EUROPEAN CENTRAL BANK

EUROSYSTEM

CCBM2
Interface Guide
Part 0
Generic Rules and Principles

March 2011 – Working Version (Public)

TABLE OF CONTENTS

1. SCOPE OVERVIEW

The Interface Guide is part of the CCBM2 functional documentation shared across five documents:

- The Business Requirements Document (BRD)
- The Detailed System Requirements (DSR)
- The Human Interface Design (HID)
- The User Manual (UM)
- The Interface Guide (IG)

The Interface Guide provides the specifications of the messages and files to be exchanged with CCBM2; therefore this document is only related to the Application-to-Application-Mode (A2A).

The BRD and DSR provide the requirements of the CCBM2-system. The BRD contains high-level requirements and the DSR detailed requirements. These requirements serve as basic information to the functional design of CCBM2-system processes.

The HID describes the GUI (Graphical User Interface) of CCBM2 user screens and is used later on for translating logical test-cases into physical test-cases.

The UM describes how the CCBM2-system can be used, so a description of the available screens and how to use them. The User Manual and Human Interface Design are therefore related to the User-to-Application-Mode (U2A).

1.1. SCOPE OF THIS RELEASE

The current version of the Interface Guide **is work in progress**. Its content envolves with the analysis provided during each release of other functional documents. Therefore, it is not intended to be a complete and updated document in advance but in synchronisation with each release of other functional documents of CCBM2.

The current version only covers the requirements for Release 1 and Release 2 of the DSR (7 DSR Releases in total), all other information has been removed for readability and testability purposes

For this release of the Interface Guide, we consider as being out of Scope:

- ISO20022 (so called MX messages), and Internet format.

Reason	Internet could also be used for A2A communication: proprietary systems of NCBs, counterparties and CSDs send messages using a CCBM2 agreed format. Examples are (bulk) files as used for credit claims, data information in the case of auto-collateralisation and tri-party collateral management and regular messages for collateral mobilisation. The current assumption is that as far as possible for these A2A-messages the ISO20022-messages will be used.

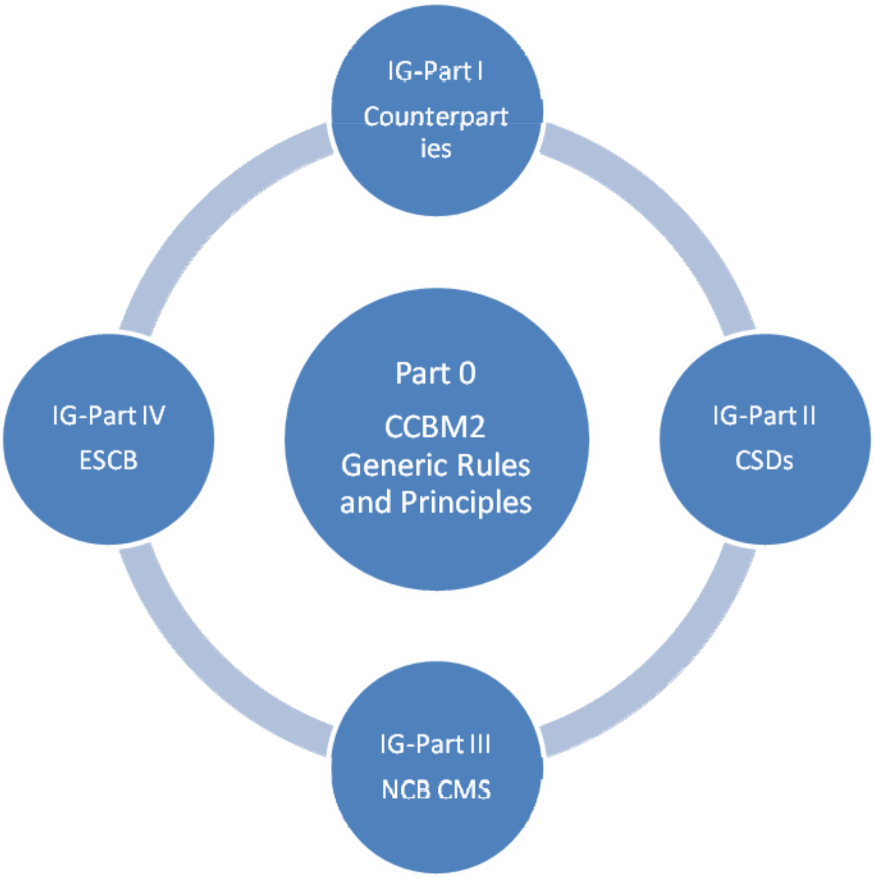
Part 0: Chapter 2 explains how to read the message templates.

Part 0: Chapter 3 and 4 have not been updated in the release and are part of this release for informative purposes only.

Important changes in subsequent releases of the Interface Guide will be referenced in order to keep track of modifications that influence communication/system interfacing with CCBM2 and facilitate information amongst the stakeholders.

1.2. INTRODUCTION

The Interface Guide consists of five parts:



- **Part 0** provides the generic rules and principles:
  - an overview of the validation rules applicable to the various incoming messages,
  - an overview of the enrichment rules applicable to the different outgoing messages,
  - the overall contextual information, as well as
  - conventions being applicable to the 3 other parts of the Interface Guide.

  Generic rules and principles are always bundled with any other part of the Interface Guide.

Scope Limitation	Useful content for DSR Release1 and Release2 only and still subject to amendments Network Specifications not to be considered as completed and updated Validation Rules are subject to amendments during subsequent releases. More detail on rules will be given in next release of Interface Guide

- **Part I** describes the messages and files to be exchanged between CCBM2 and Counterparties (CP) or its authorised sender (delegated actor). This document will represent the Counterparties message specifications based on a first version of the CCBM2 Harmonized CP format.

  This is **one of the two Public parts of the Interface Guide**[1].

- **Part II** describes the messages and files to be exchanged between CCBM2 and CSDs in the various participating countries. This is the **second of the two Public parts of the Interface Guide**.

Scope Limitation	Autocollaterisation and Triparty services related messages are out of the scope of this release The described communication is currently limited to the usage of SWIFTNet FIN service This document will represent the CSD message specifications based on the information received from NCBs to date.

Part III describes the messages and files to be exchanged between CCBM2 and NCB's. **This is the ESCB Internal part of the Interface Guide**.

- **Part IV** describes the communication between ESCB applications and CCBM2 (f.e. tender applications for importing allotment results, accounting systems etc.), as well as communication between CCBM2 and Dataproviders, Target2/PHA, Target2 Securites (T2S).**This is the ESCB Internal part of the Interface Guide**.

[1] *Note: The work on the harmonization of the messages towards counterparties has not been yet completed, hence this part will not be published in this release.*

2. MESSAGE TEMPLATE CONVENTIONS

2.1. READING SWIFT-FIN FORMAT DESCRIPTIONS (ISO15022)

2.1.1. COLUMNS

The template uses the following layout:

SOURCE										TARGET
SWIFT Format						CSD Specific Rules				
SWIFT	Tag	Option	Qualifier	Description/utilisation of field	Format	CSD M/O	Business rule	Matched	Examples	Mobilisation Instruction

The first line devides the template in 2 parts Source and Target
The source and target can switch place depending if it's an incomming or outgoing message.

The section under SWIFT format defines everything related to the SWIFT message format:

- **SWIFT** :Is the field optional (O), conditional mandatory (C/M) or mandatory (M) for SWIF. Grouping tags are indicated by 'I', 16R is used to start the block while 16S ends it.
- **Tag**: the SWIFT Tag
- **Option**: the option of the Tag; some tag have multiple options (formats)
- **Qualifier**: the tag qualifier; some Tags are used for multiple purposes
- **Description**: the SWIFT usage of the field
- **Format**: the format constraints imposed by SWIFT (see 2.3)

The section under CCBM2/CSD Specific Rules defines everything related to the field usage by the CSD/CCBM2:

- **CSD M/O**: is the field optional (O), conditional mandatory (C/M) or mandatory (M) for CSD/CCBM2
- **Business rule**: a short description of how the field will be used by CCBM2 (to match the CSD requirements)
- **Matched**: will the field be used for matching (by CSD or CCBM2)
- **Example**: an example of the tag value (the Tag and Option are not used in the example)

The section under LIM defines the logica fields as they are used in the DSR.

2.1.2. ROWS

A basic introduction about the structure of a SWIFT
Example:

O	Sequence A1 - Linkages				
I	16	R	LINK		
M	20	C	RELA	reference of the message that is confirmed	:4!c//16x
I	16	S	LINK		

This line indicates the start of a logical SWIFT block (Sequence A1 - Linkages is the name of the block). This line is not part of the SWIFT format and is for informative/layout purposes

O	Sequence A1 - Linkages

A block is (in most case) delimited with a 16R - 16S Tag. The LINK qualifier indicates that this block is Sequence A1 - Linkages.

I	16	R	LINK		

I	**16**	**S**	**LINK**		

When several options are possible for a certain tag

M	**95**	**P**	**PSET**	**Place of settlement BIC code**	**:4!c//4!a2!a2!c[3!c]**
		R	**PSET**	**Place of settlement proprietary code**	**:4!c/8c/34x**
		Q	**PSET**	**Place of settlement name and address**	**:4!c//4*35x**

The format of the lines (borders) indicates what field information should be read together. In this case the P, R and Q are options. If on the same line of R in another column a value is put, this means that that value belongs to that specific option e.g. P, R and Q will have it's own format although only 1 option is used for the 95a PSET tag.

Typographic styles are used, only for readability; no interpretation should be attached to the use of this. Next types can be found in the sheets:

- bold: when the tag is mandatory
- bold + cursive: when the tag is conditional mandatory

2.1.3. SWIFT FORMAT

The format of a field is defined as follows:

- nn = Maximum length
- nn! = Fixed-length
- nn-nn = Minimum and maximum length
- nn * nn = Maximum number of lines times maximum line length
- [...] = Brackets indicate an optional element

The allowed characters are defined as follows:

- n = Digits
- d = Digits with decimal comma
- h = Uppercase hexadecimal
- a = Uppercase letters
- c = Uppercase alphanumeric
- e = Space
- x = SWIFT character set

X S.W.I.F.T. Character Set	**Code**
Alphabetical Characters	
A to Z (upper case)	EBCDIC
a to z (lower case)	EBCDIC
Numeric Characters	
0 to 9	EBCDIC
Special Characters	
/ - ? : ( ) . , ’ + SPACE CrLf	EBCDIC

- y = Uppercase level A ISO 9735 characters

Y S.W.I.F.T. Character Set	**Code**
Alphabetical Characters	
A to Z (upper case)	EBCDIC
Numeric Characters	
0 to 9	EBCDIC
Special Characters	
SPACE . , - ( ) / = ’ + : ?	EBCDIC

Special Characters (incompatible with international telex)	
! ” % & * ; < >	EBCDIC

- z = SWIFT extended character set

Alphabetical Characters	
A to Z (upper case)	EBCDIC
a to z (lower case)	EBCDIC
Numeric Characters	
0 to 9	EBCDIC
Special Characters	
. , - ( ) / = ’ + : ? @ # Cr Lf SPACE {	EBCDIC

For details see the SWIFT documentation 'Message Format Validation Rules'.

2.2. READING THE XML FORMAT DESCRIPTIONS

The template uses the following layout:

ssp.pm.ModifyCreditLine camt.998.01.02.xsd				**Output message from CCBM2 (CCBM2 sends)**		**SOURCE**
XML Format (ssp.pm.ModifyCreditLine.camt.998.001.02.xsd @ https://target2.ecb.int/doc/udfs/ **IN** pm library**)**						**LIM reference**
XPATH	**Description/utilisation of field**	**Format**	**M/O**	**Business rule**	**Matched**	**RTGS_ModifyCreditLine**

The first line divides the template in 2 parts Source and Target. The source and target can switch place depending if it's an incomming or outgoing message.

The follwing columns are used

- **XPATH**: the field to map, using an Xpath notation
- **Description**: a short description
- **Format**: the format specification, details can be found in the XSD:
- **M/O**: is the field optional (O), conditional mandatory (C/M) or mandatory (M)
- **Business** rule: a short description of how the field will be used by CCBM2
- **Matched**: will the field be used for matching (by CCBM2)

The section under LIM defines the logical fields as they are used in the DSR.

3. A2A-COMMUNICATION CHANNELS AND STANDARDS USED BY CCBM2

Communication Channels

The following networks can be used by external parties to communicate with CCBM2 in A2A-mode:

- SWIFT (FIN, InterAct and FileAct).
- Secure Internet.
- Corenet[2] (NCBs).

The technical specifications for SWIFT and Secure Internet have been given in chapter 4.
Restrictions apply to which communication channel can be used depending on the actor and external proprietary systems. The table below shows which channel is available for which party:

Available Channels for External Parties

	SWIFT	Corenet	Internet
	A2A	A2A	A2A
CSD	✓		✓
NCB when acting on behalf of CSD	*To be confirmed*		
Counterparty	✓		✓
Third Pary Custodian	✓		✓
External CMS[3]	✓		✓
Target2/PHA	✓		
T2S	*To be confirmed*		
NCB applications		✓	

Choice of Channel

For messages/files sent in to CCBM2, parties can choose at each moment which of the available channels they want to use (so for example for the first message of a day SWIFT can be used, for the second one of the same kind of message Secure Internet etc.).

For messages/files to be received by parties from CCBM2, they have to indicate (via the static data of CCBM2) for every kind of message/file which channel always must be used by CCBM2, independent of the channel that was used to send in a message. So for example, if a Counterparty has indicated that they always want to receive a ‘Settlement Confirmation Mobilisation Request Free of Payment’ via SWIFT, CCBM2 will always send this message via SWIFT, even if the Counterparty did send in the Mobilisation Request via Secure Internet or via the User-to-Application Mode.

[2] ESCB internal network (NCB2NCB communication only)

[3] E.g. Triparty collateral management service providers

MT and MX

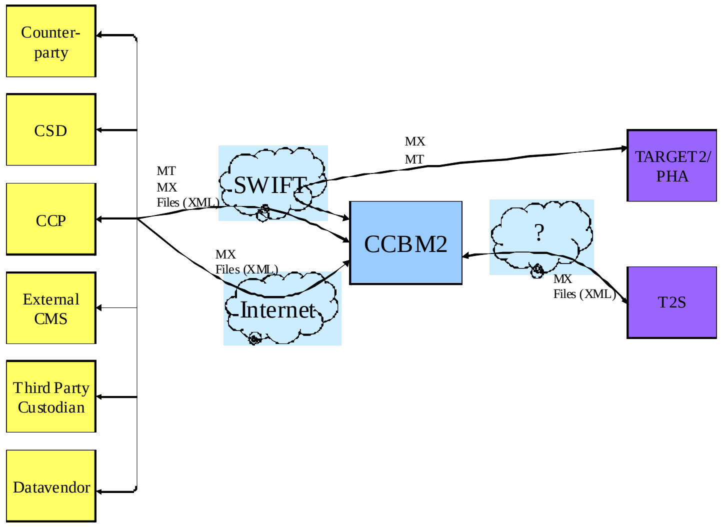
CCBM2 will support the ISO15022 (or so called MT-messages) from start on and the ISO20022 (or so called MX-messages) standard could be supported as of Go-live T2S.

Standards to be used per Channel and External Party

	SWIFT	Corenet	Internet
	A2A	A2A	A2A
CSD	MT MX[4] Files (XML)		MX Files (XML)
Counterparty	MT MX Files (XML)		MX Files (XML)
Third Party Custodian	MT MX Files (XML)		MX Files (XML)
External CMS	MT MX Files (XML)		MX Files (XML)
CCP	MT MX Files (XML)		MX Files (XML)
Target2/PHA	MT MX		
T2S			

Diagram

The information of both tables above is depicted in the diagram below.



[4] MX messages over the SWIFT network will be supported as of Go-live T2S.

Chosen approach writing the Interface Guide

For the development of the Interface Guide the approach is chosen to first develop the MT-messages, and start the discussions with the NCBs and the market on these MT-messages. Once the discussions about the MT-messages are finalised, the XML-format (for the internet channel) should be derived straight forward by using reverse engineering translation rules.

The reasons for choosing this approach are:

- The MT-messages are quite commonly known.
- By focussing on the MT-messages first, changes on request by the NCBs or market participants only need to be done to the MT-messages.

Usage of SWIFT-services

CCBM2 will make the following use of the services offered by SWIFT:

- For MT-messages SWIFTNet-FIN will be used.
- For A2A-request/response interactions SWIFTNet InterAct Store-and-Forward Messaging Mode will be used and FileAct Store-and-Forward in case of large replies from CCBM2.
- For A2A-request/response interactions with Target2 InterAct Real-time Query-and-Response will be used.
- For the delivery of files to CCBM2 containing instructions (for example the non-marketable assets bulkfile) SWIFTNet FileAct Store-and-Forward will be used.

4. TECHNICAL SPECIFICATIONS PER NETWORK

4.1. SWIFTNet FIN Messages

4.1.1. Headers, Trailers and double entry check

4.1.1.1. Basic Header

Usage

The Basic Header is used in every message type sent to or received from CCBM2 via SWIFTNet-FIN.

Structure

The Basic Header has the following structure:

Basic Header			
Status	**Field Name**	**Format**	**Use in CCBM2**
M	Block Identifier	1:	-
M	Application Identifier	F	F=FIN
M	Service Identifier	01	-
M	LT Address	4!a2!a2!c1!c3!c	BIC+LT, 12 digits Message from participant to CCBM2: o Sender's LT address Message from CCBM2 to participant: o Receiver's LT address
M	Session Number	4!n	-
M	Sequence Number	6!n	

4.1.1.2. Application Header

Usage

The Application Header is used in every message type sent to or received from CCBM2 via SWIFTNet-FIN. It has different formats depending on whether the participant sends or receives a message.

Structure when sending a Message

The Application Header has the following structure when a participant sends a message via SWIFTNet-FIN to CCBM2:

Application Header			
Status	**Field Name**	**Format**	**Use in CCBM2**
M	Block Identifier	2:	-
M	Input/Output Identifier	I	I=Input for SWIFT
M	Message Type	3!n	Examples of Message Types used in CCBM2 are: 540,541,542,202,204 etc
M	Destination Address	4!a2!a2!c1!c3!c	BIC+LT, 12 digits o Receiver's LT address
M	Message Priority	N or U	Not relevant in CCBM2
O	Delivery Monitoring	1!n	-
O	Obsolescence Period	3!n	-

Structure when receiving a Message

The Application Header has the following structure when a participant receives a message via SWIFTNet-FIN from CCBM2:

Application Header			
Status	**Field Name**	**Format**	**Use in CCBM2**
M	Block Identifier	2:	-
M	Input/Output Identifier	O	O=Output for SWIFT
M	Message Type	3!n	Examples of Message Types used in CCBM2 are: 540,541,542,598 etc.
M	Input Time	HHMM	-
M	Message	6!n4!a2!a2!c1!c3	Input date, local to the

	Input Reference	!c4!n6!n	sender, LT address of the sender, session and sequence number of the sender.
M	Date	YYMMDD	Output date, local to the receiver.
M	Time	HHMM	Output time, local to the receiver
M	Message Priority	N or U	Sender's message priority.

4.1.1.3. Trailer

Usage

The Trailer is used in every message that is exchanged with CCBM2. The content of the Trailers is different depending on whether a message is sent to or received from CCBM2.

Structure when sending a Message

The Trailer has the following structure when a participant sends a message via SWIFTNet-FIN to CCBM2:

Trailer			
Status	**Field Name**	**Content/ Options**	**Use in CCBM2**
-	Block Identifier	5:	-
M	Authentication Code	{MAC:8!h}	-
M	Checksum	{CHK:12!h}	-
O	Training	{TNG}	Only in test and training mode.
O	Possible Duplicate Emission	{PDE:[<time><mir>]}	

Structure when receiving a Message

The Trailer has the following structure when a participant receives a message via SWIFTNet-FIN from CCBM2:

Trailer			
Status	**Field Name**	**Content/ Options**	**Use in CCBM2**
-	Block Identifier	5:	-
M	Authentication Code	{MAC:8!h}	-
M	Checksum	{CHK:12!h}	-

O	Training	{TNG}	Only in test and training mode.
O	Possible Duplicate Emission	{PDE:[<time><mir>]}	
O	Possible Duplicate Message	{PDM:[<time><mor>]}	
O	Delayed Message	{DLM}	

4.1.1.4. Handling of PDM/PDE Trailer

PDM Trailer (Possible Duplicate Message Trailer)

A PDM-trailer is added to a SWIFT-message by SWIFT. It is used to warn the receiver that the same message may already have been delivered by SWIFT. The reason for sending a message with PDM-trailer is that SWIFT does not know whether the message was already sent.

If CCBM2 receives a message it checks in addition to the double-entry check whether the message is delivered twice (once with and once without a PDM-trailer):

- If the message without PDM trailer was already delivered, then the message with PDM-trailer will be ignored by CCBM2.
- If the message without PDM trailer was not delivered, the message with PDM trailer will be processed by CCBM2.
- If the message without PDM trailer is delivered after the message with PDM trailer, the message without PDM trailer will be ignored by CCBM2.

PDE Trailer (Possible Duplicate Emission Trailer)

A PDE-trailer is added by the sender of a SWIFT-message. It is used to warn the receiver that the same message may already have been received. The reason for sending a message with PDE-trailer is that the sender is not sure whether the message was already sent.

If CCBM2 receives a message it checks in addition to the double-entry check whether the message is delivered twice (once with and once without a PDE-trailer):

- If the message without PDE trailer was already delivered, then the message with PDE-trailer will be ignored by CCBM2.
- If the message without PDE trailer was not delivered, the message with PDE trailer will be processed by CCBM2.
- If the message without PDE trailer is delivered after the message with PDE trailer, the message without PDE trailer will be ignored by CCBM2.

4.1.1.5. Double Entry Check

Double Entry Check

CCBM2 carries out a check on the double receipt of messages. If the following information regarding a SWIFT-message that is received by CCBM2 is the same as in a previously received message, CCBM2 will reject the message:

- Sender of the message
- Message Type
- Sender Reference

Note: If a message was rejected by CCBM2 because of an error within the message, then the next delivery of the 'same' (i.e corrected) message is not rejected because of the double entry check, to allow the 'resending' of corrected messages.

4.1.2. SECURITY

Security Aspects

The following security aspects are of importance:

- Authenticity
  The common SWIFT-functionality using the PKI-solution for establishing the authenticity of a sender of a SWIFT-message is used.

- Non-repudiation
  The common SWIFT-functionality for the non-repudiation of a sent or received message is used.

- Integrity
  The common SWIFT-functionality using the PKI-solution for safeguarding the accuracy of the contents of messages and the standard SWIFT-functionality to guarantee the completeness of the number of sent and received messages (ACK/NACK, ISN/OSN-gap-detections) are used.

- Availibility
  See UDFS chapter 12 regarding the measures to guarantee the agreed availability of the CCBM2-platform.

- Confidentiality
  The common SWIFT-functionality for encrypting messages using the PKI-solution is used.

4.2. SWIFTNet InterAct/FileAct Messages

4.2.1. Global Structure of a XML Message

Global Structure of a XML Message

A XML message via InterAct Real-Time Messaging Mode and via InterAct/FileAct Store-and-Forward Mode (both in Push-Mode) consists of the following blocks:

Name of Block	**Optional or Mandatory by SWIFT**
Authorisation Context, Authenticator, Requestor	M
Request Control	M
Request Header	M
Payload: - Application Header - Document	O (but mandatory for CCBM2)
Cryptographic Blocks	O
Message Signature	O

4.2.1.1. Authorisation Context, Authenticator, Requestor

Usage

This block contains the ‘User Distinguished Name’ (DN) of the entity that authorised the sending of the message. The User DN is a X.500 distinguished name, ending with:

o=<BIC8>, o=swift

It is used by SWIFT to identify and authenticate the sending user. The sending user can be an operator or an application certificate.

Example: BANKNL2A sends a XML request to CCBM2 using the following User DN:

<SwSec:UserDN>cn=jweersma,o=banknl2a,o=swift</SwSec:UserDN>

4.2.1.2. Request Control

Usage

This block indicates vis-à-vis SWIFT whether the request:

- Contains cryptographic operations to be performed.
- Requires non-repudiation of emission (NRE).
- Is to be handled by means of Store-and-Forward.

- In CCBM2 all messages having ‘writing access’ require NRE. For these messages the field <NRIndication> must contain the value ‘TRUE’.

4.2.1.3. Request Header

Usage

This block contains the following information:

- Distinguished Name of the Requestor (Requestor DN).
- Distinguished Name of the Responder (Responder DN).
- Service Name.
- Request Type.
- Priority (not used by CCBM2).
- Request Reference.

Requestor DN

The Requestor DN identifies the sending party (=institution that sends the XML-message). The Requestor DN is a X.500 distinguished name, ending with:

o=<BIC8>,o=swift

No registration in the SWIFTNet directory tree of the sending party or certification of this DN is required. The value of the requestor field may be equal to the ‘user-DN’ which is used to sign the message, but this is not mandatory.

<SwInt:Requestor>o=BIC8,o=swift</SwInt:Requestor>

Responder DN

The Responder DN identifies the receiving party (=institution that receives the XML-message). The Responder DN is a X.500 distinguished name, ending with:

o=<BIC8>,o=swift

At this moment it is not yet known which Responder DNs will be used by CCBM2.

<SwInt:Responder>o=ccbmxecm,o=swift</SwInt:Responder>

Service Name

The Service Name contains the SWIFTNet service used. For CCBM2 different Service Names are in place for:

- Customer test environment

Service Name	SWIFTNet Service	Mode
Not known yet	InterAct FileAct Browse	Real-time
Not known yet	InterAct FileAct	Store-and-Forward

- Live environment

Service Name	SWIFTNet Service	Mode
Not known yet	InterAct FileAct Browse	Real-time
Not known yet	InterAct FileAct	Store-and-Forward

The tag containing the Service Name is called: <SwInt:Service>.

Request Type

The Request Type identifies the message type of the XML message using the standard code of the message, for example ‘sese.023.001.01’ for a Marketable assetsSettlementTransactionInstruction (see the SWIFT-documentation for all the codes).

The tag containing the Request Type is called: <SwInt:RequestType>

4.2.1.4. Payload

Usage

This block contains the Application Header and the so called Document (in fact the actual MX-message).

Application Header

The Application Header contains information that is relevant to the business applications that route and process the business document. This information is required before opening the actual document so that the business application can process the content properly.

The Application Header regarding CCBM2 contains the following components:

Header Element	Syntax	Purpose	Use in CCBM2
Business application from which the document is received	<From> <Type> ????? </Type> <Id> ????? </Id> </From>	Identifies the application that has created the document	This element will be filled by CCBM2 for XML-messages sent by CCBM2
Business application to which the document is sent	<To> <Type> ????? </Type> <Id> ????? </Id> </To>	Identifies the receiving application for which the document is created	Will have to be filled by the sender of the XML-message.
Message Reference	<MsgRef>	The unique identifier of the message.	
Creation Date	<CrDate>	Date and time at which the message was created	

N.B. The Application Header only contains the ‘From’ or the ‘To’ element, depending if the message is sent or received by CCBM2.

4.2.1.5. Cryptographic Blocks

Usage

This block is general optional but mandatory when NRE is applied.
There are no specific rules defined for this block by CCBM2

4.2.1.6. Message Signature

Usage

This block is not relevant for CCBM2. It might be used according to the rules defined by SWIFT.

4.2.2. Security

Security Aspects

The following security aspects are of importance:

- Authenticity
  The common SWIFT-functionality using the PKI-solution for establishing the authenticity of a sender of a SWIFT-message is used. It is not clear yet if RBAC will be used by CCBM2, and if yes, if a special A2A-role will be defined for CCBM2.

- Non-repudiation
  The common SWIFT-functionality for the non-repudiation of a sent or received message is used.

- Integrity
  The common SWIFT-functionality using the PKI-solution for safeguarding the accuracy of the contents of messages and the standard SWIFT-functionality to guarantee the completeness of the number of sent and received messages are used.

- Availibility
  See UDFS chapter 12 regarding the measures to guarantee the agreed availability of the CCBM2-platform.

- Confidentiality
  The common SWIFT-functionality for encrypting messages using the PKI-solution is used.