for filling message elements customised to be optional is derived from the concrete circumstances and purposes of the concerned messages:

- For query response messages the filled message elements for outbound messages are those necessary to convey the information requested by the corresponding query message;
- For report messages the same applies, in accordance to the concrete configuration for the subscribed reports;

For any other RTGS outbound message the filling of optional fields also depends on either:

- The corresponding inbound message with its specific intention,
- Or the purpose of the RTGS-generated outbound message in case no inbound message precedes.

The sections “The message in business context” may contain message usages and/or message samples in which the content of given fields for a specific purpose or as a reply to a specific inbound message are depicted.

13.1.3 XML character set (completed)

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 [20].

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.

20 UTF-8 uses a single byte to represent 7-bit ASCII characters. Representation of extended characters takes between two and six bytes.

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).

Further restrictions to the character set will be defined.

13.1.3.1 Schema validation (completed)

All ISO 20022 messages which arrive at the RTGS Interface for further processing are subject to validation rules related to the syntax and structure of the message itself. In this context one can distinguish between well-formedness and validity of the message sent to RTGS.

An ISO 20022 message is well-formed if it satisfies the general syntactical rules foreseen for XML documents as outlined in the above chapter. The major aspects to be respected are the following:

The message only contains properly encoded Unicode characters;

- The specific syntax characters (e.g. “<” and “&”) are not used in the message except in their function as mark-up delineation;
- The element-delimiting tags (i.e. start, end and empty-element tags) are correctly nested and paired and none of them is missing or overlapping;
- The start and end tags match exactly and are case-sensitive;
- The message has one root element which contains all other elements.

In contrast to other forms of representation the definition of XML documents is rather strict. XML processors cannot produce reasonable results if they encounter even slight violations against the principle of well-formedness. Any violation of this well-formedness automatically entails an interruption of the message processing and an error notification to the sender.

Every well-formed ISO 20022 message arriving at the RTGS interface undergoes a validity check according to the rules contained in the enriched RTGS schema files. These RTGS enriched schemas make the structure of the message visible to the user and provide all necessary explanations on the validations the message undergoes.

The RTGS enriched schema files serve different purposes:

- They provide a definition of all the elements and attributes in the message;
- They provide a definition on what elements are child elements and on their specific order and number;
- They provide a definition of the data types applicable to a specific element or attribute;
- They provide a definition of the possible values applicable to a specific element or attribute.

RTGS provides the RTGS enriched schema file description in several formats: in xsd, Excel and pdf. This shall allow the user to accommodate himself with the format of his choice while having recourse to computer processable information to the largest extent.

A short extract from an xsd schema file for exemplary purposes:

[EXAMPLE xsd schema file of RTGS will be added later on]

Based on the relevant RTGS enriched schema, the RTGS interface performs the following validations for each incoming message instance:

- validation of the XML structure (starting from the root element);
- validation of the element sequencing (i.e. their prescribed order);
- validation of the correctness of parent-child and sibling relations between the various elements;
- validation of the cardinality of message elements (e.g. if all mandatory elements are present or if the overall number of occurrences is allowed);
- validation of the choice options between the message elements;
- validation of the correctness of the used character set;
- validation of the correctness of the code list values and their format.

13.1.3.1.1 Business validation (completed)

Besides validations which verify the correctness of the ISO 20022 message as XML document itself RTGS also conducts validations which are based on the business context RTGS operates in.

This business validation in RTGS takes place on the basis of a set of pre-defined business rules which are available in the appendix to this document.

On a general level RTGS verifies the validity of the transmitted message content against its static data repository.

In case of violations against existing business rules, RTGS transmits them to the relevant RTGS actors directly via an outbound message. This message contains all the information the RTGS actor needs to fully understand why e.g. an intended step of processing could not be completed by the system.

[EXAMPLE – extract of an outbound message sent in case of business rule validation will be added later on]

13.2 Communication infrastructure

13.2.1 Envelope messages

13.2.1.1 Business Application Header (partially completed)

Regardless of any (ongoing) standardisation discussions at ISO level a business application header (BAH) is defined in general for all messages which are used in RTGS.

The BAH is not applicable when:

- Referring to the acknowledgement of the receipt (admi.007) of a message within RTGS;
- Technical validation errors identified during the “A2A Business File Validation and Splitting process” are answered from RTGS by a ReceiptAcknowledgement (admi.007)

Technically speaking, the application header is a separate XML document standing apart from the XML documents which represent the message instance itself.

The business application header facilitates the message processing as it stores the information necessary for the processing at one central place. Without business application header 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 business application header improves the routing of the message once it arrives at the addressee’s interface.

The “Request Payload” stands for the whole communication data which is exchanged between and with RTGS.

BAH and business message (XML message instance) are part of this payload.

For example, the message element contained in the application header allows identifying immediately whether a sent message is a copy of a previously sent message.

13.2.1.2 Business File Header (partially completed)

Besides the sending of single messages RTGS supports the exchange of message batches (multi messages). Therefore, it is possible for the RTGS actor to send and receive a file composed of several messages. RTGS uses a business file header to assure the appropriate processing of such message batch. The file structure within is compliant to the requirement of the “Giovannini Protocol: File Transfer Rulebook (May 2007)”.

The business file header contains information about the sender, the creation date of the file and the included number of messages. It therefore differs from the business 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 the RTGS interface entail a receipt confirmation from RTGS. After the successful authentication check RTGS divides the file into single messages. Every message undergoes a separate validation (schema validation). RTGS reports errors on message level either by the corresponding response message or by a status message.

To communicate 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.

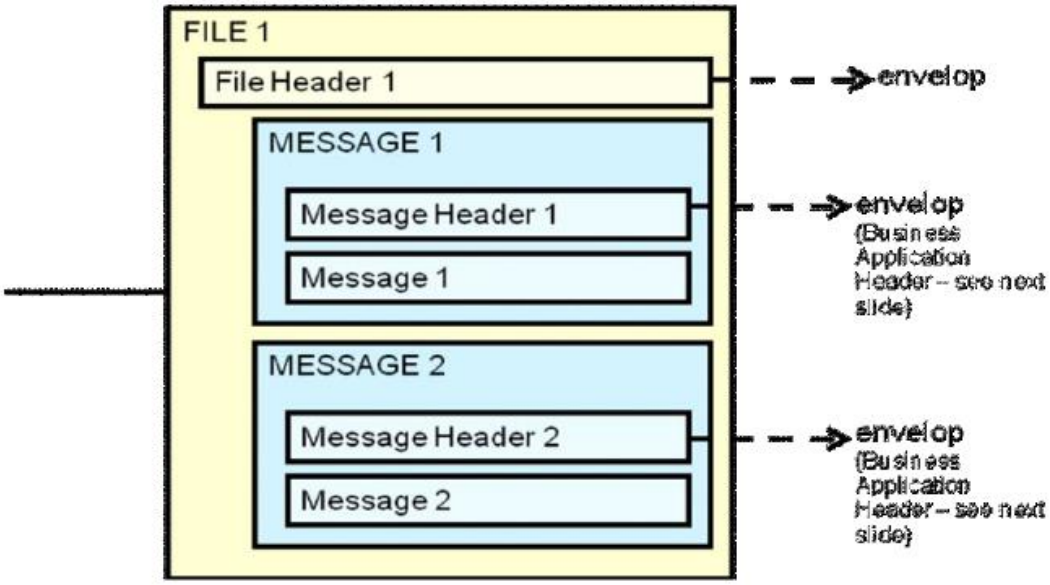


Figure 43 - Business file structure

[An example of the usage of the Business File Header will be added later]

13.2.1.2.1 Digital Signature managed within the Business Layer (partially completed)

13.2.1.3 Time zones (partially completed)

Messages exchanged between RTGS and its users consist of the business application header and the message payload. Both parts of the message contain time indications.

The relevant reference for all inbound and outbound communication in RTGS is Central European Time (CET) or Central European Summer Time (CEST). All indications contained in the payload of RTGS messages (based on given timestamps e.g.) refer to CET/CEST. The attribution of timestamps in the RTGS interface solely occurs on CET/CEST basis. All possible information related to time within the payload of mes-

sages sent to RTGS must refer to CET/CEST. The RTGS calendar as the relevant framework for all operational issues of RTGS contains CET/CEST only.

Due to the ISO definition of the application header the time indications within the application header refer to Zulu time. RTGS users must take into account the difference between the two time formats when exchanging messages with RTGS.

Example

A message sent to RTGS on 17 December 2015 at 10:30:47 CET/CEST would need to contain the following field in the application header (“ZULU time”):

```
<CreDt>2015-12-17T09:30:47Z</CreDt>
```

In case the same message contains within the payload an additional reference to the creation date of the message, it would need to contain the following information within the payload (“CET/CEST time”):

```
<CreDtTm>2015-12-17T10:30:47<CreDtTm>
```

13.2.1.4 Outbound traffic exceeding given size limitations (to be completed in iteration 4)

14 List of messages (partially completed)

Chapter	Message Code	Message Name
Administration (admi)		
ResendRequest (admi.006) [273]	admi.006	ResendRequest
ReceiptAcknowledgement (admi.007) [276]	admi.007	ReceiptAcknowledgement
Cash Management (camt)		
GetAccount (camt.003) [280]	camt.003	GetAccount
ReturnAccount (camt.004) [282]	camt.004	ReturnAccount
ReturnTransaction (camt.006) [292]	camt.006	GetTransaction
ModifyTransaction (camt.007) [299]	camt.007	ReturnTransaction
GetLimit (camt.009) [302]	camt.009	GetLimit
ReturnLimit (camt.010) [304]	camt.010	ReturnLimit
ModifyLimit (camt.011) [307]	camt.011	ModifyLimit
DeleteLimit (camt.012) [308]	camt.012	DeleteLimit
GetBusinessDayInformation (camt.018) [309]	camt.018	GetBusinessDayInformation
ReturnBusinessDayInformation (camt.019) [311]	camt.019	ReturnBusinessDayInformation
ReturnGeneralBusinessInformation (camt.021) [315]	camt.021	ReturnGeneralBusinessInformation
Receipt (camt.025) [318]	camt.025	Receipt
ResolutionOfInvestigation (camt.029) [320]	camt.029	ResolutionOfInvestigation
GetReservation (camt.046) [325]	camt.046	GetReservation
ReturnReservation (camt.047) [327]	camt.047	ReturnReservation
ModifyReservation (camt.048) [331]	camt.048	ModifyReservation
DeleteReservation (camt.049) [335]	camt.049	DeleteReservation
LiquidityCreditTransfer (camt.050) [336]	camt.050	LiquidityCreditTransfer

Chapter	Message Code	Message Name
BankToCustomerStatement (camt.053) [343]	camt.053	BankToCustomerStatement
BankToCustomerDebitCreditNotification (camt.054) [349]	camt.054	BankToCustomerDebitCreditNotification
FIToFIPaymentCancellationRequest (camt.056) [355]	camt.056	FIToFIPaymentCancellationRequest
Headers (head)		
BusinessApplicationHeader (head.001) [361]	head.001	BusinessApplicationHeader
BusinessFileHeader (head.002) [363]	head.002	BusinessFileHeader
Payments Clearing and Settlement (pacs)		
PaymentStatusReport (pacs.002) [365]	Pacs.002	PaymentStatusReport
PaymentReturn (pacs.004) [369]	Pacs.004	PaymentReturn
CustomerCreditTransfer (pacs.008) [373]	Pacs.008	CustomerCreditTransfer
FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [383]	Pacs.009	FinancialInstitutionCreditTransfer (GEN and COV)
FinancialInstitutionDirectDebit (pacs.010) [402]	Pacs.010	FinancialInstitutionDirectDebit

14.1 Account management (acmt)

14.1.1 AccountQuery (acmt.025)

14.1.1.1 Overview and scope of the message

This chapter illustrates the AccountQuery message.

The AccountQuery is sent by an actor authorised to query cash account reference data.

In response to the AccountQuery, an acmt.026 containing the requested information is returned.

14.1.1.2 Schema

Outline of the schema

The AccountQuery message is composed of the following message building blocks:

References

This block is mandatory and contains an identification used to uniquely and unambiguously identify the message.

AccountServicerIdentification

This block is mandatory. It contains the identification of the party receiving the request.

Organisation

This block is mandatory. It contains the identification of the party sending the request.

Account Search Criteria

This block is mandatory and provides with all the search criteria that must be used to filter Account records in the CRDM coverage.

References/links

The schema and the related documentation in HTML/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

14.1.2 AccountReport (acmt.026)

14.1.2.1 Overview and scope of the message

This chapter illustrates the AccountReport message.

The AccountReport is sent by CRDM to an authorised actor to provide with requested cash account information.

The AccountReport is sent in response to the acmt.025 message.

14.1.2.2 Schema

Outline of the schema

The AccountReport 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 central bank 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.

References/links

The schema and the related documentation in HTML/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

14.2 Administration (admi)

14.2.1 ResendRequest (admi.006)

14.2.1.1 Overview and scope of the message

This chapter illustrates the *ResendRequest* message.

The *ResendRequest* message is sent by directly connected RTGS participants to RTGS. It is used to request the resending of a message or a file (a duplicate of the original message/file) supported by RTGS.

Within RTGS, the *ResendRequest* message usage supports resend requests for the following messages:

- BankToCustomerStatement (camt.053) [343]

The *ResendRequest* message must provide party Technical Address of the RTGS participant to receive the resent message. This usage is described below, in the chapter “The message in business context”.

In response to the *ResendRequest* message, RTGS sends out either:

- ReceiptAcknowledgement (admi.007) advising of an error

or, simultaneously

- ReceiptAcknowledgement (admi.007) [276]advising of a successful validation
- the requested resend message (i.e. ReceiptAcknowledgement (admi.007) [276])

14.2.1.2 Schema

Outline of the schema.

The *ResendRequest* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the request message.

ResendSearchCriteria

Defines the criteria required to unambiguously identify the information to be resent.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/admi.006.001.01_RTGS

Business rules applicable to the schema

For business rules applicable to *ResendRequest* please refer to the business rules table below.

14.2.1.3 The message in business context

Usage case: Resend BankToCustomerStatement

In this usage case, the message clearly identifies a previously created BankToCustomerStatement (camt.053) [343] message which then is triggered for resending by RTGS.

Specific message requirements

The search criteria name is mandatory and at least one of either SequenceNumber or OriginalMessage-NameIdentification must be supplied.

All content must comply with the business rules for the message.

Table 110 - admi.006_ResendRequest_MessageContent

Message item	Data type/code	Utilisation
ResendRequestV01 Document/RsndReq	ResendRequestV01	
BusinessDate Docu-ment/RsndReq/RsndSchCrit/BizDt	ISODate	Original sending date of the camt.053 message/s to be resent.
SequenceNumber Docu-ment/RsndReq/RsndSchCrit/SeqNb	RTGS_Max16NumericText	Sequence number of the original camt.053 message to be resent.
OriginalMessageNameIdentification Docu-ment/RsndReq/RsndSchCrit/OrgnlMsgNmId	RTGS_RestrictedFINXMax35Text	Unique message number of the original camt.053 message to be resent.
Name RsndReq/RsndSchCrit/Rcpt/NmAndAdr/Nm	Max256Text	Original recipient name of the camt.053 message/s to be resent.

Usage case example: (placeholder)admi006.001.01_RTGS_ResendRequest_Example.xml

14.2.2 ReceiptAcknowledgement (admi.007)

14.2.2.1 Overview and scope of the message

This chapter illustrates the *ReceiptAcknowledgementV01* message.

The *ReceiptAcknowledgement* message is sent by RTGS to a directly connected RTGS participant. It is used to reject the reception of a previously sent message, or to notify the success of a ResendRequest (admi.006) [▶ 273]. Within RTGS this message is generated after a negative authentication process. It can be also sent as an error reporting response to a report query or resend request and as a validation result notification to a resend request.

This message is sent by RTGS in the following message usages:

- missing authentication (without BAH)
- inbound processing rejections
- RejectionResend
- validation result resend
- oversize and timeout

These usages are described below, in the chapter “The message in business context”.

In general, the *ReceiptAcknowledgement* message is sent by RTGS without a BAH.

14.2.2.2 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.

Report

This building block is mandatory and is composed of the individual RelatedReference and RequestHandling blocks.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/admi.007.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReceiptAcknowledgement* message.

14.2.2.3 The message in business context

Message usage: Missing authentication

The system-acknowledgement message is used in this usage to report that RTGS is not able to process an incoming message because of failed authentication of the sending party (sender authentication NOK or decryption NOK).

Specific message content

Table 111 - admi.007_MissingAuthentication_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	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. Used in case of BR short names: list TBD
Description Document/RctAck/Rpt/ReqHdlg/Desc	RestrictedFINXMax140Text	Textual description of the technical validation error specified in the status code field. Used in case of BR short names: list TBD

Message usage example: admi.007.001.01_RTGS_MissingAuthentication_Example.xml

Message usage: Inbound processing rejections

The ReceiptAcknowledgement is used in this usage by RTGS to inform the sender that an incoming message has caused an error during its processing. It reports the error which occurred in an error code and, if available, in a textual description.

Specific message content

Table 112 - admi.007_InboundProcessingRejections_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	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

Message usage example: admi.007.001.01_RTGS_InboundProcessingRejections_Example.xml

Message usage: RejectionResend

The ReceiptAcknowledgement message is used in this usage to inform the sender about the rejection (check permission resend NOK) of an incoming message.

Specific message content

Table 113 - admi.007_RejectionResend_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	MsgID of the incoming message this ReceiptAcknowledgement is sent for
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status code specifying the missing permission error
Description Document/RctAck/Rpt/ReqHdlg/Desc	RestrictedFINXMax140Text	Permission denied

Message usage example: admi.007.001.01_RTGS_Rejectionresend_Example.xml

Message usage: Validation result-resend

The ReceiptAcknowledgement validation result resend message is used in this usage to inform the sender of a message that their request for resending a message could be successfully processed by RTGS. It reports the positive status in a code.

Specific message content

Table 114 - admi.007_ValidationResultResend_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	MsgID of the incoming message this ReceiptAcknowledgement is sent for
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status code “OK”

Message usage example: admi.007.001.01_RTGS_ValidationResultResend_Example.xml

Message usage: Oversize and Timeout

The ReceiptAcknowledgement message is used in to inform the sender about an oversize and timeout scenario. The related reference indicates "NONREF". The correlation to the query has to be identified on network layer.

Specific message content

Table 115 - admi.007_OversizeAndTimeout_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	Always populated with NON-REF
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status code indicating the error which occurred during the technical validation. Used in case of BR short names: list TBD
Description Document/RctAck/Rpt/ReqHdlg/Desc	RestrictedFINXMax140Text	Textual description of the technical validation error specified in the status code field. Used in case of BR short names: list TBD

Message usage example: admi.007.001.01_RTGS_OversizeAndTimeout_Example.xml

14.3 Cash management (camt)

14.3.1 GetAccount (camt.003)

14.3.1.1 Overview and scope of the message

This chapter illustrates the GetAccount message.

The *GetAccount* message is sent by a RTGS Participant (or on their behalf by an authorised party) to RTGS. It is used to request balances including credit line of one RTGS dedicated cash account held at RGTS.

The *GetAccount* message contains the criteria which is used to select the response information

Within RTGS, the *GetAccount* message has the following usages:

- query account balance

This usage is described below, in the chapter “The message in business context”.

In response to the *GetAccount* message, a ReturnAccount (camt.004) [▶ 282] 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.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.003.001.06_RTGS

Business rules applicable to the schema

For business rules applicable to *GetAccount* please refer to the business rules table below.

14.3.1.3 The message in business context

Usage case: Query account balance

In this usage case, the message identifies the RTGS dedicated cash account for which the balances are being queried.

Specific message requirements

The search criteria name is mandatory and at least one of either SequenceNumber or OriginalMessageNameIdentification must be supplied.

All content must comply with the business rules for the message.

Table 116 - camt.003_GetAccount_MessageRequirements

Message item	Data type/code	Utilisation
Sub-Account ID Document/GetAcct/AcctQryDef/AcctCrit/NewCrit/Schcrit/AcctId/EQ/Othr/ID	Max34Text	RTGS dedicated cash account – sub-account.
Currency Document/GetAcct/AcctQryDef/AcctCrit/NewCrit/Schcrit/Ccy	ActiveOrHistoricCurrencyCode	Currency of the account / sub-account to be queried.
Account owner Document/GetAcct/AcctQryDef/AcctCrit/NewCrit/Schcrit/AcctOwnr/ID/OrgId/AnyBIC	BICFIIdentifier	BIC of the RTGS participant owning the account / sub-account to be queried.

Usage case example: camt.003.001.06_RTGS_GetAccount_Example.xml

14.3.2 ReturnAccount (camt.004)

14.3.2.1 Overview and scope of the message

This chapter illustrates the *ReturnAccountV07* message.

The *ReturnAccount* message is sent by RTGS to a RTGS participant (or a party authorised by them). It is used to provide information on the balances of one RTGS dedicated cash account held at RTGS by the RTGS participant.

Within RTGS, the *ReturnAccount* message has the following usages:

- query account balance response
- information to RTGS particicpant – floor notification
- information to RTGS particicpant – ceiling notification

These usages are described below, in the chapter “The message in business context”.

The *ReturnAccount* query account balance response message is sent in response to a GetAccount (camt.003) [▶ 280] message, which requested the information.

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.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.004.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *ReturnAccount* please refer to the business rules table below.

14.3.2.3 The message in business context

Usage case: Query account balance response

In this usage case, the message informs the sender of a GetAccount (camt.003) [▶ 280] of the account balance/s for the queried RTGS dedicated cash account. The response is sent in real time and contains the latest cash balances available.

Specific message content

To denote the usage case, the field Type contains the value'SACC'.

Table 117 - camt.004_ReturnAccountGetAccountQueryResponse_MessageContent

Message item	Data type/code	Utilisation
ReturnAccountV07 Document/RtrAcct	ReturnAccountV07	
Sub-Account ID Document/RtrAcct/RptOrErr/AcctRpt/AcctId/Othr/ID	Max34Text	RTGS account / sub-account being informed.
Reporting type Document/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/Tp/Cd	CashAccountType2Choice	'SACC'.
Currency Document/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the account / sub-account being informed.
Account owner Document/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/Ownr/ID/OrgId/AnyBIC	BICFIIdentifier	BIC of the RTGS participant owning the account / sub-account being informed.
Mutiple balance repetitions as below		

Message item	Data type/code	Utilisation
Balance amount Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/Amt	ImpliedCurrencyAndAmount	Balance amount.
Credit/Debit Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/CdtDbtInd	CreditDebitCode	'CRDT' or 'DBIT'.
Balance type Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/Tp/Cd	BalanceType8Choice	The type of balance informed in this multiple balance occurrence.
Balance status Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/Sts	BalanceStatus1Code	'PNDG' or 'STLD'.
Value date Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/ValDt/Dt	ISODate	Date of the balance amount informed.
Restriction type Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/RstrctnTp/Tp/ID	Max35Text	RTGS restriction type identification.
Processing code Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/RstrctnTp/PrcgTp/Cd	ProcessingType1Choice	RTGS restriction type processing code

Usage case example 1: camt.004.001.07_RTGS_ReturnAccountGetAccountQueryResponse_Example.xml

When RTGS needs to report an error processing the request the following fields are used.

Table 118 - camt.004_ReturnAccountGetAccountQueryResponseErr_MessageContent

Message item	Data type/code	Utilisation
ReturnAccountV07 Document/RtrAcct	ReturnAccountV07	
Error Code Document/RtrAcct/RptOrErr/OprlErr/Err/Prtry/	Max35Text	RTGS code for the problem being informed.
Error Description Document/RtrAcct/RptOrErr/OprlErr/Err/Desc	Max140Text	Description of the problem being informed.

Usage case example 2: camt.004.001.07_RTGS_ReturnAccountGetAccountQueryResponseErr_Example.xml

Usage case: Information to RTGS participant – floor notification

In this usage case, the message informs the RTGS participant that the current balance of a RTGS dedicated cash account has fallen below the defined minimum amount for the account.

Specific message content

To denote the usage case, TBA

Table 119 - camt.004_ReturnAccountFloorNotification_MessageContent

Message item	Data type/code	Utilisation
ReturnAccountV07 Document/RtrAcct	ReturnAccountV07	
MessageIdentification Document/RtrAcct/MsgHdr/OrgnlBizQry/MsgId	Max35Text	'NONREF'.

Message item	Data type/code	Utilisation
Sub-Account ID Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctId/Othr/ID	Max34Text	RTGS account / sub-account being informed.
Balance amount Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/Amt	ImpliedCurrencyAndAmount	Balance amount which is below the allowed floor value.
Balance type Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/Tp/Cd	BalanceType8Choice	The type of balance informed in this multiple balance occurrence.
Value date Docu-ment/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/ValDt/Dt	ISODate	Date of the balance amount informed.

Usage case example: camt.004.001.07_RTGS_ReturnAccountFloorNotification_Example.xml

Usage case: Information to RTGS participant – ceiling notification

In this usage case, the message informs the RTGS participant that the current balance of a RTGS dedicated cash account has risen above the defined maximum amount for the account.

Specific message content

To denote the usage case, TBA

Table 120 - camt.004_ReturnAccountCeiling Notification_MessageContent

Message item	Data type/code	Utilisation
ReturnAccountV07 Document/RtrAcct	ReturnAccountV07	
MessageIdentification Docu-ment/RtrAcct/MsgHdr/OrgnlBizQry/MsgId	Max35Text	'NONREF'.

Message item	Data type/code	Utilisation
Sub-Account ID Document/RtrAcct/RptOrErr/AcctRpt/AcctId/Othr/ID	Max34Text	RTGS account / sub-account being informed.
Balance amount Document/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/Amt	ImpliedCurrencyAndAmount	Balance amount which is above the allowed ceiling value.
Balance type Document/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/Tp/Cd	BalanceType8Choice	The type of balance informed in this multiple balance occurrence.
Value date Document/RtrAcct/RptOrErr/AcctRpt/AcctOrErr/Acct/MulBal/ValDt/Dt	ISODate	Date of the balance amount informed.

Usage case example: camt.004.001.07_RTGS_ReturnAccountCeilingNotification_Example.xml

14.3.3 GetTransaction (camt.005)

14.3.3.1 Overview and scope of the message

This chapter illustrates the *GetTransaction* message.

The *GetTransaction* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to request information about payment orders and payments held in RTGS.

The GetTransaction message can be used to request payment information based upon optional multiple criteria.

- within RTGS, the *GetTransaction* message has the following usages:
- query payment files

This usage is described below, in the chapter “The message in business context”.

In response to the *GetTransaction* message, a ReturnTransaction (camt.006) [▸ 292] 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.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.005.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *GetTransaction* please refer to the business rules table below.

14.3.3.3 The message in business context

Usage case: Query payment files

In this usage case, the message identifies the payment instruction/s which are being queried.

The following fields can be used to restrict the query. If no criteria are specified then the query returns all information consistent with the requestor's access rights:

- query type: all, changed, modified, deleted
- debit party and account
- original payment reference
- payment date
- status (within RTGS)
- instructed amount
- currency

- payment method
- payment type
- priority
- processing validity time

The construction of the data in the response can also be defined, using return criteria.

Specific message requirements

All content must comply with the business rules for the message.

Table 121 - camt.005_GetTransaction_MessageRequirements

Message item	Data type/code	Utilisation
Query type Document/GetTx/TxQryDef/QryTp	QueryTypeCode	All: 'ALLL', Changed: 'CHNG', Modified: 'MODF', Deleted: 'DELD'.
Query name Document/GetTx/TxQryDef/TxCrit/QryNm	Max35Text	A name for recalling the criteria used in this search.
Payment to BIC Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtTo/MmbId/FinInstnId/BICFI	CSLD_BIC11Text	BIC identifier of the payment receiver,
Payment to country Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtTo/Ctry	CountryCode	Country of the payment.
Payment from BIC Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtFr/MmbId/FinInstnId/BICFI	CSLD_BIC11Text	BIC identifier of the payment receiver.
Payment from country Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtFr/Ctry	CountryCode	Country of the payment.

Message item	Data type/code	Utilisation
Message ID Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/MsgId	CSDL_RestrictedFINXMax35Text	Unique message ID of the original payment instruction.
Requested execution date Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/ ReqdExctnDt/various	ISODate	Selection criteria for: From date, To date, From-to-date range, A single date.
Payment ID Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/PmtId/LngBizId/various	Various – please see MyStandards	Payment identification using: identifier, settled amount, settled date.
Status Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Sts/PmtInstrSts/various	Various – please see MyStandards	Payment status including: pending status, final status, status time & date, Status reason (failure).
Instructed amount Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/InstdAmt/various	Various – please see MyStandards	Instructed amount including: amount range, currency, credit debit indicator.
Instructed amount ccy Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/InstdAmtCcy/various	ActiveOrHistoricCurrencyCode	Currency of the instructed payment.
Credit debit ind Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/CdtDbtInd	CreditDebitCode	Credit / debit indicator of the settled payment.

Message item	Data type/code	Utilisation
Interbank settlement amount Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/IntrBkSttlmAmt/various	Various – please see MyStandards	I-bank settlement amount including: amount range, currency, credit debit indicator.
Interbank settlement amount ccy Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/IntrBkSttlmAmtCcy	ActiveCurrencyCode	Currency of the settled interbank payment.
Payment method Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/PmtMtd/various	Various – please see MyStandards	Payment method including: message types, proprietary types.
Payment type Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/PmtTp/Prtry	PaymentTypeCode	Payment type.
Priority Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Prty/Cd	PriorityCode	Priority code.
Processing validity time Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/PrcgVldtyTm/various	Various – please see MyStandards	Selection criteria for: From datetime, To datetime, From-to-datetime range,
Processing instruction Document/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Instr	Instruction1Code	Instruction code.

Message item	Data type/code	Utilisation
Parties Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/Pties/various	Various – please see MyStandards	Party information including : debtor, debtor agent, intermediary agent, creditor agent, creditor.
Account entry Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/SchCrit/PmtSch/AcctNtrySch/various	Various – please see MyStandards	Cash search options including: account ID, entry date,
Return criteria Docu-ment/GetTx/TxQryDef/TxCrit/NewCrit/RtrCrit/various	Various – please see MyStandards	Return criteria including : payment-to info, payment-from info, cash entry info, payment return info.

Usage case example: camt.005.001.07_RTGS_GetTransaction_Example.xml

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 RTGS to a RTGS participant (or a party authorised by them). It is used to provide information on the details of one or more payment orders and/or payments held in RTGS.

The *ReturnTransaction* message contains such payment information based upon dedicated cash accounts held at RTGS by the RTGS participant and upon the criteria provided in the request.

Within RTGS, the *ReturnTransaction* message has the following usages:

- response to transaction query GetTransaction (camt.005) [▸ 287]

This usage is described below, in the chapter “The message in business context”.

The *ReturnTransaction* message is sent in response to a GetTransaction (camt.005) [287] 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.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.006.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnTransaction* response message.

14.3.4.3 The message in business context

Usage case: Response to transaction query

In this usage case, the message informs the sender of a GetTransaction (camt.005) [287] of the payments information on the queried RTGS dedicated cash account. The response is sent in real time and contains the latest payment information available.

The selection of payments for this response is determined by the search criteria supplied on the query request.

Specific message content

Table 122 - camt.006_ReturnTransaction_MessageContent

Message item	Data type/code	Utilisation
Payment in queue Document/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/QId/QId	Max16Text	Payment in a queue.
Payment in queue position Document/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/QId/PosInQ	Max16Text	Position of payment in queue.
Long biz transaction ID Document/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/LngBizId/TxId	Max35Text	Long business ID – transaction ID.
Long biz i-bank settlement amount Document/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/LngBizId/IntrBkSttlmAmt	CSLD_Max14_Max2DecimalAmountImpliedCurrency	Long business ID – Interbank settlement amount.
Long biz i-bank settlement date Document/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/LngBizId/ IntrBkSttlmDt	ISODate	Long business ID – Interbank settlement date.
Long biz payment method - xml Document/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/LngBizId/PmtMtd/XMLMsgNm	RTGS_XMLMessageNamePattern	Long business ID – payment method - xml.
Long biz payment method - proprietary Document/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/LngBizId/PmtMtd/Prtry	RTGS_PaymentMethodCode	Long business ID – payment method - proprietary.
Long biz instructing agent Document/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/LngBizId/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	Long business ID – Instructing agent.

Message item	Data type/code	Utilisation
Long biz instructed agent Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/LngBizId/InstdAgt/FinInstnId/BICFI	BICFIIdentifier	Long business ID – Instructed agent.
Long biz instructed agent end-to-end id Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/LngBizId/InstdAgt/EndToEndId	Max35Text	Long business ID – Instructed agent, end-to-end iD.
Short biz transaction ID Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/ShrtBizId/TxId	Max35Text	Short business ID – transaction ID.
Short biz i-bank settlement date Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/ShrtBizId/ IntrBkSttlmDt	ISODate	Short business ID – interbank settle-ment date.
Short biz instructing agent Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/PmtId/ShrtBizId/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	Short business ID – instructing agent.
Payment to BIC Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/PmtTo/MmbId/FinInstnId/BICFI	BICFIIdentifier	Payment-to BIC.
Payment from BIC Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/PmtFr/MmbId/FinInstnId/BICFI	BICFIIdentifier	Payment-from BIC.
Credit / Debit ind Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/CrdDbtInd	CreditDebitCode	Credit / debit indicator.

Message item	Data type/code	Utilisation
Message ID Document/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/MsgId	Max35Text	Identification of the instructing message.
Status code Document/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Sts/Cd	PaymentStatusCodeChoice	Payment status code.
Status datetime Document/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Sts/DtTm/DtTm	ISODateTime	Payment status date and time.
Reject reason code Document/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Sts/Rsn/PrtryRjctn/PrtrystsRsn	Max4AlphaNumericText	Reason code for why the payment was rejected
Reject reason text Document/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Sts/Rsn/PrtryRjctn/Rsn	Max256Text	Reason text for why the payment was rejected
Instructed amount Document/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/InstdAmt/AmtWthCcy	CSLD_Max14_Max5DecimalAmount	Instructed payment amount
I-bank settlement amount Document/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/IntrBkSettlmAmt/AmtWthCcy	CSLD_Max14_Max5DecimalAmount	Interbank settlement amount
Payment method - xml Document/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/PmtMtd/XMLMsgNm	RTGS_XMLMessageNamePattern	Payment method – XML name

Message item	Data type/code	Utilisation
Payment method - proprietary Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/PmtMtd/Prtry	RTGS_PaymentMethodCode	Payment method – proprietary
Priority Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Prty/Cd	PriorityCode	Priority
Processing validity time - from Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/PrcgVldtyTm/FrDtTm	ISODateTime	Processing validity time - from
Processing validity time - to Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/PrcgVldtyTm/ToDtTm	ISODateTime	Processing validity time - to
Processing validity time – range from Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/PrcgVldtyTm/DtTmRg/FrDtTm	ISODateTime	Processing validity time – range from
Processing validity time – range to Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/PrcgVldtyTm/DtTmRg/ToDtTm	ISODateTime	Processing validity time – range to
Instruction copy Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/InstrCpy	Max20000Text	Instruction copy
Payment type Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Tp/Prtry	RTGS_PaymentTypeCode	Payment type

Message item	Data type/code	Utilisation
Transaction ID Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/TxId	Max35Text	Payment transaction ID
I-Bank settlement date Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/IntrBkSttlmDt	ISODate	Interbank settlement date
End-to-end ID Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/EndToEndId	Max35Text	Payment end-to-end ID
Debtor Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Pties/Dbtr/ Fin-InstnId/BICFI	BICFIIdentifier	Debtor party BIC
Debtor agent Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Pties/DbtrAgt/FinInstnId/BICFI	BICFIIdentifier	Debtor agent party BIC
Intermediary agent Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Pties/IntrmyAgt/FinInstnId/BICFI	BICFIIdentifier	Intermediary agent party BIC
Creditor agent Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Pties/CdtrAgt/FinInstnId/BICFI	BICFIIdentifier	Creditor agent party BIC

Message item	Data type/code	Utilisation
Creditor Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/Pmt/Pties/Cdtr/FinInstnId/BICFI	BICFIIdentifier	Creditor party BIC
Account ID Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/AcctNtry/Acct/ID/Othr/ID	Max34Text	RTGS dedicated cash account ID
Entry date Docu-ment/RtrTx/RptOrErr/BizRpt/TxRpt/TxOrErr/Tx/AcctNtry/Ntry/Dt/DtTm	ISODateTime	Entry date

Usage case example 1: camt.006.001.07_RTGS_ReturnTransaction_Example.xml

When RTGS needs to report an error processing the request the following fields are used.

Table 123 - camt.006_ReturnTransactionErr_MessageContent

Message item	Data type/code	Utilisation
Error code Docu-ment/RtrTx/RptOrErr/OprlErr/Err/Prtry/	ErrorHandlingCode	RTGS code for the problem being in-formed.
Error description Docu-ment/RtrTx/RptOrErr/OprlErr/Desc	Max140Text	Description of the problem being in-formed.

Usage case example 2: camt.006.001.07_RTGS_ReturnTransactionErr_Example.xml

14.3.5 ModifyTransaction (camt.007)

14.3.5.1 Overview and scope of the message

This chapter illustrates the *ModifyTransaction*message.

The *ModifyTransaction* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to request modification to one payment order on the RTGS participant's dedicated cash account. The *ModifyTransaction* may only be used for a payment 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 RTGS participant wants to be applied to the relevant feature of the payment order identified in the message.

Within RTGS, the *ModifyTransaction* message has the following usages:

- change order of payments in a queue
- modify a payment
- queue management/payment order amendment

These usages are described below, in the chapter "The message in business context".

In response to the *ModifyTransaction* message, a Receipt (camt.025) [▸ 318] is sent, indicating the success or rejection/failure of the modification.

To further verify the outcome of the request, the RTGS participant may submit a GetTransaction (camt.005) [▸ 287] message with the appropriate search criteria.

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 repetitive. It identifies the list of modifications to be executed.

References/Links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.007.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *ModifyTransaction* please refer to the business rules table below.

14.3.5.3 The message in business context

Usage case: ONE

In this usage case, need to find out what fields are to be used.

Specific message requirements

All content must comply with the business rules for the message.

Table 124 - camt.007_ModifyTransaction_MessageRequirements

Message item	Data type/code	Utilisation
Payment ID Document/ModfyTx/Mod/PmtId/LngBizId/TxId	Max35Text	Payment identification set by the instructing agent
I-Bank settlement amt Document/ModfyTx/Mod/PmtId/LngBizId/IntrBkSttlmAmt	CSLD_Max14_Max2DecimalAmountImpliedCurrency	Interbank settlement amount
I-Bank settlement date Document/ModfyTx/Mod/PmtId/LngBizId/IntrBkSttlmDt	ISODate	Interbank settlement date
XML message name Document/ModfyTx/Mod/PmtId/LngBizId/PmtMtd/XMLMsgNm	CSLD_XMLMessageNamePattern	XML message name which instructed the payment
Instructing agent Document/ModfyTx/Mod/PmtId/LngBizId/InstgAgt/FinInstnId/BICFI	CSLD_BIC11Text	Instructing agent BIC

Message item	Data type/code	Utilisation
Instructed agent Docu-ment/ModfyTx/Mod/PmtId/LngBizId/InstdAgt/FinInstnId/BICFI	CSLD_BIC11Text	Instructed agent BIC
End to End ID Docu-ment/ModfyTx/Mod/PmtId/LngBizId/EndToEndId	Max35Text	End-to-end identifier – set by the pay-ment initiator
New details to be applied		
Priority code Docu-ment/ModfyTx/Mod/NewPmtValSet/Prty/Cd	PriorityCode	Priority code
Priority proprietary code Docu-ment/ModfyTx/Mod/NewPmtValSet/Prty/Prtry	CSLD_QueueReordering	Proprietary priority code
Validity period - from Docu-ment/ModfyTx/Mod/NewPmtValSet/PrcgVldtyTm/FrDtTm	ISODateTime	Validity period – from datetime
Validity period - from Docu-ment/ModfyTx/Mod/NewPmtValSet/PrcgVldtyTm/TorDtTm	ISODateTime	Validity period – to datetime

Usage case example: camt.007.001.07_RTGS_ModifyTransaction_Example.xml

14.3.6 GetLimit (camt.009)

14.3.6.1 Overview and scope of the message

This chapter illustrates the *GetLimit* message.

The *GetLimit* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to request details of one or more limits set by the RTGS participant (or on their behalf by an authorised party).

Within RTGS, the *GetLimit* message has the following usages:

- query limits

This usage is described below, in the chapter "The message in business context".

In response to the *GetLimit* message, a ReturnLimit (camt.010) [› 304] message containing the requested information is returned.

14.3.6.2 Schema

Outline of the schema.

The *GetLimit* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

LimitQueryDefinition

This building block is mandatory. It contains detailed information related to the business query about limit. It includes sections related to limit type, the credit consumer identifier, the currency code, the limit amount and an attribute to specify a search criteria “=” against the date from which the credit limit is valid.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.009.001.06_RTGS

Business rules applicable to the schema

For business rules applicable to *GetLimit* please refer to the business rules table below.

14.3.6.3 The message in business context

Usage case: Query limits

In this usage case, the message identifies the RTGS dedicated cash account owner. All limits defined for this party is selected for the response message.

Specific message requirements

All content must comply with the business rules for the message.

Table 125 - camt.009_LimitQuery_MessageRequirements

Message item	Data type/code	Utilisation
Account owner Docu-ment/GetLmt/LmtQryDef/LmtCrit/NewCrit/SchCrit/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Account owner of the RTGS dedicated cash account
Account owner Docu-ment/GetLmt/LmtQryDef/LmtCrit/NewCrit/SchCrit/AcctOwnr/FinInstnId/Nm	Max140Text	Name of the account owner of the RTGS dedicated cash account

Query type example: camt.009.001.06_RTGS_LimitQuery_Example.xml

14.3.7 ReturnLimit (camt.010)

14.3.7.1 Overview and scope of the message

This chapter illustrates the *ReturnLimit*message.

The *ReturnLimit* message is sent by RTGS to a RTGS participant (or a party authorised by them). It is used to respond to a limit query.

Within RTGS, the *ReturnLimit* message has the following usages:

- response to query limits (GetLimit (camt.009) [▸ 302])

This usage is described below, in the chapter “The message in business context”.

The *ReturnLimit* message is sent in response to a GetLimit (camt.009) [▸ 302] message, which requested the information.

14.3.7.2 Schema

Outline of the schema.

The *ReturnLimit* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the message and the original business query identification.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about limit message, or an error indication. It includes sections such as limit type, the credit consumer identifier, the currency code, the limit amount, the date from which the credit limit is valid.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.010.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnLimit* response message.

14.3.7.3 The message in business context

Specific message content

A limit query response contains information on the limit, RTGS dedicated cash account and institutions involved.

Table 126 - camt.010_LimitQueryResponse_MessageContent

Message item	Data type/code	Utilisation
Counterparty ID Docu-ment/RtrLmt/RptOrErr/BizRpt/CurLmt/LmtId/BilLmtCtrPtyId/FinInstnId/BICFI	BICFIIdentifier	Limit identification – counterparty BIC
Counterparty ID Docu-ment/RtrLmt/RptOrErr/BizRpt/CurLmt/LmtId/Tp/Cd	LimitTypeCode	Limit identification – owner of RTGS dedicated cash acccount
Party ID Docu-ment/RtrLmt/RptOrErr/BizRpt/CurLmt/LmtId/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Limit identification – party BIC
Limit amount Docu-ment/RtrLmt/RptOrErr/BizRpt/CurLmt/LmtOrErr/Lmt/Amt/AmtWthCcy	ActiveCurrencyAndAmount	Limit information – amount
Credit debit ind Docu-ment/RtrLmt/RptOrErr/BizRpt/CurLmt/LmtOrErr/Lmt/CrdDbtInd	CreditDebitCode	Limit information – credit or debit indi-cator

Usage case example 1: camt.010.001.07_RTGS_LimitResponse_Example.xml

The returned business data in case of an error response.

Table 127 - camt.010_LimitQueryResponse_ErrorContent

Message item	Data type/code	Utilisation
Error code Document/RtrLmt/RptOrErr/OprlErr/Err/Prtry/Cd	ErrorHandlingCode	RTGS code for the problem being informed.
Error description Document/RtrLmt/RptOrErr/OprlErr/Desc	Max140Text	Description of the problem being informed.

Usage case example 2: camt.010.001.07_RTGS_LimitResponseErr_Example.xml

14.3.8 ModifyLimit (camt.011)

14.3.8.1 Overview and scope of the message

This chapter illustrates the ModifyLimit message.

The ModifyLimit is sent by an authorised party for instructing the update of a limit, by providing details about the limit to be updated.

In response to the modify limit message, CRDM sends a camt.025 message when the update of the limit has been successfully performed or rejected.

14.3.8.2 Schema

Outline of the schema

The ModifyLimit message is composed of the following message building blocks:

MessageHeader

This building block is mandatory. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

LimitDetails

This block contains detailed information related to the limit to be updated.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/camt.011.001.006

14.3.9 DeleteLimit (camt.012)

14.3.9.1 Overview and scope of the message

This chapter illustrates the DeleteLimit message.

The DeleteLimit is sent by an authorised actor for instructing the deletion of a limit, by providing details about the limit to be deleted.

In response to the delete limit message, CRDM sends a camt.025 message when the deletion of the limit has been successfully performed or rejected.

14.3.9.2 Schema

Outline of the schema

The DeleteLimit message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and it contains an identification assigned by the sending party to uniquely and unambiguously identify the message.

LimitDetails

This building block is mandatory it contains detailed information related to the limit to be deleted.

References/links

The schema and the related documentation in HTML/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.012.001.006

14.3.10 GetBusinessDayInformation (camt.018)

14.3.10.1 Overview and scope of the message

This chapter illustrates the *GetBusinessDayInformationV04* message.

The *GetBusinessDayInformation* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to request information on different types of administrative data linked to the RTGS system.

Within RTGS, the *GetAccount* message has the following usages:

- query system time (GetSystemTime)
- query system time (GetBusinessDayInformation)

These usages are described below, in the chapter “The message in business context”.

In response to the *GetBusinessDayInformation* message, a ReturnBusinessDayInformation (camt.019) [▶ 311] message containing the requested information is returned.

14.3.10.2 Schema

Outline of the schema.

The *GetBusinessDayInformation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.018.001.04_RTGS

Business rules applicable to the schema

For business rules applicable to *GetBusinessDayInformation* please refer to the business rules table below.

14.3.10.3 The message in business context

Usage case: Get system time

In this usage case, the message is querying for the RTGS system time.

Specific message requirements

All content must comply with the business rules for the message.

Table 128 - camt.018_GetBusinessDayInformationGetSystemTime_MessageRequirements

Message item	Data type/code	Utilisation
Document/GetBizDayInf/MsgHdr/MsgId	Max35Text	Unique ID for the message
Document/GetBizDayInf/MsgHdr/ReqTp/Enqry	ExternalEnquiryRequestTypeCode	Proprietary value for RTGS system time query - TBC

Usage case example: camt.018.001.04_RTGS_GetBusinessDayInformationGetSystemTime_Example.xml

Usage case: Get business day information

In this usage case, the message is querying for RTGS business day information.

Specific message requirements

All content must comply with the business rules for the message.

Table 129 - camt.018_GetBusinessDayInformationGetBusinessDayInfo_MessageRequirements

Message item	Data type/code	Utilisation
Document/GetBizDayInf/MsgHdr/MsgId	Max35Text	Unique ID for the message
Document/GetBizDayInf/MsgHdr/ReqTp/Enqry	ExternalEnquiryRequestTypeCode	Proprietary value for RTGS business day info query - TBC

Usage case example: camt.018.001.04_RTGS_GetBusinessDayInformationGetBusinessDayInfo_Example.xml

14.3.11 ReturnBusinessDayInformation (camt.019)

14.3.11.1 Overview and scope of the message

This chapter illustrates the *ReturnBusinessDayInformation*message.

The *ReturnBusinessDayInformation* message is sent by RTGS to a RTGS participant (or a party authorised by them). It is used to provide information on the details of on different types of administrative data linked to the RTGS system.

The *ReturnBusinessDayInformation* message contains such administrative data information based upon the criteria provided in the request.

Within RTGS, the *ReturnBusinessDayInformation* message has the following usages:

- query system time (GetSystemTime)
- query system time (GetBusinessDayInformation)

These usages are described below, in the chapter “The message in business context”.

The *ReturnBusinessDayInformation* message is sent in response to a GetBusinessDayInformation (camt.018) [▸ 309] message, which requested the information.

14.3.11.2 Schema

Outline of the schema.

The *ReturnBusinessDayInformation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about business day information, or an error indication.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.019.001.06_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnBusinessDayInformation* response message.

14.3.11.3 The message in business context

Usage case: Get system time

In this usage case, the message informs of the RTGS system time.

Specific message content

Table 130 - camt.019_ReturnBusinessDayInformationGetSystemTime_MessageContent

Message item	Data type/code	Utilisation
System ID Docu-ment/RtrBizDayInf/RptOrErr/BizRpt/SysId/MktInfrstrctrId/Cd	ExternalMarketInfrastructureCode	Identification of the RTGS component
Business date Docu-ment/RtrBizDayInf/RptOrErr/BizRpt/BizDayOrErr/BizDayInf/SysDt	ISODate	Current business date of RTGS

Usage case example 1: camt.019.001.06_RTGS_ReturnBusinessDayInformationGetSystemTime_Example.xml

When RTGS needs to report an error processing the request the following fields are used.

Table 131 - camt.004_ReturnBusinessDayInformationGetSystemTimeErr_MessageContent

Message item	Data type/code	Utilisation
Error code Document/RtrBizDayInf/RptOrErr/OprlErr/Err/Prtry/	Max4Text	RTGS code for the problem being informed.
Error description Document/RtrBizDayInf/RptOrErr/OprlErr/Err/Desc	Max140Text	Description of the problem being informed.

Usage case example 2: camt.019.001.06_RTGS_ReturnBusinessDayInformationGetSystemTimeErr_Example.xml

Usage case: Get business day information

In this usage case, the message informs of the RTGS operational information.

Specific message content

Table 132 - camt.019_ReturnBusinessDayInformationGetBusinessDayInfo_MessageContent

Message item	Data type/code	Utilisation
System ID Document/RtrBizDayInf/RptOrErr/BizRpt/SysId/MktInfrstrctrId/Cd	ExternalMarketInfrastructureCode	Identification of the RTGS component
Business date Document/RtrBizDayInf/RptOrErr/BizRpt/BizDayOrErr/BizDayInf/SysDt	ISODate	Current business date of RTGS
Multiple currency repetitions		
Currency Document/RtrBizDayInf/RptOrErr/BizRpt/BizDayOrErr/BizDayInf/SysInfPerCcy/SysCccy	ActiveCurrencyCode	Currency tor which this repetition of data refers

Message item	Data type/code	Utilisation
Event type Docu-ment/RtrBizDayInf/RptOrErr/BizRpt/BizDayOr-Err/BizDayInf/SysInfPerCcy/Evt/Tp/Prtry/ID	Max4Text	RTGS proprietary codes - TBC
Scheduled time Docu-ment/RtrBizDayInf/RptOrErr/BizRpt/BizDayOr-Err/BizDayInf/SysInfPerCcy/Evt/Tp/SchdldTm	ISODateTime	Scheduled time for the event
Effective time Docu-ment/RtrBizDayInf/RptOrErr/BizRpt/BizDayOr-Err/BizDayInf/SysInfPerCcy/Evt/Tp/FctvTm	ISODateTime	Scheduled time for the event

Usage case example 1: camt.019.001.06_RTGS_ReturnBusinessDayInformationGetBusinessDayinfo_Example.xml

When RTGS needs to report an error processing the request the following fields are used.

Table 133 - camt.004_ReturnBusinessDayInformationGetBusinessDayInfoErr_MessageContent

Message item	Data type/code	Utilisation
Error code Docu-ment/RtrBizDayInf/RptOrErr/OprlErr/Err/Prtry/	Max4Text	RTGS code for the problem being in-formed.
Error description Docu-ment/RtrBizDayInf/RptOrErr/OprlErr/Err/Desc	Max140Text	Description of the problem being in-formed.

Usage case example 2: camt.019.001.06_RTGS_ReturnBusinessDayInformationGetBusinessDayInfoErr_Example.xml

14.3.12 ReturnGeneralBusinessInformation (camt.021)

14.3.12.1 Overview and scope of the message

This chapter illustrates the *ReturnGennerалBusinessInformation* message.

The *ReturnGeneralBusinessInformation* message is sent by RTGS to a RTGS participant (or a party authorised by them). It is used to provide information on the details of processing within the RTGS system.

Within RTGS, the *ReturnGeneralBusinessInformation* message has the following usages:

- response to ancillary system orders
- start of night time procedure
- start/end of procedure initiated by ancillary system

These usages are described below, in the chapter “The message in business context”.

The *ReturnGeneralBusinessInformation* is triggered by events and processing inside RTGS. It is not a response to any form of query.

14.3.12.2 Schema

Outline of the schema.

The *ReturnGeneralBusinessInformation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains information about business day information. In RTGS there is no error usage.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.021.001.06_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnGeneralBusinessInformation* response message.

14.3.12.3 The message in business context

Usage case: Response to ancillary system orders

Specific message content

Table 134 - camt.021_ReturnGeneralBusinessInformationResponseToASOrder_MessageContent

Message item	Data type/code	Utilisation
TBDC	TBCTBD	TBCTBD

Usage case example: camt.021.001.05_RTGS_ReturnGeneralBusinessInformationResponseToASOrder_Example.xml

Usage case: Start of night time procedure

Specific message content

Table 135 - camt.021_ReturnGeneralBusinessInformationStartOfNightTimeProcedure_MessageContent

Message item	Data type/code	Utilisation
TBDC	TBCTBD	TBCTBD

Usage case example: camt.021.001.05_RTGS_ReturnGeneralBusinessInformationStartOfNightTimeProcedure_Example.xml

Usage case: Start/End of procedure initiated by ancillary system

Specific message content

Table 136 - camt.021_ReturnGeneralBusinessInformationStartEndProcedureInitiatedbyAS_MessageContent

Message item	Data type/code	Utilisation
TBDC	TBCTBD	TBCTBD

Usage case example: camt.021.001.05_RTGS_ReturnGeneralBusinessInformationStartEndProcedureInitiatedbyAS_Example.xml

14.3.13 ModifyStandingOrder (camt.024)

14.3.13.1 Overview and scope of the message

This chapter illustrates the ModifyStandingOrder message.

The ModifyStandingOrder message is sent by an actor authorised to create or modify standing orders for liquidity transfers.

The ModifyStandingOrder message is replied by a camt.025 to return a positive technical response to the sender of the message or to provide detailed information in case of an error.

14.3.13.2 Schema

Outline of the schema

The ModifyStandingOrder message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with 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 HTML/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

14.3.14 Receipt (camt.025)

14.3.14.1 Overview and scope of the message

This chapter illustrates the *Receipt* message.

The *Receipt* message is sent by RTGS to a RTGS participant (or a party authorised by them). It is used to reply to a previously sent liquidity transfer order, payment order or order-related activity.

The *Receipt* message is used to inform the RTGS participant regarding the following business activities:

- amend an immediate reservation order
- ancillary system transaction processing
- change order of payments in a queue
- create a liquidity transfer
- create an immediate reservation order
- intra-RTGS liquidity transfer
- intra-service liquidity transfer between two RTGS accounts of the same liquidity transfer group (incl. sub-account)
- liquidity reservation
- modification of standing order for liquidity transfers
- modify a payment
- queue management/payment order amendment
- queue management/payment order cancellation
- response to amendment of standing order limit
- response to deletion of standing order limit
- response to modification of intra-service standing liquidity transfer order (incl. liquidity transfer to sub-account)
- response to modification of standing liquidity transfer order to ancillary system mirror account
- revoke a payment

Within RTGS, the *Receipt* message has the following usages:

- response to a previously sent message

This usage is described below, in the chapter “The message in business context”.

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. The camt.025 is used to return a positive response to the sender of the message or to provide detailed information in case of an error.

See above business actions for details.

14.3.14.2 Schema

Outline of the schema.

The *Receipt* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReceiptDetails

This building block is mandatory and non-repetitive. It contains information relating to the status of a previous instruction, with descriptive text if the status indicates a failure.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.025.001.04_RTGS

Business rules applicable to the schema

For business rules applicable to *Receipt* please refer to the business rules table below.

14.3.14.3 The message in business context

Usage case: Response to a previously sent message

In this case, RTGS provides to the sender of an action message, an update regarding the progress of their request.

This update could be one of many things, for example:

- acknowledging the arrival of the action message
- advising that the action is pending
- advising that the action is under way, or part done
- advising that the action request has been rejected completely (with a reason)

Specific message content

The actual status value used depends upon the nature of the original requested action, based upon the following table:

Action / Status table – TBD

Table 137 - camt.025_Receipt_MessageContent

Message item	Data type/code	Utilisation
Original message Document/Rct/RctDtls/OrgnlMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Document/Rct/RctDtls/OrgnlMsgId/ReqHdlg/StsCd	Max4AlphaNumericText	Values TBD
Description Document/Rct/RctDtls/OrgnlMsgId/ReqHdlg/desc	RestrictedFINXMax140Text	Descriptive text explaining the reason for rejection of the action request.

Usage case example 1: camt.025.001.04_RTGS_Receipt_Example.xml

Usage case example 2: camt.025.001.04_RTGS_ReceiptErr_Example.xml

14.3.15 ResolutionOfInvestigation (camt.029)

14.3.15.1 Overview and scope of the message

This chapter illustrates the *ResolutionOfInvestigation* message.

The *ResolutionOfInvestigation* message is sent by RTGS to a RTGS participant (or a party authorised by them). It is used to inform of the status of a previously requested payment order or payment cancellation.

The *ResolutionOfInvestigation* message only concerns the cancellation of one payment order.

Within RTGS, the *ResolutionOfInvestigation* message has the following usages:

- successful cancellation of a payment instruction (FIToFIPaymentCancellationRequest (camt.056) [▸ 355])
- unsuccessful cancellation of a payment instruction (FIToFIPaymentCancellationRequest (camt.056) [▸ 355])
- notification of forwarding a cancellation of a payment instruction (FIToFIPaymentCancellationRequest (camt.056) [▸ 355])

These usages are described below, in the chapter “The message in business context”.

The *ResolutionOfInvestigation* message is sent in response to a FIToFIPaymentCancellationRequest (camt.056) [▸ 355] message.

14.3.15.2 Schema

Outline of the schema.

The *ResolutionOfInvestigation* message is composed of the following message building blocks:

Assignment

Identifies the assignment of an investigation case from an assigner to an assignee. The assigner must be the sender of this message and the assignee must be the receiver.

Status

Indicates the status of the investigation/cancellation.

Cancellation Details

Specifies the details of the underlying transactions being cancelled.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.029.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ResolutionOfInvestigation* response message.

14.3.15.3 The message in business context

Message usage: Successful cancellation of a payment instruction

Specific message content

A *ResolutionOfInvestigation* provides a confirmation status for the cancellation as well as some information from the transaction which was cancelled.

Table 138 - camt.029_ResolutionOfInvestigationSuccessfulPaymentCancel_MessageContent

Message item	Data type/code	Utilisation
Identification Document/RsltnOfInvstgtn/Assgnmt/ID	CSLD_RestrictedFINXMax35Text	Unique identifier for the cancellation message being responded.
Sender of the cancellation Document/RsltnOfInvstgtn/Assgnmt/Assgnr/Agt/FinInstnId/BICFI	CSLD_BIC11Text	Sending party BIC which sent the payment cancellation.
Receiver of the cancellation Document/RsltnOfInvstgtn/Assgnmt/Assgne/Agt/FinInstnId/BICFI	CSLD_BIC11Text	Receiving party BIC which received the payment cancellation.
Message Date Time Document/RsltnOfInvstgtn/Assgnmt/CreDtTm	ISODateTime	Date and Time when the cancellation message was created.
Status Document/RsltnOfInvstgtn/Sts/Conf	CSLD_CancellationStatusCode	“CNCL” (Cancelled as per request)
Status ID Document/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/CxlStsId	CSLD_RestrictedFINXMax35Text	Unique ID for this cancellation request status.

Message item	Data type/code	Utilisation
Original message ID Document/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/OrgnlGrpInf/OrgnlMsgId	CSLD_RestrictedFINXMax35Text	Message ID of the original payment instruction message.
Original message name Document/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/OrgnlGrpInf/OrgnlMsgNmId	CSLD_XMLMessageNamePattern	Name of message used for the original payment instruction message.
Original message date/time Document/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/OrgnlGrpInf/OrgnlCreDtTm	ISODateTime	Date and time of the original payment instruction message.
Payment ID Document/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/OrgnlInstrId	Max35Text	Unique ID of original payment instruction.
Payment end to end ID Document/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/OrgnlEndToEnd	Max35Text	Unique end-to-end ID of original payment instruction.
Payment transaction ID Document/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/OrgnlTxId	Max35Text	Unique ID of original payment instruction, as set by the original instructing agent.
Payment clearing system ID Document/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/OrgnlClrSysRef	Max35Text	Unique ID of original payment instruction, as set by the original clearing system agent.

Message item	Data type/code	Utilisation
Cancellation originator Docu-ment/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/CxlStsRsnInf/Orgtr	PartyIdentification	Originator of the payment cancellation.
Cancellation reason code Docu-ment/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/CxlStsRsnInf/Rsn/Cd	ExternalCancellationReason1Code	Reason for the payment cancellation.
Additional information Docu-ment/RsltnOfInvstgtn/CxlDtls/TxInfAndSts/CxlStsRsnInf/AddtlInf	Max105Text	Returned payment reference.

Usage case example: camt.029.001.08_RTGS_ResolutionOfInvestigationSuccessfulPaymentCancel_Example.xml

Message usage: Unsuccessful cancellation of a payment instruction

Specific message requirements

A *ResolutionOfInvestigation* contains information on the transaction which was requested for cancellation; as well as the status of the cancellation which may also contain the reason why cancellation has not happened.

Table 139 - camt.029_ResolutionOfInvestigationUnsuccessfulPaymentCancel_MessageContent

Message item	Data type/code	Utilisation
As for 'successful payment cancel above, except:		
Status Document/RsltnOfInvstgtn/Sts/Conf	CSLD_CancellationStatusCode	"RJCR" (Rejected cancellation request)

Usage case example: camt.029.001.08_RTGS_ResolutionOfInvestigationUnsuccessfulPaymentCancel_Example.xml

Message usage: Notification of forwarding a cancellation of a payment instruction

Specific message requirements

A *ResolutionOfInvestigation* contains information on the transaction which was requested for cancellation; as well as the status of the cancellation which may also contain the reason why cancellation has not happened.

Table 140 - camt.029_ResolutionOfInvestigationForwardedPaymentCancel_MessageContent

Message item	Data type/code	Utilisation
As for 'successful payment cancel above, except:		
Status Document/RsltnOfInvstgtn/Sts/Conf	CSLD_CancellationStatusCode	"PTNA" (Passed to next agent)

Usage case example: camt.029.001.08_RTGS_ResolutionOfInvestigationForwardedPaymentCancel_Example.xml

14.3.16 GetReservation (camt.046)

14.3.16.1 Overview and scope of the message

This chapter illustrates the *GetReservation*message.

The *GetReservation* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to request details of one or more reservation facilities set by the RTGS participant (or on their behalf by an authorised party).

The *GetReservation* message can be used to request reservation information based on several criteria.

Within RTGS, the *GetReservation* message has the following usages:

- query reservations

This usage is described below, in the chapter "The message in business context".

In response to the *GetReservation* message, a ReturnReservation (camt.047) [327] message containing the requested information is returned.

14.3.16.2 Schema

Outline of the schema.

The *GetReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

ReservationQueryDefinition

Definition of the reservation query.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.046.001.06_RTGS

Business rules applicable to the schema

For business rules applicable to *GetReservation* please refer to the business rules table below.

14.3.16.3 The message in business context

Usage case: Query reservations

In this usage case, the message identifies the RTGS dedicated cash account owner for which the reservations are being queried.

Specific message requirements

All content must comply with the business rules for the message.

Table 141 - camt.046_GetReservation_MessageRequirements

Message item	Data type/code	Utilisation
Query name Document/GetRsvatn/ RsvatnQryDef/ RsvatnCrit/QryNm	Max35Text	A name for recalling previous search criteria.
New query name Document/GetRsvatn/ RsvatnQryDef/ RsvatnCrit/Newcrit/NewQryNm	Max35Text	A name for saving the criteria used in this search.
New query name Document/GetRsvatn/ RsvatnQryDef/ RsvatnCrit/Newcrit/SchCrit/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Identification of the party owning the RTGS dedicated cash account.

Usage case example: camt.046.001.04_RTGS_GetReservation_Example.xml

14.3.17 ReturnReservation (camt.047)

14.3.17.1 Overview and scope of the message

This chapter illustrates the *ReturnReservation* message.

The *ReturnReservation* message is sent by RTGS to a RTGS participant (or a party authorised by them). It is used to respond to a reservation query.

The *ReturnReservation* message provides details of one or more reservation facilities set by the RTGS participant (or on their behalf by and authorised party).

Within RTGS, the *ReturnTransaction* message has the following usages:

- response to query reservations (GetReservation (camt.046) [325])

This usage is described below, in the chapter “The message in business context”.

The *ReturnReservation* message is sent in response to a GetReservation (camt.046) [325] message which requested the information.

14.3.17.2 Schema

Outline of the schema.

The *ReturnReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the message and the original business query identification.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about limit message, or an error indication. It includes sections such as limit type, the credit consumer identifier, the currency code, the limit amount, the date from which the credit limit is valid.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.047.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *ReturnReservation* response message.

14.3.17.3 The message in business context

Usage case: Response to query reservations

In this usage case, the message informs the sender of a GetReservation (camt.046) [▶ 325] of the reservations made against RTGS dedicated cash accounts for the specified account owner.

Specific message content

Table 142 - camt.047_ReturnReservation_MessageContent

Message item	Data type/code	Utilisation
Current reservation - type Docu-ment/RtrRsvatn/RptOrErr/BizRpt/CurRsvatn/RsvatnId/Tp/Cd	ReservationType2Code	Current reservation – reservation type
Current reservation – account owner Docu-ment/RtrRsvatn/RptOrErr/BizRpt/CurRsvatn/RsvatnId/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Current reservation -RTGS dedicated cash account owner
Current reservation – amount Docu-ment/RtrRsvatn/RptOrErr/BizRpt/CurRsvatn/RsvatnOrErr/Rsvatn/Amt/AmtWthCcy	ActiveCurrencyAndAmount	Current reservation – amount & cur-rency
Current reservation – status Docu-ment/RtrRsvatn/RptOrErr/BizRpt/CurRsvatn/RsvatnOrErr/Rsvatn/Sts/Cd	ReservationStatusCode	Current reservation – status
Default reservation - identification Docu-ment/RtrRsvatn/RptOrErr/BizRpt/DfltRsvatn/RsvatnId/Tp/Cd	Max35Text	Default reservation – reservation identi-fier

Message item	Data type/code	Utilisation
Default reservation - type Docu-ment/RtrRsvatn/RptOrErr/BizRpt/DfltRsvatn/RsvatnId/Tp/Cd	ReservationTypeCode	Default reservation – reservation type
Default reservation – account owner Docu-ment/RtrRsvatn/RptOrErr/BizRpt/DfltRsvatn/RsvatnId/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Default reservation – RTGS dedicated cash account owner
Default reservation – amount Docu-ment/RtrRsvatn/RptOrErr/BizRpt/DfltRsvatn/RsvatnOrErr/Rsvatn/Amt/AmtWthtCcy	ImpliedCurrencyAndAmount	Default reservation – amount & curren-cy

Usage case example 1: camt.047.001.05_RTGS_ReservationResponse_Example.xml

When RTGS needs to report an error processing the request the following fields are used.

Table 143 - camt.047_ReturnReservation_ErrorContent

Message item	Data type/code	Utilisation
Error code Docu-ment/RtrRsvatn/RptOrErr/OprlErr/Err/Prtry/Cd	ErrorHandlingCode	RTGS code for the problem being in-formed.
Error description Docu-ment/RtrRsvatn/RptOrErr/OprlErr/Desc	Max140Text	Description of the problem being in-formed.

Usage case example 2: camt.047.001.05_RTGS_ReservationResponseErr_Example.xml

14.3.18 ModifyReservation (camt.048)

14.3.18.1 Overview and scope of the message

This chapter illustrates the *ModifyReservation* message.

The *ModifyReservation* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to request modifications to the details of one particular reservation set by the RTGS participant (or on their behalf by an authorised party).

The *ModifyReservation* message contains the new value that the RTGS participant wants to be applied to the reservation facility identified in the message.

Within RTGS, the *ModifyReservation* message has the following usages:

- liquidity reservation (create)
- liquidity reservation (amend)
- standing order for reservation (create)
- standing order for reservation (amend)

These usages are described below, in the chapter “The message in business context”.

In response to the *ModifyReservation* message, a Receipt (camt.025) [▶ 318] is sent, indicating the success or rejection/failure of the modification.

To further verify the outcome of the request, the RTGS participant may submit a GetReservation (camt.046) [▶ 325] message with the appropriate search criteria.

14.3.18.2 Schema

Outline of the schema.

The *ModifyReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReservationIdentification

Identification of the reservation (current or default).

NewReservationValueSet

New reservation values.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.048.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *ModifyReservation* please refer to the business rules table below.

14.3.18.3 The message in business context

Usage case: Liquidity reservation (create)

In this usage case, the message defines a required liquidity reservation.

Specific message requirements

All content must comply with the business rules for the message.

Table 144 - camt.048_ModifyReservationCreateLiquidityReservation_MessageRequirements

Message item	Data type/code	Utilisation
Current reservation type Document/ModfyRsvatn/RsvatnId/Cur/Tp/Cd	ReservationTypeCode	Current reservation identification – reservation type
Current reservation account owner Document/ModfyRsvatn/RsvatnId/Cur/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Current reservation identification – RTGS dedicated cash account owner
Default reservation type Document/ModfyRsvatn/RsvatnId/Dflt/Tp/Cd	ReservationTypeCode	Default reservation – reservation type
Default reservation account owner Document/ModfyRsvatn/RsvatnId/Dflt/AcctOwnr/FinInstnId/BICFI	BICFIIdentifier	Default reservation – RTGS dedicated cash account owner

Message item	Data type/code	Utilisation
Default reservation account owner branch Docu-ment/ModfyRsvatn/RsvatnId/Dflt/AcctOwnr/BrnchId/ID	Max35Text	Default reservation – RTGS dedicated cash account owner branch ID
Default reservation account owner branch Docu-ment/ModfyRsvatn/RsvatnId/Dflt/AcctOwnr/BrnchId/Nm	Max140Text	Default reservation – RTGS dedicated cash account owner branch name
Default reservation account owner postal address Docu-ment/ModfyRsvatn/RsvatnId/Dflt/AcctOwnr/BrnchId/Nm	PostalAddress	Default reservation – RTGS dedicated cash account owner postal address
Default reservation account Docu-ment/ModfyRsvatn/RsvatnId/Dflt/AcctId/Othr/ID	Max34Text	Default reservation – RTGS dedicated cash account number
New reservation amount Docu-ment/ModfyRsvatn/NewRsvatnValSet/Amt/AmtWthCcy	ActiveCurrencyAndAmount	New reservation amount required

Usage case example: camt.048.001.04_RTGS_ModifyReservationCreateLiquidityReservation_Example.xml

Usage case: Liquidity reservation (amend)

In this usage case, the message defines an amendment to an existing liquidity reservation.

Specific message requirements

All content must comply with the business rules for the message.

Table 145 - camt.048_ModifyReservationAmendLiquidityReservation_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.048.001.04_RTGS_ModifyReservationAmendLiquidityReservation_Example.xml

Usage case: standing order for reservation (create)

In this usage case, the message defines a required standing order for reservation.

Specific message requirements

All content must comply with the business rules for the message.

Table 146 - camt.048_ModifyReservationCreateReservationSO_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.048.001.04_RTGS_ModifyReservationCreateReservationSO_Example.xml

Usage case: Standing order for reservation (amend)

In this usage case, the message defines an amendment to an existing reservation standing order.

Specific message requirements

All content must comply with the business rules for the message.

Table 147 - camt.048_ModifyReservationAmendReservationSO_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.048.001.04_RTGS_ModifyReservationAmendReservationSO_Example.xml

14.3.19 DeleteReservation (camt.049)

14.3.19.1 Overview and scope of the message

This chapter illustrates the *DeleteReservation* message.

The *DeleteReservation* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to request the deletion of one particular reservation set by the RTGS participant (or on their behalf by an authorised party).

The *DeleteReservation* message allows for the deletion of only one reservation facility.

Within RTGS, the *DeleteReservation* message has the following usages:

- delete an immediate liquidity order

This usage is described below, in the chapter “The message in business context”.

In response to the *DeleteReservation* message, a Receipt (camt.025) [▸ 318] is sent, indicating the success or rejection/failure of the deletion.

To further verify the outcome of the request, the RTGS participant may submit a GetReservation (camt.046) [▸ 325] message with the appropriate search criteria.

14.3.19.2 Schema

Outline of the schema.

The *DeleteReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

CurrentReservation

Identifies the current reservation to delete.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.049.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *DeleteReservation* please refer to the business rules table below.

14.3.19.3 The message in business context

Usage case: Delete an immediate liquidity order

In this usage case, the message identifies an existing immediate liquidity order that is to be deleted.

Specific message requirements

All content must comply with the business rules for the message.

Table 148 - camt.049_DeleteReservation_MessageRequirements

Message item	Data type/code	Utilisation
Reservation type Document/DelRsvatn/CurRsvatn/Tp/Cd	ReservationTypeCode	Reservation type
Account owner Docu-ment/DelRsvatn/CurRsvatn/AcctOwnr/FinIntsnId/BICFI	BICFIIdentifier	Owner of the RTGS dedicated cash account

Usage case example: camt.049.001.04_RTGS_DeleteReservation_Example.xml

14.3.20 LiquidityCreditTransfer (camt.050)

14.3.20.1 Overview and scope of the message

This chapter illustrates the *LiquidityCreditTransfer* message.

The *LiquidityCreditTransfer* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS.

The *LiquidityCreditTransfer* message is used to request a transfer of funds between

- two RTGS dedicated cash accounts belonging to the RTGS participant, or
- two RTGS dedicated cash accounts within the same liquidity group of dedicated cash accounts, defined within RTGS identified via account IDs.

Within RTGS, the *LiquidityCreditTransfer* message has the following usages:

- inter-service liquidity transfer
- intra-service liquidity transfer (between two RTGS accounts of the same liquidity transfer group (incl. sub-account))
- ancillary system transaction processing

These usages are described below, in the chapter “The message in business context”.

In response to the *LiquidityCreditTransfer* message, a Receipt (camt.025) [▸ 318] message containing the requested information is returned.

14.3.20.2 Schema

Outline of the schema.

The *LiquidityCreditTransfer* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

LiquidityTransferIdentification

Used to uniquely identify the liquidity transfer.

Creditor

Owner of the account to be credited.

CreditorAccount

Account to be credited as a result of a transfer of liquidity.

TransferredAmount

Amount of money that the transaction administrator transfers from one account to another.

Debtor

Owner of the account to be debited.

DebtorAccount

Account to be debited as a result of a transfer of liquidity.

SettlementDate

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.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.050.001.04_RTGS

Business rules applicable to the schema

For business rules applicable to *LiquidityCreditTransfer* please refer to the business rules table below.

14.3.20.3 The message in business context

Usage case: Inter-service liquidity transfer

In this usage case, the message instructs the movement of liquidity from one RTGS dedicated cash account into dedicated cash account of a different service.

Specific message requirements

All content must comply with the business rules for the message.

Table 149 - camt.050_LiquidityCreditTransferInterService_MessageRequirements

Message item	Data type/code	Utilisation
Identification Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/LqdtyTrfId/InstrId	RestrictedFINXMax16Text	Unique ID set by the instructing party.
Identification Document/ LqdtyCdtTrf/LqdtyCdtTrf/LqdtyTrfId/EndToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party.

Message item	Data type/code	Utilisation
Creditor ID Document/LqdtyCdtTrf/LqdtyCdtTrf/Cdtr/FinInstnId/BICFI	CSLD_BIC11Text	Account owner of dedicated cash account to be credited.
Creditor account Document/LqdtyCdtTrf/LqdtyCdtTrf/CdtrAcct/ID/Othr/ID	RestrictedFINX2Max34Text	Dedicated cash account to be credited.
Creditor account type Document/LqdtyCdtTrf/LqdtyCdtTrf/CdtrAcct/Tp/Cd	ExternalCashAccountType1Code	Type of dedicated cash account to be credited.
Amount Document/LqdtyCdtTrf/LqdtyCdtTrf/TrfdAmt/AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred.
Debtor ID Document/LqdtyCdtTrf/LqdtyCdtTrf/Dbtr/FinInstnId/BICFI	CSLD_BIC11Text	Account owner of RTGS dedicated cash account to be debited.
Debtor account Document/LqdtyCdtTrf/LqdtyCdtTrf/DbtrAcct/ID/Othr/ID	RestrictedFINX2Max34Text	RTGS dedicated cash account to be debited.
Debtor account type Document/LqdtyCdtTrf/LqdtyCdtTrf/DbtrAcct/Tp/Cd	ExternalCashAccountTypeCode	Type of RTGS dedicated cash account to be debited.
Settlement date Document/LqdtyCdtTrf/LqdtyCdtTrf/SettlmDt	ISODate	Intended settlement date.

Usage case example: camt.050.001.04_RTGS_LiquidityCreditTransferInterService_Example.xml

Usage case: Intra-service liquidity transfer

In this usage case, the message instructs the movement of liquidity from one RTGS dedicated cash account into another RTGS dedicated cash account in the same liquidity transfer group.

Specific message requirements

All content must comply with the business rules for the message.

Table 150 - camt.050_LiquidityCreditTransferIntraService_MessageRequirements

Message item	Data type/code	Utilisation
Identification Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/LqdtyTrfId/InstrId	RestrictedFINXMax16Text	Unique ID set by the instructing party.
Identification Document/ LqdtyCdtTrf/LqdtyCdtTrf/LqdtyTrfId/EndToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party.
Creditor ID Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/Cdtr/FinInstnId/BICFI	CSLD_BIC11Text	Account owner of RTGS dedicated cash account to be credited.
Creditor account Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/CdtrAcct/ID/Othr/ID	RestrictedFINX2Max34Text	RTGS dedicated cash account / sub-account to be credited.
Creditor account type Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/CdtrAcct/Tp/Cd	ExternalCashAccountTypeCode	Type of RTGS dedicated cash account to be credited.
Amount Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/TrfdAmt/AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred.

Message item	Data type/code	Utilisation
Debtor ID Document/LqdtyCdtTrf/LqdtyCdtTrf/Dbtr/FinInstnId/BICFI	CSLD_BIC11Text	Account owner of RTGS dedicated cash account to be debited.
Debtor account Document/LqdtyCdtTrf/LqdtyCdtTrf/DbtrAcct/ID/Othr/ID	RestrictedFINX2Max34Text	RTGS dedicated cash account to be debited.
Debtor account type Document/LqdtyCdtTrf/LqdtyCdtTrf/DbtrAcct/Tp/Cd	ExternalCashAccountTypeCode	Type of RTGS dedicated cash account to be debited.
Settlement date Document/LqdtyCdtTrf/LqdtyCdtTrf/SettlmDt	ISODate	Intended settlement date.

Usage case example: camt.050.001.04_RTGS_LiquidityCreditTransferIntraService_Example.xml

Usage case: Ancillary service transaction processing

In this usage case, the message instructs the movement of liquidity from one RTGS dedicated cash account into another RTGS dedicated cash account, but is sent from an ancillary service.

Specific message requirements

All content must comply with the business rules for the message.

Table 151 - camt.050_LiquidityCreditTransferAncillaryService_MessageRequirements

Message item	Data type/code	Utilisation
Identification Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/LqdtyTrfId/InstrId	RestrictedFINXMax16Text	Unique ID set by the instructing party.
Identification Document/ LqdtyCdtTrf/LqdtyCdtTrf/LqdtyTrfId/EndToEndId	RestrictedFINXMax16Text	Unique ID set by the initiating party.
Creditor ID Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/Cdtr/FinInstnId/BICFI	CSLD_BIC11Text	Account owner of RTGS dedicated cash account to be credited.
Creditor account Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/CdtrAcct/ID/Othr/ID	RestrictedFINX2Max34Text	RTGS dedicated cash account / sub-account to be credited.
Creditor account type Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/CdtrAcct/Tp/Cd	ExternalCashAccountTypeCode	Type of RTGS dedicated cash account to be credited.
Amount Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/TrfdAmt/AmtWthCcy	CSLD_Max14_Max2DecimalAmount	Amount to be transferred.
Debtor ID Docu-ment/LqdtyCdtTrf/LqdtyCdtTrf/Dbtr/FinInstnId/BICFI	CSLD_BIC11Text	Account owner of RTGS dedicated cash account to be debited.

Message item	Data type/code	Utilisation
Debtor account Document/LqdtyCdtTrf/LqdtyCdtTrf/DbtrAcct/ID/Othr/ID	RestrictedFINX2Max34Text	RTGS dedicated cash account to be debited.
Debtor account type Document/LqdtyCdtTrf/LqdtyCdtTrf/DbtrAcct/Tp/Cd	ExternalCashAccountTypeCode	Type of RTGS dedicated cash account to be debited.
Settlement date Document/LqdtyCdtTrf/LqdtyCdtTrf/SettlmDt	ISODate	Intended settlement date.

Usage case example: camt.050.001.04_RTGS_LiquidityCreditTransferAncillaryService_Example.xml

14.3.21 BankToCustomerStatement (camt.053)

14.3.21.1 Overview and scope of the message

This chapter illustrates the *BankToCustomerStatement*message.

The *BankToCustomerStatement* message is sent by RTGS to a RTGS participant (or a party authorised by them). It is used to inform of the entries booked to an account and account balance information at a given point in time.

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 transactions that have been included in the entry.

Within RTGS, the *BankToCustomerStatement* message has the following usages:

- account statement

This usage is described below, in the chapter “The message in business context”.

The *BankToCustomerStatement* message is produced automatically at end of day.

14.3.21.2 Schema

Outline of the schema.

The *BankToCustomerStatement* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Statement

This building block is mandatory and repetitive. It contains information on booked entries and balances for a RTGS dedicated cash account.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.053.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *BankToCustomerStatement* message.

14.3.21.3 The message in business context

Usage case: Account statement

The data content covers a single RTGS dedicated cash account and shows:

- opening and closing balances
- list of executed cash entries
- entry summary information

Specific message content

Table 152 - camt.053_BankToCustomerStatement_MessageRequirements

Message item	Data type/code	Utilisation
Statement ID Document/BkToCstmrStmt/Stmt/ID	Max35Text	Statement number.
Reporting sequence Document/BkToCstmrStmt/Stmt/RptgSeq/FrSeq	Max35Text	Reporting sequence –from.
Reporting sequence Document/BkToCstmrStmt/Stmt/RptgSeq/ToSeq	Max35Text	Reporting sequence –to.
Reporting sequence Document/BkToCstmrStmt/Stmt/RptgSeq/FrToSeq/FrSeq	Max35Text	Reporting sequence –range from.
Reporting sequence Document/BkToCstmrStmt/Stmt/RptgSeq/FrToSeq/ToSeq	Max35Text	Reporting sequence –range to.
Reporting sequence Document/BkToCstmrStmt/Stmt/RptgSeq/EqSeq	Max35Text	Reporting sequence –single sequence.
Reporting sequence Document/BkToCstmrStmt/Stmt/RptgSeq/NEQSeq	Max35Text	Reporting sequence –excluding a sequence.
Creation date/time Document/BkToCstmrStmt/Stmt/CreDtTm	ISODateTime	Timestamp when the statement was created.

Message item	Data type/code	Utilisation
Account Document/BkToCstmrStmt/Stmt/Acct/ID/Othr/ID	Max34Text	RTGS dedicated cash account number.
Currency Document/BkToCstmrStmt/Stmt/Acct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS dedicated cash account.
Account owner Document/BkToCstmrStmt/Stmt/Acct/Ownr/ID/OrgId/AnyBIC	AnyBICIdentifier	Owner of the RTGS dedicated cash account.
Account owner country Document/BkToCstmrStmt/Stmt/Acct/Ownr/ID/CtryOfRes	CountryCode	Country of residence, of the owner of the RTGS dedicated cash account.
Multiple repetitions of Balance information		
Balance type Document/BkToCstmrStmt/Stmt/Bal/Tp/CdOrPrty/Cd	CSLD_BalanceTypeCode	Type of balance.
Balance amount Document/BkToCstmrStmt/Stmt/Bal/Amt/	ActiveOrHistoricCurrencyAndAmount	Amount of balance.
Balance credit/debit Document/BkToCstmrStmt/Stmt/Bal/Amt	CreditDebitCode	Credit or debit indicator for the balance amount.
Balance date Document/BkToCstmrStmt/Stmt/Bal/Dt/Dt	ISODate	Date of the balance.
Balance availability Document/BkToCstmrStmt/Stmt/Bal/Avlbty	CashAvailability	Availability of balance. Might be needed for non-EURO – TBD.

Message item	Data type/code	Utilisation
Transactions summary		
Number of all entries Document/BkToCstmrStmt/Stmt/TxsSummry/TtlNtries/NbOfNtries	Max15NumericText	Total number of entries on statement.
Sum of all entries Document/BkToCstmrStmt/Stmt/TxsSummry/TtlNtries/Sum	DecimalNumber	Total sum of all of entries on statement.
Net sum of all entries Document/BkToCstmrStmt/Stmt/TxsSummry/TtlNtries/TtlNetNtry/Amt	NonNegativeDecimalNumber	Net total sum of all of entries on statement.
Credit debit ind Document/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 Document/BkToCstmrStmt/Stmt/TxsSummry/TtlCdtNtries/Sum	DecimalNumber	Total sum of all of credit entries on statement.
Sum of all debit entries Document/BkToCstmrStmt/Stmt/TxsSummry/TtlDbtNtries/Sum	DecimalNumber	Total sum of all of debit entries on statement.
Multiple repetitions of Entry information		
Entry ID Document/BkToCstmrStmt/Stmt/Ntry/NtryRef	Max35Text	Unique reference for the entry.
Amount Document/BkToCstmrStmt/Stmt/Ntry/Amt	CSLD_Max14_Max2DecimalAmount	Entry amount.

Message item	Data type/code	Utilisation
Credit debit ind Docu-ment/BkToCstmrStmt/Stmt/Ntry/CrdDbtInd	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/BookgDt/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/BkTxCd/Prtry/Cd	CSLD_BankTransactionCode	Transaction code.
Message ID Docu-ment/BkToCstmrStmt/Stmt/Ntry/NtryDtls/TxDtls/Refs/MsgId	Max35Text	Unique message reference of the in-structing message.
Instruction ID Docu-ment/BkToCstmrStmt/Stmt/Ntry/NtryDtls/TxDtls/Refs/InstrId	Max35Text	Unique reference set by the instructing party.

Message item	Data type/code	Utilisation
End to end ID Docu-ment/BkToCstmrStmt/Stmt/Ntry/NtryDtls/TxDtls/Refs/EndToEndId	Max35Text	Additional unique reference set by the initiating party.
Transaction ID Docu-ment/BkToCstmrStmt/Stmt/Ntry/NtryDtls/TxDtls/Refs/TxID	Max35Text	Transaction ID set by the instructing agent.
Transaction ID Docu-ment/BkToCstmrStmt/Stmt/Ntry/NtryDtls/TxDtls/Amt	CSLD_Max14_Max2DecimalAmount	Entry detail amount
Transaction ID Docu-ment/BkToCstmrStmt/Stmt/Ntry/NtryDtls/TxDtls/CrdDbtInd	CreditDebitCode	Credit debit indicator of entry detail amount
Local instrument code Docu-ment/BkToCstmrStmt/Stmt/Ntry/NtryDtls/TxDtls/LclInstrm/Cd	ExternalLocalInstrumentCode	Local instrument code
Local instrument proprietary code Docu-ment/BkToCstmrStmt/Stmt/Ntry/NtryDtls/TxDtls/LclInstrm/Prtry	Max35Text	Local instrument – proprietary code

Message example: camt.053.001.07_RTGS_BankToCustomerStatement_Example.xml

14.3.22 BankToCustomerDebitCreditNotification (camt.054)

14.3.22.1 Overview and scope of the message

This chapter illustrates the *BankToCustomerDebitCreditNotification*message.

The *BankToCustomerDebitCreditNotification* message is sent by RTGS to RTGS participants (or a party authorised by them). It is used to confirm the credit or the debit of a certain amount on one of their RTGS dedicated cash accounts.

The *BankToCustomerDebitCreditNotification* message is sent by RTGS when the account-owner was not the instructor of the movement.

The *BankToCustomerDebitCreditNotification* message is only concerned with one single debit or credit movement on one single RTGS dedicated cash account.

Within RTGS, the *BankToCustomerDebitCreditNotification* message has the following usages:

- notification of credit entry

This usage is described below, in the chapter “The message in business context”.

The *BankToCustomerDebitCreditNotification* message is sent in response to a debit/credit movement activity within CLM.

14.3.22.2 Schema

Outline of the schema.

The *BankToCustomerDebitCreditNotification* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the message.

Notification

This building block is mandatory and repetitive. Each repetition notifies of a debit or credit entry for the RTGS dedicated cash account.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.054.001.07_RTGS

Business rules applicable to the schema

No business rules are applicable to a *BankToCustomerDebitCreditNotification* message.

14.3.22.3 The message in business context

Message usage: Notification of credit entry

This message usage enables RTGS to send a confirmation of credit to a RTGS participant if the respective RTGS dedicated cash account was credited.

Specific message content

In the rules and further descriptions, the confirmation contains always the exact amount and the reason for the credit. The following requirements apply:

Table 153 - camt.054_BankToCustomerDebitCreditNotificationCredit_MessageContent

Message item	Data type/code	Utilisation
Identification Document/BkToCstmrDbtCdtNtfctn/NtFctn/ID	Max35Text	Unique ID for this notification, set by RTGS.
Notification date Document/BkToCstmrDbtCdtNtfctn/NtFctn/CreDtTm	ISODateTime	Date time when the notification was created.
Account Document/BkToCstmrDbtCdtNtfctn/NtFctn/Acct/ID/Othr/ID	Max34Text	RTGS dedicated cash account ID.
Currency Document/BkToCstmrDbtCdtNtfctn/NtFctn/Acct/Ccy	ActiveOrHistoricCurrencyCode	Currency of the RTGS dedicated cash account.
Account owner Document/BkToCstmrDbtCdtNtfctn/NtFctn/Acct/Ownr/ID/OrgId/AnyBIC	CSLD_BIC11Text	Party owning the RTGS dedicated cash account.
Amount Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/Amt	CSLD_Max14_Max2DecimalAmount	Amount credited to the RTGS dedicated cash account.

Message item	Data type/code	Utilisation
Credit debit ind Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/CrdDbtInd	CreditDebitCode	“CRDT”.
Status Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/Sts/Cd	ExternalEntryStatus1Code	“BOOK”.
Booking date time Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/BookgDt/DtTm	ISODateTime	Time when the credit entry was booked.
Value date time Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/ValDt/DtTm	ISODateTime	Time when the credit entry amount became available.
Transaction code Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/BnkTxCd/prtry/Cd	CSLD_BankTransactionCode	Bank transaction code.
Entry amount type Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/AmtDtls/PrtryAmt/Tp	Max35Text	Entry amount type.
Entry amount Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/AmtDtls/PrtryAmt/Amt	ActiveOrHistoricCurrencyAndAmount	Entry amount.
Transaction msg ID Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Refs/MsgId	CSLD_RestrictedFINXMax35Text	Message ID of the underlying instruction which caused the credit entry.

Message item	Data type/code	Utilisation
Instruction ID Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Refs/InstrId	Max35Text	Identification of the underlying instruction which caused the credit entry.
End to end ID Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Refs/EndToEndId	Max35Text	End-to-end ID of the underlying instruction which caused the credit entry.
Transaction ID Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Refs/TxId	Max35Text	Transaction ID. Of the underlying instruction which caused the credit entry.
Transaction amount Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/Amt	CSLD_Max14_Max5DecimalAmount	Transaction amount.
Credit debit ind Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/CrdDbtInd	CreditDebitCode	“CRDT”. Indicator for transaction amount.
Transaction debit account - IBAN Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdPties/DbtrAcct/ID/IBAN	IBAN2007Identifier	Debtor account in the underlying transaction.
Transaction debit account - RTGS Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdPties/DbtrAcct/ID/Othr/ID	Max34Text	Debtor account in the underlying transaction.
Transaction credit account - IBAN Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdPties/CdtrAcct/ID/IBAN	IBAN2007Identifier	Creditor account in the underlying transaction.

Message item	Data type/code	Utilisation
Transaction credit account - RTGS Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdPties/CdtrAcct/ID/Othr/ID	Max34Text	Creditor account in the underlying transaction.
Transaction instructing agent Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdAgts/InstgAgt/FinInstnId/BICFI	CSLD_BIC11Text	Instructing agent BIC.
Transaction instructed agent Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdAgts/InstdAgt/FinInstnId/BICFI	CSLD_BIC11Text	Instructed agent BIC.
Transaction debtor agent Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdAgts/DbtrAgt/FinInstnId/BICFI	CSLD_BIC11Text	Debtor agent BIC.
Transaction creditor agent Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdAgts/CdtrAgt/FinInstnId/BICFI	CSLD_BIC11Text	Creditor agent BIC.
Transaction intermed-1 agent Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdAgts/IntrmyAgt1/FinInstnId/BICFI	CSLD_BIC11Text	First intermediary agent BIC.
Transaction intermed-2 agent Document/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdAgts/IntrmyAgt2/FinInstnId/BICFI	CSLD_BIC11Text	Second intermediary agent BIC.

Message item	Data type/code	Utilisation
Transaction intermed-3 agent Docu-ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/RltdAgts/IntrmyAgt3/FinInstnId/BICFI	CSLD_BIC11Text	Third intermediary agent BIC.
Local instrument code Docu-ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/LclInstrm/Cd	ExternalLocalInstrument1Code	Local instrument code.
Local instrument proprietary code Docu-ment/BkToCstmrDbtCdtNtfctn/NtFctn/Ntry/NtryDtls/TxDtls/LclInstrm/Prtry	Max35Text	Local instrument proprietary code.

Message usage example: camt.054.001.07_RTGS_BankToCustomerDebitCreditNotificationCredit_Example.xml

14.3.23 FIToFIPaymentCancellationRequest (camt.056)

14.3.23.1 Overview and scope of the message

This chapter illustrates the *FiToFiPaymentCancellationRequest* message.

The *FIToFIPaymentCancellationRequest* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to request the cancellation of an original payment order.

The *FIToFIPaymentCancellationRequest* message is used to cancel the following types of original payment order:

- CustomerCreditTransfer (pacs.008) [▸ 373]
- FinancialInstitutionDirectDebit (pacs.010) [▸ 402]
- FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 383]

The *FIToFIPaymentCancellationRequest* message concerns only one original payment order.

Within RTGS, the *FIToFIPaymentCancellationRequest* message has the following usages:

- revoke a payment

These usages are described below, in the chapter “The message in business context”.

In response to the FIToFIPaymentCancellationRequest message, a ResolutionOfInvestigation (camt.029) [▸ 320] is sent, indicating the success or rejection/failure of the cancellation.

14.3.23.2 Schema

Outline of the schema.

The *FIToFIPaymentCancellationRequest* message is composed of the following message building blocks:

Assignment

Identifies the assignment of an investigation case from an assigner to an assignee. The assigner must be the sender of this message and the assignee must be the receiver.

Underlying

Identifies the payment instruction to be cancelled.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/camt.056.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *FIToFIPaymentCancellationRequest* please refer to the business rules table below.

14.3.23.3 The message in business context

Usage case: Revoke a payment

In this usage case, the message identifies an existing payment instruction which is to be revoked from execution.

Specific message requirements

All content must comply with the business rules for the message.

Table 154 - camt.056_FiToFiPaymentCancellationRequestRevokeAPayment_MessageRequirements

Message item	Data type/code	Utilisation
Identification Docu-ment/FIToFIPmtCxlReq/Assgnmt/ID	CSLD_RestrictedFINXMax35Text	Unique identifier for this cancellation message.
Sender Docu-ment/FIToFIPmtCxlReq/Assgnmt/Assgnr/Agt/FinInstnId/BICFI	CSLD_BIC11Text	Sending party BIC.
Receiver Docu-ment/FIToFIPmtCxlReq/Assgnmt/Assgne/Agt/FinInstnId/BICFI	CSLD_BIC11Text	Receiving party BIC.
Message date time Docu-ment/FIToFIPmtCxlReq/Assgnmt/CreDtTm	ISODateTime	Date and time when the cancellation message was created.
Cancellation ID Docu-ment/FIToFIPmtCxlReq/Undrlyg/TxInf/CxlId	CSLD_RestrictedFINXMax35Text	Unique payment cancellation reference from the sender.
Original message ID Docu-ment/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlGrpInf/OrgnlMsgId	CSLD_RestrictedFINXMax35Text	Message ID of the original payment instruction message.
Original message name Docu-ment/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlGrpInf/OrgnlMsgNmId	CSLD_XMLMessageNamePattern	Name of message used for the original payment instruction message.
Original message date/time Docu-ment/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlGrpInf/OrgnlCreDtTm	ISODateTime	Date and time of the original payment instruction message.

Message item	Data type/code	Utilisation
Payment ID Document/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlInstrId	Max35Text	Unique ID of original payment instruction.
Payment end to end ID Document/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlEndToEnd	Max35Text	Unique end-to-end ID of original payment instruction.
Payment transaction ID Document/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlTxId	Max35Text	Unique ID of original payment instruction, as set by the original instructing agent.
Payment clearing system ID Document/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlClrSysRef	Max35Text	Unique ID of original payment instruction, as set by the original clearing system agent.
Payment amount Document/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlIntrBkSttlmAmt	ActiveOrHistoricCurrencyAndAmount	Payment amount of the original payment instruction
Payment date Document/FIToFIPmtCxlReq/Undrlyg/TxInf/OrgnlIntrBkSttlmDt	ISODate	Payment date of the original payment instruction
Cancellation originator Document/FIToFIPmtCxlReq/Undrlyg/TxInf/CxlRsnInf/Orgtr	PartyIdentification	Originator of the payment cancellation.
Cancellation reason code Document/FIToFIPmtCxlReq/Undrlyg/TxInf/CxlRsnInf/Rsn/Cd	ExternalCancellationReasonCode	Reason for the payment cancellation.

Usage case example: camt.056.001.07_RTGS_FiToFiPaymentCancellationRevokeAPayment_Example.xml

14.3.24 GetStandingOrder (camt.069)

14.3.24.1 Overview and scope of the message

This chapter illustrates the GetStandingOrder message.

The GetStandingOrder message is sent by an authorised actor to retrieve standing order information.

The GetStandingOrder message is replied by a camt.070 to return the retrieved standing order information or to provide detailed information in case of an error (e.g. no rows retrieved).

14.3.24.2 Schema

Outline of the schema

The GetStandingOrder message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message Identification provided by the requesting actor.

RequestType

This block is optional and can be used to specify which kind of query must be performed.

StandingOrderQueryDefinition

This block is mandatory and provides with all the search criteria that must be used to filter standing order records in the CRDM coverage. Possible criteria are account and BIC.

References/links

The schema and the related documentation in HTML/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

14.3.25 ReturnStandingOrder (camt.070)

14.3.25.1 Overview and scope of the message

This chapter illustrates the ReturnStandingOrder message.

The ReturnStandingOrder message is sent by CRDM to an authorised actor to provide with requested standing order information.

The ReturnStandingOrder message is sent as a response to a previously sent camt.069.

14.3.25.2 Schema

Outline of the schema

The ReturnStandingOrder message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message identification provided by the requesting actor as well as the original business query message identification and the request type specifying the kind of query that has been performed.

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.

References/links

The schema and the related documentation in HTML/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

14.3.26 DeleteStandingOrder (camt.071)

14.3.26.1 Overview and scope of the message

This chapter illustrates the DeleteStandingOrder message.

The DeleteStandingOrder message is sent by an actor authorised to delete standing orders for liquidity transfers.

The DeleteStandingOrder message is replied by a camt.025 to return a positive technical response to the sender of the message or to provide detailed information in case of an error.

14.3.26.2 Schema

Outline of the schema

The DeleteStandingOrder message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message identification provided by the requesting actor.

StandingOrderDetails

This block is mandatory and provides with all the key information to identify an existing standing order to be deleted. Both identification and account identification must be provided.

References/links

The schema and the related documentation in HTML/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

14.4 Headers (head)

14.4.1 BusinessApplicationHeader (head.001)

14.4.1.1 Overview and scope of the message

This chapter illustrates the *BusinessApplicationHeader* message.

For payment messages between bank A and bank B, FROM identifies bank A and TO identifies bank B. For service messages between bank A and the MI (e.g. pacs.009 connected payment, liquidity messages etc.), FROM identifies bank A and TO identifies the MI.

14.4.1.2 Schema

Outline of the schema

The BAH message is composed of the following message building blocks:

FROM

The sender that has created this message for the receiver that processes this message. FROM BIC must have exactly 11 characters.

TO

The receiver designated by the sender who ultimately processes this message. TO BIC must have exactly 11 characters.

BusinessMessageIdentifier

Identifies unambiguously the message. The BusinessMessageIdentifier has maximum 35 characters.

MessageDefinitionIdentifier

Contains the MessageIdentifier that defines the message. It must contain a MessageIdentifier published on the ISO 20022 website.

CreationDate

Date and time when this message (header) was created.

CopyDuplicate (optional)

Indicates whether the message is a copy, a duplicate or a copy of a duplicate of a previously sent ISO 20022 message.

PossibleDuplicate (optional)

Is a flag indicating if the message exchanged between sender and receiver is possibly a duplicate.

Signature (optional)

Contains the digital signature of the business entity authorised to sign this message.

Related (optional)

Specifies the BAH of the message to which this message relates. It can be used when replying to a query; it can also be used when canceling or amending.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/head.001.001.01_RTGS

Business rules applicable to the schema

For business rules applicable to *BusinessApplicationHeader* please refer to the business rules table below.

14.4.1.3 The message in business context

The BAH contains information to correctly process the message payload by means that every messages exchanged between RTGS and the participants respectively RTGS and the other services includes such an information. The relation between BAH and message payload is exactly 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)

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 directly connected RTGS to send several business messages within one file to RTGS.

Under a single *BusinessFileHeader*, every message within a file has to be an ISO 20022 Message together with its business application header (business message). A file can contain one or several business messages.

Within RTGS, the BusinessFileHeader information is used for:

- consistency and completeness checks

In response to an incoming file which fails validation, RTGS sends a ReceiptAcknowledgement (admi.007) [▶ 276] message containing information on negative validation.

Results from validation which is performed at file level, are sent by RTGS without BAH information.

14.4.2.2 Schema

Outline of the schema.

The *BusinessFileHeader* is composed of the following building blocks:

PayloadDescription

The PayloadDescription is a mandatory block and contains the following information tags:

- PayloadDetails: with PayloadIdentifier; CreationDateAndTime and PossibleDuplicateFlag
- ApplicationSpecificInformation: which contains information about the total number of instances (messages) within the file
- PayloadTypeDetails: which declares the payload content (describes the type of business document being exchanged)
- ManifestDetails: with information to each DocumentType and the number of instances (messages) for each declared type.

Payload

The payload is a mandatory block and contains the set of business messages, each built of an ISO 20022 message together with its business application header.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/head.002.001.01_RTGS

Business rules applicable to the schema

For business rules applicable to *BusinessFileHeader* please refer to the business rules table below.

14.4.2.3 The message in business context

Message example: head.002.001.01_RTGS_IncomingMessageFileWithinRTGS_Example.xml

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 RTGS to a RTGS participant (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.

The *FIToFIPaymentStatusReport* message is used as a response/update for the following business activities:

- settlement of customer payment
- settlement of direct debit
- settlement of interbank payment
- settlement of interbank payment (cover for customer payments)
- settlement of payment return
- settlement of ancillary system (former procedure 2 and 3)
- liquidity transfer from RTGS dedicated cash account to sub-account (initiated by settlement bank day and night)

Within RTGS, the *FIToFIPaymentStatusReport* message has the following usages:

- success response to a previously sent message
- rejection response to a previously sent message

These usages are described below, in the chapter “The message in business context”.

The *FIToFIPaymentStatusReport* 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. See above business actions for details.

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

Information concerning the original transactions, to which the status report message refers.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.002.001.09_RTGS

Business rules applicable to the schema

For business rules applicable to *FIToFIPaymentStatusReport* please refer to the business rules table below.

14.5.1.3 The message in business context

Usage case: Success response to a previously sent message

In this case, RTGS provides to the sender of an action message, to inform that the action request was successful.

Specific message content

Table 155 - pacs.002_FIToFIPaymentStatusReport_MessageContent

Message item	Data type/code	Utilisation
FIToFIPaymentStatusReport09 Document/FIToFIPmtStsRpt	FIToFIPaymentStatusReport09	
Identification Document/FIToFIPmtStsRpt/TxInfAndSts/StsId	Max35Text	Unique ID for the status.
Identification Document/FIToFIPmtStsRpt/TxInfAndSts/OrgnlGrpInf/OrgnlMsgId	Max35Text	Message ID of original instruction.
XML message name Document/FIToFIPmtStsRpt/TxInfAndSts/OrgnlGrpInf/OrgnlMsgNmId	RTGS_XMLMessageNamePattern	Message name of the original instruction.
Original Instruction ID Document/FIToFIPmtStsRpt/TxInfAndSts/OrgnlInstrId	Max35Text	Identification of the original instruction.
Original transaction ID Document/FIToFIPmtStsRpt/TxInfAndSts/OrgnlInstrId	Max35Text	Transaction ID of the original instruction.
Status Document/FIToFIPmtStsRpt/TxInfAndSts/TxSts	RTGS_TransactionStatusCode	“ACSC” (Accepted settlement completed).

Message item	Data type/code	Utilisation
Status reason Docu-ment/FIToFIPmtStsRpt/TxInfAndSts/StsRsnInf/Rsn/Prtry	Max35Text	Status reason.
Additional info Docu-ment/FIToFIPmtStsRpt/TxInfAndSts/StsRsnInf/Addtlinf	Max105Text	Additional information.
Clearing system ID Docu-ment/FIToFIPmtStsRpt/TxInfAndSts/ClrSysRef	Max105Text	Identification from clearing system.

Message example: pacs.002.001.09_RTGS_FIToFIPaymentStatusReportSuccessful_Example.xml

Usage case: Rejection response to a previously sent message

In this case, RTGS provides to the sender of an action message, to inform that the action request was rejected.

Specific message content

Table 156 - pacs.002_FIToFIPaymentStatusReport_MessageContent

Message item	Data type/code	Utilisation
As for 'successful action request above, except:		
Status Docu-ment/FIToFIPmtStsRpt/TxInfAndSts/TxSts	RTGS_TransactionStatusCode	"RJCT" (Rejected).

Message example: pacs.002.001.09_RTGS_FIToFIPaymentStatusReportRejected_Example.xml

14.5.2 PaymentReturn (pacs.004)

14.5.2.1 Overview and scope of the message

This chapter illustrates the *PaymentReturn* message.

The *PaymentReturn* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to undo a previously settled payment.

The *PaymentReturn* message concerns only one payment.

Within RTGS, the *PaymentReturn* message has the following usages:

- settlement of payment return

This usage is described below, in the chapter “The message in business context”.

In response to the *PaymentReturn* message, a PaymentStatusReport (pacs.002) [▶ 365] message containing the status of the payment return is returned to the sending RTGS participant.

In addition, if the payment return is successful, the *PaymentReturn* message is forwarded to the recredited RTGS participant (or a party authorised by them).

14.5.2.2 Schema

Outline of the schema.

The *PaymentReturn* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

TransactionInformation

Information concerning the original transactions, to which the return message refers.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.004.001.08_RTGS

Business rules applicable to the schema

For business rules applicable to *PaymentReturn* please refer to the business rules table below.

14.5.2.3 The message in business context

Usage case: Settlement of payment return

In this usage case, the message identifies an already settled payment, which is to be reversed and the funds re-credited to the original payer.

Specific message requirements

All content must comply with the business rules for the message.

Table 157 - pacs.004_PaymentReturn_MessageRequirements

Message item	Data type/code	Utilisation
PaymentReturnV08 Document/PmtRtr	PaymentReturnV08	
Identification Document/PmtRtr/TxInf/OrgnlGrpInf/OrgnlMsgId	Max35Text	Message ID of original instruction.
XML message name Document/PmtRtr/TxInf/OrgnlGrpInf/OrgnlMsgNmId	RTGS_XMLMessageNamePattern	Message name of the original instruction.
Original Instruction ID Document/PmtRtr/TxInf/OrgnlInstrId	Max35Text	Identification of the original instruction.
Original end-to-end ID Document/PmtRtr/TxInf/OrgnlEndToEndId	Max35Text	End-to-end identification of the original instruction.
Original i-bank settlement amount Document/PmtRtr/TxInf/OrgnlIntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Interbank settlement amount of the original instruction.

Message item	Data type/code	Utilisation
Original i-bank settlement date Document/PmtRtr/TxInf/OrgnlIntrBkSttlmDt	ISODate	InteSettlement date of the original instruction.
Returned i-bank settlement amount Document/PmtRtr/TxInf/RtrdIntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Settlement amount of the return payment.
I-bank settlement date Document/PmtRtr/TxInf/IntrBkSttlmDt	ISODate	Interbank settlement date of the return payment.
Return priority Document/PmtRtr/TxInf/SttlmPrty	Priority3Code	Priority for the return payment.
Returned instructed amount Document/PmtRtr/TxInf/RtrdInstdAmt	RTGS_Max14_Max5DecimalAmount	Gross amount of return payment.
Exchange rate Document/PmtRtr/TxInf/XchgRate	BaseOneRate	Exchange rate.
Compensation amount Document/PmtRtr/TxInf/CompstnAmt	RTGS_Max14_Max5DecimalAmount	Compensation amount.
Charge bearer Document/PmtRtr/TxInf/ChrgBr	ChargeBearerType1Code	Charge bearer for the return payment.
Charges amount Document/PmtRtr/TxInf/ChrgsInf/Amt	RTGS_Max14_Max5DecimalAmount	Charges amount.
Charges agent Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/BICFI	BICFIIdentifier	Charges agent (BIC).
Charges agent clearing system Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/ClrSysMmbId/ClrSysId/Cd	ExternalClearingSystemIdentification1Code	Charges agent (Clearing system).
Charges agent clearing system Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/ClrSysMmbId/ClrSysId/Prtry	Max35Text	Charges agent (Clearing system proprietary).

Message item	Data type/code	Utilisation
Charges agent clearing system member Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/ClrSysMmbId/MmbId	RTGS_RestrictedFINXMax28Text	Charges agent (Clearing system member).
Charges agent clearing system member name Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/ClrSysMmbId/Nm	Max140Text	Charges agent (Clearing system member name).
Charges agent postal address Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/PstlAdr	PostalAddress6	Charges agent postal address.
Charges agent other ID - identification Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/Othr/ID	Max35Text	Charges agent other identification.
Charges agent other ID – scheme code Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/Othr/SchmeNm/Cd	ExternalFinancialInstitutionIdentification1Code	Charges agent other ID - scheme code.
Charges agent other ID – scheme code Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/Othr/SchmeNm/Prtry	Max35Text	Charges agent other ID – proprietary code.
Charges agent other ID – scheme code issuer Document/PmtRtr/TxInf/ChrgsInf/Agt/FinInstnId/Othr/Issr	Max35Text	Charges agent other ID – proprietary code issuer.
Clearing system ref Document/PmtRtr/TxInf/ClrSysRef	Max35Text	Reference at the clearing system for the return payment.

Message item	Data type/code	Utilisation
Instructing agent Docu-ment/PmtRtr/TxInf/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	Party instructing the return payment.
Instructed agent Docu-ment/PmtRtr/TxInf/InstdAgt/FinInstnId/BICFI	BICFIIdentifier	Party receiving the return payment instruction.
Return chain	various	Full analysis to follow.
Reason party Document/PmtRtr/TxInf/RtrRsnInf/Orgtr	PartyIdentification125	Party issuing the return payment in-struction.
Reason code Docu-ment/PmtRtr/TxInf/RtrRsnInf/Rsn/Cd	ExternalReturnReason1Code	Reason code for return payment.
Reason additional Docu-ment/PmtRtr/TxInf/RtrRsnInf/AddtlInf	Max105Text	Additional reason information.

Usage case example: pacs.004.001.08_RTGS_Paymentreturn_Example.xml

14.5.3 CustomerCreditTransfer (pacs.008)

14.5.3.1 Overview and scope of the message

This chapter illustrates the *FIToFICustomerCreditTransfer* message.

This message type is used in RTGS component to execute a payment order if the debtor or the creditor or both are non-financial institutions.

The payment order message can be sent by a

- direct RTGS participant
- BIC of the multi-addressee access and
- central bank as a direct participant

Credited and debited RTGS dedicated cash accounts must be denominated in the same currency.

Within RTGS, the *FIToFICustomerCreditTransfer* message has the following usages:

- settlement of a customer payment

This usage is described below, in the chapter “The message in business context”.

In response to the *FIToFICustomerCreditTransfer* message, a PaymentStatusReport (pacs.002) [▸ 365] is returned.

14.5.3.2 Schema

Outline of the schema.

The *FIToFICustomerCreditTransfer* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

CreditTransferTransactionInformation

Set of elements providing information specific to the individual credit transfer(s).

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.008.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *FIToFICustomerCreditTransfer* please refer to the business rules table below.

14.5.3.3 The message in business context

Usage case: Settlement of customer payment

In this usage case, the message describes a payment to or from involving a non-financial institution.

Specific message requirements

All content must comply with the Business Rules for the message.

Table 158 - pacs.008_FIToFICustomercreditTransfer_MessageRequirements

Message item	Data type/code	Utilisation
FIToFICustomerCreditTransferV07 Document/FIToFICstmrCdtTrf	FIToFICustomerCreditTransferV07	
Payment ID Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PmtId/InstrId	Max35Text	Payment ID – instruction ID
Payment ID Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PmtId/EndToEndId	Max35Text	Payment ID – end to end ID
Payment ID Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PmtId/TxId	Max35Text	Payment ID – transaction ID
Payment ID Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PmtId/ClrSysRef	Max35Text	Payment ID – RTGS system reference
Payment type priority Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PmtTpInf/InstrPrty	Priority2Code	Payment type priority
Payment service level code Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PmtTpInf/SvcLvl/Cd	ExternalServiceLevel1Code	Service level code
Payment local instrument code Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PmtTpInf/LclInstrm/Cd	ExternalLocalInstrument1Code	Local instrument code

Message item	Data type/code	Utilisation
Payment category purpose code Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PmtTpInf/CtgyPurp/Cd	ExternalCategoryPurpose1Code	Category purpose code
Amount Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/IntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Payment amount
Date Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/IntrBkSttlmDt	ISODate	Payment date
Settlement priority Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/SttlmPrty	Priority3Code	Settlement priority
Settlement date time Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/SttlmTmIndctn/CrdDtTm	ISODateTime	Settlement date time
Settlement time request - till Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/SttlmTmReq/TillTm	ISOTime	Settlement time request – till time
Settlement time request - from Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/SttlmTmReq/FrTm	ISOTime	Settlement time request – from time
Settlement time request - reject Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/SttlmTmReq/RjctTm	ISOTime	Settlement time request – reject time

Message item	Data type/code	Utilisation
Instructed amount Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/InstdAmt	ActiveOrHistoricCurrencyAndAmount	Instructed amount
Exchange rate Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/XchgRate	BaseOneRate	Exchange rate
Charge bearer Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgBr	ChargeBearerType1Code	Charge bearer
Charge amount Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Amt	RTGS_Max14_Max5DecimalAmount	Charge amount
Charge agent Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/BICFI	BICFIIdentifier	Charge agent - BIC
Charge agent Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/ClrSysMmbId/ClrSysId/Cd	ExternalClearingSystemIdentifica-tion1Code	Charge agent – clearing system code
Charge agent Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/ClrSysMmbId/MmbId	RTGS_RestrictedFINXMax28Text	Charge agent – clearing system mem-ber ID
Charge agent Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/Nm	Max140Text	Charge agent – name

Message item	Data type/code	Utilisation
Charge agent Document/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/PstlAdr	PostalAddress6	Charge agent – postal address
Charge agent Document/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/Othr/ID	Max35Text	Charge agent – other ID
Charge agent Document/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/Othr/SchmeNm/Cd	ExternalFinancialInstitutionIdentification1Code	Charge agent - other ID, scheme code.
Charge agent Document/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/Othr/SchmeNm/Prtry	Max35Text	Charge agent - other ID, proprietary code.
Charge agent Document/FIToFICstmrCdtTrf/CdtTrfTxInf/ChrgsInf/Agt/FinInstnId/Othr/Issr	Max35Text	Charge agent - other ID, proprietary code issuer.
Prev instructing agt1 Document/FIToFICstmrCdtTrf/CdtTrfTxInf/PrvsInstgAgt1	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 1
Prev instructing agt1 account Document/FIToFICstmrCdtTrf/CdtTrfTxInf/PrvsInstgAgt1Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Document/FIToFICstmrCdtTrf/CdtTrfTxInf/PrvsInstgAgt2	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 2

Message item	Data type/code	Utilisation
Prev instructing agt2 account Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PrvsInstgAgt2Acct	CashAccount24	Previous instructing agent 2 account
Prev instructing agt3 Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PrvsInstgAgt3	BranchAndFinancialInstitutionIdentifi-cation5	Previous instructing agent 3
Prev instructing agt3 account Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/PrvsInstgAgt3Acct	CashAccount24	Previous instructing agent 3 account
Instructing agt Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	Instructing agent
Instructed agt Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/InstdAgt/FinInstnId/BICFI	BICFIIdentifier	Instructed agent
Intermed. Agt1 Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/IntrmyAgt1	BranchAndFinancialInstitutionIdentifi-cation5	Intermediary agent 1
Intermed. Agt1 account Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/IntrmyAgt1Acct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/IntrmyAgt2	BranchAndFinancialInstitutionIdentifi-cation5	Intermediary agent 2

Message item	Data type/code	Utilisation
Intermed. Agt2 account Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/IntrmyAgt2Acct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/IntrmyAgt3	BranchAndFinancialInstitutionIdentifi-cation5	Intermediary agent 3
Intermed. Agt3 account Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/IntrmyAgt3Acct	CashAccount24	Intermediary agent 3 account
Ultimate debtor Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/UltmtDbtr	PartyIdentification125	Ultimate debtor
Initiating party Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/InitgPty	PartyIdentification125	Initiating party
Debtor Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/Dbtr	PartyIdentification125	Debtor
Debtor account Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Docu-ment/FIToFICstmrCdtTrf/CdtTrfTxInf/DbtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Debtor agent

Message item	Data type/code	Utilisation
Debtor agt account Document/FIToFICstmrCdtTrf/CdtTrfTxInf/DbtrAgtAcct	CashAccount24	Debtor agent account
Creditor agt Document/FIToFICstmrCdtTrf/CdtTrfTxInf/CdtrAgt	BranchAndFinancialInstitutionIdentification5	Creditor agent
Creditor agt account Document/FIToFICstmrCdtTrf/CdtTrfTxInf/CdtrAgtAcct	CashAccount24	Creditor agent account
Creditor Document/FIToFICstmrCdtTrf/CdtTrfTxInf/Cdtr	PartyIdentification125	Creditor
Creditor account Document/FIToFICstmrCdtTrf/CdtTrfTxInf/CdtrAcct	CashAccount24	Creditor account
Ultimate creditor Document/FIToFICstmrCdtTrf/CdtTrfTxInf/UltmtCdtr	PartyIdentification125	Ultimate creditor
Instruction for creditor agt Document/FIToFICstmrCdtTrf/CdtTrfTxInf/InstrForCdtrAgt/Cd	Instruction3Code	Instruction for creditor agent - code
Instruction for creditor agt Document/FIToFICstmrCdtTrf/CdtTrfTxInf/InstrForCdtrAgt/InstrInf	Max30Text	Instruction for creditor agent - information

Message item	Data type/code	Utilisation
Purpose code Document/FIToFICstmrCdtTrf/CdtTrfTxInf/Purp/Cd	ExternalPurpose1Code	Purpose code
Reg reporting – Db/Crd indicator Document/FIToFICstmrCdtTrf/CdtTrfTxInf/RgltryRptg/DbtCdtRptgInd	RegulatoryReportingType1Code	Regulatory reporting – db/crd indicator
Reg reporting – authority Document/FIToFICstmrCdtTrf/CdtTrfTxInf/RgltryRptg/Authrty/Nm	Max140Text	Regulatory reporting – authority
Reg reporting – authority country Document/FIToFICstmrCdtTrf/CdtTrfTxInf/RgltryRptg/Authrty/Ctry	CountryCode	Regulatory reporting – authority country
Reg reporting – details Document/FIToFICstmrCdtTrf/CdtTrfTxInf/RgltryRptg/Dtls	StructuredRegulatoryReporting3	Regulatory reporting – details
Related remittance information Document/FIToFICstmrCdtTrf/CdtTrfTxInf/RltdRmtInf	RemittanceLocation4	Related remittance information
Remittance information Document/FIToFICstmrCdtTrf/CdtTrfTxInf/RmtInf	RemittanceInformation15	Remittance information

Usage case example: pacs.008.001.07_RTGS_FIToFICustomerCreditTransfer_Example.xml

14.5.4 FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009)

14.5.4.1 Overview and scope of the message

This chapter illustrates the *FinancialInstitutionCreditTransfer* message.

This message type can be used for different RTGS services:

- high value payments
- ancillary systems transactions

High value payments can be sent by a

- direct RTGS participant
- BIC of the multi-addressee access and
- central bank as a direct participant or on behalf of a RTGS participant (mandated payments)

Transactions to serve ancillary system settlement procedures can be sent by

- ancillary systems for procedures real-time settlement and bilateral settlement sent in batch
- settlement banks to provide liquidity for settlement on dedicated ancillary systems liquidity account (technical account for real-time settlement procedure)

Credited and debited RTGS dedicated cash accounts must be denominated in the same currency.

Within RTGS, the *FinancialInstitutionCreditTransfer* message has the following usages:

- settlement of an interbank payment
- settlement of an interbank payment (customer cover)
- settlement of ancillary system movement
- liquidity transfer from RTGS dedicated cash account to sub-account

These usages are described below, in the chapter “The message in business context”.

In response to the *FinancialInstitutionCreditTransfer* message, a PaymentStatusReport (pacs.002) is returned.

14.5.4.2 Schema

Outline of the schema.

The *FinancialInstitutionCreditTransfer* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

CreditTransferTransactionInformation

Set of elements providing information specific to the individual credit transfer(s).

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.009.001.07_RTGS

Business rules applicable to the schema

For business rules applicable to *FinancialInstitutionCreditTransfer* please refer to the business rules table below.

14.5.4.3 The message in business context

Usage case: Settlement of an interbank payment

In this usage case, the message describes a payment between two financial institutions.

Specific message requirements

All content must comply with the business rules for the message.

Table 159 - pacs.009_FinancialInstutionCreditTransferIBPayment_MessageRequirements

Message item	Data type/code	Utilisation
FinancialInstitutionCreditTransferV07 Document/FICdtTrf	FinancialInstitutionCreditTransferV07	
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/InstrId	Max35Text	Payment ID – instruction ID
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/EndToEndId	Max35Text	Payment ID – end to end ID

Message item	Data type/code	Utilisation
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/TxId	Max35Text	Payment ID – transaction ID
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/ClrSysRef	Max35Text	Payment ID – RTGS system reference
Payment service level code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/SvcLvl/Cd	ExternalServiceLevel1Code	Service level code
Payment local instrument code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/LclInstrm/Cd	ExternalLocalInstrument1Code	Local instrument code
Payment category purpose code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/CtgyPurp/Cd	ExternalCategoryPurpose1Code	Category purpose code
Amount Document/FICdtTrf/CdtTrfTxInf/IntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Payment amount
Date Document/FICdtTrf/CdtTrfTxInf/IntrBkSttlmDt	ISODate	Payment date
Settlement priority Document/FICdtTrf/CdtTrfTxInf/SttlmPrty	Priority3Code	Settlement priority
Settlement date time Document/FICdtTrf/CdtTrfTxInf/SttlmTmIndctn/CrdDtTm	ISODateTime	Settlement date time

Message item	Data type/code	Utilisation
Settlement time request - CLS Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/CLSTm	ISOTime	Settlement time request – CLS time
Settlement time request - till Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/TillTm	ISOTime	Settlement time request – till time
Settlement time request - from Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/FrTm	ISOTime	Settlement time request – from time
Settlement time request - reject Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/RjctTm	ISOTime	Settlement time request – reject time
Prev instructing agt1 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt1	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 1
Prev instructing agt1 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt1Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt2	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 2
Prev instructing agt2 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt2Acct	CashAccount24	Previous instructing agent 2 account

Message item	Data type/code	Utilisation
Prev instructing agt3 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt3	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 3
Prev instructing agt3 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt3Acct	CashAccount24	Previous instructing agent 3 account
Instructing agt Document/FICdtTrf/CdtTrfTxInf/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	Instructing agent
Instructed agt Document/FICdtTrf/CdtTrfTxInf/InstdAgt/FinInstnId/BICFI	BICFIIdentifier	Instructed agent
Intermed. Agt1 Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt1	BranchAndFinancialInstitutionIdentification5	Intermediary agent 1
Intermed. Agt1 account Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt1Acct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt2	BranchAndFinancialInstitutionIdentification5	Intermediary agent 2
Intermed. Agt2 account Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt2Acct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt3	BranchAndFinancialInstitutionIdentification5	Intermediary agent 3

Message item	Data type/code	Utilisation
Intermed. Agt3 account Docu-ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3Acct	CashAccount24	Intermediary agent 3 account
Debtor Document/FICdtTrf/CdtTrfTxInf/Dbtr	PartyIdentification125	Debtor
Debtor account Docu-ment/FICdtTrf/CdtTrfTxInf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Document/FICdtTrf/CdtTrfTxInf/DbtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Debtor agent
Debtor agt account Docu-ment/FICdtTrf/CdtTrfTxInf/DbtrAgtAcct	CashAccount24	Debtor agent account
Creditor agt Document/FICdtTrf/CdtTrfTxInf/CdtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Creditor agent
Creditor agt account Docu-ment/FICdtTrf/CdtTrfTxInf/CdtrAgtAcct	CashAccount24	Creditor agent account
Creditor Document/FICdtTrf/CdtTrfTxInf/Cdtr	PartyIdentification125	Creditor
Creditor account Docu-ment/FICdtTrf/CdtTrfTxInf/CdtrAcct	CashAccount24	Creditor account
Remittance information Document/FICdtTrf/CdtTrfTxInf/RmtInf	RemittanceInformation2	Remittance information

Usage case example: pacs.009.001.07_RTGS_FinancialInstitutionCreditTransferIBPayment_Example.xml

Usage case: Settlement of an interbank customer cover payment

In this usage case, the message describes a payment between two financial institutions, performed as a cover for an underlying customer payment.

Specific message requirements

All content must comply with the business rules for the message.

Table 160 - pacs.009_FinancialInstutionCreditTransferIBPayment_MessageRequirements

Message item	Data type/code	Utilisation
Institutional information as above PLUS underlying customer payment information below		
Ultimate debtor Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/UltmtDbtr	PartyIdentification125	Ultimate debtor
Initiating party Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/InitgPty	PartyIdentification125	Initiating party
Debtor Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/Dbtr	PartyIdentification125	Debtor
Debtor account Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/DbtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Debtor agent

Message item	Data type/code	Utilisation
Debtor agt account Document/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/DbtrAgtAcct	CashAccount24	Debtor agent account
Prev instructing agt1 Document/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/PrvsInstgAgt1	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 1
Prev instructing agt1 account Document/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/PrvsInstgAgt1Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Document/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/PrvsInstgAgt2	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 2
Prev instructing agt2 account Document/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/PrvsInstgAgt2Acct	CashAccount24	Previous instructing agent 2 account
Prev instructing agt3 Document/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/PrvsInstgAgt3	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 3
Prev instructing agt3 account Document/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/PrvsInstgAgt3Acct	CashAccount24	Previous instructing agent 3 account
Intermed. Agt1 Document/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/IntrmyAgt1	BranchAndFinancialInstitutionIdentification5	Intermediary agent 1

Message item	Data type/code	Utilisation
Intermed. Agt1 account Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/IntrmyAgt1Acct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/IntrmyAgt2	BranchAndFinancialInstitutionIdentifi-cation5	Intermediary agent 2
Intermed. Agt2 account Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/IntrmyAgt2Acct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/IntrmyAgt3	BranchAndFinancialInstitutionIdentifi-cation5	Intermediary agent 3
Intermed. Agt3 account Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/IntrmyAgt3Acct	CashAccount24	Intermediary agent 3 account
Creditor agt Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/CdtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Creditor agent
Creditor agt account Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/CdtrAgtAcct	CashAccount24	Creditor agent account
Creditor Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/Cdtr	PartyIdentification125	Creditor

Message item	Data type/code	Utilisation
Creditor account Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/CdtrAcct	CashAccount24	Creditor account
Ultimate creditor Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/UltmtCdtr	PartyIdentification125	Ultimate creditor
Instruction for creditor agent Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/InstrForCdtrAgt	InstructionForCreditorAgent1	Instruction for creditor agent
Remittance information Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/RmtInf	RemittanceInformation15	Remittance information
Instructed amount Docu-ment/FICdtTrf/CdtTrfTxInf/UndrlygCstmrCdtTrf/InstdAmt	RTGS_Max14_Max5DecimalAmount	Instructed amount

Usage case example: pacs.009.001.07_RTGS_FinancialInstitutionCreditTransferIBCustomerCover_Example.xml

Usage case: Settlement of ancillary system movement

In this usage case, the message describes a payment movement instructed by an ancillary system.

Specific message requirements

All content must comply with the business rules for the message.

Table 161 - pacs.009_FinancialInstutionCreditTransferASMovement_MessageRequirements

Message item	Data type/code	Utilisation
FinancialInstitutionCreditTransferV07 Document/FICdtTrf	FinancialInstitutionCreditTransferV07	
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/InstrId	Max35Text	Payment ID – instruction ID
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/EndToEndId	Max35Text	Payment ID – end to end ID
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/TxId	Max35Text	Payment ID – transaction ID
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/ClrSysRef	Max35Text	Payment ID – RTGS system reference
Payment service level code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/SvcLvl/Cd	ExternalServiceLevel1Code	Service level code
Payment local instrument code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/LclInstrm/Cd	ExternalLocalInstrument1Code	Local instrument code
Payment category purpose code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/CtgyPurp/Cd	ExternalCategoryPurpose1Code	Category purpose code
Amount Document/FICdtTrf/CdtTrfTxInf/IntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Payment amount

Message item	Data type/code	Utilisation
Date Document/FICdtTrf/CdtTrfTxInf/IntrBkSttlmDt	ISODate	Payment date
Settlement priority Document/FICdtTrf/CdtTrfTxInf/SttlmPrty	Priority3Code	Settlement priority
Settlement date time Document/FICdtTrf/CdtTrfTxInf/SttlmTmIndctn/CrdDtTm	ISODateTime	Settlement date time
Settlement time request - CLS Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/CLSTm	ISOTime	Settlement time request – CLS time
Settlement time request - till Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/TillTm	ISOTime	Settlement time request – till time
Settlement time request - from Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/FrTm	ISOTime	Settlement time request – from time
Settlement time request - reject Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/RjctTm	ISOTime	Settlement time request – reject time
Prev instructing agt1 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt1	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 1

Message item	Data type/code	Utilisation
Prev instructing agt1 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt1Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt2	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 2
Prev instructing agt2 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt2Acct	CashAccount24	Previous instructing agent 2 account
Prev instructing agt3 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt3	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 3
Prev instructing agt3 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt3Acct	CashAccount24	Previous instructing agent 3 account
Instructing agt Document/FICdtTrf/CdtTrfTxInf/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	Instructing agent
Instructed agt Document/FICdtTrf/CdtTrfTxInf/InstdAgt/FinInstnId/BICFI	BICFIIdentifier	Instructed agent
Intermed. Agt1 Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt1	BranchAndFinancialInstitutionIdentification5	Intermediary agent 1

Message item	Data type/code	Utilisation
Intermed. Agt1 account Docu-ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt1Acct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Docu-ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt2	BranchAndFinancialInstitutionIdentifi-cation5	Intermediary agent 2
Intermed. Agt2 account Docu-ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt2Acct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Docu-ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3	BranchAndFinancialInstitutionIdentifi-cation5	Intermediary agent 3
Intermed. Agt3 account Docu-ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3Acct	CashAccount24	Intermediary agent 3 account
Debtor Document/FICdtTrf/CdtTrfTxInf/Dbtr	PartyIdentification125	Debtor
Debtor account Docu-ment/FICdtTrf/CdtTrfTxInf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Document/FICdtTrf/CdtTrfTxInf/DbtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Debtor agent
Debtor agt account Docu-ment/FICdtTrf/CdtTrfTxInf/DbtrAgtAcct	CashAccount24	Debtor agent account
Creditor agt Document/FICdtTrf/CdtTrfTxInf/CdtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Creditor agent
Creditor agt account Docu-ment/FICdtTrf/CdtTrfTxInf/CdtrAgtAcct	CashAccount24	Creditor agent account

Message item	Data type/code	Utilisation
Creditor Document/FICdtTrf/CdtTrfTxInf/Cdtr	PartyIdentification125	Creditor
Creditor account Docu-ment/FICdtTrf/CdtTrfTxInf/CdtrAcct	CashAccount24	Creditor account
Remittance information Document/FICdtTrf/CdtTrfTxInf/RmtInf	RemittanceInformation2	Remittance information

Usage case example: pacs.009.001.07_RTGS_FinancialInstitutionCreditTransferASMovement_Example.xml

Usage case: Liquidity transfer to sub-account

In this usage case, the message describes a payment movement instructed by an ancillary system.

Specific message requirements

All content must comply with the business rules for the message.

Table 162 - pacs.009_FinancialInstutionCreditTransferLTtoSubaccount_MessageRequirements

Message item	Data type/code	Utilisation
FinancialInstitutionCreditTransferV07 Document/FICdtTrf	FinancialInstitutionCreditTransferV07	
Payment ID Docu-ment/FICdtTrf/CdtTrfTxInf/PmtId/InstrId	Max35Text	Payment ID – instruction ID
Payment ID Docu-ment/FICdtTrf/CdtTrfTxInf/PmtId/EndToEndId	Max35Text	Payment ID – end to end ID
Payment ID Docu-ment/FICdtTrf/CdtTrfTxInf/PmtId/TxId	Max35Text	Payment ID – transaction ID

Message item	Data type/code	Utilisation
Payment ID Document/FICdtTrf/CdtTrfTxInf/PmtId/ClrSysRef	Max35Text	Payment ID – RTGS system reference
Payment service level code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/SvcLvl/Cd	ExternalServiceLevel1Code	Service level code
Payment local instrument code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/LclInstrm/Cd	ExternalLocalInstrument1Code	Local instrument code
Payment category purpose code Document/FICdtTrf/CdtTrfTxInf/PmtTpInf/CtgyPurp/Cd	ExternalCategoryPurpose1Code	Category purpose code
Amount Document/FICdtTrf/CdtTrfTxInf/IntrBkSttlmAmt	RTGS_Max14_Max2DecimalAmount	Payment amount
Date Document/FICdtTrf/CdtTrfTxInf/IntrBkSttlmDt	ISODate	Payment date
Settlement priority Document/FICdtTrf/CdtTrfTxInf/SttlmPrty	Priority3Code	Settlement priority
Settlement date time Document/FICdtTrf/CdtTrfTxInf/SttlmTmIndctn/CrdDtTm	ISODateTime	Settlement date time

Message item	Data type/code	Utilisation
Settlement time request - CLS Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/CLSTm	ISOTime	Settlement time request – CLS time
Settlement time request - till Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/TillTm	ISOTime	Settlement time request – till time
Settlement time request - from Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/FrTm	ISOTime	Settlement time request – from time
Settlement time request - reject Document/FICdtTrf/CdtTrfTxInf/SttlmTmReq/RjctTm	ISOTime	Settlement time request – reject time
Prev instructing agt1 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt1	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 1
Prev instructing agt1 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt1Acct	CashAccount24	Previous instructing agent 1 account
Prev instructing agt2 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt2	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 2
Prev instructing agt2 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt2Acct	CashAccount24	Previous instructing agent 2 account

Message item	Data type/code	Utilisation
Prev instructing agt3 Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt3	BranchAndFinancialInstitutionIdentification5	Previous instructing agent 3
Prev instructing agt3 account Document/FICdtTrf/CdtTrfTxInf/PrvsInstgAgt3Acct	CashAccount24	Previous instructing agent 3 account
Instructing agt Document/FICdtTrf/CdtTrfTxInf/InstgAgt/FinInstnId/BICFI	BICFIIdentifier	Instructing agent
Instructed agt Document/FICdtTrf/CdtTrfTxInf/InstdAgt/FinInstnId/BICFI	BICFIIdentifier	Instructed agent
Intermed. Agt1 Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt1	BranchAndFinancialInstitutionIdentification5	Intermediary agent 1
Intermed. Agt1 account Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt1Acct	CashAccount24	Intermediary agent 1 account
Intermed. Agt2 Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt2	BranchAndFinancialInstitutionIdentification5	Intermediary agent 2
Intermed. Agt2 account Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt2Acct	CashAccount24	Intermediary agent 2 account
Intermed. Agt3 Document/FICdtTrf/CdtTrfTxInf/IntrmyAgt3	BranchAndFinancialInstitutionIdentification5	Intermediary agent 3

Message item	Data type/code	Utilisation
Intermed. Agt3 account Docu-ment/FICdtTrf/CdtTrfTxInf/IntrmyAgt3Acct	CashAccount24	Intermediary agent 3 account
Debtor Document/FICdtTrf/CdtTrfTxInf/Dbtr	PartyIdentification125	Debtor
Debtor account Docu-ment/FICdtTrf/CdtTrfTxInf/DbtrAcct	CashAccount24	Debtor account
Debtor agt Document/FICdtTrf/CdtTrfTxInf/DbtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Debtor agent
Debtor agt account Docu-ment/FICdtTrf/CdtTrfTxInf/DbtrAgtAcct	CashAccount24	Debtor agent account
Creditor agt Document/FICdtTrf/CdtTrfTxInf/CdtrAgt	BranchAndFinancialInstitutionIdentifi-cation5	Creditor agent
Creditor agt account Docu-ment/FICdtTrf/CdtTrfTxInf/CdtrAgtAcct	CashAccount24	Creditor agent account
Creditor Document/FICdtTrf/CdtTrfTxInf/Cdtr	PartyIdentification125	Creditor
Creditor account Docu-ment/FICdtTrf/CdtTrfTxInf/CdtrAcct	CashAccount24	Creditor account
Remittance information Document/FICdtTrf/CdtTrfTxInf/RmtInf	RemittanceInformation2	Remittance information

Usage case example: pacs.009.001.07_RTGS_FinancialInstitutionCreditTransferLTtoSubaccount_Example.xml

14.5.5 FinancialInstitutionDirectDebit (pacs.010)

14.5.5.1 Overview and scope of the message

This chapter illustrates the *FinancialInstitutionDirectDebit* message.

The *FinancialInstitutionDirectDebit* message is sent by a RTGS participant (or on their behalf by an authorised party) to RTGS. It is used to move an amount from the RTGS dedicated cash account of another RTGS participant, to a dedicated cash account of the sending RTGS participant.

The *FinancialInstitutionDirectDebit* message concerns only one direct debit movement.

Within RTGS, the *FinancialInstitutionDirectDebit* message has the following usages:

- settlement of a direct debit
- generation of a direct debit
- settlement of an ancillary system

These usages are described below, in the chapter “The message in business context”.

In response to the *FinancialInstitutionDirectDebit* message, a PaymentStatusReport (pacs.002) message containing the status of the movement is returned to the sending RTGS participant.

In addition, if the movement is successful, the *FinancialInstitutionDirectDebit* message is forwarded to the debited RTGS participant (or a party authorised by them).

14.5.5.2 Schema

Outline of the schema.

The *FinancialInstitutionDirectDebit* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

CreditInstruction

Characteristics that apply to the credit side of the payment transaction(s) included in the message.

References/links

The RTGS-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/RTGS/pacs.010.001.02_RTGS

Business rules applicable to the schema

For business rules applicable to *FinancialInstitutionDirectDebit* please refer to the business rules table below.

14.5.5.3 The message in business context

Usage case: Settlement of a direct debit

In this usage case, the message describes a payment to be received under a previously created direct debit arrangement.

Specific message requirements

All content must comply with the business rules for the message.

Table 163 - pacs.010_FinancialInstutionDirectDebitSettlementofDD_MessageRequirements

Message item	Data type/code	Utilisation
FinancialInstitutionDirectDebitV02 Document/FIDrctDbt	FinancialInstitutionDirectDebitV02	
Identification Document/FIDrctDbt/CdtInstr/CdtId	Max35Text	Identification of credit movement
Instructing agt Document/FIDrctDbt/CdtInstr/InstgAgt/FinInstgId/BICFI	BICFIIdentifier	Instructing agent BIC
Instructed agt Document/FIDrctDbt/CdtInstr/InstdAgt/FinInstgId/BICFI	BICFIIdentifier	Instructed agent BIC

Message item	Data type/code	Utilisation
Creditor agt Docu-ment/FIDrctDbt/CdtInstr/CdtrAgt/FinInstgId/BICFI	BICFIIdentifier	Creditor agent BIC
Creditor agt acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAgtAcct/ID/IBAN	IBAN2007Identifier	Creditor agent account - IBAN
Creditor agt acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAgtAcct/ID/Othr/ID	Max34Text	Creditor agent account – other ID
Creditor agt acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAgtAcct/Tp/Cd	ExternalCashAccountType1Code	Creditor agent account – type
Creditor agt acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAgtAcct/Ccy	ActiveOrHistoricCurrencyCode	Creditor agent account – currency
Creditor Docu-ment/FIDrctDbt/CdtInstr/Cdtr/FinInstgId/BICFI	BICFIIdentifier	Creditor BIC
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/ID/IBAN	IBAN2007Identifier	Creditor account - IBAN
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/ID/Othr/ID	Max34Text	Creditor account – other ID

Message item	Data type/code	Utilisation
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/Tp/Cd	ExternalCashAccountType1Code	Creditor account – type
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/Ccy	ActiveOrHistoricCurrencyCode	Creditor account – currency
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/InstrId	Max35Text	DD ID – Instruction ID
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/EndToEndId	Max35Text	DD ID – End-to-end ID
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/TxId	Max35Text	DD ID – Transaction ID
DD Local instrument Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtTpInf/LclInstrm/Cd	ExternalLocalInstrument1Code	DD – Local instrument code
DD settlement amount Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/IntrBkSttlmAmt	ActiveCurrencyAndAmount	DD – Amount
DD settlement date Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/IntrBkSttlmDt	ISODate	DD – Date

Message item	Data type/code	Utilisation
DD settlement priority Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SettlmPrty	Priority3Code	DD – Settlement priority
DD Settlement time request - till Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/TillTm	ISOTime	Settlement time request – till time
DD Settlement time request - from Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/FrTm	ISOTime	Settlement time request – from time
DD Settlement time request - reject Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/RjctTm	ISOTime	Settlement time request – reject time
DD Debtor Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/Dbtr/FinInstgId/BICFI	BICFIIdentifier	Debtor BIC
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/ID/IBAN	IBAN2007Identifier	Debtor account - IBAN
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/ID/Othr/ID	Max34Text	Debtor account – other ID
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/Tp/Cd	ExternalCashAccountType1Code	Debtor account – type

Message item	Data type/code	Utilisation
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/Ccy	ActiveOrHistoricCurrencyCode	Debtor account – currency
DD Debtor agt Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgt/FinInstgId/BICFI	BICFIIdentifier	Debtor agent BIC
DD Debtor agt acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/ID/IBAN	IBAN2007Identifier	Debtor agent account - IBAN
DD Debtor agt acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/ID/Othr/ID	Max34Text	Debtor agent account – other ID
DD Debtor agt acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/Tp/Cd	ExternalCashAccountType1Code	Debtor agent account – type
DD Remittance info Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/RmtInf	ActiveOrHistoricCurrencyCode	DD Remittance information

Usage case example: pacs.010.001.02_RTGS_FinancialInstitutionDirectDebitSettlementofDD_Example.xml

Usage case: Generation of a direct debit

In this usage case, the message describes a direct debit arrangement.

Specific message requirements

All content must comply with the business rules for the message.

Table 164 - pacs.010_FinancialInstutionDirectDebitGenerationofDD_MessageRequirements

Message item	Data type/code	Utilisation
FinancialInstitutionDirectDebitV02 Document/FIDrctDbt	FinancialInstitutionDirectDebitV02	
Identification Document/FIDrctDbt/CdtInstr/CdtId	Max35Text	Identification of credit movement
Instructing agt Document/FIDrctDbt/CdtInstr/InstgAgt/FinInstgId/BICFI	BICFIIdentifier	Instructing agent BIC
Instructed agt Document/FIDrctDbt/CdtInstr/InstdAgt/FinInstgId/BICFI	BICFIIdentifier	Instructed agent BIC
Creditor agt Document/FIDrctDbt/CdtInstr/CdtrAgt/FinInstgId/BICFI	BICFIIdentifier	Creditor agent BIC
Creditor agt acct Document/FIDrctDbt/CdtInstr/CdtrAgtAcct/ID/IBAN	IBAN2007Identifier	Creditor agent account - IBAN
Creditor agt acct Document/FIDrctDbt/CdtInstr/CdtrAgtAcct/ID/Othr/ID	Max34Text	Creditor agent account – other ID
Creditor agt acct Document/FIDrctDbt/CdtInstr/CdtrAgtAcct/Tp/Cd	ExternalCashAccountType1Code	Creditor agent account – type
Creditor agt acct Document/FIDrctDbt/CdtInstr/CdtrAgtAcct/Ccy	ActiveOrHistoricCurrencyCode	Creditor agent account – currency

Message item	Data type/code	Utilisation
Creditor Docu-ment/FIDrctDbt/CdtInstr/Cdtr/FinInstgId/BICFI	BICFIIdentifier	Creditor BIC
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/ID/IBAN	IBAN2007Identifier	Creditor account - IBAN
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/ID/Othr/ID	Max34Text	Creditor account – other ID
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/Tp/Cd	ExternalCashAccountType1Code	Creditor account – type
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/Ccy	ActiveOrHistoricCurrencyCode	Creditor account – currency
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/InstrId	Max35Text	DD ID – Instruction ID
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/EndToEndId	Max35Text	DD ID – End-to-end ID
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/TxId	Max35Text	DD ID – Transaction ID

Message item	Data type/code	Utilisation
DD Local instrument Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtTpInf/LclInstrm/Cd	ExternalLocalInstrument1Code	DD – Local instrument code
DD settlement amount Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/IntrBkSttlmAmt	ActiveCurrencyAndAmount	DD – Amount
DD settlement date Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/IntrBkSttlmDt	ISODate	DD – Date
DD settlement priority Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SettlmPrty	Priority3Code	DD – Settlement priority
DD Settlement time request - till Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/TillTm	ISOTime	Settlement time request – till time
DD Settlement time request - from Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/FrTm	ISOTime	Settlement time request – from time
DD Settlement time request - reject Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/RjctTm	ISOTime	Settlement time request – reject time
DD Debtor Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/Dbtr/FinInstgId/BICFI	BICFIIdentifier	Debtor BIC

Message item	Data type/code	Utilisation
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/ID/IBAN	IBAN2007Identifier	Debtor account - IBAN
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/ID/Othr/ID	Max34Text	Debtor account – other ID
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/Tp/Cd	ExternalCashAccountType1Code	Debtor account – type
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/Ccy	ActiveOrHistoricCurrencyCode	Debtor account – currency
DD Debtor agt Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgt/FinInstgId/BICFI	BICFIIdentifier	Debtor agent BIC
DD Debtor agt acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/ID/IBAN	IBAN2007Identifier	Debtor agent account - IBAN

Message item	Data type/code	Utilisation
DD Debtor agt acct Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/ID/Othr/ID	Max34Text	Debtor agent account – other ID
DD Debtor agt acct Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/Tp/Cd	ExternalCashAccountType1Code	Debtor agent account – type
DD Remittance info Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/RmtInf	ActiveOrHistoricCurrencyCode	DD Remittance information

Usage case example: pacs.010.001.02_RTGS_FinancialInstitutionDirectDebitGenerationofDD_Example.xml

Usage case: Settlement of an ancillary system

In this usage case, the message describes a direct debit payment instructed from an ancillary system.

Specific message requirements

All content must comply with the business rules for the message.

Table 165 - pacs.010_FinancialInstutionDirectDebitSettlementofAS_MessageRequirements

Message item	Data type/code	Utilisation
FinancialInstitutionDirectDebitV02 Document/FIDrctDbt	FinancialInstitutionDirectDebitV02	
Identification Document/FIDrctDbt/CdtInstr/CdtId	Max35Text	Identification of credit movement
Instructing agt Docu-ment/FIDrctDbt/CdtInstr/InstgAgt/FinInstgId/BICFI	BICFIIdentifier	Instructing agent BIC

Message item	Data type/code	Utilisation
Instructed agt Docu-ment/FIDrctDbt/CdtInstr/InstdAgt/FinInstgId/BICFI	BICFIIdentifier	Instructed agent BIC
Creditor agt Docu-ment/FIDrctDbt/CdtInstr/CdtrAgt/FinInstgId/BICFI	BICFIIdentifier	Creditor agent BIC
Creditor agt acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAgtAcct/ID/IBAN	IBAN2007Identifier	Creditor agent account - IBAN
Creditor agt acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAgtAcct/ID/Othr/ID	Max34Text	Creditor agent account – other ID
Creditor agt acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAgtAcct/Tp/Cd	ExternalCashAccountType1Code	Creditor agent account – type
Creditor agt acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAgtAcct/Ccy	ActiveOrHistoricCurrencyCode	Creditor agent account – currency
Creditor Docu-ment/FIDrctDbt/CdtInstr/Cdtr/FinInstgId/BICFI	BICFIIdentifier	Creditor BIC
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/ID/IBAN	IBAN2007Identifier	Creditor account - IBAN

Message item	Data type/code	Utilisation
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/ID/Othr/ID	Max34Text	Creditor account – other ID
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/Tp/Cd	ExternalCashAccountType1Code	Creditor account – type
Creditor acct Docu-ment/FIDrctDbt/CdtInstr/CdtrAcct/Ccy	ActiveOrHistoricCurrencyCode	Creditor account – currency
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/InstrId	Max35Text	DD ID – Instruction ID
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/EndToEndId	Max35Text	DD ID – End-to-End ID
DD identification Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtId/TxId	Max35Text	DD ID – Transaction ID
DD Local instrument Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/PmtTpInf/LclInstrm/Cd	ExternalLocalInstrument1Code	DD – Local instrument code
DD settlement amount Docu-ment/FIDrctDbt/CdtInstr/DrctDbtTxInf/IntrBkSttlmAmt	ActiveCurrencyAndAmount	DD – Amount

Message item	Data type/code	Utilisation
DD settlement date Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/IntrBkSttlmDt	ISODate	DD – Date
DD settlement priority Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SettlmPrty	Priority3Code	DD – settlement priority
DD Settlement time request - till Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/TillTm	ISOTime	Settlement time request – till time
DD Settlement time request - from Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/FrTm	ISOTime	Settlement time request – from time
DD Settlement time request - reject Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/SttlmTmReq/RjctTm	ISOTime	Settlement time request – reject time
DD Debtor Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/Dbtr/FinInstgId/BICFI	BICFIIdentifier	Debtor BIC
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/ID/IBAN	IBAN2007Identifier	Debtor account - IBAN
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/ID/Othr/ID	Max34Text	Debtor account – other ID

Message item	Data type/code	Utilisation
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/Tp/Cd	ExternalCashAccountType1Code	Debtor account – type
DD Debtor acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAcct/Ccy	ActiveOrHistoricCurrencyCode	Debtor account – currency
DD Debtor agt Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgt/FinInstgId/BICFI	BICFIIdentifier	Debtor agent BIC
DD Debtor agt acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/ID/IBAN	IBAN2007Identifier	Debtor agent account - IBAN
DD Debtor agt acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/ID/Othr/ID	Max34Text	Debtor agent account – other ID
DD Debtor agt acct Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/DbtrAgtAcct/Tp/Cd	ExternalCashAccountType1Code	Debtor agent account – type
DD Remittance info Document/FIDrctDbt/CdtInstr/DrctDbtTxInf/RmtInf	ActiveOrHistoricCurrencyCode	DD Remittance information

Usage case example: pacs.010.001.02_RTGS_FinancialInstitutionDirectDebitSettlementofAS_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 reda.017 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.

References/links

The schema and the related documentation in HTML/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

14.6.2 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 reda.015 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.

References/links

The schema and the related documentation in HTML/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

IV Appendixes

15 Index and digital signature (partially completed)

15.1 Index of business rules and error codes (partially completed)

BR name	Description	Inbound message	Reply message	Reason code	Error text
	A message structure is valid according to the schema defined for a message.	any message	admi.007		The message is not valid. //Dynamic error including element name.//
	A message type has to be supported	head.001	admi.007		The received single message type is not known in RTGS
	The system user sending the inbound A2A communication has to be known.	head.002	admi.007		The System User is not known
	The system user sending the inbound A2A communication must not be locked.	head.002	admi.007		The System User is blocked due to lockout.
	The file header tags which are necessary for authentication processing must be valid according to the XML schema.	head.002	admi.007		At least one BFH tag for authentication is not valid. //Dynamic error including element name.//
	The system user sending the inbound A2A communication has to be known.	head.001	admi.007		The System User sending the inbound A2A communication has to be known.

BR name	Description	Inbound message	Reply message	Reason code	Error text
	The digital signature has to be valid for the business sending user.	head.002	admi.007		Digital signature is not valid for the Business Sending User.
	The business sending user has to be known.	head.002	admi.007		The Business Sending User is not known.
	The business application header tags which are necessary for authentication processing must be valid according to the XML schema.	head.001	admi.007		At least one BAH tag for authentication is not valid. //Dynamic error including element name.//
	The digital signature has to be valid for the business sending user.	head.001	admi.007		Digital signature is not valid for the Business Sending User.
	Business sending user is allowed to send for the system user reference.	head.002	admi.007		Business Sending User is not allowed to send for the system user reference.
	The business sending user has to be known.	head.001	admi.007		The Business Sending User is not known.

BR name	Description	Inbound message	Reply message	Reason code	Error text
	The file must be valid according to the XML schema.	head.002	admi.007		The file is not valid. //Dynamic error including element name.//
	The file must contain at least one individual message.	head.002	admi.007		The file could not be processed, because it does not contain any individual message.
	The file must not have been already processed. The file was sent twice or the reference number of the file was used before by the same business sending party.	head.002	admi.007		The file was sent twice or the reference number of the file was used before. It could only be processed once.

15.2 Index of status value and codes (to be completed in iteration 4)

15.3 Index of instruction references (to be completed in iteration 4)

15.4 Digital signature on business layer (to be completed in iteration 4)

16 Glossary (partially completed)

Term	Definition	Acronym	Source [21]
4CB	The Deutsche Bundesbank (BBk), the Banco de España (BdE), the Banque de France (BdF) and the Banca d'Italia (BdI), collectively, in their capacity as the national central banks responsible for building, maintaining and running T2 services 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
A2A	See application-to-application.		CLM/RTGS
Account holder	Individual or entity which is authorised to perform transactions on behalf of an account.		CLM/RTGS
Account monitoring group	An optional clustering of accounts for liquidity purposes, e.g. consolidated monitoring, liquidity management.		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
Actor	User defined a dedicated distinguished name which is allowed to interact with one or more T2 service.		CLM/RTGS

21 In general definitions are taken from the sources provided in this column. Where small variations to the original text have been made, the source is marked with "*".

Term	Definition	Acronym	Source [21]
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.	ALG	RTGS
Ancillary system	A system in which payments or securities are exchanged and/or cleared, while the ensuing monetary obligations are settled in another system, typically an RTGS system. Ancillary systems are e.g.: –retail payment systems (RS) –large value payment systems (LVPS) –foreign exchange (FX) systems –money market systems –clearing houses –securities settlement systems (SSS).	AS	RTGS
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.	A2A	CLM/RTGS
AS	See ancillary system.		RTGS
Authentication	The methods used to verify the origin of a message or to verify the identity of a participant connected to a system and to confirm that a message has not been modified or replaced in transit.		CLM/RTGS
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

Term	Definition	Acronym	Source [21]
Backup payments	In the event of a technical system outage a direct participant (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
BAH	See business application header.		CLM/RTGS
Beneficiary	A recipient of funds (payee) or securities. Depending on the context, a beneficiary can be a direct participant in CLM or RTGS and/or a final recipient.		CLM/RTGS
BIC	See business identifier code.		CLM/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
BIC directory	Directory published by SWIFT, part. 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
Bilateral/multilateral limit	Instruction of a direct participant to define a bilateral/multilateral limit of a fixed amount within RTGS on a regular basis (time or event triggered).		RTGS
Broadcast	Information message simultaneously available to all or a selected group of participants in CLM and RTGS.		CLM/RTGS

Term	Definition	Acronym	Source [21]
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.	BAH	CLM/RTGS
Business day	The business day comprises and defines the opening times and specific phases per T2 service.		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.	BIC	CLM/RTGS
Bypass FIFO	See FIFO by-passing		
CB	See central bank		CLM/RTGS
CBO	See central bank operations		CLM/RTGS
CBS	See central bank services		CLM/RTGS
CCP	See central counterparty		RTGS
Ceiling	An upper threshold of an account balance defined by the participant for initiating a service-specific action.		CLM/RTGS
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.	CB	CLM/RTGS
Central bank operations	Operations initiated by central banks in their capacity as central bank of issue, e.g. monetary policy operations, changes of the credit line.	CBO	CLM/RTGS
Central bank services	Business service managing central bank operations and meeting monetary policy requirements.	CBS	CLM/RTGS

Term	Definition	Acronym	Source [21]
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.	CCP	RTGS
Central European Time	Standard time which is one hour ahead of Coordinated Universal Time (UTC).	CET	CLM/RTGS
Central liquidity management	Business component of the T2 services managing and showing funds and credit lines for direct participants and central bank operations. In addition, central component for funding the RTGS component and T2S and TIPS.	CLM	CLM/RTGS
CET	See Central European Time.		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 may act as central counterparty for the participants and therefore assume significant financial risks.		CLM/RTGS
CLM	see central liquidity management.		CLM/RTGS
CLS	See continuous linked settlement.		CLM/RTGS
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

Term	Definition	Acronym	Source [21]
Common reference data management	Business component managing centrally the reference data for all TARGET services and common components.	CRDM	CLM/RTGS
Connected payment	Payments by a central bank or an ancillary system 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 services	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.	CLS	RTGS
COT	See Cut-off time		CLM/RTGS
CRDM	See common reference data management.		CLM/RTGS
Credit line	A commitment to grant intra-day credit on demand based on collateral provided to a central bank.		CLM/RTGS
Credit transfer	A payment order or, sometimes, a sequence of payment orders 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

Term	Definition	Acronym	Source [21]
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.	COT	CLM/RTGS
Data warehouse	Centralised collection of data from operational business applications in which data are aggregated and optimised for reporting and analysis.	DWH	CLM/RTGS
DCA	See Dedicated Cash Account.		CLM/RTGS
Dedicated cash account	An account dedicated for a single service/component e.g. TIPS, T2S, RTGS.	DCA	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 remunerated at a prespecified interest rate.		CLM
Direct participant	A participant in T2 services that directly carries out transactions with other participants in the system. He can perform all activities allowed in the T2 services without intermediary.		CLM/RTGS
Distinguished name	A name given to a person, company or element within a computer system or network that uniquely identifies it from everything else	DN	CLM/RTGS
DN	See distinguished name.		CLM/RTGS
DWH	See data warehouse.		CLM/RTGS
EBA	Euro Banking Association.		RTGS
ECB	European Central Bank.		CLM/RTGS
End of Day	End of the defined business day.	EOD	CLM/RTGS

Term	Definition	Acronym	Source [21]
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
EOD	See end of day.		CLM/RTGS
ESMIG	See Eurosystem single market infrastructure gateway.		CLM/RTGS
Eurosystem single market infrastructure gateway	The common entry point for all interaction with the T2 services, 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.	ESMIG	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.	XML	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. It may include zero, one or many single individual messages.		CLM/RTGS
Final (finality)	Irrevocable, unconditional, or not annullable.		CLM/RTGS
Final settlement	Settlement which is irrevocable, unconditional, or not annullable.		CLM/RTGS

Term	Definition	Acronym	Source [21]
Floor	A lower threshold of an account balance defined by the participant for initiating a component-specific action.		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.	GUI	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 predefined 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 system or a central bank on its behalf.		RTGS
Guarantor	Owner of the guarantee funds account.		RTGS
GUI	See graphical user interface.		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 T2 services.		CLM/RTGS

Term	Definition	Acronym	Source [21]
Indirect participant	A participant in a funds or securities transfer system with tiering arrangement using a direct participant as intermediary to perform some of the activities allowed in the T2 services.		CLM/RTGS
Instructions	Orders for a service/component e.g. payment order, liquidity transfer order, tasks.		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	International Organization for Standardization		CLM/RTGS
ISO 20022	The international standard for financial services messaging, maintained by the International Organization for Standardization (ISO).		CLM/RTGS
Limit	Amount for normal payments a direct participant 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 direct participant it is possible to establish standing orders or current bilateral (respectively multilateral) limits.		RTGS
Liquidity transfer	Liquidity transfer is a payment, the main purpose of which is to transfer liquidity between different accounts of the same participant.	LT	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

Term	Definition	Acronym	Source [21]
Liquidity transfer order	Liquidity transfer order is a payment order, the main purpose of which is to transfer liquidity between different accounts of the same participant. A liquidity transfer order is still not settled.	LTO	CLM/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.	MCA	CLM
Mandated payment	Payment initiated by an entity that is not party to the transaction (typically by a central bank or an ancillary system in connection with ancillary system settlement) on behalf of another entity. A central bank sends a credit transfer (with specific message structure) on behalf of the failed direct participant (only in case of contingency situations).	MP	CLM/RTGS
Market infrastructure services	Services offered – in this case - by the Eurosystem in the area of payments and security settlements.	MIS	CLM/RTGS
MCA	See main cash account.		CLM/RTGS
Messages	Messages part of the interactive communication between user and service/component		CLM/RTGS
MIS	See market infrastructure services.		CLM/RTGS
Network service provider	A business entity, licensed – in this case - by the Eurosystem, that provides the technical infrastructure, including hardware and software, to establish a secure and encrypted network connection permitting the exchange of information between actors.	NSP	CLM/RTGS
Night-time settlement	Procedure during night time phase.	NTS	RTGS
NSP	See network service provider.		CLM/RTGS

Term	Definition	Acronym	Source [21]
NTS	See night-time settlement.		/RTGS
Offsetting	Offsetting in the RTGS aims at increasing the capacity of the system to settle payments, thereby reducing queues, speeding up the settlement process and reducing the need of intraday liquidity. A bilateral or multilateral offsetting mechanism considers payments in the queues of participants and tries to settle them simultaneously on a gross basis within one legal and logical second.		RTGS
Opening day	See TARGET opening day.		CLM/RTGS
Overnight credit	See marginal lending facility.		CLM
Overnight deposit	See deposit facility.		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
Participant	An entity which is identified/recognized by the system, is bound by rules of the system and is allowed to send and capable to receive transfer orders, either directly (as a direct participant) or indirectly (as an indirect participant).		CLM/RTGS
Party	Any entity defined in the system. This includes: central banks, payment banks, participants and ancillary systems.		CLM/RTGS
Payee	See beneficiary.		CLM/RTGS
Payer	The party to a payment transaction which issues the payment order 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

Term	Definition	Acronym	Source [21]
Payment order	An order or message to initiate a payment .The order may relate either to a credit transfer or to a direct debit.		CLM/RTGS
Payment system	A payment system consists of a set of instruments, banking procedures and, typically, interbank 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).	PvP	RTGS
Priority	In general, payments are settled immediately, if sufficient liquidity is available on the cash account of the participant. Considering their urgency, they can be submitted and managed by the sender using different priorities.		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 component, 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.		CLM/RTGS
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

Term	Definition	Acronym	Source [21]
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
PvP	See payment versus payment.		RTGS
Query	A function to retrieve information from a database using selection criteria to fulfil ad hoc information demands.		CLM/RTGS
Real-time	At the same time as events 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.	RTGS	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
Receiver	A participant who obtains the respective message.		CLM/RTGS
Report	An event-driven or time-triggered publishing of information in a defined standard format to specific recipients.		CLM/RTGS
RTGS	See real-time gross settlement.		RTGS
RTGS component	Comprises the processing of high-value payments and ancillary system settlement.		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.	SSS	RTGS

Term	Definition	Acronym	Source [21]
Sender	A participant who initiates the process by sending the respective message to the T2 services.		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.	SLM	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 bank	Direct participant who pertains to one or more ancillary systems. 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
SoD	Start of day.		CLM/RTGS
SSS	See securities settlement system.		RTGS
Standing liquidity transfer order	Instruction of a direct participant to transfer regularly a fixed amount (time or event triggered) between different accounts (main cash accounts, dedicated cash accounts) of the same participant.		CLM/RTGS

Term	Definition	Acronym	Source [21]
STP	See straight-through processing.		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.	STP	RTGS
Sub account	Specific account, belonging to an RTGS dedicated cash account, holding dedicated liquidity to allow the settlement of an ancillary system using the interfaced settlement procedure.		RTGS
Systemic risk	The risk that the inability of one institution to meet its obligations when due causes other institutions to be unable to meet their obligations when due. Such failure may cause significant liquidity or credit problems and, as a result, could threaten the stability of or confidence in markets.		CLM/RTGS
T2	See TARGET2.		CLM/RTGS
T2S	See TARGET2-Securities.		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).	T2	CLM/RTGS
T2 services	T2 services contains of CLM and RTGS.		CLM/RTGS

Term	Definition	Acronym	Source [21]
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.	T2S	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 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
Tasks	Tasks are activities in a task queue which need to be performed.		CLM/RTGS
Technical account	Account used in the context of ancillary systems settlement as intermediary account for the collection of debits/credits.		CLM/RTGS
TIPS	Target Instant Payment Settlement: real-time settlement system for retail payments settled in central bank money.		CLM/RTGS
Transaction Reference Number	A unique reference number used to identify each payment instruction.	TRN	CLM/RTGS
Transit account	(Technical) account maintained in CLM and RTGS component, T2S and TIPS for the processing of liquidity transfers.		CLM/RTGS

Term	Definition	Acronym	Source [21]
TRN	See transaction reference number.		CLM/RTGS
U2A	See user-to-application.		CLM/RTGS
UI	See user interaction.		CLM/RTGS
URD	See user requirements document.		CLM/RTGS
User	A user can be an individual person or technical user interacting with the T2 services.		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.	UI	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.	URD	CLM/RTGS
User-to-application	A connectivity mode for the exchange of information through a graphical user interface.	U2A	CLM/RTGS
UTC	See coordinated universal time.		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
XML	See Extensible Mark-up Language.		CLM/RTGS