



Standards

Standards MX

General Information

This document describes the SWIFT Standards MX messages (MXs) and explains the concept of XML for MX messages and the structure and function of these messages. This document is for customers that use MXs, and software developers that implement SWIFT Solutions that use MXs.

29 April 2011

Table of Contents

| | |
|--|----|
| Preface | 3 |
| 1 Standards MX Message Concepts | 4 |
| 1.1 Message Granularity | 4 |
| 1.2 Message Variants | 5 |
| 1.3 Message Versions and Maintenance | 7 |
| 1.4 Message Classification and Naming | 7 |
| 1.5 Message Definition | 14 |
| 1.6 SWIFT Standards Financial Dictionary | 16 |
| 2 XML for Standards MX Messages | 18 |
| 2.1 Introduction to XML | 18 |
| 2.2 XML Schemas and Instances | 18 |
| 2.3 XML Syntax | 20 |
| 2.4 XML Document Validation | 29 |
| 2.5 SWIFT Standards Supported Character Sets and Encoding Scheme | 32 |
| 2.6 xs:any | 34 |
| 3 MX Messages and InterAct | 37 |
| 3.1 Message Structure | 37 |
| 3.2 Message Headers | 38 |
| 3.3 Message Instance | 46 |
| 4 MX Messages and FileAct | 48 |
| 4.1 File Structure | 48 |
| 4.2 File Header | 49 |
| 5 Understanding the Documentation | 54 |
| 5.1 Standards MX Message Reference Guide - HTML | 54 |
| 5.2 Standards MX Message Reference Guide - PDF | 70 |
| 6 Financial Glossary | 75 |
| Legal Notices | 81 |

Preface

About this document

This guide provides a general description of Standards MX (MXs) concepts, as follows:

- "Standards MX Message Concepts" describes general concepts which include message frameworks, message granularity, message variants and versions, message classification, message naming, and definitions. This section also introduces the SWIFT Standards dictionary.
- "XML for Standards MX Messages" describes how MXs use XML. This section includes information about XML schemas, instances, and syntax, discusses tags and elements, and describes XML document standards, supported characters, validation, and namespaces.
- "MX Messages and InterAct" describes the structure of an InterAct message and describes how to use the application header.
- "MX Messages and FileAct" describes the structure of a FileAct message and describes how to use the file header.
- "Understanding the Documentation" describes how to use the Standards MX documentation in both the HTML and the PDF formats.

For the latest available version of this guide, see swift.com > Support > [Documentation](#).

Audience

Customers that use MXs, and software developers that implement SWIFT Solutions that use MXs, are the intended audience for this document.

Reading conventions

This guide contains some terms that are abbreviated as follows:

- SWIFT Standards XML message is abbreviated to MX
- SWIFT Standards FIN message, Message Type, is abbreviated to MT

Significant changes

The following table lists the significant changes to the content of the *Standards MX General Information* since the September 2008 edition.

| New and updated information | Location |
|---|---------------------------------|
| Message maintenance process | "Maintenance Process" on page 7 |
| New ISO 20022 business application header | "Message Headers" on page 38 |

Related documentation

- *Standards MX Message Reference Guides*
- *SWIFTNet Service Description*
- *SWIFTNet Messaging Operations Guide*

1 Standards MX Message Concepts

Introduction to MX message concepts

This section contains information about the following topics:

- A description of "Message Granularity". This maximises consistency in the structure of messages that support the same message functions. Examples of messages that support the same message functions are an instruction, a query, a report, or a confirmation.
- Information about "Message Variants" and "Message Versions" that enable customer to fulfil specific business requirements and allow messages to evolve over time.
- A definition of the "Maintenance Process".
- A list of "Message Classification and Naming" that ensure that the network and business applications can identify each MX uniquely.

1.1 Message Granularity

Granular message types

SWIFT strictly controls the design of all MXs by using granular message types. A granular message defines only one type of business function.

Examples

| Task | Number of message types |
|--|---|
| Update account data, and cancel a standing order on the same account. | This requires 2 message types. These are 2 different business functions. |
| Initiate a set of bulk payments. | This requires 1 message type. This is more than 1 action but only 1 business function. |
| Transfer a payment to a beneficiary by crediting its account or by issuing a cheque. | This requires 1 message type. This task uses 2 different financial instruments but is only 1 business function. |

Optional fields

Each granular message type has the minimum number of optional fields. These optional fields are limited and usage rules limit their conditions for use.

Example of an optional field

Payment initiation has an optional field, charges account. This field must only be used if charges have been booked to an account other than the debtor's account.

Advantages of granular message types

The advantages of granular message types are as follows:

- Granular message types only impact business applications that deal directly with the changes. Maintenance of multipurpose message types can affect other business applications.
- Granular message types are simpler than multipurpose message types and require fewer fields and business rules as they cater for fewer functions.

- Granular message types are easier to document than multipurpose message types as they are always used for the same purpose.
- Granular message types have a positive impact on Straight-Through Processing (STP) rates because they reduce the risk of an incorrect implementation or interpretation.
- Granular message types require fewer bilateral agreements than multipurpose message types because their usage is predictable.
- A single business application usually processes the same message type making easier to route incoming messages to the correct business application. Multipurpose message types may need to be routed to several applications depending on their content.
- Granular message types fit into end-to-end transactions more easily because it is easier to know which message to use and when to use it. It is also easier to know the status of a transaction based on the received message type.

Note The granularity principle can lead to an increase in the number of message types when moving from SWIFT Standards MT messages (MTs) to Standards MX messages (MXs). This is especially the case when MTs are very generic.

For example, MT n95 Queries and MT n96 Answers are used to process exceptions and investigations. This requires a larger number of MXs because MXs allow more automation.

The examples in this document do not always reflect the latest available version of the messages or the message components, and are presented for illustration purposes only.

1.2 Message Variants

Explicit and implicit variants

A message variant defines a specific way of using a message. Message variants cater for specific requirements in local market practices and restrictions required by the level of automation, for example Straight-Through Processing (STP).

For more information about STP, see "Extensible messages with Straight-Through Processing (STP)" on page 36.

Message variants can be explicit or implicit:

- **Explicit variant**

An explicit message variant is derived from an existing global message by adding restrictions. A specific message definition supports each message variant.

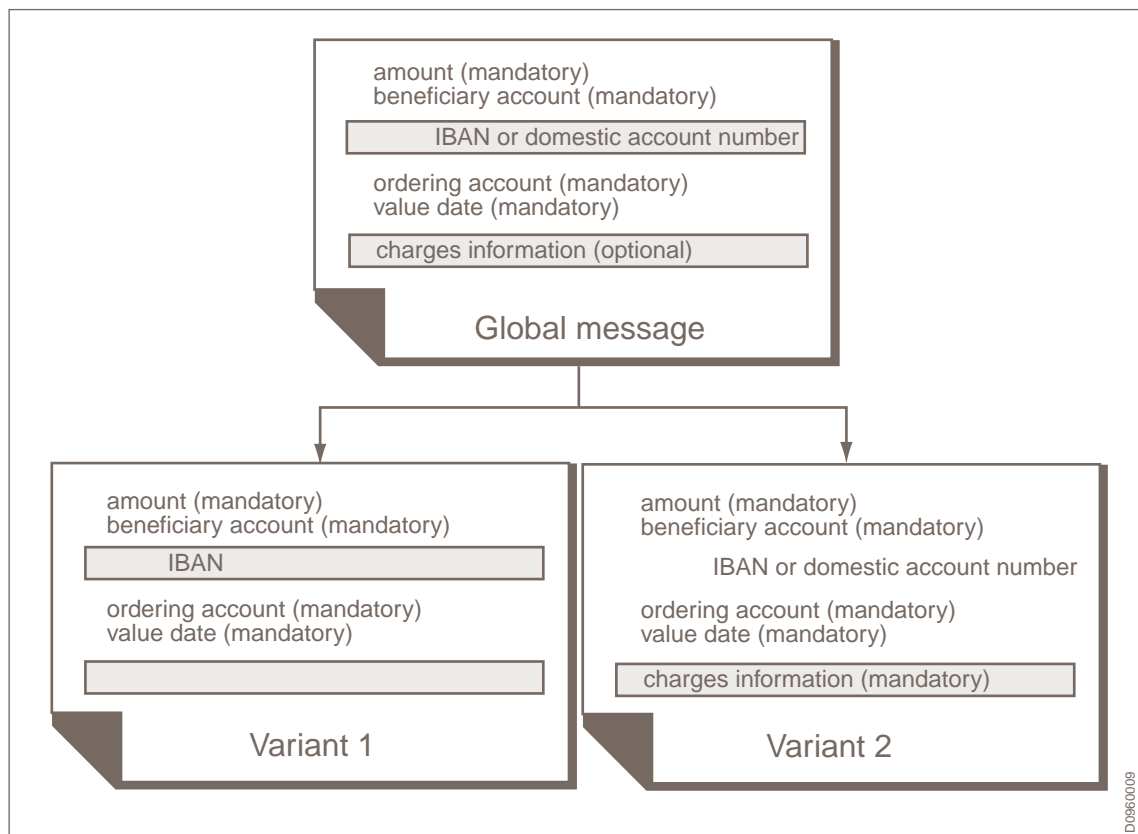
- **Implicit variant**

An implicit message variant is embedded in a single global message with specific usage guidelines, market practices, or bilateral agreements.

Examples of message variants

This example shows how variants are derived from a single global message. The first variant limits the choice to one option (only the IBAN is accepted) and optional information (charges information) has been removed. The second variant makes the optional information (charges information) mandatory.

Message variants derived from a single global message



Rules for the creation of message variants

For message variants, SWIFT imposes the following rules:

- SWIFT only creates message variants if there is a valid business justification.
- SWIFT limits the lifetime of message variants.
- SWIFT regularly reassesses the business justification for the message variants that exist.
- Message variants must never be in conflict with the global message.

All the mandatory fields in the global message must always be present in the message variants, and no new fields must be added. This ensures that standards can be implemented based on the structure and contents of the global message rather than the variants.

Fields that are optional in the global message can appear in some variants. These fields can be optional or mandatory in the variants. Fields that define a choice in the global message can restrict this choice in a variant.

1.3 Message Versions and Maintenance

1.3.1 Message Versions

Definition of message versions

Message versions enable customers to migrate from one way of using a message to another way of using that message. Customers can change the way that they use the message according to changes in the business or technical requirements.

A new version of an MX has a version number that is part of the MX name and an identifier increment. SWIFT supports coexistence of two versions of the same message for a limited time. At an agreed date, SWIFT no longer supports the older version of the message.

Note The difference between message variants and message versions is that message variants exist concurrently and independently of one another. Message versions enable customers to switch from one use for a message to another use of the message to accommodate a change in business requirements.

Example of message versions

When it is mandatory to include the IBAN in the message, a new version of the message is created.

1.3.2 Maintenance Process

MX maintenance

The *Standards MX Development and Maintenance Process* document describes the creation and maintenance process used by the SWIFT Standards Department to create and maintain MX message standards.

ISO 20022 maintenance

In most of the cases when there is a maintenance to an ISO 20022 message, SWIFT follows the the ISO maintenance process.

Related information

For more information about the MX maintenance process, see the *Standards MX Development and Maintenance Process* at swift.com > Support > [Documentation](#).

For more information about the ISO 20022 maintenance process, see iso20022.org > [Registration Process](#)

1.4 Message Classification and Naming

Introduction to message names, identifiers, and business areas

Each MX has a message name and a message identifier. The message name is human-readable. The message identifier is a unique structured computer-readable identifier for use by systems and applications.

A message name and its identifier must be considered in the context of its business area.

Related information

"Business Area" on page 8

"Message Identifier" on page 13

1.4.1 Business Area**Definition of a business area**

A business area is the area where an MX is exchanged. A business area is composed of a set of strongly related activities, for example, payments clearing and settlement, or securities settlement. Each message is associated with a business area. It is important that customers identify any messages that belong to the same message area when they develop new applications. This is because the message area can be used to route messages.

Example of a message that exists in only one business area

A message that only exists in one business area is the CustomerCreditTransferInitiationV02 (`pain.002.001.02`) message. This exists only in the context of the Payment Initiation business area.

Example of a message that exists across business areas

A message that exists across business areas is the PaymentsStatusReportV02 message. This message exists in the context of the Payment Initiation business area (`pain.002.001.02`) and the Payments Clearing and Settlement business area (`pacs.002.001.02`).

1.4.2 Business Area Codes**Background**

The following table lists business areas that cover the FIN domains, the related codes, and the existing SWIFT Standards MX messages. The FIN MTs are examples only, to clarify the business area. The examples are not exhaustive, and there is no intention to imply that there are current plans to create equivalent messages in XML. The development of new XML messages in a particular business area requires the approval of the SWIFT Board of Directors.

Note The business area codes mentioned in this table in **bold** are non-ISO codes. They can be used on the SWIFT network to refer to a category rather than a specific business area, where appropriate.

| Business area code | Business area | Description | Examples of MTs classification |
|--------------------|---------------------------------------|---|--------------------------------|
| cash | PAYMENTS & CASH MANAGEMENT | | |
| <code>pain</code> | Payments Initiation | Messages that support the initiation of a payment from the ordering customer to a financial institution that services a cash account and reporting its status | 101, 104(RFDD), 207 |
| <code>pacs</code> | Payments Clearing and Settlement | Messages that support the clearing and settlement processes for payment transactions between financial institutions | 102-4, 107, 200-5, 400, 405 |

| Business area code | Business area | Description | Examples of MTs classification |
|---------------------------|-----------------------------|---|--|
| camt | Cash Management | Messages that support the reporting and advising of the cash side of any financial transactions, including cash movements, transactions and balances, plus any exceptions and investigations related to cash transactions. | 210, 400, 900, 910, 920, 935, 940-2, 950, 97n, 990-2, 995-6, 190-2, 195-6, 290-2, 295-6 |
| trad | TRADE SERVICES | | |
| tsin | Trade Services Initiation | Messages that support the request for a trade service, including any related application, instruction, request, acknowledgement, or advice. | |
| tsrv | Trade Services | Messages that support the issuance of a trade services instrument, execution and/or settlement of a trade transaction, including any related reimbursement, acceptance, authorisation, claims, enquiries, invoicing, financing, or other undertaking. | "410,412,416, 430, 700,701,705,707, 710,711,720, 721,730,732,734, 740,742,747, 752,754,756,760, 767,768" |
| tsmt | Trade Services Management | Messages that support ancillary commercial trade services functions, including checking, matching and reporting, plus any exceptions and investigations related to trade services transactions. | "420, 422, 490-2, 495-6 750,769, 790-2, 795-6" |
| secs | SECURITIES | | |
| seis | Securities Issuance | Messages that support the process of issuing (issuance/creation) securities. | |
| seti | Securities Trade Initiation | Messages that support the process of trade initiation, including indication of interest in buying or selling a security and the related quoting process. | |
| setr | Securities Trade | Messages that support trade and post-trade processes for securities, including order to buy or sell, trade execution, affirmation, confirmation, allocation, and notification. | 502, 509, 513-5, 517-8, 576, 584 |
| secl | Securities Clearing | Messages that support the clearing process for securities, including management of post-trading, pre-settlement credit exposure, netting, margining, borrowing, conformance with market settlement rules. | 516, 526 |
| sese | Securities Settlement | Messages that support the settlement process for securities and report its status and confirmation. | 528-9, 530, 540-9, 578, 587-9 |
| semt | Securities Management | Messages that support post-settlement processes for securities (including reporting on securities movements, trades, and balances), the processes required to protect beneficial owner's rights throughout settlement, plus any | 508, 524, 535-8, 575, 549, 500-1, 510, 519, 574, 577, 579, 582, 586, 590-2, 595-6 |

| Business area code | Business area | Description | Examples of MTs classification |
|---------------------------|-----------------------------------|--|---------------------------------------|
| | | exceptions and investigations related to securities transactions. | |
| seev | Securities Events | Messages that support asset servicing, including proxy voting, income and corporate actions | 564-8 |
| secs forx | SECURITIES FX | | |
| fxti | Foreign Exchange Trade Initiation | Messages that support the process of trade initiation, including indication of interest in buying or selling a foreign-exchange contract and the related quoting process. | |
| fxtr | Foreign Exchange Trade | Messages that support trade and post-trade processes for foreign-exchange contracts, including order to buy or sell, execution, affirmation, confirmation, allocation, and notification. | 380-1,300, 303-4, 307-8 |
| fxmt | Foreign Exchange Management | Messages that support management processes for foreign-exchange contracts, including reporting on trades and balances, plus any exceptions and investigations related to foreign-exchange contracts. | 390-2, 395-6 |
| bklo | BANK LOAN/DEPOSIT | | |
| blti | Bank Loan Trade Initiation | Messages that support the process of trade initiation, including indication of interest in buying or selling a bank loan or deposit and the related quoting process. | |
| bltr | Bank Loan Trade | Messages that support trade and post-trade processes for bank loans and deposits, including order to buy or sell, execution, affirmation, confirmation, allocation, and notification. | 320-1, 330, 350 |
| blmt | Bank Loan Management | Messages that support management processes for bank loans and deposits, including reporting on trades and balances, plus any exceptions and investigations related to bank loans and deposits. | 390-2, 395-6 |
| deri | DERIVATIVES | | |
| deti | Derivatives Trade Initiation | Messages that support the process of trade initiation, including indication of interest in buying or selling a derivative contract and the related quoting process. | |
| detr | Derivatives Trade | Messages that support trade and post-trade processes for derivative contracts, including order to buy or sell, trade execution, affirmation, confirmation, allocation, and notification. | 305-6, 340-1,360-5 |
| demt | Derivatives Management | Messages that support management processes for derivative contracts, | 390-2, 395-6 |

| Business area code | Business area | Description | Examples of MTs classification |
|---------------------------|--|--|---------------------------------------|
| | | including reporting on trades and balances, plus any exceptions and investigations related to derivative contracts. | |
| comm | COMMODITIES | | |
| coti | Commodities Trade Initiation | Messages that support the process of trade initiation, including indication of interest in buying or selling a commodity and the related quoting process. | |
| cotr | Commodities Trade | Messages that support trade and post-trade processes for commodities, including order to buy or sell, execution, affirmation, confirmation, allocation, and notification. | 600-1, 620 |
| comt | Commodities Management | Messages that support management processes for commodities, including reporting on trades and balances, plus any exceptions and investigations related to commodities. | 604-9, 690-2, 695-6 |
| sylo | SYNDICATED LOANS | | |
| syin | Syndicated Loans Initiation | Messages that support the process of initiation, including indication of interest in buying or selling a syndicated loan and the related quoting process. | |
| synd | Syndicated Loans | Messages that support post-trade processes for syndicated loans, including execution, affirmation, confirmation, allocation, and notification. | 643,644,646 |
| symt | Syndicated Loans Management | Messages that support management processes for syndicated loans, including notices between the agent bank and lenders to manage drawdowns, rate set, and interest and fee payments related to syndicated loans. | 643-6, 649, 690-2, 695-6 |
| card | CARD PAYMENTS | | |
| caaa | Acceptor to Acquirer Card Transactions | Messages that support any card payment-related transactions and services between a card acceptor and a card transaction acquirer. It includes the authorisation, the cancellation, and the capture of card transactions. | |
| cain | Acquirer to Issuer Card Transaction | Messages that support any card payment-related transactions and services between a card transaction acquirer and a card issuer. It includes the authorisation, reversal, and financial presentment of card transactions. | |
| casp | Sale to POI Card Transactions | Messages that support any card-related transactions and services between a sale system and a Point of Interaction (POI) system. | |

| Business area code | Business area | Description | Examples of MTs classification |
|---------------------------|---|--|---------------------------------------|
| caad | Card Administration | Messages that support any card-related administrative services between financial institutions and their agents. | |
| catm | Terminal Management | Messages that support any card-related terminal management services between a Terminal Management System (TMS) and a Point of Interaction (POI) system. | |
| cacs | Card Clearing and Settlement | Messages that support the clearing and settlement processes for card payment transactions between financial institutions | |
| cafc | Free Collection | Messages that support the reporting and advising of card payment transactions, including the collection of fees and the processing of charge-backs. | |
| misc | MISCELLANEOUS/ OTHER/GENERIC | | |
| reda | Reference Data | Messages that support the communication of reference data related to financial instruments, parties, accounts, prices, and other business information required to support financial activities | |
| acmt | Account Management | Messages that support the management of account related activities, such as the opening and the maintenance of an account. | |
| colr | Collateral | Messages that support collateral management actions. | 503-7, 527, 558, 569, 581 |
| admi | Administration | Generic messages (for example, system event notifications and generic rejections) | |
| | OTHERS (non-ISO) | | |
| defp | (derivatives-fpml on SWIFT) | Used for current fpml messages on SWIFT network. | |
| xrma | SWIFT system messages for RMA | Relationship Management Application system messages. | |
| xsys | SWIFT system messages | System messages for use on the SWIFT network. | (similar to FIN system messages) |
| xoom | SWIFT-to-user traffic | Data (for example, reports) sent from SWIFTNet Online Operations Manager | |
| | OBSOLETE (being phased out) | | |
| clpt | (commercial loans) | Originally used for some loans messages. | |

1.4.3 Message Name

Structure of a message name

The structure of the message name is as follows.

| Structure | Definition |
|--------------------|---|
| Message Definition | Specifies the function of the message in the business area. |
| Variant | Identifies a particular market practice. The variant name is optional and is only used if there is more than 1 variant. |
| Version | The version number is identified explicitly in the message name. |

Example message name structure

| Structure | Example |
|--|--------------------------|
| Message Definition [Variant Name] Version Identifier | BulkCreditTransferSTPV01 |

1.4.4 Message Identifier

Structure of a message identifier

Each MX has a message identifier. The message identifier is a unique structured identifier intended for use by systems and applications. Dots separate the four elements of the message identifier. There is a one-to-one relationship between the message name and the message identifier.

| Format | Function |
|--------|---|
| MX | Stands for XML message type. MX is omitted when the message identifier is used in applications or in the SWIFTNet Request Header. |
| 4!a | Mandatory code that identifies the business area. For example, <i>pacs</i> stands for Payments Clearing and Settlement. This code has a fixed length of 4 alphabetic characters. See page 8. |
| 3!c | Mandatory element that identifies the message number. This element has a fixed length of 3 alphanumeric characters. |
| 3!n | Mandatory element that identifies a variant. This element has a fixed length of 3 numeric characters. |
| 2!n | Mandatory element that identifies the version number. The version number is always incremented starting from version 01. This element has a fixed length of 2 numeric characters. |

Example message identifier structure

| Structure | Example |
|----------------------|--------------------|
| [MX] 4!a.3!c.3!n.2!n | MX pacs.001.001.01 |

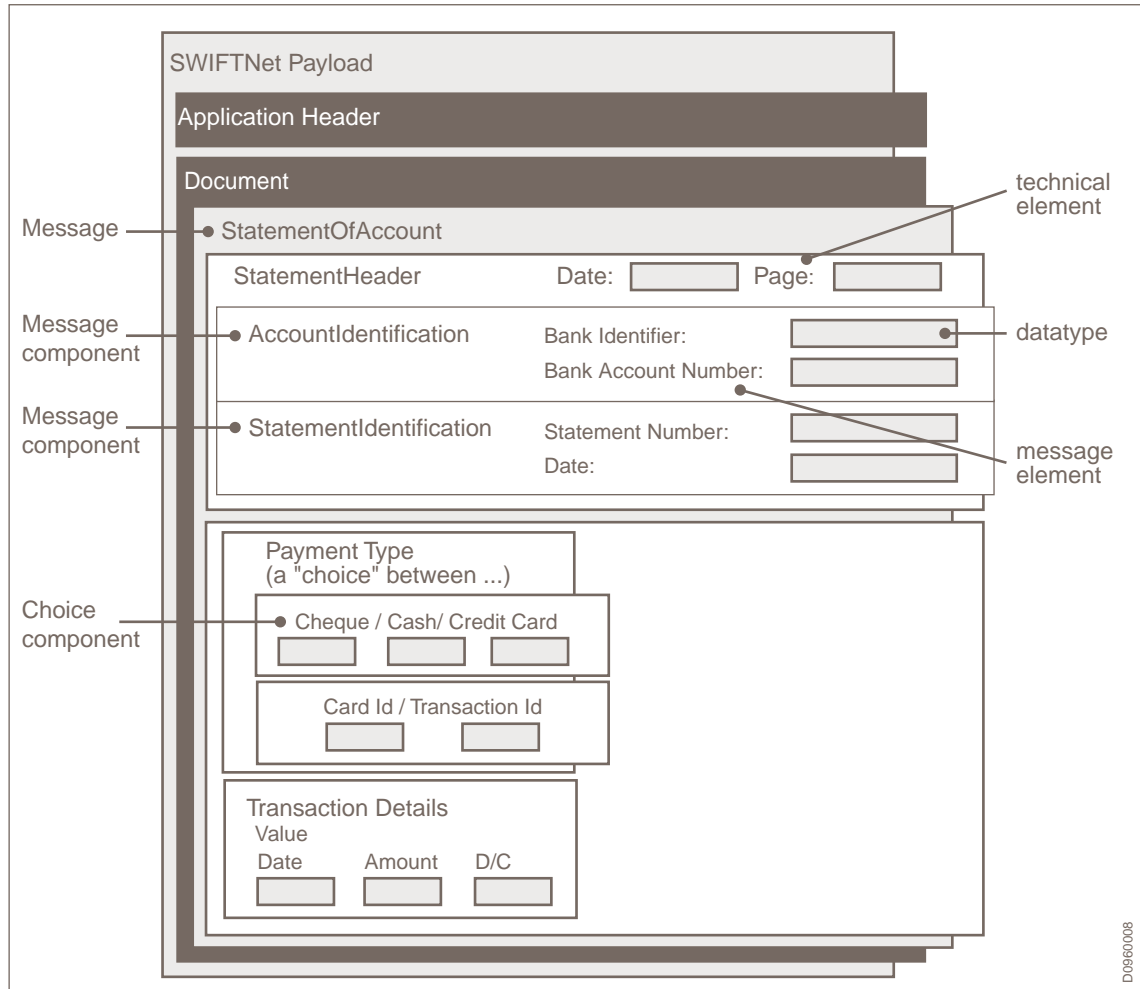
1.5 Message Definition

1.5.1 Message Structure

Structure of a message

A message definition represents the formal description of the MX structure. The structure is composed of message components, choice components, and message elements. Each MX has a unique identifier in the SWIFT Standards domain.

Example StatementOfAccount message structure



The StatementOfAccount message example shows the message structure.

Message components

| Definition | Function |
|-------------------|---|
| Message item | Messages are composed of message items. A message item can be either a message building block at the first level of a message or a message element. A message item can appear at any level of nesting in the message definition. The XML schema of the message defines the message items. Message items appear as XML tags in the message instance. |
| Message component | Reusable item that is used for assembling a message. Message components contain specific message elements that are normally linked to a business |

| Definition | Function |
|--------------------------|--|
| | component. The financial dictionary contains a complete list of message components. |
| Message building block | Message item that is specific to a particular message and not defined in the financial dictionary. A message building block is defined in the XML schema as an immediate child of a message. |
| Choice component | Represents a choice between several possibilities. An instance of a message can only contain one of the possibilities. |
| Message element | Characteristic of a message component. Each message element is uniquely identified in its message component and linked to a business element. Either a SWIFT Standards-defined data type or a message component types a message element. |
| Technical element | Message element that only has a function in the context of a message. It can have a data type but is not related to a business element. |
| Data type | Specifies the values of a business element or a message element. |
| Data type representation | Group of similar data types. |

Related information

"XML Schema" on page 18

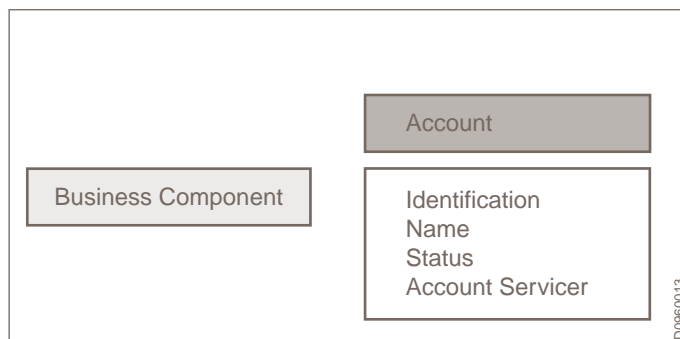
1.5.2 Message Component Reusability

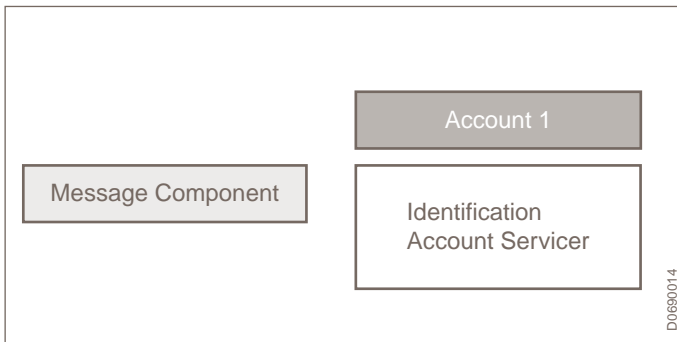
Links between message components and business components

There is a link between the various message components and the underlying business components and elements. This link promotes the reuse of components and enable customers to assess the impact of updates and changes to messages. Business concepts are defined by means of reusable business components.

Example of a message component derived from a reusable business component

This example shows that the message component Account1 is derived from the generic reusable business component Account. Account1 contains a subset (Identification and Account Service) of the four elements that characterise an Account.





Related information

"SWIFT Standards Financial Dictionary" on page 16

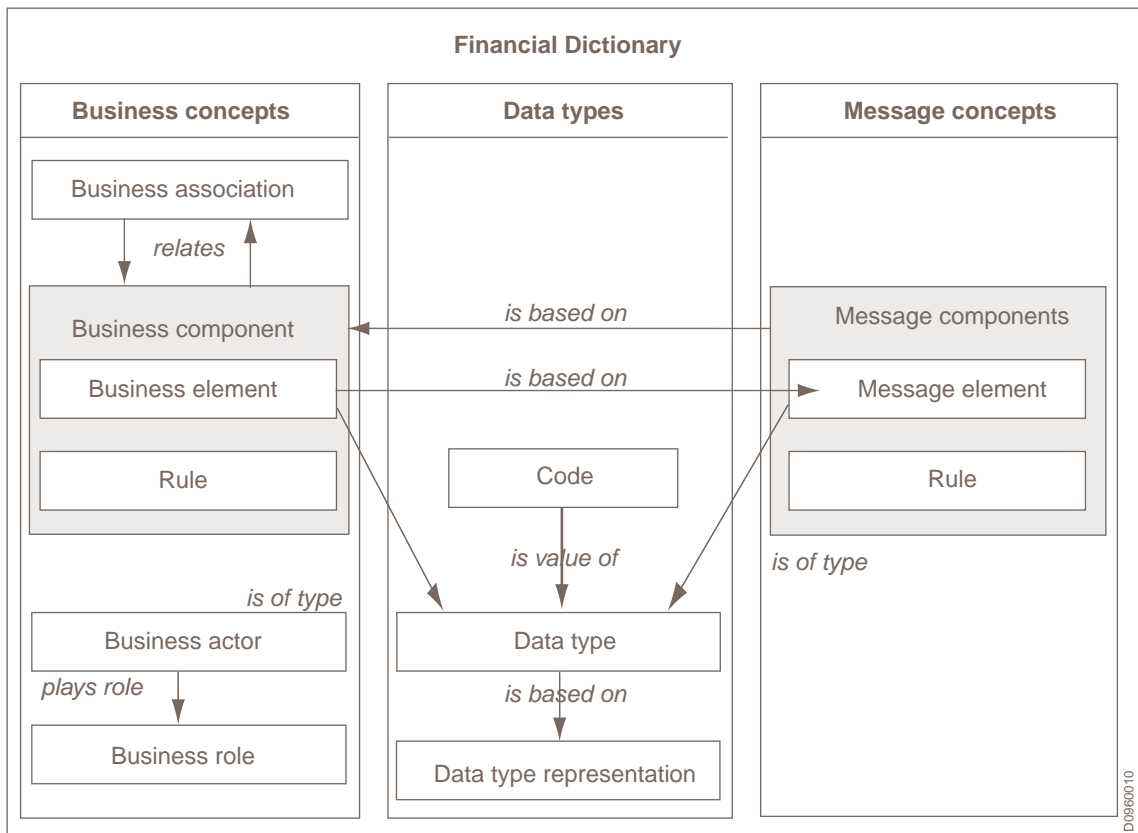
1.6 SWIFT Standards Financial Dictionary

Dictionary items

The items in the *SWIFT Standards Financial Dictionary* can be classified in the following three categories: business concepts, data types, and message concepts.

The following diagram shows the conceptual structure of the *SWIFT Standards Financial Dictionary*. This diagram shows all the types of dictionary items and the relationships between them.

The SWIFT Standards Financial Dictionary



| | | |
|------------------|--|---|
| Business concept | Dictionary item with a business semantic meaning | The following items are part of this category: business associations, business components, rules, business elements, and business actors/roles. |
| Data type | Dictionary item that unambiguously specifies a set of values for a business element or a message element | Data types are grouped into representation classes by characteristics that are common to a number of data types. |
| Message concept | Dictionary item used for message definitions | The following items are part of this category: message components, rules, and message elements. |

Related information

For more information about international financial message standards, see www.iso20022.org.

2 XML for Standards MX Messages

Introduction to Standards MX messages

This section contains information about the following topics:

- a general introduction to the XML language
- a description of the "XML Schemas and Instances"
- an overview of the "XML Syntax"
- a description of the "XML Document Validation"
- a list of the "SWIFT Standards Supported Character Sets and Encoding Scheme"
- an overview of the function of "xs:any"

2.1 Introduction to XML

Overview

SWIFT uses eXtensible Markup Language (XML) to represent its Standards MX messages (MXs). XML is a standard of the WorldwideWeb Consortium (W3C). XML is a formal, machine-processable language understood by computers. XML facilitates data exchange by ensuring that data is defined clearly and interpreted in the same way by both the sending and the receiving parties. This enables users to transport structured information using XML. XML can also specify the structure of the message, for example, the sequence and format of fields.

2.2 XML Schemas and Instances

2.2.1 XML Schema

Functions of an XML schema

The XML schema is a formal, machine-processable language used for defining, describing and constraining the structure, contents, and semantics of XML documents. An XML document that conforms with its corresponding schema is known as a valid XML instance of the schema.

Standards MX schema

The MX schema defines the formal MX structure including the data and values allowed in an MX instance.

Standards XML Schema namespaces

There are several namespace declarations used in the XML schema.

- The targetNamespace

The URN consists of the following parts:

- A fixed part that consists of the URN of the namespace for ISO documents:
urn:iso:std:iso:2002:tech:xsd or URN of the namespace for SWIFT documents:
urn:swift:xsd.
- A part unique to each schema: **xsd:[BusinessService\$]MessageDefinitionIdentifier** where BusinessService with its "\$" separator is optional and specifies the service that is supporting the underlying business transaction.

Note The **BusinessService\$** is no longer specified for new services and will be removed from older services when they are next maintained. The schemas are decoupled from the service in which they are used. Most XML schemas only contain the **MessageIdentifier**.

- The XML schema namespace, that uses the namespace prefix 'xs'
- The namespace to which all MX elements and types belong. This is the default namespace, which is the same as the TargetNamespace.

Example of namespace declarations in an MX schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="urn:iso:std:iso:2002:tech:xsd:camt.007.002.01" elementFormDefault="qualified" targetNamespace="urn:iso:std:iso:2002:tech:xsd:camt.007.002.01">
```

Example MX Schema snippet

```
...
<xs:complexType name="CaseAssignment">
  <xs:sequence>
    <xs:element name="Id" type="Max35Text"/>
    <xs:element name="Assignr" type="AnyBICIdentifier"/>
    <xs:element name="Assigne" type="AnyBICIdentifier"/>
    <xs:element name="CreDtTm" type="ISODatetime"/>
  </xs:sequence>
</xs:complexType>
...
```

MX message tree structure

SWIFT Standards documentation represents MX schema content, properties, and hierarchy as a structure called the message tree structure.

This structure resembles the Format Specifications Table for SWIFT Standards MT messages (MTs).

| Index | Or | Message Item | <XML Tag> | Mult. | Represent./ Type | Rule/ Guid. No. |
|-------------|----|--------------------------------|---------------------|--------|------------------|-----------------|
| <u>1.0</u> | | GroupHeader | <GrpHdr> | [1..1] | | |
| <u>1.1</u> | | MessageIdentification | <MsgId> | [1..1] | Text | |
| <u>1.2</u> | | CreationDateTime | <CreDtTm> | [1..1] | Date Time | |
| <u>1.3</u> | | BatchBooking | <BtchBookg> | [0..1] | Indicator | |
| <u>1.4</u> | | NumberOfTransactions | <NbOfTx> | [1..1] | Text | |
| <u>1.5</u> | | ControlSum | <CtrlSum> | [0..1] | Quantity | |
| <u>1.6</u> | | TotalInterbankSettlementAmount | <TtlIntrBkSttlmAmt> | [0..1] | Amount | |
| <u>1.7</u> | | InterbankSettlementDate | <IntrBkSttlmDt> | [0..1] | Date Time | |
| <u>1.8</u> | | SettlementInformation | <SttlmInf> | [1..1] | | |
| <u>1.9</u> | | SettlementMethod | <SttlmMtd> | [1..1] | Code | |
| <u>1.10</u> | | SettlementAccount | <SttlmAcct> | [0..1] | ± | |

2.2.2 XML Instance

Relationship between schemas and instances

An MX Schema describes what the allowed, valid MX instances can be. An MX instance refers to the XML Schema to which it complies.

Example of an extract from the AdditionalPaymentInformation message instance

```
...  
<Assgnmt>  
<Id>CUSTA20050003</Id>  
<Assignr>CUASGB2L</Assignr>  
<Assigne>AAAAGB2L</Assigne>  
<CreDtTm>2006-07-21T08:35:30</CreDtTm>  
</Assgnmt>  
...
```

2.3 XML Syntax

2.3.1 XML Tags and Data

XML Tags

XML tags are used in nested pairs, with matching start-tags and end-tags that enclose pieces of data. These are called XML elements (see "Elements" on page 20). The tags define the structure and meaning of the data that they enclose. XML tags are case sensitive and are written using abbreviated upper CamelCase.

SWIFT Standards XML tags

SWIFT uses abbreviated XML tags in schemas and instances.

Example abbreviated XML tags

```
<FinInstnId>  
<ClrSysMmbld>
```

2.3.2 XML Elements

Elements

An XML instance or document contains data in elements and nested elements corresponding to the hierarchy imposed by the XML schema.

Naming conventions

An XML element is named according to the following rules:

- The name can contain letters, numbers, and other characters, but must not start with a number or punctuation mark.
- The name must not start with XML, xml, or Xml.
- The name must not contain any spaces.

MX naming conventions

There are some generic naming rules that apply to most items in the database.

- The names of all items in the database use the upper CamelCase convention, as follows:
 - Each word starts with a capital letter.
 - There are no white spaces between words.
- A name may be made up of multiple words, each consisting of alphanumeric characters.
- Words use British English vocabulary.
- All names must start with an alphabetic character.
- All characters that follow the first characters must be letters or numbers.

MX message element properties

An MX message element can specify cardinality using `minOccurs` `maxOccurs`. `minOccurs=1` is the default value, and `maxOccurs=unbounded` is the unlimited value. This corresponds to the message item multiplicity or the cardinality.

Example of an optional message element

```
<xs:element name="StrtNm" type="Max70Text" minOccurs="0" maxOccurs="1"/>
```

Example in an instance

```
<StrtNm>OxfordStreet</StrtNm>
```

2.3.3 XML Element Types

XML element types

All message elements have a type. The XML schema describes the XML types. XML types define the content and value of XML elements. Types can be either complex, simple, or primitive. Simple types are based on primitive types.

SWIFT Standards XML complex types

Complex types are specified in the schema by `complexType`. Complex types provide the structure of an XML element.

Each message component or choice component corresponds to a complex type in the MX schema.

A SWIFT Standards-defined complex type uses the following XML schema features:

- sequence
- choice

Example of a complex type that corresponds to a message component in a schema extract

This schema extract shows a complex type that corresponds to a message component in the *SWIFT Standards Financial Dictionary*.

```

<xs:complexType name="CaseAssignment">
  <xs:sequence>
    <xs:element name="Id" type="Max35Text"/>
    <xs:element name="Assgnr" type="AnyBICIdentifier"/>
    <xs:element name="Assgne" type="AnyBICIdentifier"/>
    <xs:element name="CreDtTm" type="ISODatetime"/>
  </xs:sequence>
</xs:complexType>

```

This message component is represented in the *SWIFT Standards Financial Dictionary*, as follows.

| Message Component Details | | | |
|--|----------------------------------|--|----------------------------------|
| Repository Repository: R 4.9.6.2 - 09 October 2006 Simple Repository Search Advanced Dictionary Search Advanced Catalog Search Print friendly | | | |
| M-Component: CaseAssignment Registration Status: Registered | | Is based on: InvestigationCaseAssignment | |
| Definition Represents the assignment of a case to a party. Assignment is a step in the overall process of managing a case. | | | |
| M-Element | Type | Defined in | Is based on |
| Identification [1..1] | Max35Text | CaseAssignment | Identification |
| Assigner [1..1] | AnyBICIdentifier | CaseAssignment | Assigner |
| Assignee [1..1] | AnyBICIdentifier | CaseAssignment | Assignee |
| CreationDateTime [1..1] | ISODatetime | CaseAssignment | CreationDateTime |

Example of a complex type that corresponds to a choice component in a schema extract

This complex type corresponds to a choice component in the *SWIFT Standards Financial Dictionary*.

The property **choice** indicates that only one of the following elements can appear in an instance:

```

<xs:complexType name="AccountIdentificationChoice">
  <xs:sequence>
    <xs:choice>
      <xs:element name="IBAN" type="IBANIdentifier"/>
      <xs:element name="BBAN" type="BBANIdentifier"/>
      <xs:element name="UPIC" type="UPICIdentifier"/>
      <xs:element name="DmstAcct" type="SimpleIdentificationInformation"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

This choice component is represented in the *SWIFT Standards Financial Dictionary*, as follows.

| Message Component Details | | | |
|--|--------------------------------|--|--|
| Repository Repository: R 4.9.6.2 - 09 October 2006 Simple Repository Search Advanced Dictionary Search Advanced Catalog Search Print friendly | | | |
| M.Component: AccountIdentificationChoice | | Is based on: AccountIdentification | |
| Registration Status: Provisionally Registered | | | |
| Definition Unique identifier of an account, as assigned by the account servicer. | | | |
| M-Element | Type | Defined in | Is based on |
| {Or IBAN [1..1] | IBANIdentifier | AccountIdentificationChoice | IBAN |
| Or BBAN [1..1] | BBANIdentifier | AccountIdentificationChoice | BBAN |
| Or UPIC [1..1] | UPICIdentifier | AccountIdentificationChoice | UPIC |
| Or) DomesticAccount [1..1] | Max35Text | AccountIdentificationChoice | ProprietaryIdentification |
| M-Elements using this M-Component as a Type: 4 | | | Appearing in |
| AccountIdentification | | | AccountCashEntryDetails |
| AccountIdentification | | | AccountReport |
| AccountIdentification | | | NostroAccountCashEntrySearch |
| AccountIdentification | | | NostroAccountSearchCriteria |

XML schema-defined primitive types

An XML schema-defined primitive type species the set of data and the set of operations that can be performed on the data specified in the schema.

XML Schema Data Types are encoded as defined by the WorldWideWeb Consortium (W3C), www.w3.org.

Standards MX defined data types

Standards MX defined data types have the following characteristics:

- They use primitive data types.
- They are grouped into representation classes.
All the data types in the representation class inherit the characteristics of that representation class.
- They can be constrained using facets and by rules.
- They are globally unique across MXs.
- They have names that are suffixed by the representation class name, for example `Max128Text`.

SWIFT-supported XML schema data types

| XML name | Used by SWIFT Standards | Description |
|----------|-------------------------|---|
| string | Yes | A finite sequence of UTF-8 characters. |
| boolean | Yes | Has a value space for Boolean constants <code>TRUE</code> or <code>FALSE</code> . |
| decimal | Yes | Has precise decimal numbers. |
| date | Yes | The date in ISO 8601 format. Format: <code>CCYY-MM-DD</code> Example: 10 June 2006 is <code>2006-06-10</code> |

| XML name | Used by SWIFT Standards | Description |
|-----------|-------------------------|---|
| time | Yes | <p>The time in ISO 8601 format.</p> <p>The recommended format is the following: ⁽¹⁾HH:MM:SSZ.</p> <p>Example: 12:34:23Z</p> |
| dateTime | Yes | <p>Corresponds to a date and a time in ISO 8601 format. The letter T separates the date and time. hh represents the hours, mm represents the minutes, and ss represents the seconds. Additional digits can be used to increase the precision of fractional seconds if required. The format ss.sss... with any number of digits after the decimal point is supported, but the fractional seconds are optional. Other parts of the lexical form are not optional.</p> <p>This dateTime representation can be immediately followed by a ⁽¹⁾Z to indicate Coordinated Universal Time (UTC). The time zone is represented as the difference between the local time and UTC immediately followed by a plus or minus sign (+ or -). It is then followed by the difference from UTC represented as hh:mm.</p> <p>Note: If the time zone is included, then both hours and minutes must be present.</p> <p>See the ISO 8601 Date and Time Formats for details about the legal values used in the various fields.</p> <p>In general, the order relation on dateTime is a partial order since there is an ambiguous relationship between certain instants. For example, there is no determinate ordering between:</p> <ul style="list-style-type: none"> a. 2000-01-20T12:00:00 b. 2000-01-20T12:00:00Z <p>Based on time zones currently in use, (c) can vary from 2000-01-20T12:00:00+12:00 to 2000-01-20T12:00:00-13:00.</p> <p>Format: ⁽¹⁾CCYY-MM-DDTHH:MM:SSZ+- offset to UTC</p> <p>This is the recommended format: CCYY-MM-DDTHH:MM:SSZ</p> <p>Example: 2006-11-01T11:34:00Z</p> |
| duration | No | <p>A period of time in ISO 8601 format.</p> <p>Format: PnYnMnDTnHnMnS</p> |
| gMonthDay | No | <p>A set of 1-day long, annually periodic instances (for example, a birthday).</p> <p>The time zone must be UTC.</p> <p>Format: --MM-DD</p> <p>Example: 25 August is --08-25</p> |
| gMonth | No | <p>A specific month from midnight on the first day till midnight on the Last Day.</p> <p>Format: --MM--</p> <p>Example: February is --02--</p> |
| gYear | Yes | <p>A time period that starts at the midnight that starts the first day of the year. This time period ends at the midnight that ends the Last Day of the year.</p> <p>This is a set of one-year long, non-periodic instances.</p> <p>Format: CCYY</p> |

| XML name | Used by SWIFT Standards | Description |
|--------------|-------------------------|---|
| | | Example: 2006 |
| gDay | No | A set of 1-day long, monthly periodic instances. The time zone must be UTC. Format: ---DD Example: 29th is ---29 |
| gYearMonth | Yes | A specific month in a specific year in the UTC time zone. Format CCYY-MM Example: February 2006 is 2006-02 |
| base64Binary | No | Base64-encoded arbitrary binary data. |

(1) The Z indicates a Universal Time Code (UTC) time. +hh:mm or +hh indicates a local time ahead of UTC. -hh:mm or -hh indicates a local time behind UTC.

How to use dates and times

Users can select from several different data types to express a time. Examples of these data types are ISODateTime, ISOTime, and ISODate.

- **UTC time**

This is a moment in time expressed as Co-ordinated Universal Time (UTC). If a user opts to use UTC time, then the user generates an unambiguous timestamp.

| Data type | Format | Example |
|-------------|--------------------------|----------------------|
| ISOTime | hh:mm:ss.sssZ | 13:20:00.5Z |
| ISODateTime | YYYY-MM-DDThh:mm:ss.sssZ | 2007-10-10T17:00:00Z |

- **Local time and UTC offset**

This is a moment in time expressed in local time with either a plus (+) or a minus (-). A plus (+) indicates a positive time difference from UTC. A minus (-) indicates a negative time difference from UTC. If a user opts to use a combination of local time and UTC offset, then the user generates an unambiguous timestamp.

| Data type | Format | Example |
|-------------|---------------------------------|-------------------------------|
| ISOTime | hh:mm:ss.sss+/-hh:mm | 13:20:00-05:00 |
| ISODateTime | YYYY-MM-DDThh:mm:ss.sss+/-hh:mm | 2007-10-10T12:00:00.001-05:00 |

- **Local time**

This is a moment in time expressed in local time without an explicit indication of the time zone. If a user opts to use local time, then the user generates an ambiguous timestamp unless all the parties involved know about the time zone.

| Data type | Format | Example |
|-------------|-------------------------|---------------------|
| ISOTime | hh:mm:ss.sss | 15:28:30.05 |
| ISODateTime | YYYY-MM-DDThh:mm:ss.sss | 2007-09-17T16:30:00 |

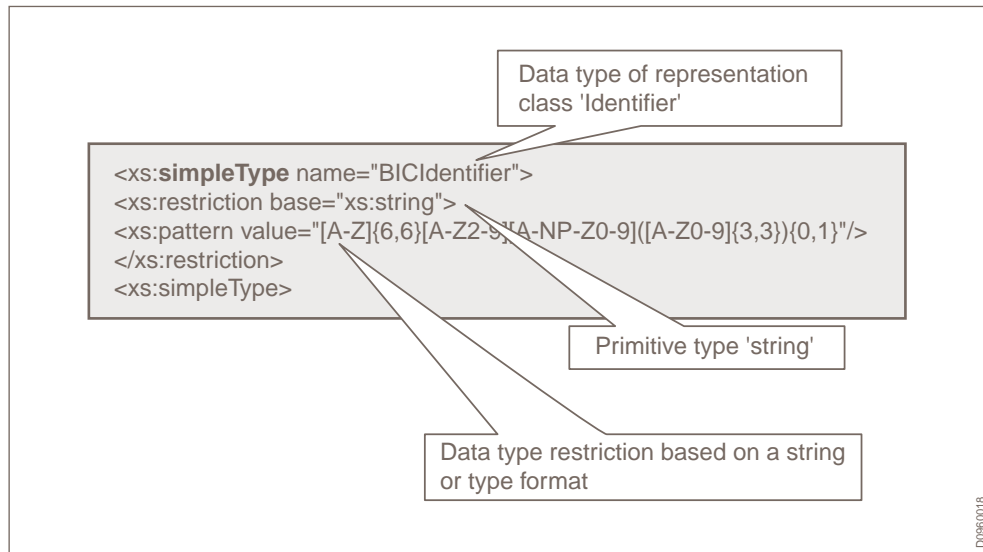
-
- Note**
1. 00:00:00 is the beginning of a calendar day.
24:00:00 is the end of a calendar day.
 2. Users can include decimal fractions of a second. All the parties involved must agree the number of digits that are allowed.
-

Data type representation class

SWIFT Standards data types are grouped into representation classes. Representation classes group characteristics that are common to a number of data types. All the data types that use that representation class inherit the characteristics of that representation class.

| Class name | Definition | Data type |
|------------|---|--|
| Identifier | A character string used to distinguish uniquely one instance of an object within an identification scheme from all other instances of the objects within the same scheme. | string |
| Code | A list of all coded values for a particular field and the meanings they represent. | string |
| Indicator | 2 alternative values that indicate a condition. | boolean |
| Text | A finite set of characters. | string |
| Quantity | A number of non-monetary units. | decimal |
| Amount | A number of monetary units specified in a currency where the currency is explicit or implied. | decimal |
| DateTime | Describes a time measurement. For more information about <code>dateTime</code> formats, see "SWIFT-supported XML schema data types" on page 23. | date time dateTime gDay gMonth gYear gMonthday duration |
| Rate | A quantity or amount specified with respect to another quantity or another amount, or a fixed or appropriate charge, cost, or value. | decimal |

Example of an MX schema extract that shows a SWIFT Standards-defined data type



Example of a Standards MX defined data type in the SWIFT Standards Financial Dictionary

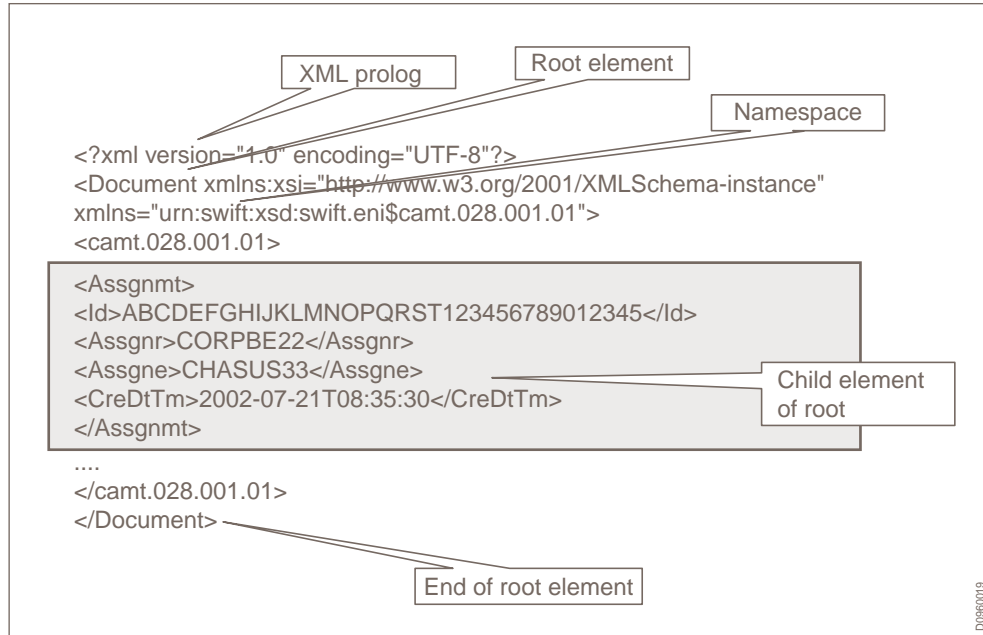
| Data Type Details | |
|--|--|
| Repository Repository: R 4.9.6.2 - 09 October 2006 Simple Repository Search Advanced Dictionary Search Advanced Catalog Search Print friendly | |
| Data Type: BICIdentifier | Representation: Identifier XML Type: string |
| Registration Status: Registered | |
| Definition Bank Identifier Code. code allocated to a financial institution by a Registration Authority under an international identification scheme, as described in the latest edition of the international standard ISO 9362 "Banking - Banking telecommunication messages - Bank identifier codes". | |
| Property | Value |
| IdentificationSchemeName | SWIFT; BICIdentifier |
| XML Facet | Value |
| pattern | [A-Z]{6,6}[A-Z2-9][A-NP-Z0-9]([A-Z0-9]{3,3}){0,1} |
| Rule | Description |
| * ALG BIC | Valid BICs are registered with the ISO 9362 Registration Authority, and consist of eight (8) or eleven (11) contiguous characters comprising the first three or all four of the following components: BANK CODE, COUNTRY CODE, LOCATION CODE, BRANCH CODE. The bank code, country code and location code are mandatory, while the branch code is optional. |
| Business Elements using this Data Type: 3 | Defined in |
| BIC | FinancialInstitutionIdentification |
| BIC | OrganisationIdentification |
| VoteThroughNetwork | MeetingNotice |
| Message Elements using this Data Type: 55 | Defined in |
| AccountOwner | LimitIdentificationDetails1 |

Example of message element typed by the BICIdentifier data type in an MX message instance

```
<BIC>CHASUS33C</BIC>
```

2.3.4 XML Document

SWIFT Standards XML document or MX instance example



XML prolog

The first line of the document or message instance defines the XML version and the character encoding used in the document. This declaration is not an XML element and not part of the XML document or message instance. The XML prolog does not have a closing tag.

Root element

All XML documents must have a root element. The root element is at the top of the XML element hierarchy. The root element of an XML instance is `<Document>`. The last line of the document defines the end of the root element `</Document>`.

Namespace

A namespace is a container that provides context for the XML elements and XML attributes that it qualifies and allows XML elements and attributes that have the same name to be easily distinguished.

| Namespace prefix | Definition |
|---|--|
| <code>xsi</code> | The XML schema instance namespace. |
| absent (default) or defined by the Sender | The namespace to which all XML elements of this MX instance belong that are defined in the corresponding MX Schema. It defines all XML elements and attributes that do not belong to the <code>xsi</code> namespace. |

2.4 XML Document Validation

Message validation

MX message validation is performed in the following two parts:

- schema validation
- extended validation

2.4.1 Schema Validation

Schema validation

Schema validation is performed in two parts:

- the **MX instance** is well-formed XML
- the **MX instance** is valid

Well-formed XML

The schema verifies that the document is well formed. An XML instance or XML document with correct syntax is called a well-formed XML document.

Well-formed XML has the following characteristics:

- Each XML document must be a single-rooted hierarchical structure of elements.
- All elements are correctly paired.
- The element opening tag and closing tag are the same.
- Element names must start with an alphabetic character. Subsequent characters may be alphanumeric.

Examples of well-formed XML

| Well-formed | Not well-formed | Option |
|-----------------|-----------------|---|
| <Ref>1234</Ref> | | |
| | <Ref>1234<Ref> | end tag / is missing |
| | <R@f>1234</R@f> | element name contains illegal character @ |

Valid schema

The schema checks the validity of the XML instance. An instance validated against its corresponding schema is called a valid XML instance.

MX structure

The schema validates the following items:

- the MX structure starting from the root element (top level element)
- the allowed elements, and the order in which they must be used (sequence)
- simple XML element relations, for example, parent-child and siblings

- the cardinality of the XML elements, for example, mandatory or optional, or the number of occurrences allowed
- choices among XML elements

XML element content

The schema verifies the allowed content of the following elements:

- the allowed values within the elements
- the allowed format
- the allowed characters (character set)

2.4.2 Non Schema Validation

Extended validation

Extended validation ensures that Standards MX messages are validated to the same standards as Standards MT messages. Extended validation can be performed when the XML schema does not validate the document completely. A user can also decide to perform some validation tasks separately, for example, Business Identifier Code (BIC) validation.

Rules for extended validation

| Type of validation rule | Example |
|----------------------------------|--|
| Cross-element | The element Entry Date must only be present if entry status has the value booked. |
| Intra-element | The fractional part of an amount is checked against its currency. |
| Calculations or derived elements | The comparison of 2 dates or the comparison of 2 amounts. International Bank Account Number (IBAN) validation of the check digits. For more information about IBANs, see "Example of calculation validation" on page 30. |
| External tables | Currency codes (ISO 4217). Country codes (ISO 3166). For more information about country codes, see "Example of external table validation" on page 31. Business Identifier Code (BIC) |

Example of calculation validation

| | |
|------------|---|
| Definition | An identifier used internationally by financial institutions to identify the account of a customer at a financial institution. This is described in the ISO 13616 standard, Banking, and related financial services International Bank Account Number (IBAN). |
| Format | [a-zA-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,20} |
| Rules | IBAN A valid IBAN consists of all 3 components, the Country Code, check digits and Basic Bank Account Number (BBAN). (fatal) error code: sw.Stds.D000003 |
| Example | AT611904300234573201 |

Example of external table validation

| | |
|------------|--|
| Definition | This code identifies a country, a dependency, or another area of particular geopolitical interest. This code is based on the list of country names obtained from the United Nations (ISO 3166 alpha-2 code). |
| Format | [A-Z]{2,2} |
| Rules | Country The code is checked against the list of country names obtained from the United Nations (ISO 3166. Alpha-2 code). (fatal) error code: Sw.Stds.D00004 |
| Example | BE |

Non-validated rules

It is not always possible to validate some rules. These rules are known as non-validated rules. A non-validated rule usually occurs when the infrastructure does not have enough information to be able to validate the rule correctly. This does not mean that these rules cannot be followed.

An example of non-validated rules is inter-document rules. XML schema validation and extended validation only cover intra-document validation. An example of an inter-document rule that cannot be validated is checking that relevant references are passed in all messages relating to the same transaction.

SWIFT only maintains a limited set of code and identifier lists. Most other identifier lists are referred to by the standard. In these cases, SWIFTNet cannot verify that a value (such as ISIN, CUSIP, SEDOL, or RIB) taken from such list, is correct.

Guidelines

An MX can also have guidelines in addition to validated and non-validated rules.

Important Unlike validated and non-validated rules, guidelines are not mandatory.

Guidelines cannot be imposed in the same way as rules because there are rare cases when a guideline cannot be followed. For example, there is a guideline that advises a user to use an International Securities Identification Number (ISIN). This is a guideline not a rule because some securities do not have an ISIN or are not uniquely identified. This also applies to the guideline to use a BIC to identify a party. There are some financial institutions or business entities that do not have a BIC.

2.4.3 Error Codes

Generic error codes

The errors that are produced during XML schema validation generate and display generic error codes. Any payload can generate these errors.

Note The XML schema validation engine uses third-party software to generate error codes. SWIFT recommends that customers do not include these error codes in any customer programs since the error codes may change without notice.

Possible instance validation errors

The placeholders in these error messages ("_TypeName", "_Value_", "_ElementName_") must be replaced with values specific to the error context at runtime.

For the latest list of generic error codes, see the *SWIFTNet Link Error Codes*.

MX or message component-specific errors

Refer to the message description in the relevant *Standards MX Message Reference Guide* for the description and error code for a specific error.

2.5 SWIFT Standards Supported Character Sets and Encoding Scheme

2.5.1 SWIFT Standards Supported Character Sets

Character sets list

The default rule for MX-supported characters and languages is that the user must always use English and Basic Latin, except if a closed user group is set up in which other principles are agreed. This mechanism allows any user to decide whether the user wants to be part of the closed user group or not. It reduces the risk that local languages / character sets are sent *by accident* to a wrong destination. If the service description of a closed user group does not specify language or character set, then it means that the user must **only** use English and Basic Latin within that closed user group. If the user must use other characters (for example, Cyrillic), then this must be explicitly mentioned within the service description of that closed user group.

The following table includes the most relevant characters used in Basic Latin, compared to the SWIFTNet FIN character sets X, Y and Z.

Basic Latin character set

| MT | | | MX | | |
|----|---|---|-------------|-----------|---|
| X | Y | Z | Basic Latin | Character | Description |
| * | | * | * | a - z | 26 lower case letters of the Latin alphabet |
| * | * | * | * | A - Z | 26 upper case letters of the Latin alphabet |
| * | * | * | * | 0 - 9 | 10 numerical characters |
| * | * | * | * | / | slash |
| * | * | * | * | - | hyphen |
| * | * | * | * | ? | question mark |
| * | * | * | * | : | colon |
| * | * | * | * | (| open parentheses |
| * | * | * | * |) | close parentheses |
| * | * | * | * | . | full stop |
| * | * | * | * | , | comma |
| * | * | * | * | ' | apostrophe |
| * | * | * | * | + | plus |
| * | * | * | * | | space |
| | * | * | * | = | equals |

| MT | | | MX | | |
|----|---|---|-------------|-----------|----------------------|
| X | Y | Z | Basic Latin | Character | Description |
| | * | * | * | ! | exclamation mark |
| | * | * | * | " | quotation mark |
| | * | * | * | % | percent |
| | * | * | * | And | ampersand |
| | * | * | * | * | asterisk |
| | * | * | * | < | less than |
| | * | * | * | > | more than |
| | * | * | * | ; | semi-colon |
| | | * | * | @ | at |
| | | * | * | # | hash |
| | | * | * | \$ | dollar |
| * | * | | * | { | open curly bracket |
| * | * | | * | } | close curly bracket |
| * | * | * | * | CR | carriage return |
| * | * | * | * | LF | line feed |
| | | | * | [| left square bracket |
| | | | * |] | right square bracket |
| | | | * | \ | back slash |
| | | | * | _ | under score |
| | | | * | ^ | circumflex |
| | | | * | ` | grave accent |
| | | | * | | vertical line |
| | | | * | ~ | tilde |
| | | | * | | control character |

Unicode text-based data types

This is a nonexhaustive list of the data types that currently support the Unicode character set.

| | | | |
|-----------|------------|------------|--------------|
| Max3Text | Max16Text | Max128Text | Max1025Text |
| Max6Text | Max35Text | Max140Text | Max2000Text |
| Max8Text | Max70Text | Max256Text | Max20000Text |
| Max10Text | Max105Text | Max350Text | Max1MText |

Use of normalised characters

When an application creates an XML instance, characters must not be represented as character entities. Except for the five mandatory characters which must be represented as shown in the following table.

| Special character | XML representation |
|-------------------|--------------------|
| " | " ; |
| ' | &apos ; |
| < | < ; |
| > | > ; |
| & | & ; |

Important This method for handling special characters applies irrespective of whether the full Unicode character set, or only the restricted Basic Latin character set, is used.

Even characters that are shorter when represented as a character entity must use UTF-8 encoding. This includes all the UTF-8 encoded characters that use more than 4 bytes. For example, `CrLf` must be normalised into `L`. (For more information, see the *XML 1.0 specification, paragraph 2.11.*)

Example of normalised characters

XML entities are stored in computer files which are organised into lines. Typically these lines use a combination of the carriage-return (`#xD`) characters and line-feed (`#xA`) characters to separate each line.

These end-of-line characters can be normalised. The 2-character sequence `#xD #xA`, and any `#xD` that is not followed by `#xA`, are translated into a single `#xA` character.

| Action performed | Character sequence | Normalised to |
|----------------------------|----------------------|------------------------------|
| carriage-return, line-feed | <code>#xD #xA</code> | <code>#xA</code> (line-feed) |
| carriage-return | <code>#xD</code> | <code>#xA</code> (line-feed) |

2.5.2 Use of the UTF-8 Encoding Scheme

UTF-8 encoding

UTF-8 encoding is a technical implementation in XML schemas that can specify almost any character. The schema is based on Unicode and the Universal Character Set (ISO 10646). UTF-8 encoding represents any Unicode character (or any character of the Universal Character Set) as a sequence of bytes.

Important XML instances based on MX schemas must be encoded in UTF-8.

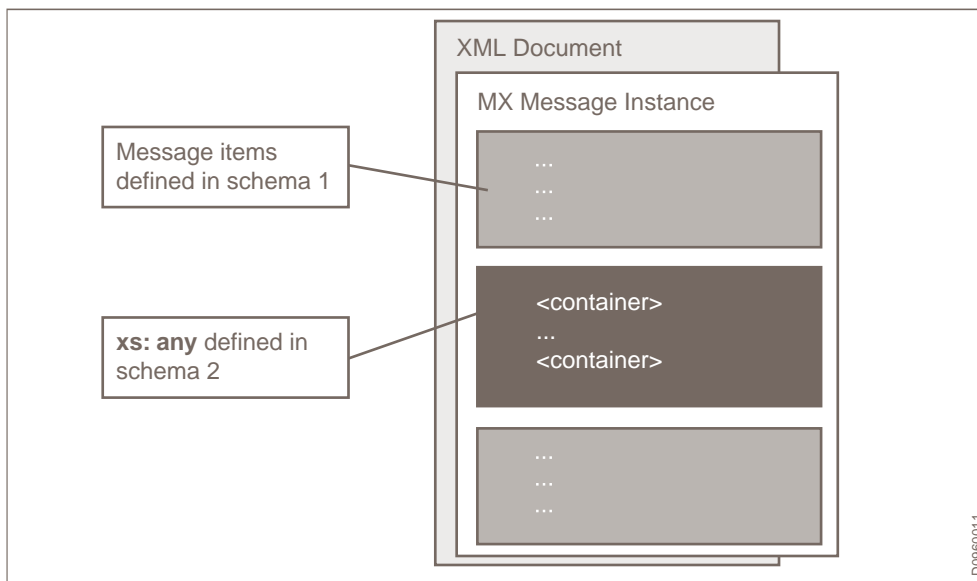
2.6 xs:any

Function of xs:any

The XML schema function `xs:any` allows Standards MX message instances to contain additional information while retaining as much control as possible over the content. Customers can use this function to insert a copy of a message into another message.

When the `xs:any` schema function is used, there are two XML schemas. The parent schema must be an MX schema. The parent schema declares the open content by using `xs:any`. The child schema describes the contents of `xs:any`.

Parent and child schemas



A valid XML document must be valid for both the parent schema (for the part outside the `<any>`) and the child schema (for the part inside the `<any>`). The parent schema can use the parameter `processContents` to set the validation level inside `xs:any`. For more information about the `processContents` parameter, see "processContents parameter" on page 35.

Namespace parameter

The value of the `namespace` parameter specifies the type of element content permitted in the namespace. The message instance must specify either one namespace from a list of namespaces, or a predefined parameter value.

| Namespace parameter value | Permitted contents of the element |
|--------------------------------|---|
| <code>##any</code> | Any well-formed XML from any namespace. This is the default parameter value. |
| <code>##other</code> | Any well-formed XML from a namespace other than the target namespace of the type being defined. Unqualified elements are not allowed. |
| <code>##targetNamespace</code> | Any well-formed XML belonging to any namespace in list. <code>##targetNamespace</code> represents the target namespace being defined. |

processContents parameter

The parent schema can use the parameter `processContents` to set the validation level for `xs:any`.

| processContents parameter value | Effect on validation |
|---------------------------------|--|
| <code>strict</code> | The sender must specify the schema used by <code>xs:any</code> in the XML document. If the receiver cannot locate the schema specified in the document, then the document fails validation. |
| <code>lax</code> | Any <code>xs:any</code> elements that are defined in the matching schema are validated. Elements in the parent schema that are not defined in the matching schema are allowed to pass through unvalidated. |
| <code>skip</code> | <code>xs:any</code> content is not validated. |

Extensible messages with Straight-Through Processing (STP)

If the `processContents` parameter is set to `strict`, then the receiver of the message can understand and process the extension. This specifies that only XML schemas known to the receiver can be used.

Extensible message with less or without STP

If the `processContents` parameter is set to `lax` or `skip`, then the receiver may or may not be able to process the document. Documents with XML schemas unknown to the receiver still pass validation.

3 MX Messages and InterAct

Introduction to the MX structure

This section contains information about the following topics:

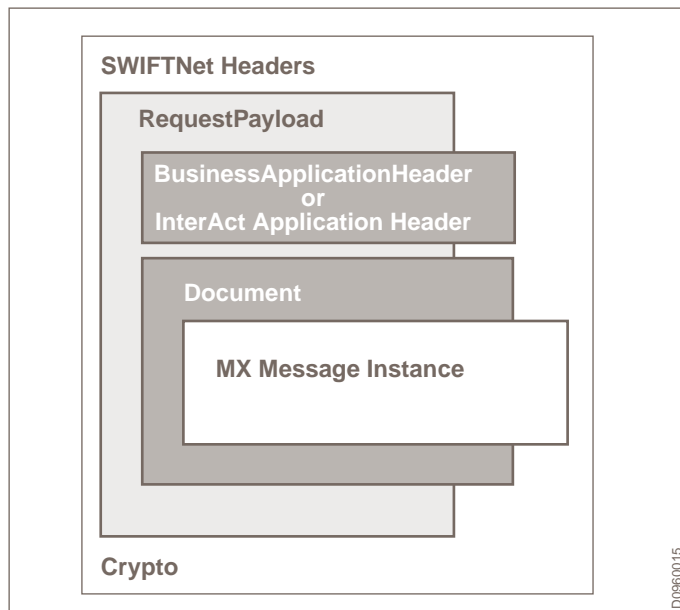
- structure and functions of the business application header with some typical scenarios that show how to use them to route messages
- the structure of an MX message using the InterAct messaging service
- more details of the MX message instance using the InterAct messaging service

Related information

"The Core SWIFTNet Messaging Services" in the *SWIFTNet Service Description*
SWIFTNet Messaging Operations Guide

3.1 Message Structure

MX message blocks using InterAct



| | |
|---------------------------|--|
| SWIFTNet Headers | Among others, SWIFTNet headers include <code>RequestControl</code> , <code>RequestResponse</code> , and <code>RequestHeader</code> . |
| RequestPayload | The <code>RequestPayload</code> consists of the business application header and the document (MX message instance). |
| BusinessApplicationHeader | The SWIFTNet InterAct ApplicationHeader or the ISO 20022 BusinessApplicationHeader. |
| XML Document | The XML document corresponds to the message instance (message body). The XML schema defines the XML document. The XML document has a maximum payload of 100 000 bytes in store-and-forward mode. |
| Crypto | <code>Crypto</code> contains any message signature and verification information. |

3.2 Message Headers

Introduction

This section contains general information applicable to any MX. For more information about the specific usage of the application header in the context of a service, see the relevant service description or message usage guidelines.

What are headers?

Applications use headers to manipulate messages. A header enables an application to process messages regardless of their content.

There can be several headers, each addressing a different application that processes part of the message:

- **Business application header**

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

- **Network headers**

The network headers contain information that is relevant for the networking application transporting the messages.

- **Security header**

The security header contains information that is relevant to the application in charge of the encryption or the signature of the messages.

ISO 20022 business application header or InterAct application header?

The ISO 20022 business application (BAH) header is replacing the InterAct application header (`AppHdr`) mandated by SWIFTNet.

The deployment principles for the BAH are as follows:

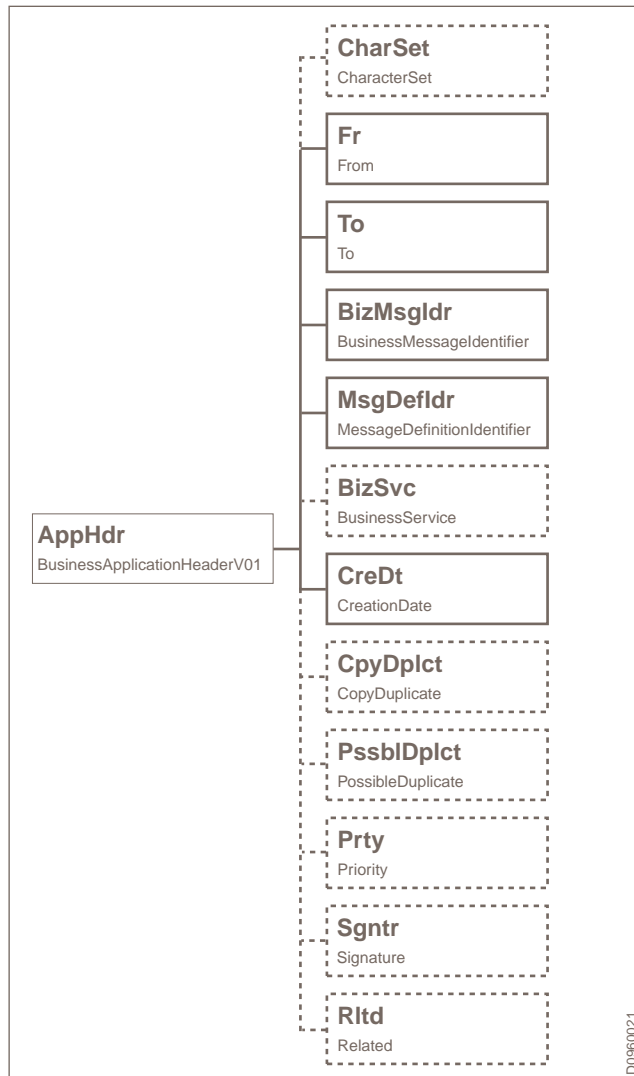
- The BAH will be deployed on a service per service basis.
- The decision on when a service will start using the new BAH will be driven by the user community of that service.

Note For existing SWIFT Solutions, a duplication of information between the business application header and the MX message is possible. In a future release of the MX messages set, the relevant message elements will be removed to avoid any duplication.

3.2.1 ISO 20022 Business Application Header

Structure of the ISO 20022 business application header

The business application header is the first element of the `RequestPayload` element. The payload contains the document which contains the business document. This may be an ISO 20022 message or a SWIFT proprietary message.



Components of the ISO 20022 business application header

| Header element | Syntax | Purpose |
|-------------------------|-----------|--|
| CharacterSet | <CharSet> | The additional character set(s) that some of the text-based elements may contain. Example: Basic Latin |
| SendingBusinessEntity | <Fr> | The sending business application that has created this business message for the receiving business application that will process this business message. |
| ReceivingBusinessEntity | <To> | The Business Application designated by the sending Business Application to be the recipient who will ultimately process this Business Message. Note the receiving Business Application might be different from |

| Header element | Syntax | Purpose |
|-----------------------------|--------------|--|
| | | the receiving address potentially contained in the transport header (as defined in the transport layer). |
| BusinessMessageIdentifier | <BizMsgIdr> | The identification of the business message, unique to the SendingBusinessEntity. Example: ref0001. |
| MessageDefinitionIdentifier | <MsgDefIdr> | The identification of message definition. Example: camt.001.001.01 |
| BusinessService | <BizSvc> | The business service (SWIFT's Solutions or non-SWIFT service) agreed between the two business applications under which rules this business message is exchanged. To be used when there is a choice of processing services or processing service levels. Example for a SWIFT's Solution such as Exceptions and Investigations: swift.eni |
| CreationDateTime | <CreDt> | The date and time when this business message (header) was created. Times must be normalised, using the Z annotation. |
| CopyDuplicate | <CpyDplct> | Indicates whether the business message is a copy, a duplicate or a copy of a duplicate of a previously sent business message. |
| PossibleDuplicate | <PssblDplct> | Flag indicating if the business message exchanged between the business applications is possibly a duplicate. If the receiving MessagingEndpoint did not receive the original, then this business message should be processed as if it were the original. If the receiving MessagingEndpoint did receive the original, then it should perform necessary actions to avoid processing this business message again. This guarantees business idempotent behaviour. |
| Priority | <Prty> | Relative indication of the processing precedence of the message over a (set of) business messages with assigned priorities. |
| Signature | <Sgntr> | The digital signature of the Business Entity authorised to sign this business message. |
| RelatedReference | <Rltd> | The Business Application Header of the original business message |

Related information

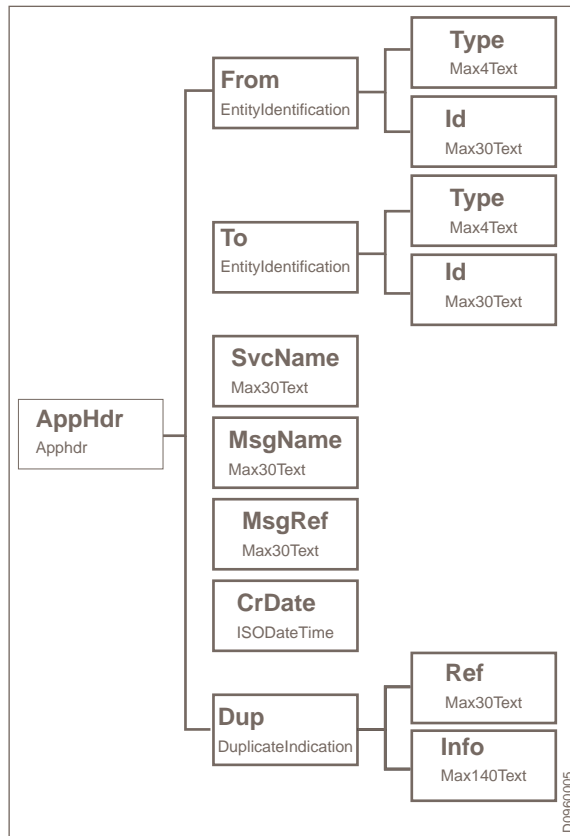
The ISO 20022 BusinessApplicationHeader XML schema, the Message Definition Report and the Message Usage Guide are available at iso20022.org > [Catalogue of ISO 20022 Messages](#).

The ISO 20022 BusinessApplicationHeader enriched XML schema and MS Excel representation are part of the Standards Developer Toolkit. For more information about this product, see swift.com > Products & services > Standards > Tools & Services > [Standards Developer Toolkit](#)

3.2.2 InterAct Application Header

Structure of the InterAct application header

The application header is the first element of the `RequestPayload` element. The payload contains the document which contains the business document. This may be an ISO 20022 message or a SWIFT proprietary message.



Components of the InterAct application header

| Header element | Syntax | Purpose |
|--|-----------|--|
| Business application from which the document is received | <From> | Identifies the application that has created the document. |
| Business application to which the document is sent | <To> | Identifies the receiving application for which the document is created. |
| Service name | <SvcName> | Identifies the SWIFTNet service to which the message belongs. |
| Message identifier | <MsgName> | A unique structured identifier that identifies the message. |
| Message reference | <MsgRef> | The sending application defines this unique identifier for the message. |
| Creation date | <CrDate> | Date and time at which the message was created. |
| Duplicate | <Dup> | Used when the sending application has already tried to send the document to the receiving application. |

The InterAct application header schema

```

<?xml version = "1.0" encoding = "UTF-8"?>
<xs:schema xmlns = "urn:swift:xsd:$ahV10"
  targetNamespace = "urn:swift:xsd:$ahV10"
  xmlns:xs = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified">
  <xs:element name = "AppHdr" type = "AppHdr"/>
  <xs:complexType name = "AppHdr">
    <xs:sequence>
      <xs:element name = "From" type = "EntityIdentification" minOccurs = "0"/>
      <xs:element name = "To" type = "EntityIdentification" minOccurs = "0"/>
      <xs:element name = "SvcName" type = "Max30Text" minOccurs = "0"/>
      <xs:element name = "MsgName" type = "Max30Text" minOccurs = "0"/>
      <xs:element name = "MsgRef" type = "Max30Text"/>
      <xs:element name = "CrDate" type = "ISODateTime"/>
      <xs:element name = "Dup" type = "DuplicateIndication" minOccurs = "0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name = "EntityIdentification">
    <xs:sequence>
      <xs:element name = "Type" type = "Max4Text"/>
      <xs:element name = "Id" type = "Max30Text"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name = "DuplicateIndication">
    <xs:sequence>
      <xs:element name = "Ref" type = "Max30Text"/>
      <xs:element name = "Info" type = "Max140Text"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name = "Max140Text">
    <xs:restriction base = "xs:string">
      <xs:maxLength value = "140"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name = "Max30Text">
    <xs:restriction base = "xs:string">
      <xs:maxLength value = "30"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name = "Max4Text">
    <xs:restriction base = "xs:string">
      <xs:maxLength value = "4"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name = "ISODateTime">
    <xs:restriction base = "xs:dateTime"/>
  </xs:simpleType>
</xs:schema>

```

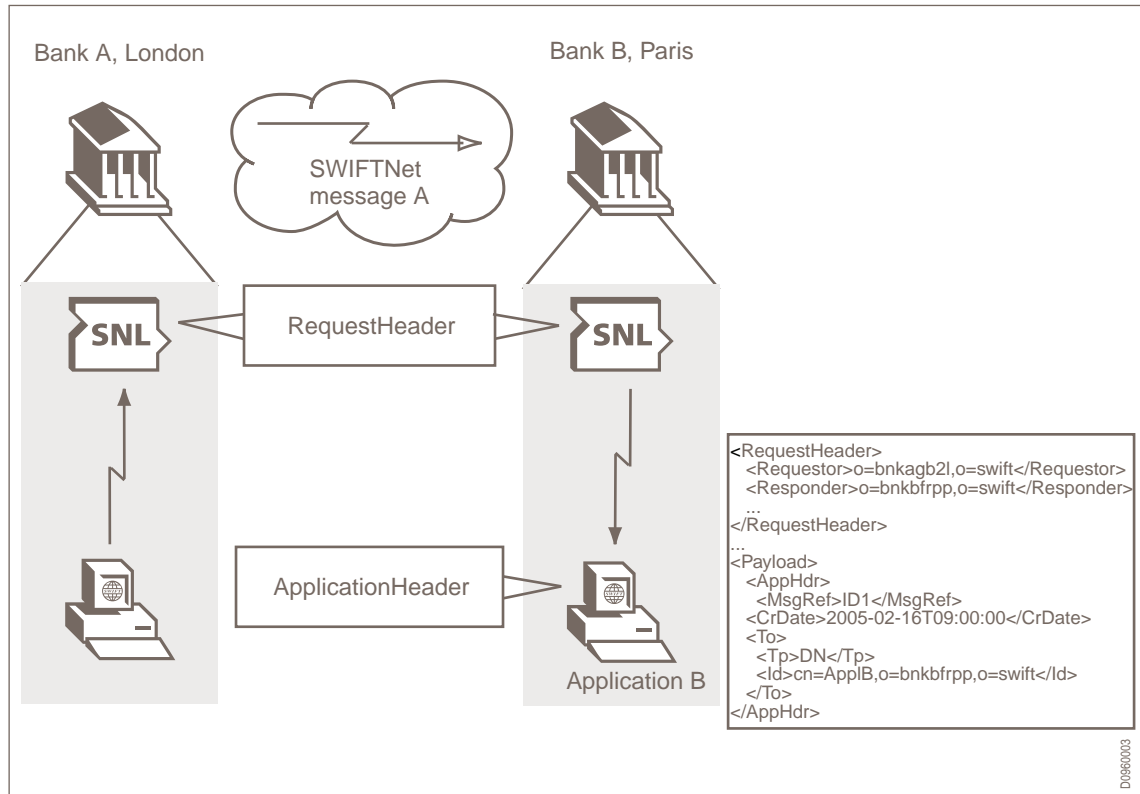
InterAct application header scenarios

This section contains a number of examples of how the application header is used to route an MX in different scenarios, as follows:

- "Scenario one: the default application header".
- "Scenario two: a SWIFTNet service bureau header".
- "Scenario three: a Member-Administered Closed User Group header".
- "Scenario four: a relay header".

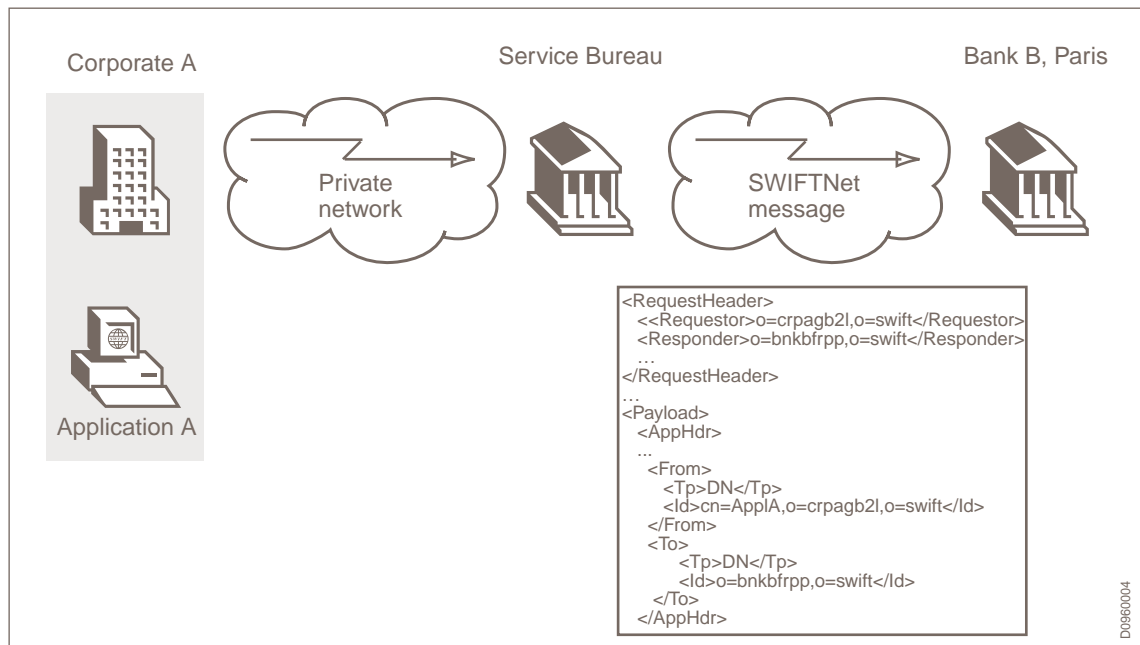
Scenario one: the default application header

A generic messaging application, at bank A in London, sends an MX to business application B, at bank B in Paris.



Scenario two: a SWIFTNet service bureau header

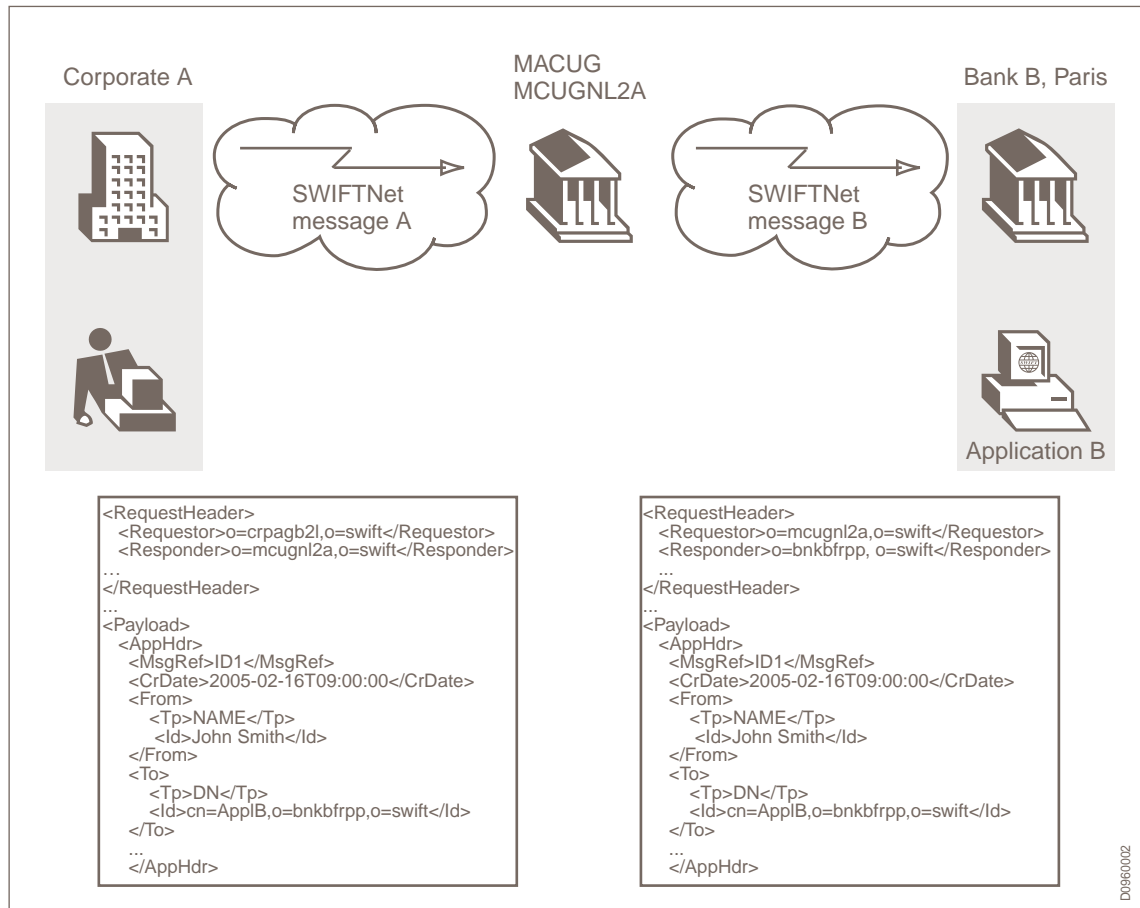
Application A, at corporation A, sends an MX to bank B in Paris. Corporation A is not directly connected to SWIFTNet, so it sends the MX through a service bureau.



Scenario three: a Member-Administered Closed User Group header

Corporate A is part of a Member-Administered Closed User Group administered by MCUGNL2A. A person in corporate A sends a message to application A, at bank B, in Paris. MCUGNL2A only forwards the contents of the <Payload> part of the message without processing it. <Payload> remains unchanged between corporate A and bank B. The application header contains no reference to MCUGNL2A. MCUGNL2A does appear in the request header.

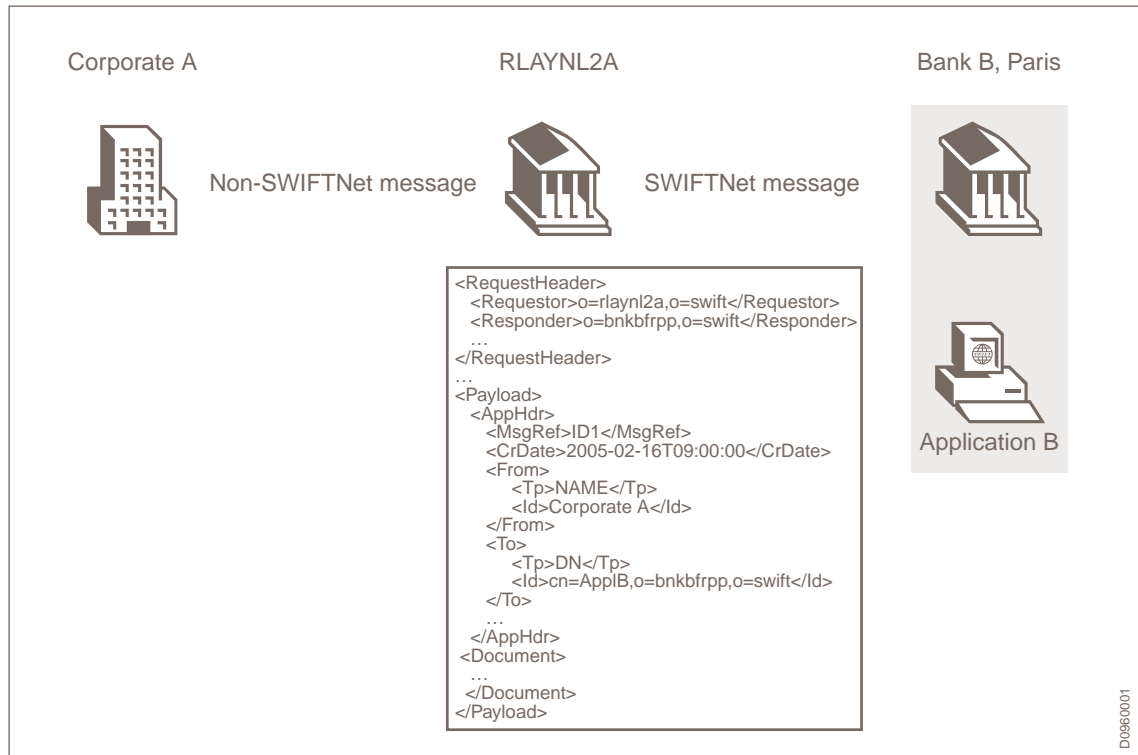
Note The application header has several formats that identify a business entity: DN, BIC, and NAME.



D0960002

Scenario four: a relay header

Corporate A sends a message to application B, at bank B, in Paris. Corporate A is not connected to SWIFTNet, so it sends the message through RLAYN2A which is a relay institution. RLAYN2A forwards the contents of the <Payload> section of the message. <Payload> remains unaltered between corporate A and bank B. Corporate A creates the application header. The relay institution does not change the application header.



3.2.3 Correspondence between the ISO Header and the InterAct Header

Mapping table

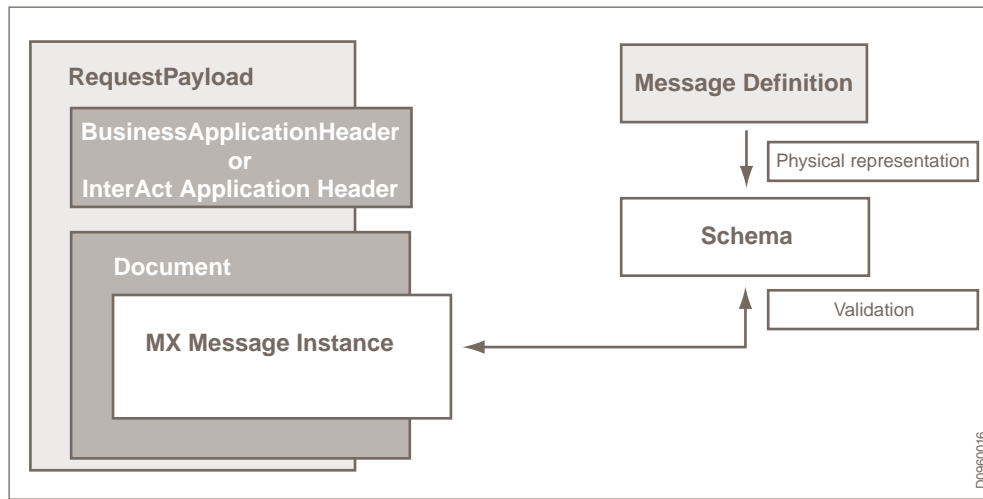
| ISO BusinessApplicationHeader | InterAct AppHdr |
|-------------------------------|-----------------|
| <CharSet> | - |
| <Fr> | <Fr> |
| <To> | <To> |
| <BizMsgIdr> | <MsgRef> |
| <Msg DefIdr> | <MsgName> |
| <BizSvc> | <SvcName> |
| <CrDate> | <CrDate> |
| <CpyDplct> | - |
| <PssblDplct> | <Dup> |
| <Prty> | - |

| ISO BusinessApplicationHeader | InterAct AppHdr |
|-------------------------------|-----------------|
| <Sgntr> | - |
| <Rltd> | - |

3.3 Message Instance

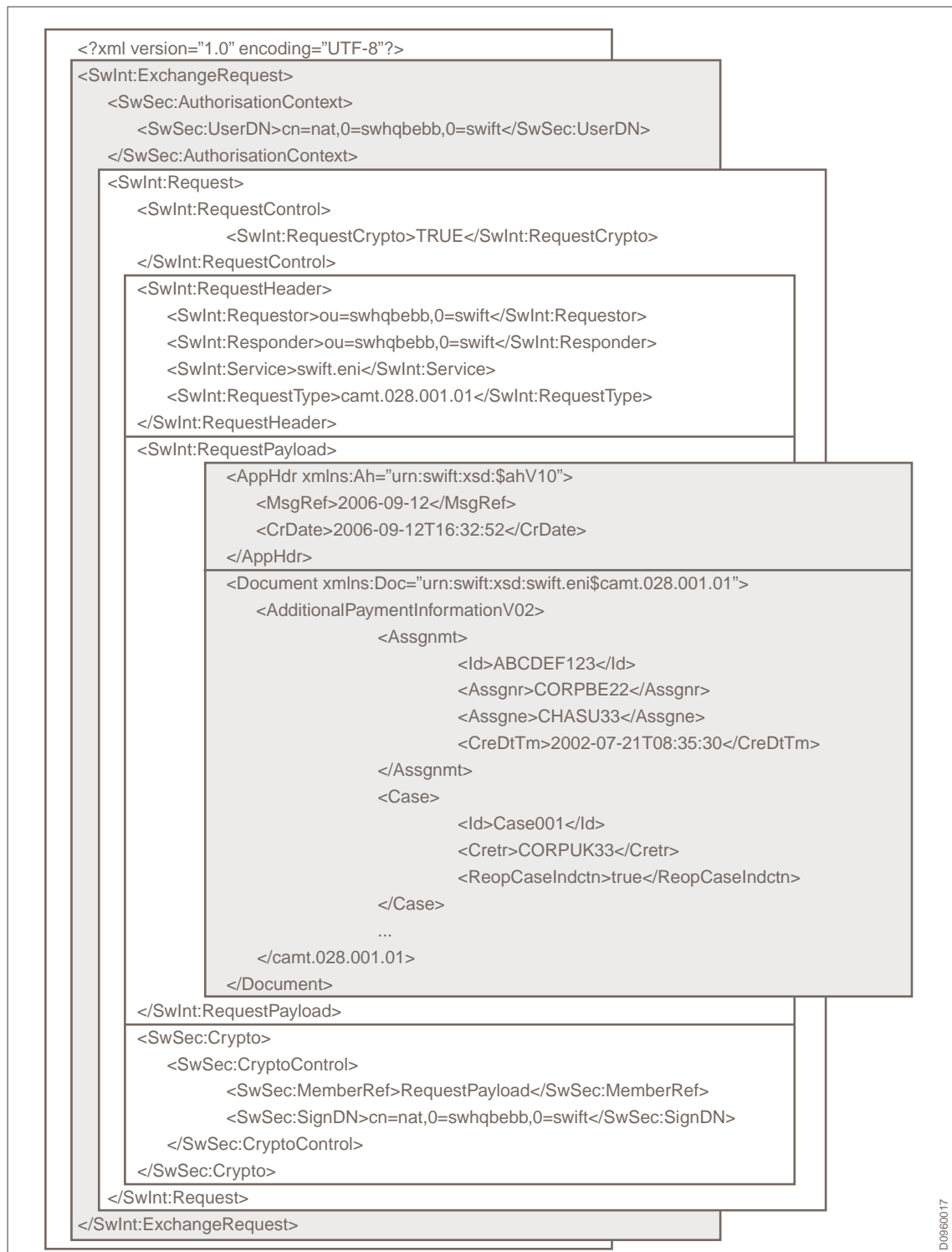
Message definition, schema, and instance

The following diagram shows the link between a message definition, the message schema, and the message instance.



This diagram shows an extract from the AdditionalPaymentInformation message camt . 028 . 001 . 01 sent with InterAct

Example of a camt.028.001.01 message sent in the scope of the SWIFT Exceptions and Investigations solution with InterAct:



D0960017

4 MX Messages and FileAct

Introduction to the FileAct message structure

SWIFTNet uses dedicated XML structures in the header of the FileAct file transfer primitives to allow customers to specify key parameters. This also allows the SWIFTNet central systems or the receiver's applications, or both to act upon them.

This section contains the following information:

- the structure of an MX message using the FileAct messaging service
- the structure and functions of the FileAct HeaderInfo field

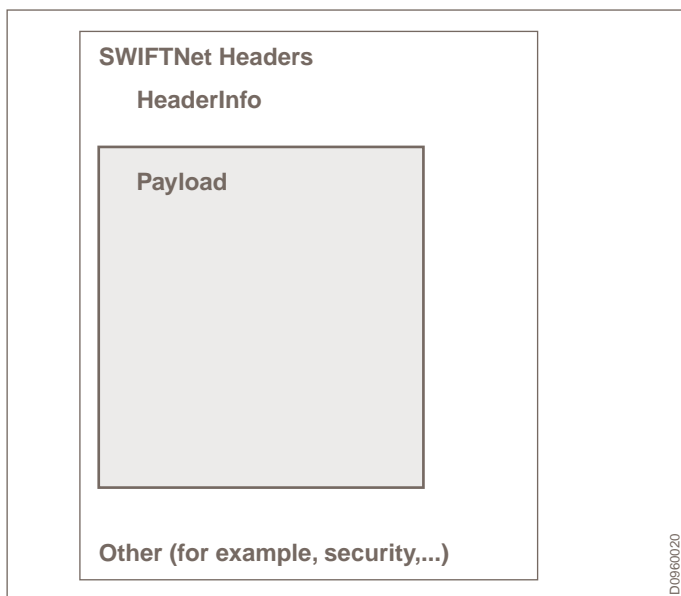
Related information

"The Core SWIFTNet Messaging Services" in the *SWIFTNet Service Description SWIFTNet Messaging Operations Guide*

4.1 File Structure

FileAct message structure

A message sent using the FileAct messaging service is either referred to as a **FileAct message** or a **File**.



| | |
|------------------|---|
| SWIFTNet Headers | Among others, SWIFTNet headers include information elements that are necessary to perform the file transfer (for example requestor, responder, or service). |
| HeaderInfo | The <code>HeaderInfo</code> is an optional XML field of the SWIFTNet headers. |
| Payload | The content of the file transfer. |
| Other | Other elements such as message signature and verification information. |

4.2 File Header

Introduction

The FileAct HeaderInfo field may be structured according to several service profiles. The term *service profile* describes the rules that apply in the context of a specific SWIFT Solution. This includes the documentation and XML schemas used in the HeaderInfo XML structure.

The following common service profiles are available:

- Service profile 1 - Transaction Count
- Service profile 2 - Payment File Summary

Other service profiles specific to SWIFT Solutions are available. For more information, see the relevant service description.

4.2.1 Service Profile 1 - Transaction Count

Customer requirements

Customers implement this data structure when there is a need to identify only the number of items (for example, payments) in a file. SWIFT can require this for example to benefit from a pricing per transaction offered in some SWIFTNet FileAct-based services.

Technical requirements

Applications that send files must obtain the number of transactions in each file and specify this in the TotalNumberOfTransactions data element for each file.

Data elements

The following elements and characteristics are available in this service profile.

| Element Name | Cardinality | XML Tag | Data type | Description | Comment |
|---------------------------|-------------|--------------|--------------------|---|--|
| TotalNumberOfTransactions | [1..1] | <TtlNbOfTxS> | Max15 Numeric Text | Total number of individual transactions contained in the file | SWIFT also centrally uses this field to apply the pricing. |

Example

```
<ApplSpfcfc xmlns="urn:swift:xsd:ApplSpfcfc.TxsCntr.01">
  <TxsCntr>
    <TtlNbOfTxS>12345678</TtlNbOfTxS>
  </TxsCntr>
</ApplSpfcfc>
```

Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns="urn:swift:xsd:ApplSpfcfc.TxsCntr.01" xmlns:xs="http://
www.w3.org/2001/XMLSchema" targetNamespace="urn:swift:xsd:ApplSpfcfc.TxsCntr.
01" elementFormDefault="qualified">
<xs:element name="ApplSpfcfc" type="ApplSpfcfc" />
  <xs:complexType name="ApplSpfcfc">
    <xs:sequence>
      <xs:element name="TxsCntr"
type="FileTransferHeaderTransactionsCounter1" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FileTransferHeaderTransactionsCounter1">
    <xs:sequence>
      <xs:element name="TtlNbOfTxs" type="Max15NumericText" />
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="Max15NumericText">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{1,15}" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

4.2.2 Service Profile 2 - Payment File Summary

Customer requirement

This profile allows a third party to the file exchange to receive a copy, for information or for authorisation, of information related to the business content and value of the file being transmitted.

The following scenarios are possible:

- the authorisation by the head office of disbursement files sent by subsidiaries (risk management)
- the authorisation by corporate treasurer of a file of salary payments sent by its HR department to the bank (confidential details not to be shared with treasury employees)
- the copy for information of the proceeds of the file for audit, tracking, or recovery purposes

The copied information needs to contain the main characteristics of the content of the file, including for example file identification, amount, value date, or number of transactions.

Note If other fields are required in the context of some SWIFT Solutions or specific situations, then the service administration will need to use (or create) a different service profile.

Technical requirement

The elements that need to be copied to the third party must be included in the FileAct HeaderInfo field because the file content itself is not copied to a third party.

It is assumed that the business sender and receiver of the file are the requestor and responder indicated in the FileAct file headers.

Business applications

The Payment File Summary service profile is available for use by communities exchanging low-value payments.

Data elements

The following elements and characteristics are available in this service profile.

| Element Name | Cardinality | XML Tag | Data type | Description | Comment |
|---------------------------|-------------|-------------------------|---------------------------|---|---|
| InterbankSettlementAmount | [1..1] | <TtlIntrBkStt lmAmt> | Currency and Amount | Total amount of money transferred between instructing | Total amount (arithmetic sum) of the Credit Transfers |

| Element Name | Cardinality | XML Tag | Data type | Description | Comment |
|---------------------------|-------------|-----------------|--------------------|---|--|
| | | | | agent and instructed agent. | (CT) or Direct Debit (DT) items, or both in the file. Payments files exchanged between the members are exclusively single currency. |
| TotalNumberOfTransactions | [1..1] | <TtlNbOfTxS> | Max15 Numeric Text | Total number of individual transactions contained in the file. | |
| CreditDebitCode | [1..1] | <CdtDbtInd> | CRDT or DBIT | Specifies whether a transaction is an increase (credit) or a decrease (debit). | This field specifies the amount as being a debit amount or a credit amount. |
| InterbankSettlementDate | [1..1] | <IntrBkSttlmDt> | ISODate | Date on which the amount of money ceases to be available to the agent that owes it and when the amount of money becomes available to the agent to which it is due. | Date on which the settlement takes place. |
| BankOperationType | [0..1] | <BkOprTp> | Max35Text | Textual information or code indicating the nature of the instrument. Usage: the code may only appear once, and is valid for the entire file. This optional element allows senders to further define the nature of the transactions by means of a code. | This code can for example specify a specific operation type used in a given community or in between specific partners (for example, SEPA AOS). |
| FileIdentification | [1..1] | <FileId> | Max35Text | Point to point reference assigned by the instructing party and sent to the next party in the chain to unambiguously identify the file. | Reference of the file, which is assigned by the sender. Usage: the instructing party has to make sure that FileIdentification is unique per instructed party for a pre-agreed period. |

| Element Name | Cardinality | XML Tag | Data type | Description | Comment |
|---------------|-------------|----------|-----------|--|---------|
| Service Level | [0..1] | <SvcLvl> | Max35Text | Proprietary identification of a pre-agreed level of service between the parties. | |

Example

```
<ApplSpfcfc xmlns="urn:swift:xsd:ApplSpfcfc.PmtSummry.01">
  <PmtSummry>
    <TtlIntrBkSttlmAmt Ccy="EUR">1912000.00</TtlIntrBkSttlmAmt>
    <TtlNbOfTxS>12345678</TtlNbOfTxS>
    <CdtDbtInd>CRDT</CdtDbtInd>
    <IntrBkSttlmDt>2007-07-14</IntrBkSttlmDt>
    <FileId>6</FileId>
  </PmtSummry>
</ApplSpfcfc>
```

Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:swift:xsd:ApplSpfcfc.PmtSummry.02" xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:swift:xsd:ApplSpfcfc.PmtSummry.02" elementFormDefault="qualified">
  <xs:element name="ApplSpfcfc" type="ApplSpfcfc"/>
  <xs:complexType name="ApplSpfcfc">
    <xs:sequence>
      <xs:element name="PmtSummry" type="FileTransferHeaderPaymentSummary2"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="CreditDebitCode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CRDT"/>
      <xs:enumeration value="DBIT"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CurrencyAndAmount_SimpleType">
    <xs:restriction base="xs:decimal">
      <xs:minInclusive value="0"/>
      <xs:fractionDigits value="5"/>
      <xs:totalDigits value="18"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="CurrencyAndAmount">
    <xs:simpleContent>
      <xs:extension base="CurrencyAndAmount_SimpleType">
        <xs:attribute name="Ccy" type="CurrencyCode" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="CurrencyCode">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z]{3,3}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="FileTransferHeaderPaymentSummary2">
    <xs:sequence>
      <xs:element name="TtlIntrBkSttlmAmt" type="CurrencyAndAmount"/>
      <xs:element name="TtlNbOfTxS" type="Max15NumericText"/>
      <xs:element name="CdtDbtInd" type="CreditDebitCode"/>
      <xs:element name="IntrBkSttlmDt" type="ISODate"/>
      <xs:element name="FileId" type="Max35Text"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
        <xs:element maxOccurs="1" minOccurs="0" name="BkOprTp"
type="Max35Text" />
        <xs:element maxOccurs="1" minOccurs="0" name="SvcLvl"
type="Max35Text" />
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="ISODate">
    <xs:restriction base="xs:date" />
</xs:simpleType>
<xs:simpleType name="Max15NumericText">
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9]{1,15}" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Max35Text">
    <xs:restriction base="xs:string">
        <xs:minLength value="1" />
        <xs:maxLength value="35" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>
```

5 Understanding the Documentation

Introduction to Standards MX documentation

This section contains information about the following topics:

- The *Standards MX Message Reference Guide* in HTML format.
The HTML format contains information for implementors and users.
- The *Standards MX Message Reference Guide* in PDF format.
The PDF format contains information for users.

This chapter describes how to navigate around the HTML and the PDF versions of a *Standards MX Message Reference Guide*. The diagrams show extracts from the *Payments Clearing and Settlement*. All the *Standards MX Message Reference Guides* share the same structure, layout, and navigation methods.

5.1 Standards MX Message Reference Guide - HTML

Introduction to the Standards MX Message Reference Guide in HTML

The *Standards MX Message Reference Guide* in HTML contains the following chapters and sections:

- Overview
- List of Messages
 - Message Scope and Usage
 - Message Structure and Contents
 - Business Examples
- Message Flow
- General Indexes
 - Index of Business Actors/Roles
 - Index of Business Information
 - Index of Synonyms
 - Index of Message Item Types
 - Data Types
 - End Points
 - Index of XML Table
 - Index of Codes

5.1.1 The Home Page

How to display the home page

Users can click the book title to display the home page.

MX Message Reference Guide home page



Users can select an option that corresponds to a profile. The option displays a subset of standards information that is appropriate for the user.

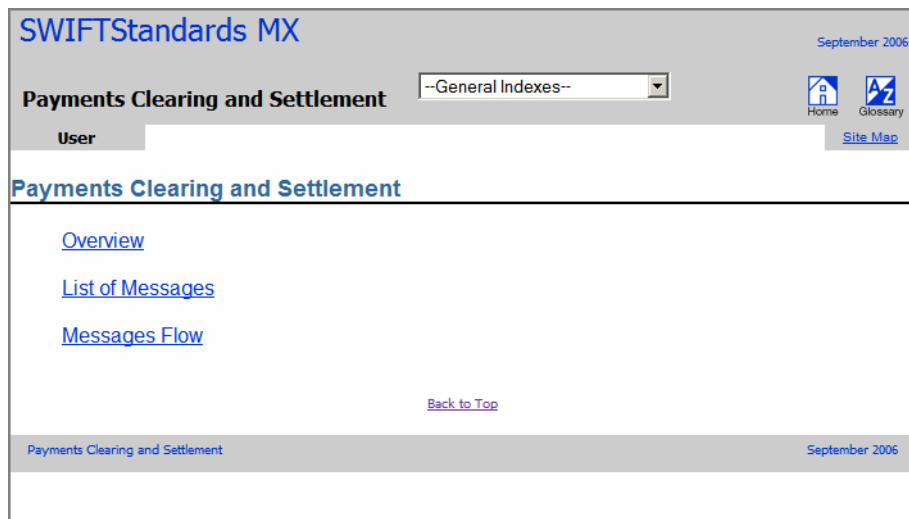
| | |
|-------------|--|
| User | This profile is designed for business analysts and operators. It describes SWIFT Standards and how to use them. |
| Implementor | This profile is designed for information providers and vendors. It contains the information available in the user profile, and additional information about how to implement the messages. |

5.1.2 User Profile

How to display the user profile main page

If the user selects the User profile, then the User profile main page screen window appears.

User profile main page screen



User profile main menu options

| | |
|------------------|--|
| Overview | Describes the scope of the full set of messages in the guide and provides some background information about the business area where the messages are used. |
| List of Messages | Provides detailed information about each message. |
| Messages Flow | Describes the sequence and relationship between messages listed in the guide. |

Top banner and navigation

The [Back](#) and [Forward](#) browser buttons move between the previous and next screens. Other navigation buttons and links are listed in table "Banner and navigation options".

Banner



Banner and navigation options

| | |
|----------|--|
| Top Bar | Displays the profile in current use. |
| Home | Returns to the main menu. |
| Glossary | Displays an overview of SWIFT Standards and explains the terms used. |
| Site Map | Provides an overview of all the topics in the guide. |

5.1.3 List of Messages

How to display the list of messages and individual message menu

If the user clicks List of Messages, then a list of the messages available appears.

Selecting one of the messages displays a window with additional information about the message. This information includes scope, structure and content, and any relevant business examples available for that message. This type of information resembles the information available for SWIFT Standards MT messages (MTs) in the *SWIFT Standards MT Message Reference Guides* of the *SWIFT User Handbook*.

List of Messages screens

The image displays two screenshots of the SWIFT Standards MX website interface. The top screenshot shows the 'List of Messages' page for 'Payments Clearing and Settlement'. It features a navigation bar with 'SWIFTStandards MX', 'September 2006', a dropdown menu set to '--General Indexes--', and icons for 'Home' and 'Glossary'. Below the navigation, there is a 'User' field and a 'Site Map' link. The main content area is titled 'List of Messages' and contains a list of message types: FIToFICustomerCreditTransferV01 (pacs.008.001.01), FIToFICustomerDirectD..., FIToFIPaymentReversa..., FinancialInstitutionCredi..., PaymentCancellationRe..., PaymentReturnV01 (pa..., and PaymentStatusReportV... The bottom screenshot shows the 'Individual View' for 'FIToFICustomerCreditTransferV01 (pacs.008.001.01)'. It has the same navigation bar as the top screenshot. The main content area is titled 'FIToFICustomerCreditTransferV01 (pacs.008.001.01)' and contains links for 'Message Scope and Usage', 'Structure and Contents', 'Business Example 1', and 'Business Example 2'. At the bottom of this view, there is a 'Back to Top' link and a footer with 'FIToFICustomerCreditTransferV01 (pacs.008.001.01)' and 'September 2006'.

List of Messages options

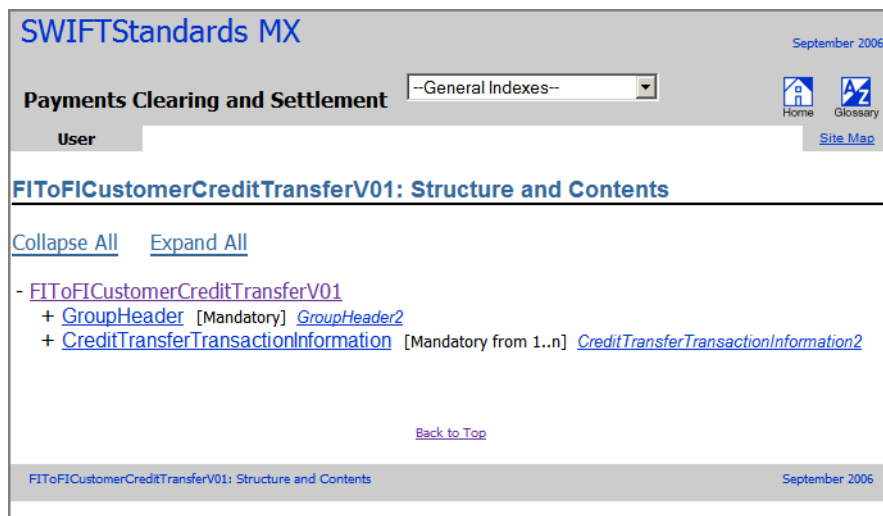
| | |
|------------------|--|
| List of Messages | Displays an overview of all the messages documented in the <i>Standards MX Message Reference Guide</i> . |
| Individual View | Displays the scope, structure, content, and relevant business examples for each message. |

5.1.4 Message Structure and Contents

How to display the message structure and contents

If the user clicks Structure and Contents display, then the screen shown in "Message structure and contents screens" appears. This screen shows how the message is structured. It is comparable to the Format Specifications tables for MTs. The screen lists the message items that comprise the message.

Message structure and contents screens



Users can expand a block in the tree structure by clicking the sign, or expand the whole tree by clicking [Expand All](#).

This displays the main message building blocks. This resembles the blocks and sequences for MTs. Each message building block contains a number of message items (like the fields in MT blocks). Each message item has a type that either a message component or a data type specifies. Message components and data types appear in *italic* in the tree structure.

Expanded Structure and Contents screen

The screenshot shows the SWIFTStandards MX website interface. At the top, it says "SWIFTStandards MX" and "September 2006". Below that, there's a navigation bar with "Payments Clearing and Settlement" and a dropdown menu set to "--General Indexes--". There are also icons for "Home" and "Glossary", and a "Site Map" link. The main content area is titled "FIToFICustomerCreditTransferV01: Structure and Contents". Below the title, there are links for "Collapse All" and "Expand All". The structure is listed as follows:

- [FIToFICustomerCreditTransferV01](#)
 - [GroupHeader](#) [Mandatory] [GroupHeader2](#)
 - [MessageIdentification](#) [Mandatory] [Max35Text](#)
 - [CreationDateTime](#) [Mandatory] [ISODateTime](#)
 - [BatchBooking](#) [Optional] [BatchBookingIndicator](#)
 - [NumberOfTransactions](#) [Mandatory] [Max15NumericText](#)
 - [ControlSum](#) [Optional] [DecimalNumber](#)
 - [TotalInterbankSettlementAmount](#) [Optional] [CurrencyAndAmount](#)
 - [InterbankSettlementDate](#) [Optional] [ISODate](#)
 - [SettlementInformation](#) [Mandatory] [SettlementInformation1](#)
 - [SettlementMethod](#) [Mandatory] [SettlementMethod1Code](#)
 - [SettlementAccount](#) [Optional] [CashAccount7](#)
 - [Identification](#) [Mandatory] [AccountIdentification3Choice](#)
 - Choice amongst
 - [IBAN](#) [Mandatory] [IBANIdentifier](#)
 - [BBAN](#) [Mandatory] [BBANIdentifier](#)
 - [UPIC](#) [Mandatory] [UPICIdentifier](#)

Message Structure and Content options

| | |
|-------------------|--|
| Message Item | The message item appears on the left-hand side of the screen. It can be either a message element or a message building block. |
| Multiplicity | <p>This indicates whether a message item is optional or mandatory and how many times the element can be repeated. The multiplicity is shown in square brackets.</p> <p>For example:</p> <ul style="list-style-type: none"> • [0..1] This shows that the element can be present either 0 time or 1 time. The element is optional and repeatable. • [1..1] This shows that the element can only be present 1 time. • [1..n] This shows that the element is always present 1 time and can be repeated n times. |
| Message Item Type | A type can be either a data type, or a message component. A message component is further composed of message items. The type is shown in <i>italics</i> . It provides information about how to complete the message item. This resembles the Content/options/format column in SWIFT Standards MT Format Specifications. |

5.1.5 Message Item Information

How to display detailed message item information

This provides a detailed description of each message item.

Detailed message item information screen

SWIFTStandards MX
September 2006

Payments Clearing and Settlement
--General Indexes--

[Home](#)
[Glossary](#)

User [Site Map](#)

Message Element:
MessageIdentification

Defined in: [GroupHeader2](#)

Definition

Point to point reference assigned by the instructing party and sent to the next party in the chain to unambiguously identify the message.

Usage: The instructing party has to make sure that 'MessageIdentification' is unique per instructed party for a pre-agreed period.

Is based on: (Technical)

XML Tag: <MsgId>

Multiplicity: [Mandatory]

Type: [Max35Text](#)

[Back to Top](#)

MessageIdentification
September 2006

Detailed message item options

| | |
|--|---|
| Message Message Building Block Message Element | Displays whether a message item is a message element, a message building block or message if it is the root item. |
| Defined In | If the item is a message element, then this displays the name of the message component to which the message element belongs. If the item is a building block, then this displays the name of the message to which the block belongs. |
| Definition | Definition of the message, message building block, or message element. |
| Is Based On | Describes the business item in the "SWIFT Standards Financial Dictionary" on which a message item is based. |
| XML Tag | Short reusable name for the message element which appears in the XML schemas and XML instances. |

| | |
|--------------|---|
| Multiplicity | <p>Indicates whether an element is optional or mandatory and how many times the element can be repeated. The multiplicity is shown in square brackets.</p> <p>For example:</p> <ul style="list-style-type: none"> • [0..1] This shows that the element can be present either 0 time or 1 time. The element is optional and repeatable. • [1..1] This shows that the element can only be present 1 time. • [1..n] This shows that the element is always present 1 time and can be repeated n times. |
| Type | <p>Provides information about how to complete the message item. A type can be either a data type, or a message component. A message component is further composed of message items. The type is shown in <i>italics</i>. It provides information about how to complete the message item. This resembles the Content/options/format column in SWIFT Standards MT Format Specifications.</p> |

5.1.6 Message Item Types

How to display details of message item types

Message item type information can be displayed from the Structure and Contents page or from the Message Items screen. A message item type can be either a data type or a message component.

For more information about types, see "XML Element Types" on page 21.

Date type screens

The image shows two screenshots of the SWIFTStandards MX website. The left screenshot displays the 'Max35Text' data type screen, and the right screenshot displays the 'AccountStatusCode' data type screen. Both screens include a header with the site name, date, and navigation links, and a main content area with a definition, XML type, and XML facets.

Max35Text Data Type Screen:

- Data Type:** Max35Text
- Representation:** Text
- Definition:** Specifies a character string with a maximum length of 35 characters.
- XML Type:** string
- XML Facet:**

| XML Facet | Value |
|-----------|-------|
| maxLength | 35 |
| minLength | 1 |

AccountStatusCode Data Type Screen:

- Data Type:** AccountStatusCode
- Representation:** Code
- Definition:** Specifies the current state of an account.
- XML Type:** string
- Code Table:**

| Code | Code Name | Definition |
|------|--------------------------|---|
| DELE | Deleted | Account cannot be used any longer. |
| DISA | Disabled | Account cannot be used for its intended purpose, either temporarily or permanently. |
| ENAB | Enabled | Account can be used for its intended purpose. |
| FORM | ProForma | Account is temporary and can be partially used for its intended purpose. The account will be fully available for use when the account servicer has received all relevant documents. |

The Data Types screen displays a list of codes with a definition for each code.

Data type options

| | |
|----------------|---|
| Date Type | Defines the contents of the data in the message item. |
| Representation | Groups data according to its characteristics. For example, data can be grouped by Identifier, Code, Text, Rate, DateTime, Amount, Quantity, or Indicator. All the data types that belong to the same representation class inherit the same characteristics. |
| Definition | A definition of the data type. |
| XML Type | Based on this XML primitive type. |
| XML Facet | XML types that specify format restrictions on data types. |
| Example | Provides an example of permitted values. |

Data type of the representation class

| | |
|------------|--|
| Code | Lists the code used in a data type. |
| Code Name | Displays the name of the code used in the data type. |
| Definition | Provides a definition of the code used in the data type. |

5.1.7 Message Component Information

How to display message components

This screen shows details of the message components.

Message component information screen

SWIFTStandards MX September 2006

Payments Clearing and Settlement --General Indexes--

User: [Home](#) [Glossary](#) [Site Map](#)

Message Component: CashAccount7 **Is based on:** [CashAccount](#)

Definition

Information used for identifying an account.

| Message Element | Type | Example | Defined in | Is based on |
|--|--|---------|--------------|--|
| Identification [Mandatory] | AccountIdentification3Choice | | CashAccount7 | Account (Identification) |
| Type [Optional] | CashAccountType2 | | CashAccount7 | CashAccountType |
| Currency [Optional] | CurrencyCode | | CashAccount7 | CashAccount (Currency) |
| Name [Optional] | Max70Text | | CashAccount7 | Account (Name) |

[Back to Top](#)

Message component options

| | |
|-------------------|--|
| Message Component | Shows whether a message item is a message element or a message building block. |
| Is Based On | Describes the business item in the "SWIFT Standards Financial Dictionary" on which a message component is based. |
| Definition | Provides a definition of the message component. |
| Or | Indicates whether the message element is part of a choice. { indicates the start of the choice. } indicates the end of the choice. |
| Type | Provides information about how to complete the message item. A type can be either a data type, or a message component. A message component is composed of message items. The type is shown in <i>italics</i> . It provides information about how to complete the message item. This resembles the Content/options/format column in SWIFT Standards MT Format Specifications. |

5.1.8 Business Example

How to display an example

Each message has one or more relevant business examples that show the use and content of the message in a business scenario. This resembles the example section available in the *SWIFT Standards MT Message Reference Guides* of the *SWIFT User Handbook*.

Business example screen

The screenshot shows the SWIFTStandards MX website interface. At the top left, the logo "SWIFTStandards MX" is displayed. To the right, the date "September 2006" is shown. Below the logo, the text "Payments Clearing and Settlement" is visible, along with a dropdown menu set to "--General Indexes--". A "User" field is present, and navigation links for "Home", "Glossary", and "Site Map" are on the right. The main content area is titled "FIToFICustomerCreditTransferV01 (pacs.008.001.01): Business Example 1" and contains a section for "Narrative-Step 1".

BBB Bank, New York, has received a payment initiation from its client ABC Corporation, New York, that holds an account 00125574999 with them for payment of the following invoices:

1. An invoice with number 4562, dated 08 September 2006 from DEF Electronics, London: 10 million JPY needs to be paid to DEF Electronics account 23683707994215 with AAAA Bank, London (AAAAGB2L). ABC Corporation assigns reference ABC/4562/2006-09-08 to the payment. Payment transaction charges are shared between ABC Corporation and DEF Electronics.
2. An invoice with number ABC-13679, dated 15 September 2006 from GHI Semiconductors, Brussels: 500,000 EUR needs to be paid to GHI Semiconductors account BE30001216371411 with DDDD Bank, Belgium (DDDDBEBB). ABC Corporation assigns reference ABC/ABC-13679/2006-09-15 to the payment. The accounts receivable department of GHI Semiconductors needs to be advised when the funds have been credited to the account on telephone number +32/2/2222222. GHI Semiconductors will bear all payment transaction charges.
3. An invoice with number 987-AC, dated 27 September 2006 from their branch ABC Corporation, California: 1 million USD needs to be paid on the branch account 4895623 with BBBB Bank, San Francisco (BBBBUS66). ABC assigns reference ABC/987-AC/2006-09-27 to the payment. Payment transaction charges are shared.

5.1.9 Message Flows

How to display a message sequence

The message flow is a description and a graphical display of a sequence of messages that industry participants exchange. It resembles the message flow diagrams available in the *SWIFT Standards MT Message Reference Guides* of the *SWIFT User Handbook*.

Message flow screen

The screenshot displays the SWIFTStandards MX website interface. The main content area is titled "Messages Flow" and lists several message types, with "FIToFI Customer Credit Transfer" selected. The selected message type is further detailed with an "Introduction" section. The introduction explains that a FIToFICustomerCreditTransfer is the inter-bank movement of an amount from a party bank account (the Debtor) to a beneficiary party (the Creditor). It also notes that these transfers result in one or more cash transfers between Debtors and Creditors through correspondent banks or infrastructures.

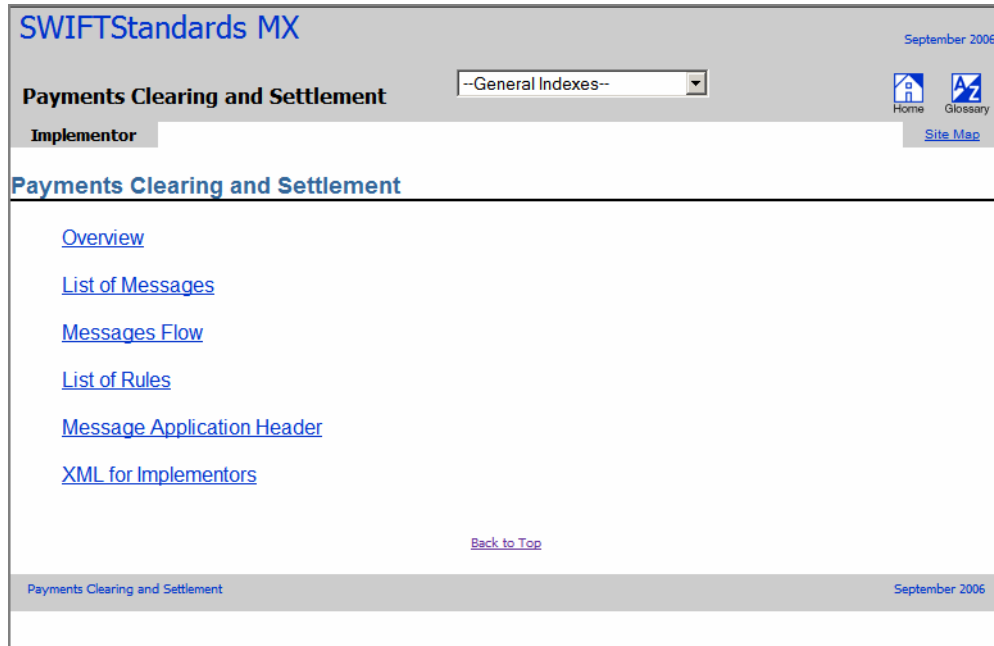
Below the text is a sequence diagram illustrating the message flow. The diagram shows five participants: Debtor : Application, Debtor's Agent : FinancialInstAppli..., Intermediary Agent/Settlem..., Creditor's Agent : FinancialInstAppli..., and Creditor : Application. The diagram shows two messages: "FIToFICustomerCreditTransfer" sent from the Debtor's Agent to the Creditor's Agent, and "PaymentStatusReport" sent from the Creditor's Agent back to the Debtor's Agent. A blue callout box states: "Only potential messages part of the Credit Transfer Project Scope are illustrated in the Sequence Diagrams." A yellow callout box notes: "Scenario1: Debtor's Agent and Creditor's Agent are correspondents".

5.1.10 Implementor Profile

How to display the implementor's profile main menu

To access the implementor profile from the main menu screen, users click the Implementor. The implementor's profile accesses the same information as the user profile. In addition, it provides the information as listed in "Implementor's profile options".

Implementor's profile main menu screens



Implementor's profile options

| | |
|----------------------------|---|
| List of Rules | Lists the defined error codes. |
| Message Application Header | Provides information about the application header. (For more information about the SWIFTNet headers, see the <i>SWIFTNet Service Description</i> and the <i>SWIFTNet Messaging Operations Guide</i> .) |
| XML for Implementors | Provides information about how to implement Standards MX messages. |

Implementor's individual message menu

Individual message screen

The screenshot shows the SWIFTStandards MX website interface. At the top left, it says "SWIFTStandards MX" and "September 2006". Below this, there is a navigation bar with "Payments Clearing and Settlement" and a dropdown menu set to "--General Indexes--". There are also icons for "Home" and "Glossary", and a "Site Map" link. The main content area is titled "Implementor" and "FIToFICustomerCreditTransferV01 (pacs.008.001.01)". Below this, there is a list of links: "Message Scope and Usage", "Structure and Contents", "Business Example 1", "Business Example 2", "XML Schema", "XML Samples", and "Message Indexes". At the bottom of the list, there is a "Back to Top" link.

Individual message options

| | |
|-----------------|--|
| XML Schema | Displays the message schema. The schema is a formal representation of the message structure in XML syntax. Schema files are provided separately to the documentation. |
| XML Samples | Displays a number of XML documents that comply with a specific XML schema. These are technical examples. Business examples contain additional material such as a scenario, narrative, and the related XML instance. |
| Message Indexes | List indexes by individual message. General indexes list information for all the messages in the <i>Standards MX Message Reference Guide</i> . |

Implementor's message index

Implementor's message index screen

The screenshot shows the SWIFTStandards MX website interface. At the top left, it says "SWIFTStandards MX" and "September 2006" is on the top right. Below the header, there is a navigation bar with "Payments Clearing and Settlement" and a dropdown menu set to "--General Indexes--". There are also icons for "Home" and "Glossary", and a "Site Map" link. The main content area is titled "FIToFICustomerCreditTransferV01: Message Indexes" and contains several blue hyperlinks: "XML Tags", "Codes", "Message Item Types" (with sub-links "Data Types" and "Other Types"), "Synonyms", "Rules in this message", "Guidelines in this message", "Rules by Error Code", and "Rules by Message Item".

Implementor's message index options

| | |
|---|---|
| XML Tags | An alphabetical overview of all the XML tags defined for message items in the message. |
| Codes | An alphabetical list of all the codes used the date type code list present in this message. |
| Data Types | An alphabetical overview of all the data types present in this message. |
| Synonyms | An alphabetical overview of all the synonyms present in this message. |
| Rules in this message Guidelines in this message | An alphabetical list of rules and guidelines present in this message. |
| Rules by Error Code | A list of rules sorted by the error codes that appear in the message. They indicate the item in the message tree structure affected by the error. |
| Rules by Message Item | A list of the message items present in the message that a rule affects. |

5.1.11 Indexes

How to display general Indexes

General indexes are alphabetical lists of information about the complete message set for the business domain described in the guide.

General indexes screen

The screenshot shows the SWIFTStandards MX website interface. At the top, it says 'September 2006'. The main heading is 'Payments Clearing and Settlement'. Below that, there's a 'User' field. A dropdown menu is open, showing the following options: --General Indexes--, Business Actors/Roles, Business Information, Synonyms, -- Types used in Messages--, Data Types, Other Types, XML Tags, and Codes. The main content area displays a tree view for 'FIToFICustomerCreditTransferV01'. It includes links for 'Collapse All' and 'Expand All'. The tree view shows the following structure:

- FIToFICustomerCreditTransferV01
 - GroupHeader [Mandatory] [GroupHeader2](#)
 - [MessageIdentification](#) [Mandatory] [Max35Text](#)
 - [CreationDate/Time](#) [Mandatory] [ISODate/Time](#)
 - [BatchBooking](#) [Optional] [BatchBookingIndicator](#)
 - [NumberOfTransactions](#) [Mandatory] [Max15NumericText](#)
 - [ControlSum](#) [Optional] [DecimalNumber](#)
 - [TotalInterbankSettlementAmount](#) [Optional] [CurrencyAndAmount](#)
 - [InterbankSettlementDate](#) [Optional] [ISODate](#)
 - SettlementInformation [Mandatory] [SettlementInformation1](#)
 - [SettlementMethod](#) [Mandatory] [SettlementMethod1Code](#)
 - SettlementAccount [Optional] [CashAccount7](#)
 - Identification [Mandatory] [AccountIdentification3Choice](#)
 - Choice amongst
 - [IBAN](#) [Mandatory] [IBANIdentifier](#)
 - [BBAN](#) [Mandatory] [BBANIdentifier](#)

This table lists the types of indexes available.

General indexes options

| | |
|------------------------|--|
| Business Actors/Roles | An alphabetical list of all the business actors defined in the <i>SWIFT Standards Financial Dictionary</i> . |
| Business Information | An alphabetical list of business components, business elements, and associations. |
| Synonyms | An alphabetical overview of all synonyms. |
| Types used in Messages | An alphabetical overview of all message item types. |
| Data Types | An alphabetical overview of all data types. |
| Other Types | An alphabetical overview of all data types that consist of message components. |
| XML Tags | An alphabetical overview of all XML tags defined for message items. |
| Codes | An alphabetical list of all codes used in data type code lists. |

5.2 Standards MX Message Reference Guide - PDF

5.2.1 Introduction to the Standards MX Message Reference Guide

Introduction to the Standards MX Message Reference Guide in PDF

The *Standards MX Message Reference Guide* in PDF contains the following sections:

- Over view
- List of Messages
 - Message Scope and Usage
 - Message Structure and Contents
 - Business Examples
- Message Flow
- General Indexes
 - Index of Business Actors/Roles
 - Index of Business Information
 - Index of Synonyms
 - Index of Message Item Types
 - Data Types
 - End Points
 - Index of XML Table
 - Index of Codes

5.2.2 List of Messages

Message scope and usage

This section describes message scope, structure, and content for each message. It also provides relevant business examples.

Message structure

This section describes the message structure and hierarchy.

| Message Structure | | | | | | |
|-------------------|-----|--------------------------------------|------------------------|--------|---------------------|-----------------------|
| Index | Or | Message Item | <XML Tag> | Mult. | Represent./ Type | Rule/ Guid. No. |
| 1.0 | | GroupHeader | <GrpHdr> | [1..1] | | |
| 1.1 | | MessageIdentification | <MsgId> | [1..1] | Text | |
| 1.2 | | CreationDateTime | <CreDtIm> | [1..1] | DateTime | |
| 1.3 | | BatchBooking | <BtchBookg> | [0..1] | Indicator | |
| 1.4 | | NumberOfTransactions | <NbOfTxs> | [1..1] | Text | |
| 1.5 | | ControlSum | <CtrlSum> | [0..1] | Quantity | |
| 1.6 | | TotalInterbankSettlementAmount | <TtlIntrBkSttlmAmt> | [0..1] | Amount | |
| 1.7 | | InterbankSettlementDate | <IntrBkSttlmDt> | [0..1] | DateTime | |
| 1.8 | | SettlementInformation | <SttlmInf> | [1..1] | | |
| 1.9 | | SettlementMethod | <SttlmMtd> | [1..1] | Code | |
| 1.10 | | SettlementAccount | <SttlmAcct> | [0..1] | ± | |
| 1.11 | | ClearingSystem | <ClrSys> | [0..1] | | |
| 1.12 | {Or | ClearingSystemIdentification | <ClrSysId> | [1..1] | Code | |
| 1.13 | Or} | Proprietary | <Prtry> | [1..1] | Text | |
| 1.14 | | InstructingReimbursementAgent | <InstgRmbrsmntAgt> | [0..1] | ± | |
| 1.15 | | InstructingReimbursementAgentAccount | <InstgRmbrsmntAgtAcct> | [0..1] | ± | |

Message structure options

| | |
|---------------------|---|
| Index | Number that refers to the related item description in the Message Items Description section. |
| Or | Indicates whether the message element is part of a choice. { indicates the start of the choice. } indicates the end of the choice. |
| Message Item | Part of a message. It can be a message element or a message building block. (This can be compared to a field in a SWIFT Standards MT message.) Each message item is completed with a Type. Individual message items are described in the Message Item Description section. |
| XML Tag | Short reusable name for the message element which appears in the XML schemas and XML instances. |
| Multiplicity | <p>Indicates whether an element is optional or mandatory and how many times the element can be repeated. The multiplicity is shown in square brackets.</p> <p>For example:</p> <ul style="list-style-type: none"> [0..1] This shows that the element can be present either 0 time or 1 time. The element is optional and repeatable. [1..1] This shows that the element can only be present 1 time. [1..n] This shows that the element is always present 1 time and can be repeated n times. |
| Representation Type | Message item type. (This resembles the Content or Option column in SWIFTNet FIN format.) |

| | |
|-------------------|--|
| | <p>If the cell contains one of the data type representations, then the type consists of a data type. This is the lowest level of type. The data type specifies the possible values that a message element can have. This can be a format specification such as the data format .-MM-DD, or an enumeration that lists all possible codes. Data type representations can be one of Identifier, Code, Text, Rate, DateTime, Amount, Quantity, or Indicator.</p> <p>If the cell is shaded grey, then the type consists of a message component. Message components are composed of message items. Message items appear at the next nesting level in the message item column.</p> <p>If a cell contains a plus sign, then types consist of a frequently reused message component such as PartyChoice. This message component is fully described in the End Points section.</p> |
| Role Guideline | A reference to the rule or the guideline in the rules section of the guide. |

Message items description

| |
|--|
| <p>1.1 MessageIdentification <MsgId></p> <p>Presence: [1..1]</p> <p>Definition: Point to point reference assigned by the instructing party and sent to the next party in the chain to unambiguously identify the message.</p> <p>Usage: The instructing party has to make sure that 'MessageIdentification' is unique per instructed party for a pre-agreed period.</p> <p>Data Type: Max35Text</p> <p>Format: maxLength: 35 minLength: 1</p> |
|--|

Message items options

| | |
|------------|--|
| Presence | <p>Indicates whether an element is optional or mandatory and how many times the element can be repeated. The multiplicity is shown in square brackets.</p> <p>For example:</p> <ul style="list-style-type: none"> • [0..1] <p>This shows that the element can be present either 0 time or 1 time. The element is optional and repeatable.</p> <ul style="list-style-type: none"> • [1..1] <p>This shows that the element can only be present 1 time.</p> <ul style="list-style-type: none"> • [1..n] <p>This shows that the element is always present 1 time and can be repeated n times.</p> <p>This value also indicates whether a rule affects the message item.</p> |
| Definition | Contains the definition of the message, message building block, or message element. |
| Type | Provides information about how to complete the message item. The type can be either a data type or a message component. Message components are composed of further message items. |
| Format | Specifies the permitted values, format, and permitted characters. |

5.2.3 Message Item Types

Data types

This section lists and describes all the data types used in the *Standards MX Message Reference Guide*. The data types are classified by representation.

Data types screen

| |
|--|
| Data Types |
| Introduction |
| Data Types Index |
| 1 Amount |
| 1.1 CurrencyAndAmount |
| 1.2 ImpliedCurrencyAndAmount |
| 2 Date Time |
| 2.1 ISODate |
| 2.2 ISODateTime |
| 2.3 ISOTime |
| 3 Identifier |
| 3.1 AustrianBankleitzahlIdentifier |
| 3.2 BBANIdentifier |
| 3.3 BEIdentifier |
| 3.4 BICIdentifier |
| 3.5 CHIPSParticipantIdentifier |
| 3.6 CHIPSUniversalIdentifier |
| 3.7 CanadianPaymentsARNIdentifier |
| 3.8 CountryCode |
| 3.9 CurrencyCode |

Related information

"Data type representation class" on page 26

5.2.4 End Points

Reused components

This section lists and describes components that are reused frequently. A plus sign in the message structure representation, or type column, indicates frequently reused components.

End points screen

| |
|--|
| End Points |
| Introduction |
| End Points Index |
| 1 Account identification |
| 1.1 CashAccount7 |
| 2 Charge |
| 2.1 ChargesInformation1 |
| 3 Financial institution identification |
| 3.1 BranchAndFinancialInstitutionIdentification3 |

End points

This section uses the same structure and description as "Message structure" on page 71 and "Message items description" on page 72.

End points description

| 2 Charge | | | | | |
|---|------|------------------------------------|---------------|--------|------------------|
| 2.1 ChargesInformation1 | | | | | |
| <i>ChargesInformation1 is used in message FIToFICustomerCreditTransferV01 p.45, message FIToFICustomerDirectDebitV01 p.115, message FIToFIPaymentReversalV01 p.168, message PaymentReturnV01 p.322, message PaymentStatusReportV02 p.392.</i> | | | | | |
| Definition: Information on the charges related to the payment transaction. | | | | | |
| Type: The following <i>ChargesInformation1</i> element(s) must be used: | | | | | |
| Ref | Or | Message Item | <XML Tag> | Mult. | Represent./ Type |
| 2.1.0 | | ChargesAmount | <ChrgsAmt> | [1..1] | Amount |
| 2.1.1 | | ChargesParty | <ChrgsPty> | [1..1] | |
| 2.1.2 | | FinancialInstitutionIdentification | <FinInstnId> | [1..1] | |
| 2.1.3 | {Or | BIC | <BIC> | [1..1] | Identifier |
| 2.1.4 | Or | ClearingSystemMemberIdentification | <ClrSysMmbld> | [1..1] | |
| 2.1.5 | {{Or | Identification | <Id> | [1..1] | Code |
| 2.1.6 | Or}} | Proprietary | <Prtry> | [1..1] | Text |
| 2.1.7 | Or | NameAndAddress | <NmAndAdr> | [1..1] | |
| 2.1.8 | | Name | <Nm> | [1..1] | Text |
| 2.1.9 | | PostalAddress | <PstlAdr> | [1..1] | |
| 2.1.10 | | AddressType | <AdrTp> | [0..1] | Code |

Indexes

Index options

| | |
|-----------------------------|---|
| Index of Message Items | An alphabetical list of all the message items that are listed in the <i>Standards MX Message Reference Guide</i> . |
| XML Tags | An alphabetical list of all the XML tags defined for message items. |
| Index of Message Item Types | An alphabetical list of all the data types and message components used in the <i>Standards MX Message Reference Guide</i> . |
| Index of Error Codes | An alphabetical list of all error codes. |

6 Financial Glossary

Introduction to the financial glossary

This section provides a glossary of the terms used in this document.

Application Header

The part of the Payload that specifies information related to the message that is meaningful to the "From" and "To" application. This information is not captured either in the swiftNetHeader (if sent over the SWIFTNet network), or in the message itself.

Business Actor/Role

A role that an industry participant (person, organisation, or infrastructure) plays in a particular Business Process.

Example: Bank, Corporate, AccountServicer

Business Area

A set of strongly related business activities that provide a self-standing business value to the industry.

Business Application

The application that processes a business document at the sending and at the receiving side.

Business Association

A relation between two Business Components.

Example: a Party services an Account

Business Component

A representation of a (part of a) key business notion and characterised by specific Business Elements.

Example: Account, Trade, Party

Business Concept

A generic term, covering Business Actors/Roles, Business Components, Business Elements, Business Associations, and Rules.

Business Document

The information exchanged between two business applications.

Business Element

The representation of a business characteristic of a Business Component.

Example: an AccountIdentification is a characteristic of an Account, an IBAN is one of the characteristics of an AccountIdentification.

Business Example

The description of the use and the content of a Message in real business circumstances.

Business Information

A generic term, covering Business Components, Business Elements and Business Associations.

Business Process

A main business activity that helps the industry participants to achieve their business objectives. A business process always yields an observable result to one or more industry participants.

Closed user group

A subset of customers that SWIFT has grouped for using certain SWIFT services and products when accessing a specific value-added business service.

Code

A character string that for brevity or language independence is used to represent or replace a definitive value or text.

Example: "INDI" is a possible value for the Data Type "Rate Status Code" and represents an indicative Rate

Data Type

A Data Type specifies the possible values that a Business Element or a Message Element can have. This can be done with a format specification (for example, a date format such as YYYY-MM-DD) or by means of an enumeration (for example, a list of all possible codes).

Data Type Representation

A Data Type Representation is used to group similar Data Types, such as Codes, Dates, Text, and so on.

Distinguished Name (DN)

A SWIFTNet standard identifier.

Market Practice

A set of rules that are derived from specific (usually regional) business or regulatory agreements and common practices.

Member-Administered Closed User Group

A member managed service operated by SWIFT using SWIFTNet Messaging Services. It is designed to enable users participating in the closed user group to exchange SWIFT messages bilaterally with the service administrator, or with the administrator-related participant.

Message

A set of structured information that is exchanged between industry participants, in the scope of a Business Process.

Message Building Block

A Message Item that is specific to a particular Message and therefore not defined in the Financial Dictionary. A Message Building Block is defined in the XML Schema as an immediate child of a Message.

Message Component

A reusable building block for assembling a message. It is normally linked to a Business Component and is stored in the Data Dictionary.

Example: Trade Details, which contains a number of the properties of the related Business Component "*Trade*"

Message Component (choice)

A Message Component that represents a choice between different possibilities. An instance of a Message may contain one and only one of the possibilities.

Message Concept

A generic term, covering Message Components, Message Elements and Rules.

Message Definition

The formal description of the structure of a Message.

Message Element

A characteristic of a Message Component.

Example: InstructedAmount is one characteristic of PaymentTransaction.

Message Item

A Message Item is a composing part of a Message. It can appear at any level of nesting in the Message Definition. A Message Item is defined in the XML Schema of a Message and appears as an XML tag in the message instance. A Message Item is either a Message Element or a Message Building Block.

Message Item type

A Message Item Type specifies the structure or the set of valid values of a Message Item. It is a Type, a Message Component, or a Message Construct. A Message Item Type is defined in the XML Schema of a Message and is not visible in a message instance.

Message reception registry

A SWIFTNet routing component that delivers messages to a specific SWIFTNet Link server based on the service name, requestor DN, responder DN, and request type.

Multiplicity

Indication whether something is optional, mandatory, or repetitive. Notation between square brackets represents the multiplicity.

Examples of multiplicity:

- [0..1] this element can be present 0 or 1 time, that is, this element is optional and not repeatable.
- [1] this element is present exactly once. This is the default if no multiplicity is shown.

- [1..n] or [1..*] this element is always present at least once, and may be repeated any number of times.

Payload

The useful content of an information transfer or a message within a messaging system such as SWIFTNet InterAct. This does not include the network-specific header, such as the SWIFTNet Header.

Relay

A third party that forwards a business document between two business applications.

Rule

A constraint on a Business Component, a Message, a Message Construct or a Message Component.

SWIFTNet Header

The portion of a message that contains control information, including Addressee, SWIFT address, processing, and security.

SWIFTNet Validated Rule

Rule that the SWIFTNet Messaging Service validates. The SWIFTNet Messaging Service reports any errors.

SWIFT Standards Financial Dictionary

The *SWIFT Standards Financial Dictionary* provides you with all information about the business items and message items that are used within the financial industry communication. The business items in the *SWIFT Standards Financial Dictionary* are the foundation blocks used to build messages and support the execution of financial business transactions. The benefit of this global Financial Dictionary is to enable all industry players to work on the same basis and improve straight-through processing.

The *SWIFT Standards Financial Dictionary* is built through an ongoing co-operation between the SWIFT community and major financial industry bodies. This dictionary represents a major step towards convergence to a single financial Standard.

The Dictionary contains Business Actors/Roles, Business Components, Business Elements, Business Associations, Message Components, Message Elements, Data Types, and Rules.

Synonym

There are two types of synonyms:

Alternative names of an item in a particular business context. The business context is specified between brackets.

Example: `Strike price` is an alternative name for the `ExercisePrice` Business Element in the `Option` Business Component.

Possible representations of an item in a particular syntax. The syntax is specified between brackets.

Example: the synonym `:19A::REPA` is the representation of the Business Element `RepurchaseAmount` in the Business Component `RepurchaseAgreement` for the `Enhanced-7775` syntax (ISO 15022).

Synonym context

The synonym context identifies the context in which a synonym is used.

Technical Message Element

A message element that has been added for technical reasons. It has therefore no related business element and is message-specific.

Type

A generic name, covering Data Types, Business Components, Message Components, and Message Constructs. They are used to define the allowed structure or allowed set of values of a Business Element or of a Message Item.

UML

UML (Unified Modelling Language) is a standard of the OMG (Object Management Group) that can be used for business process modelling and for information modelling. SWIFT uses this language during the development of Messages to describe all relevant standards-related items in a syntax independent way.

UML Name

The name that SWIFT gives to standards-related items when they are represented in UML.

Variant

There are sometimes multiple ways to use a particular message. The correct way depends on a number of criteria, such as the market practice or the required STP-level. SWIFT uses variants to deal with this situation. Each variant is a separate message definition with a corresponding XML Schema, tree structure and set of rules that formally describes the full details of one particular way to use the message. To maximise the reuse and to ease the implementation, all variants of a particular message are based on a single template. This guarantees that parts that are common across two or more variants are always defined in the same way in these variants.

XML

XML (eXtensible Markup Language) is a standard of the W3C (WorldWideWeb Consortium) that SWIFT uses for the physical representation of its Messages.

XML Attribute

An XML attribute is used in an XML instance to provide additional information for certain types of Data Types.

Example: it is used to specify the currency code for Data Types of the Data Type Representation Amount.

XML Facet or XML Format

An XML facet is used in an XML Schema to specify format restrictions on a Data Type.

Examples: the number of digits of an amount, the maximum length of a text string, a pattern for a text string (for example, three alphanumeric characters).

XML Instance

A document that is structured according to the rules of the XML syntax. An XML instance is validated against a schema.

See also XML schema.

XML Message Name

The name that is used to identify the type of XML Message on SWIFTNet. It also identifies the XML Schema that describes the structure of this XML Message type.

XML Name

A specific name assigned to a Message Construct, Message Component, or Data Type. This appears in XML Schemas of Messages reusing this Message Construct, Message Component, or Data Type. An XML name does not appear in an XML instance.

XML Sample

An example of an XML instance that is technically correct but has no real business meaning.

XML Schema

An XML document that formally describes the allowed structure and content of an XML Message.

An XML instance is validated on SWIFTNet against its corresponding XML schema.

XML Tag

A specific name assigned to a Message Item and that appears in the XML Schema and in XML instances that use this Message Item.

Applies to: Message Elements, and message building blocks

Example: <Bal>

XML Type

The primitive type (as defined in XML) that describes the basic set of possible values for a particular Data Type Representation. Examples: string is the XML-type for Codes, Identifiers, and Text, Boolean is the XML-type of Indicators.

Legal Notices

Copyright

SWIFT © 2011. All rights reserved.

You may copy this publication within your organisation. Any such copy must include these legal notices.

Confidentiality

This publication contains SWIFT or third-party confidential information. Do not disclose this publication outside your organisation without the prior written consent of SWIFT.

Disclaimer

The information in this publication may change from time to time. You must always refer to the latest available version on www.swift.com.

SWIFT Standards Intellectual Property Rights (IPR) Policy - End-User License Agreement

SWIFT Standards are licensed subject to the terms and conditions of the *SWIFT Standards IPR Policy - End-User License Agreement*, available at www.swift.com > About SWIFT > Legal > [SWIFT Standards IPR Policy](#).

Translations

The English version of SWIFT documentation is the only official version.

Trademarks

SWIFT is the trade name of S.W.I.F.T. SCRL. The following are registered trademarks of SWIFT: SWIFT, the SWIFT logo, 3SKey, Innotribe, Sibos, SWIFTNet, SWIFTReady, and Accord. Other product, service, or company names in this publication are trade names, trademarks, or registered trademarks of their respective owners.