ISO 20022
Business Application Header
Message Usage Guide
Version 3.0

Date: 2021-11-03

This Message Usage Guide for the Business Application Header was drafted by the ISO 20022 Technical Support Group and approved by the Registration Management Group.

Contents

1 Introduction 3
1.1 How this Guide was created 3
1.2 The ISO 20022 Standard 3
1.3 Terminology 3
1.4 Separation of layers 4
1.5 Link between a Business Application Header and its Message 5
1.6 Related Documents and Guides 6
2 Scenarios 7
2.1 Business Application Header deployment options 8
2.2 Business Application A sends a Business Message to Business Application B 9
2.3 Business Application A informs Business Application B that Text based MessageElements (in the BAH or the BusinessMessage) may contain non-Basic-Latin characters 11
2.4 Business Application A informs Business Application B of the Business Service within which this BusinessMessage is exchanged 12
2.5 Business Application A sends a Business Message with implementation specification 13
2.6 Business Application A sends a Business Message with instruction of business processing date 14
2.7 Business Application A sends a copy of a previously sent Business Message 16
2.8 Business Application A sends a duplicate of a previously sent Business Message 17
2.9 Business Application A sends a duplicate of a previously sent copy of a Business Message 18
2.10 Business Application A suspects Business Application B has not received the BusinessMessage 19
2.11 Business Application A sends a Business Message to BusinessApplication B with a pre-agreed priority 20
2.12 Business Application A sends a signed Business Message to BusinessApplication B 21
2.13 Business Application B sends a Business Message B that relates to Business Message A 21
3 Mapping of the BAH to other headers 24
3.1 Introduction 24
3.2 BAH to SWIFTNet Application Header 24
3.3 BAH to ebXML/ebMS Header 26
3.4 BAH to FpML 27
4 ANNEX A - Signature Guidelines 28
4.1 Preface 28
4.2 Introduction 28
4.3 Structure of the Signature 31
4.4 Process for signing and verifying 33
4.5 Example 36
Disclaimer: 38

1 Introduction

This guide helps business analysts and software developers understand how to use the ISO 20022 Business Application Header (BAH) in various business scenarios. It provides a comprehensive view of how the BAH complements any ISO 20022 Message. It acts as a supplement to the Message Definition Report and the XML schema, which are published on the ISO 20022 website (www.iso20022.org).

Additional documents, published by individual user communities, may be available that discuss the implementation of the BAH in a more specific context. This guide should serve as the general basis for these more specific community implementation guides.

Currently, the included descriptions and examples that are used in this document are based exclusively on the ISO 20022 XML syntax. In the future, there may be a requirement to include descriptions or examples in other syntaxes such as ASN.1 or, if approved as an ISO 20022 syntax, JSON depending on the demand for this in the ISO 20022 community.

1.1 How this Guide was created

This guide was created through the combined efforts of the ISO 20022 Standards Evaluation Groups (SEG) for the collection of the requirements, the ISO 20022 Technical Support Group (TSG) for drafting the solution and the ISO 20022 Registration Management Group (RMG) for approving this document.

This guide is maintained by the TSG and published on the ISO 20022 website.

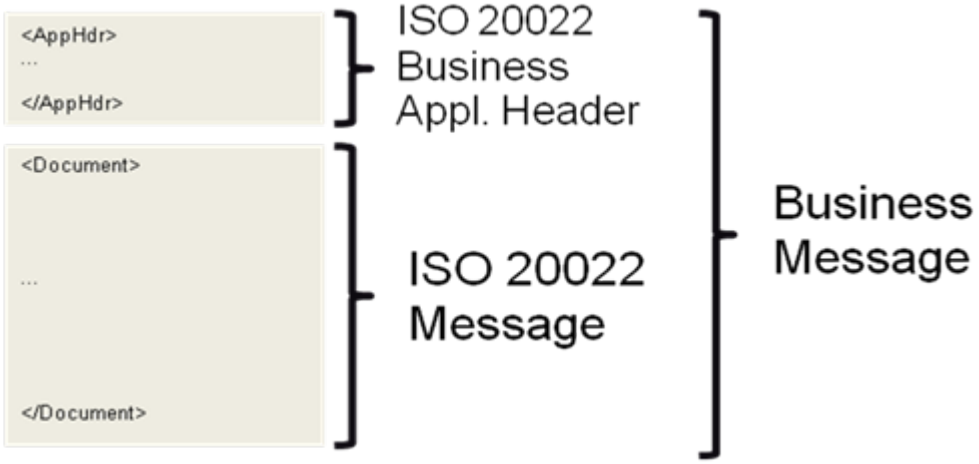
1.2 The ISO 20022 Standard

ISO 20022 – the Universal financial industry message scheme - defines a scalable, methodical process to ensure consistent descriptions of messages throughout the financial services industry. It is an international standard published by the International Organization for Standardization (ISO), and prepared by Technical Committee 68 (TC68) for Financial Services.

This process results in models and artefacts stored in a central repository, serviced by a Registration Authority. This repository is available on the World Wide Web and offers public access for browsing at www.iso20022.org. For more information on ISO and the standard itself, please see www.iso.org.

1.3 Terminology

An ISO 20022 Message together with its Business Application Header forms a Business Message.



1.4 Separation of layers

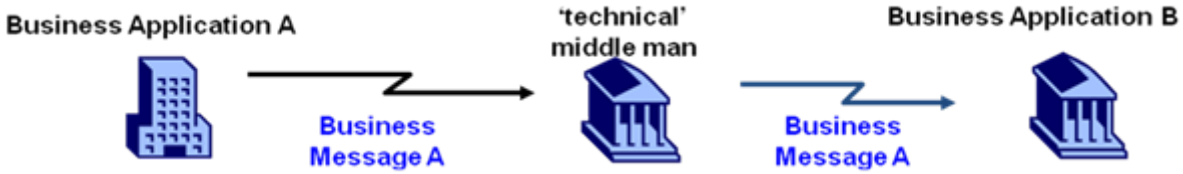
ISO 20022 messages and the BAH are designed to be transport protocol independent. The ISO 20022 standard does not provide any message transport conventions of its own (including header or trailer).

The Business Application Header is a business header and should not be confused with a file or transport header. The Business Application Header is created and applied to its ISO 20022 Message to form a Business Message. Subsequently, a file or transport routing header, if any, is applied to the Business Message.

So, any parties between the two business applications that do not perform a business function are not mentioned in the BAH. Such 'technical' middle men do not open or change the Business Message; they only forward it to the correct business application.

So, a transport scenario like below

is from a business point of view the same as a transport scenario like this:



because the technical middle man is forwarding the message, without changing the contents.

However, as soon as a Business Application is in the middle (i.e. an Application that processes the Business Message), it is identified in the BAH of the first message as the recipient, and in the BAH of the second message as the sender, and therefore sends a different business message.

Although the BAH is not the transport header, data in the BAH, as data in the ISO 20022 message itself, can be used by transport applications to determine the routing header since it does contain the business sender, receiver and document details. It can also be used by the business applications to determine the appropriate process to perform on the business message.

1.5 Link between a Business Application Header and its Message

A Business Application Header is typically packaged with its ISO 20022 Message root element following the BAH's root element in an XML envelope. Other packaging is possible, such as zip files, or mime multi part.

Although the standard does not specify the order of the Business Application Header (AppHdr XML element) and the ISO 20022 Message (Document XML element), it is common best practice that the Business Application Header precedes the ISO 20022 message.

EXAMPLE

Below XML Schema describes an element RequestPayload (used on SWIFTNet) that contains two elements: AppHdr and any other element.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Generated by Standards Editor (build:R1.6.16) on 2021 Nov 03 11:09:56, ISO 20022 version : 2013-->
<xs:schema xmlns="SWIFTNetBusinessEnvelope" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="SWIFTNetBusinessEnvelope" elementFormDefault="qualified">
    <xs:import namespace="urn:iso:std:iso:20022:tech:xsd:head.001.001.03" schemaLocation="head.001.001.03.xsd"/>
    <xs:import namespace="urn:iso:std:iso:20022:tech:xsd:pacs.008.001.08" schemaLocation="pacs.008.001.08.xsd"/>
    <xs:element name="RequestPayload" type="ISO20022BusinessMessage1"/>
    <xs:complexType name="ISO20022BusinessMessage1">
        <xs:sequence>
            <xs:any namespace="urn:iso:std:iso:20022:tech:xsd:head.001.001.03" processContents="strict"/>
            <xs:any processContents="strict"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

A valid XML instance would be structured as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<n1:RequestPayload                                    xmlns:n1="SWIFT
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:n2="urn:iso:std:iso:20022:tech:xsd:head.001.001.03"
xsi:schemaLocation="SWIFTNetBusinessEnvelope AppHdrBizMsg.xsd">
        <n2:AppHdr xmlns:n2="urn:iso:std:iso:20022:tech:xsd:head.001.001.03">
                ...
        </n2:AppHdr>
        <Document xmlns="urn:iso:std:iso:20022:tech:xsd:pacs.008.001.08">
                ...
        </Document>
</n1:RequestPayload>
```

Refers to the envelope schema

Refers to the ISO 20022 Business Application Header

Refers to the ISO 20022 Message

1.6 Related Documents and Guides

The complete catalogue of ISO 20022 messages, including the Message Definition Reports and XML schemas, is available on the ISO 20022 website: www.iso20022.org. Current and historical versions of the schemas are available free of charge. Other useful documentation available from the ISO 20022 website includes:

- ISO 20022 Repository - Data Dictionary.
- Introduction to ISO 20022 – Universal financial industry message scheme.
- An introductory PowerPoint on the ISO 20022 standard family.

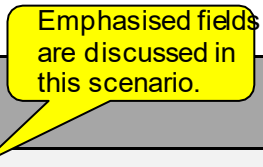
Useful documents are available from the following sources:

- An in-depth knowledge of XML can be found at: http://www.w3c.org/XML
- An in-depth knowledge of XML Schema can be found at: https://www.w3.org/XML/Schema#resources
- The UNICODE character set database can be found at: http://unicode.org/Public/UNIDATA/Blocks.txt

2 Scenarios

This section explains how to use the BAH in specified scenarios. The list of scenarios is not exhaustive and new scenarios may be added.

For clarity, each scenario includes a table showing optional fields in light grey and mandatory fields in dark grey, the fields in bold emphasise the fields under discussion.



MessageElement Name	Usage
CharacterSet	**Light grey fields are optional.**
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	
Priority	
Signature	
Related	

2.1 Business Application Header deployment options

The choice of whether to send a BAH with a given message is taken by each implementing community.

Depending on how a given message was originally designed and/or any maintenance changes that may have occurred over time, an implementing community can be confronted with the following deployment options:

Message deployment	ISO 20022 message …	
	with duplicated BAH elements	without duplicated BAH elements[1]
with BAH	Business rules and market practice should determine the use of BAH elements in the BAH itself and any duplicated BAH elements in the ISO 20022 message	Only BAH elements within BAH are available for use
without BAH	Only BAH elements duplicated within ISO 20022 message are available for use	BAH elements are not available for use, neither in ISO 20022 message nor through the BAH

In the case of ISO 20022 messages without duplicated BAH elements, message deployment without the BAH can produce challenging situations (i.e. bottom right scenario in above table).

In these scenarios, essential BAH elements like the unique message identifier (BAH element: BusinessMessageIdentifier) or the message creation date (BAH element: CreationDate) cannot be conveyed through the ISO 20022 message, nor through the BAH.

If for certain implementation communities the deployment of the BAH does not constitute a valid choice, implementation communities are requested to get in contact with the ISO 20022 Registration Authority (RA) for advice on how to proceed in order to employ a fully ISO 20022-compliant implementation (e-mail: iso20022ra@iso20022.org).

Implementation communities may have alternative means to carry this information through non-ISO 20022 mechanisms (e.g. a network header). It should be noted that, in case of non-ISO 20022 mechanisms, not all scenarios covered in this BAH MUG could be fully implemented. For further information, see chapter 2.3.

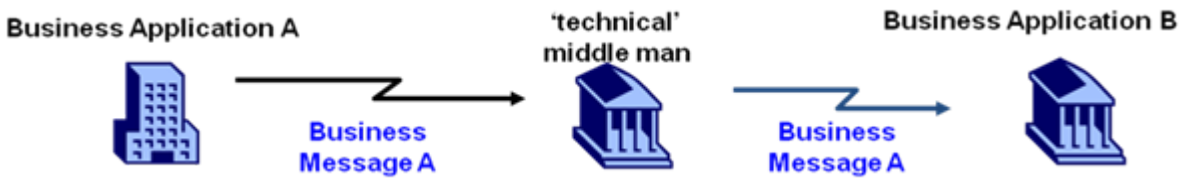
[1] ISO 20022 messages without duplicated BAH elements are labelled, in the Business domain catalogues on the ISO 20022 web-site, with the sentence "The message definitions below are intended for use with the Business Application Header.".

2.2 Business Application A sends a Business Message to Business Application B

Above is the most common, vanilla scenario, i.e. no special features of this BAH are used.

When there is a 'middle man' between the two Business Applications, it is the function/role of that middle man that will determine whether the Business Message from the middle man to Business Application B is a new Business Message.

If the middle man only forwards the Business Message, i.e. it does not process the Business Message, then only the transport header changes, but the Business Message (with its BAH) remains the same.



Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	**Id of BusinessApplication A**
To	**Id of BusinessApplication B**
BusinessMessageIdentifier	**Identification of the BusinessMessage A**
MessageDefinitionIdentifier	**Identification of the MessageDefinition**
BusinessService	
MarketPractice	
CreationDate	**Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)**
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	
Priority	
Signature	
Related	

NOTE In the case of ISO 20022 messages without duplicated BAH elements, the 'technical middle man' scenario can pose implementation challenges, if the implementer community has chosen not to use the BAH.

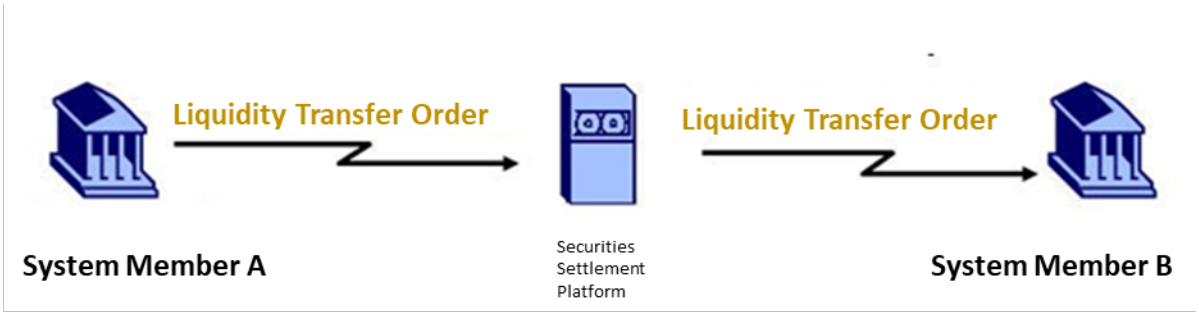
If the 'technical middle man' were to realise the link between two separate networks (e.g. each with its distinct network header), it may be that the non-ISO 20022 mechanisms employed are not sufficiently compatible in order for the business elements, otherwise conveyed through the BAH and/or the ISO 20022 message to travel from Business Application A to Business Application B.

In order to avoid such situation, the process outlined in chapter 2.2 can be employed to realise a fully ISO 20022-compliant implementation without requiring the mandatory deployment of the BAH.

If the middle man processes the Business Message, then the middle man is considered a Business Application and hence a new Business Message is created and sent to Business Application B (see scenario 2.11).

EXAMPLE

A Securities Settlement platform serves as technical middle man:




```
<AppHdr>
        <Fr>System Member A</Fr>
        <To>System Member B</To>
        …
        <BizSvc>SnR</BizSvc>
        …
</AppHdr>
```

Example code for technical middle man

There may be several CreationDate elements but they may not have the same definition.

As a general rule, the CreationDate in the BusinessApplicationHeader contains the date and time at which the Business Message was produced by the BusinessApplication. The term BusinessApplication must be interpreted in a broad sense: it may be a payments factory application; it may also be an operator at a screen who is manually inputting the business information.

Depending on the definition that is given to a CreationDate in the ISO 20022 Message (Document XML element) itself, the date and time could vary from the date and time in the Business Application Header. The creation date in the BAH applies to the entire Business Message whereas other creation dates apply to only parts of the Business Message.

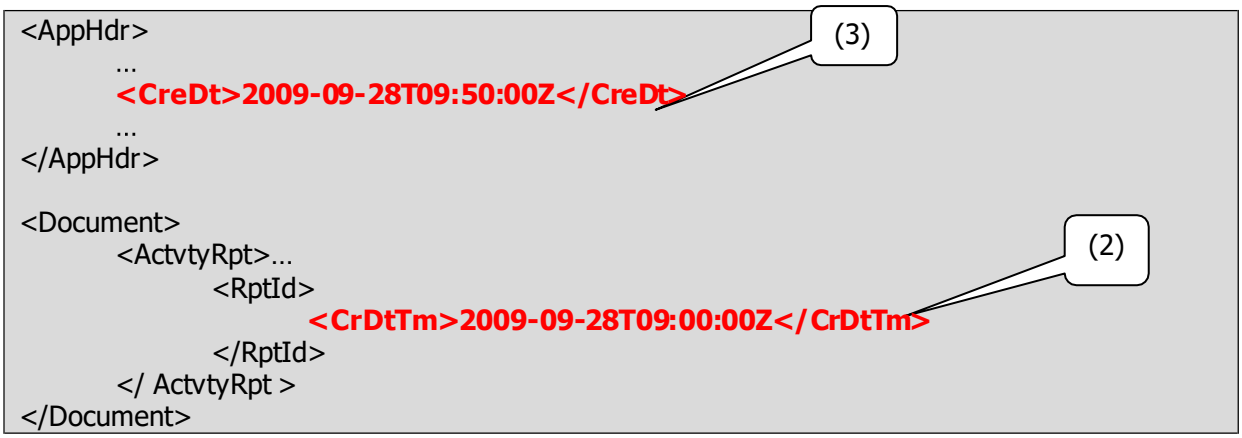
EXAMPLE

1 John Doe at Bank A prepares the payments;

2 The same person creates an activity report on September 28th at 09:00AM UTC;

3 The same person signs it at 9:50AM UTC with the signature in the BAH;

4 The BAH contains the CreationDate of the BusinessMessage, the Document contains the CreationDate of the activity report.




```
<AppHdr>
        …
        <CreDt>2009-09-28T09:50:00Z</CreDt>
        …
</AppHdr>

<Document>
        <ActvtyRpt>…
                <RptId>
                        <CrDtTm>2009-09-28T09:00:00Z</CrDtTm>
                </RptId>
        </ ActvtyRpt >
</Document>
```

2.3 Business Application A informs Business Application B that Text based MessageElements (in the BAH or the BusinessMessage) may contain non-Basic-Latin characters

In this case, each additional character set will be specified in the CharacterSet MessageElement, separated by a semicolon.

All relevant Text based Datatypes may then contain characters belonging to the character sets specified in this MessageElement.

Some Text based DataTypes may be further constrained than what is specified here in which case the character set restrictions specified in the BAH do not apply.

Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	**Character set 1; Character set 2**
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	
Priority	
Signature	
Related	

2.4 Business Application A informs Business Application B of the Business Service within which this BusinessMessage is exchanged

This enables the Receiver to unambiguously relate the BusinessMessage to the BusinessService in which it is used.

Normally this MessageElement is used when this BusinessMessage is used in multiple services.

It can be the service identified by the service provider or a bilaterally / multilaterally agreed service.
Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	**Identification of the Service within which this Message is exchanged**
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	
Priority	
Signature	
Related	

2.5 Business Application A sends a Business Message with implementation specification

Market practices are a set of rules agreed between parties and specify the business context in which business messages are exchanged.

A market practice specification consists of restrictions of the underlying message definition combined with optional extensions of the underlying message definition by using extensions or supplementary data of this underlying message.

MarketPractice is identified by reference via the following sub-elements:

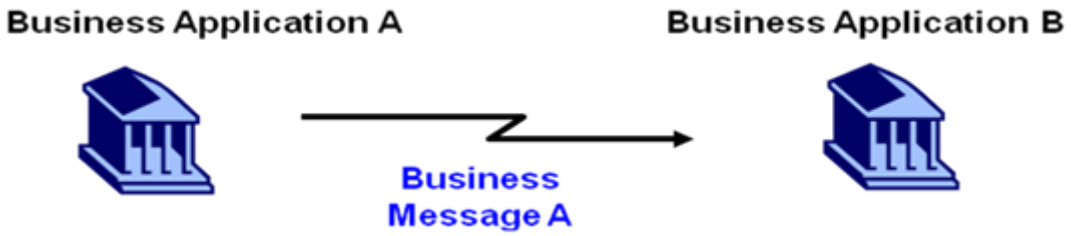
- Registry

  Name of the implementation specification registry in which the implementation specification of the ISO 20022 message is maintained, for example, "MyStandards".

- Identification

  Identification, which unambiguously identifies, within the implementation specification registry, the implementation specification to which the ISO 20022 message is compliant. This can be done via a URN. It can also contain a version number or date.

  For instance, "2018-01-01 – Version 2" or "urn:uuid:6e8bc430-9c3a-11d9-9669-0800200c9a66".



Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	**The market practice to which the message conforms**
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	
Priority	
Signature	
Related	

EXAMPLE

```
<AppHdr>
        <Fr>Securities Settlement Platform</Fr>
        <To>System Member</Fr>
        …
        <MktPrctc>
              <Regy>MyStandards</Regy>
              <Id>urn:uuid:6e8bc430-9c3a-11d9-9669-0800200c9a66</Id>
       </MktPrctc>
        …
</AppHdr>
<Document>
        <LqdtyCdtTfr>…..</LqdtyCdtTfr>
</Document>
```

2.6 Business Application A sends a Business Message with instruction of business processing date

The purpose of the timestamp in the BusinessProcessingDate is to indicate which business processing date the message refers to. The business processing date may be different from the date provided in the CreationDate element.

A market practice or bilateral agreement should specify how this element should be used.

EXAMPLE

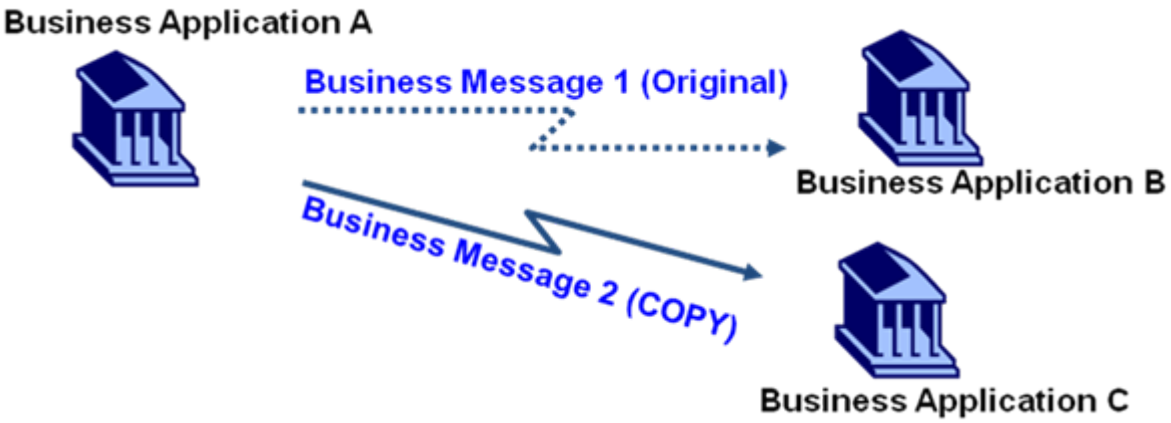
1. If a transaction was reported overnight Friday (i.e. after 12:00 am Saturday), the transaction would likely be deemed complete on Monday (Business Processing Date), not Saturday (effective Processing Date);
2. After-hours transactions or transfers that are executed on a calendar date different to the legal business date;
3. Extended processing beyond the calendar date – especially to avoid doubt when there are time zone changes relative to the UTC Date and Time;
4. Forward dated requests for migration or scheduled events.

Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	**Processing date and time indicated by the sender for the receiver of the business message**
CopyDuplicate	
PossibleDuplicate	
Priority	
Signature	
Related	

```
<AppHdr>
        <Fr>Securities Settlement Platform</Fr>
        <To>System Member</To>
        ...
        <CreDt>2020-09-25T09:00:00Z</CreDt>
        <BizPrcgDt>2020-09-28T09:00:00Z</BizPrcgDt>
        ...
</AppHdr>
<Document>
        <LqdtyCdtTfr>...</LqdtyCdtTfr>
</Document>
```

2.7 Business Application A sends a copy of a previously sent Business Message



Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	**COPY**
PossibleDuplicate	
Priority	
Signature	
Related	**Copy of the relevant MessageElements of the BusinessApplicationHeader of the original BusinessMessage sent to Business Application B**

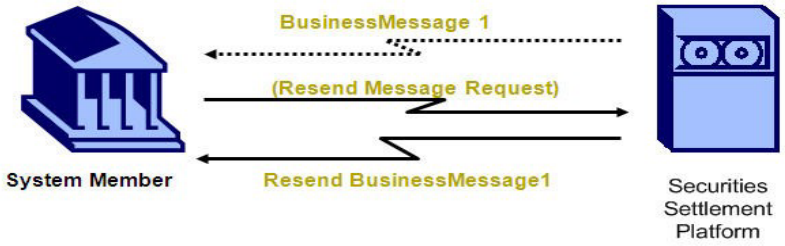
2.8 Business Application A sends a duplicate of a previously sent Business Message

Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	**DUPL**
PossibleDuplicate	
Priority	
Signature	
Related	**Copy of the relevant MessageElements of the BusinessApplicationHeader of the original BusinessMessage sent to Business Application B**

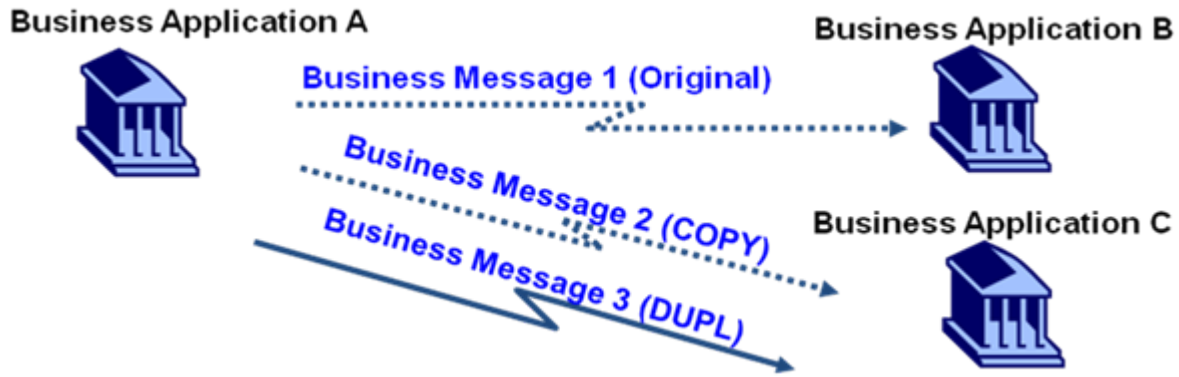
EXAMPLE

Upon request, the Securities Settlement platform resends BusinessMessage1




```
<AppHdr>
        <Fr>Securities Settlement Platform</Fr>
        <To>System Member</To>
        ...
        <CpyDplct>DUPL</CpyDplct>
        <Rltd>relevant MessageElements of original BAH</Rltd>
        ...
</AppHdr>
<Document>
```

2.9 Business Application A sends a duplicate of a previously sent copy of a Business Message



Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	**CODU**
PossibleDuplicate	
Priority	
Signature	
Related	**Copy of the relevant MessageElements of the BusinessApplicationHeader of the copy BusinessMessage sent to Business Application B**

2.10 Business Application A suspects Business Application B has not received the BusinessMessage

PossibleDuplicate is used when the Business Application that sent the message has not received any reply (because of technical or other problems).

It will therefore resend THE SAME BusinessMessage, adding the relevant header elements from the original BusinessMessage in the 'Related' MessageElement.

If the receiver did receive the original BusinessMessage identified in 'Related', then this BusinessMessage MUST BE IGNORED.

If the receiver did NOT receive the original BusinessMessage, then it must treat this BusinessMessage as if it was the original BusinessMessage.

Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	**YES**
Priority	
Signature	
Related	**Copy of the relevant MessageElements of the BusinessApplicationHeader of the original BusinessMessage sent to Business Application B**

2.11 Business Application A sends a Business Message to BusinessApplication B with a pre-agreed priority

The standard does not define the meaning of the values. The meaning is service/market/business area depended and must be pre-agreed within a specific context like the BusinessService, MarketPractice, BusinessArea, etc.

Where required, the BusinessService and/or MarketPractice should be used to identify that specific context (i.e. when there may be ambiguity) about the meaning of the value in Priority.

Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	
Priority	**The priority as defined within the BusinessService**
Signature	
Related	

2.12 Business Application A sends a signed Business Message to BusinessApplication B

The signature must be structured as per the W3C XML Signature specification and any additional constraints stated for the service / market practice / business area within which this BusinessMessage is exchanged.

Refer to the Signature Annex for more information.

Following MessageElements must be used:

MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	
Priority	
Signature	**Signature specification containing the signature of the Sending Business Entity**
Related	

2.13 Business Application B sends a Business Message B that relates to Business Message A

In this type of scenario, BusinessMessage B is neither a copy nor a duplicate of BusinessMessage A.

This scenario can be witnessed in a number of ISO 20022 domains. Whenever an ISO 20022 business process foresees that multiple business messages of potentially different message definitions are employed, a particular business message may refer, for instance, to a message previously received and/or a message previously sent.

This pattern is particularly common when a business process extends over a chain of multiple intermediaries, for example, a chain of investment funds order execution intermediaries, a chain of securities settlement agents, a securities custody chain or a chain of payment correspondent banks.

As depicted in the diagram, this scenario is used when it is relevant for the recipient of BusinessMessage B to know about the underlying BusinessMessage A which triggered the creation of this BusinessMessage B. BusinessMessage B will indicated in its Related element all details of the Business Application Header of BusinessMessage A, for instance, the entity who created the original BusinessMessage A (in the From element of the Related element).

CopyDuplicate and PossibleDuplicate are not used in this scenario.

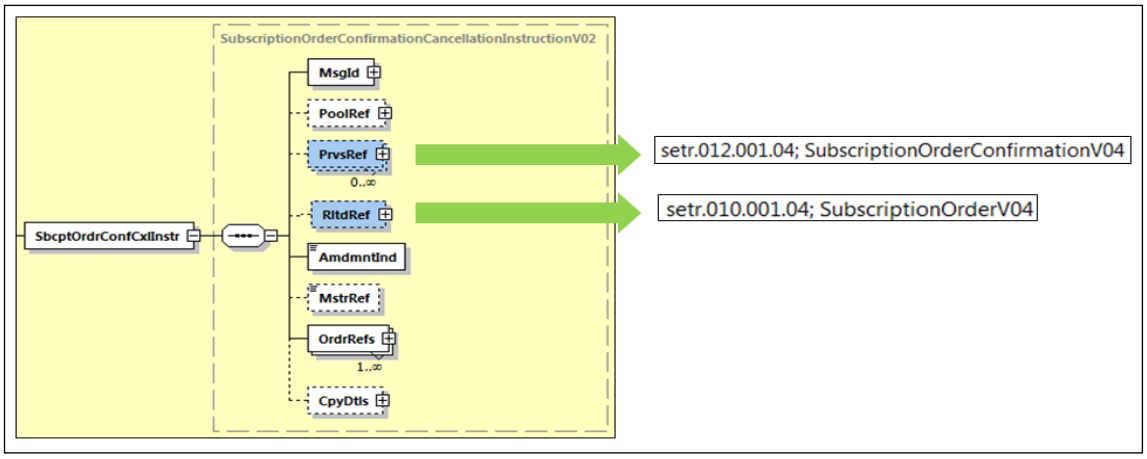
MessageElement Name	Usage
CharacterSet	
From	Id of BusinessApplication A
To	Id of BusinessApplication B
BusinessMessageIdentifier	Identification of the BusinessMessage A
MessageDefinitionIdentifier	Identification of the MessageDefinition
BusinessService	
MarketPractice	
CreationDate	Date (and time) of the creation of this BusinessApplicationHeader (and BusinessMessage)
BusinessProcessingDate	
CopyDuplicate	
PossibleDuplicate	
Priority	
Signature	
Related	**Copy of the relevant MessageElements of the BusinessApplicationHeader of BusinessMessage A**

As a business message may be related to multiple other business messages, it is necessary to be able to distinguish the function of each of the business messages referred to. For this purpose, the MessageDefinitionIdentifier of the business messages referred to can be used.

EXAMPLE

Consider the following scenario of the investment funds order execution business process.

The SubscriptionOrderConfirmationCancellationInstruction (setr.047) cancels a previously sent SubscriptionOrderConfirmation (setr.012) and is related to a previously received SubscriptionOrder (setr.010).

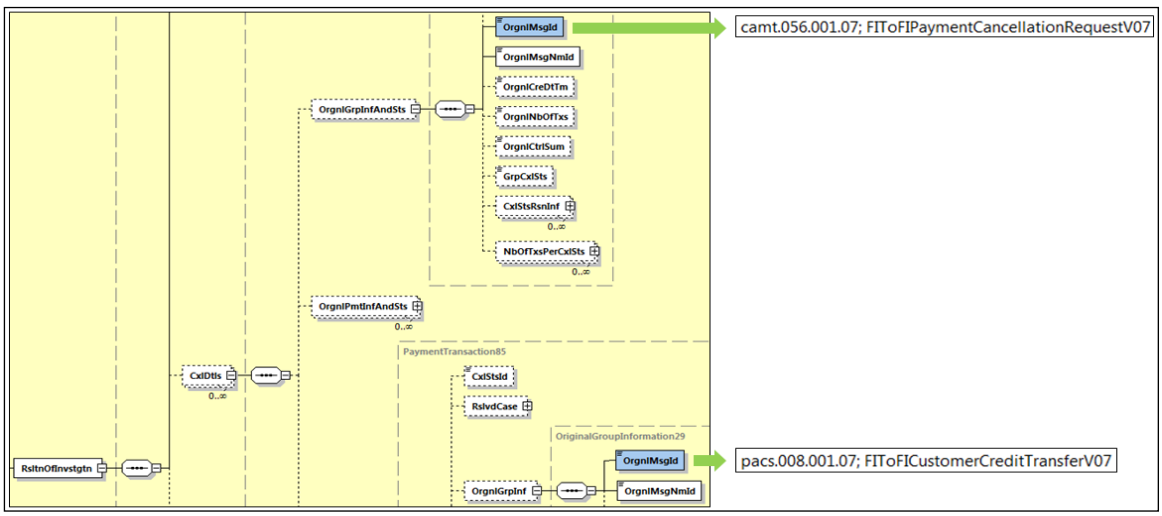


If in this example the three messages involved are used with the BAH, one instance of the Related element of the BAH with the setr.047 message can be used to identify the previously sent business message (setr.012) and the other instance can be used to identify the previously received business message (setr.010).

The mandatory MessageDefinitionIdentifier, which is indicated for each of the two business messages in their respective Related element, can be used to distinguish the function of the two related business messages.

EXAMPLE

In the payments domain, consider the ResolutionOfInvestigationV08 message (camt.029). The elements OriginalGroupInformationAndStatus / OriginalMessageIdentification and OriginalGroupInformation / OriginalMessageIdentification refer to other linked messages, i.e. the FIToFIPaymentCancellationRequestV07 message (camt.056) and the FIToFICustomerCreditTransferV07 message (pacs.008), respectively. These messages are used for exceptions and investigations.



3 Mapping of the BAH to other headers

In a number of situations, ISO 20022 (business) messages are employed in heterogeneous environments where the end-to-end use of the BAH cannot be taken for granted or where over time the migration from proprietary headers to the BAH is foreseen.

In these situations, it can be useful to understand the equivalencies of the elements of the various headers in use.

3.1 Introduction

Below list contains all MessageElements of the BusinessApplicationHeader.

For reference, this table lists the top MessageElement names of the BAH and their equivalent XML abbreviation.

MessageElement Name	abbreviation
CharacterSet	CharSet
From	Fr
To	To
BusinessMessageIdentifier	BizMsgIdr
MessageDefinitionIdentifier	MsgDefIdr
BusinessService	BizSvc
MarketPractice	MktPrctc
CreationDate	CreDt
BusinessProcessingDate	BizPrcgDt
CopyDuplicate	CpyDplct
PossibleDuplicate	PssblDplct
Priority	Prty
Signature	Sgntr
Related	Rltd

3.2 BAH to SWIFTNet Application Header

The table below states the equivalencies of the BAH elements in comparison to the elements of the proprietary SWIFTNet Application Header.

This mapping can be useful in a number of cases:

a) Introducing the BAH for ISO 20022 message sets modelled for use with and without the BAH

As such message sets do not require the use of the BAH, it is often the case that such message sets are deployed with a non-ISO 20022 header, for instance, the SWIFTNet Application Header.

As and when such communities decide to migrate from the proprietary header to the BAH, the mapping table can be employed as a migration support tool.

b) ISO 20022 message sets employed in a multiple network environment

It is not uncommon that in an end-to-end scenario the exchange of ISO 20022 message sets spans multiple underlying networks, amongst others, SWIFTNet. SWIFTNet may still foresee its own proprietary Application Header whilst others may already have migrated to using the BAH.

In order to ensure that the business information of the BAH elements can be correctly conveyed to and from their sibling elements of the SWIFTNet Application Headers, the mapping table can be used to support the implementation of the information exchange at the gateways between SWIFTNet and the other networks.

Business Application Header MessageElement	**SWIFTNet Application Header**
character set the additionally used character set(s) in the BusinessMessage for Text datatypes	-
Sending Business Entity the Business Entity that **created** the BusinessMessage	<RequestorDN>
Receiving Business Entity the Business Entity that will **process** the BusinessMessage	<ResponderDN>
BusinessMessageIdentifier the identification of the BusinessMessage, unique to the SendingBusinessEntity	<MessageReference>
MessageDefinitionIdentifier the identification of MessageDefinition	<MessageIdentifier>
BusinessService Example: EnI	<ServiceName>
Market Practice Set of rules agreed between parties that restricts the usage of the messages in order to achieve better STP (Straight Through Processing) rates.	-
CreationDate Creation date (and time) of the Business Message	<CreationDate>
Business Processing Date Processing date and time indicated by the sender for the receiver of the business message.	-
Copy / Duplicate / Functionality to indicate this message is a copy / duplicate of another	<SnFCopy>
Possible Duplicate The BusinessMessage may have been sent before.	<PossibleDuplicate>
Priority Relative indication of the processing precedence of the message over a (set of) BusinessMessages with assigned priorities	<Priority>
Signature Contains the digital signature of the person authorised to sign this BusinessMessage (based on W3C's XML Signature standard)	<Signed>
Related Reference Reference of the original BusinessMessage	<Related>
BusinessMessage	

3.3 BAH to ebXML/ebMS Header

Business Application Header element	ebMS 3.0
Sending Business Entity the Business Entity that **created** the BusinessMessage	<eb:From>
Receiving Business Entity the Business Entity that will **process** the BusinessMessage	<eb:To>
character set the additionally used character set(s) in the BusinessMessage for Text datatypes	_
BusinessService Example: E&I	<eb:Service>
MessageDefinitionIdentifier the identification of MessageDefinition	<eb:Property>
BusinessMessageIdentifier the identification of the BusinessMessage, unique to the SendingBusinessEntity	<eb:MessageId>
Copy / Duplicate / Functionality to indicate this message is a copy / duplicate of another BusinessMessage	_
Related Reference Reference of the original BusinessMessage	<eb:RefToMessageId>
Possible Duplicate The BusinessMessage may have been sent before.	_
Priority Relative indication of the processing precedence of the message over a (set of) BusinessMessages with assigned priorities	_
Signature Contains the digital signature of the person authorised to sign this BusinessMessage (based on W3C's XML Signature standard)	_
CreationDate Creation date (and time) of the BusinessMessage	<eb:Timestamp>
Market Practice Set of rules agreed between parties that restricts the usage of the messages in order to achieve better STP (Straight Through Processing) rates.	_
Business Processing Date Processing date and time indicated by the sender for the receiver of the business message.	_
BusinessMessage	

3.4 BAH to FpML

Business Application Header element	Message Header
Sending Business Entity the Business Entity that **created** the BusinessMessage	<sentBy>
Receiving Business Entity the Business Entity that will **process** the BusinessMessage	<sendTo>
character set the additionally used character set(s) in the BusinessMessage for Text datatypes	<?xml version="1.0" encoding="UTF-8">
BusinessService Example: E&I	-
MessageDefinitionIdentifier the identification of MessageDefinition	v5: root element v4: xsi:type of Document
BusinessMessageIdentifier the identification of the BusinessMessage, unique to the SendingBusinessEntity	<messageId>
Copy / Duplicate / Functionality to indicate this message is a copy / duplicate of another BusinessMessage	<copyTo>
Related Reference Reference of the original BusinessMessage	<inReplyTo>
Possible Duplicate The BusinessMessage may have been sent before.	
Priority Relative indication of the processing precedence of the message over a (set of) BusinessMessages with assigned priorities	_
Signature Contains the digital signature of the person authorised to sign this BusinessMessage (based on W3C's XML Signature standard)	<dsig:Signature>
CreationDate Creation date (and time) of the Business Message	<creationTimestamp>
Market Practice Set of rules agreed between parties that restricts the usage of the messages in order to achieve better STP (Straight Through Processing) rates.	<implementationSpecification>
Business Processing Date Processing date and time indicated by the sender for the receiver of the business message.	_
BusinessMessage	

4 ANNEX A - Signature Guidelines

4.1 Preface

It is recommended that implementers create clear guidelines for messaging scenarios and related use cases, in which the BAH signature mechanism is employed. Such guidelines can ensure a common understanding of the way the signature is structured, and they can constitute a shared agreement of how the creator of the signature and all verifiers of the signature must apply the signature mechanism.

The guidelines in the remainder of this annex can serve as an illustrative model. Based on the model, specific guidelines can be created for each distinct use case in which the BAH signature mechanism is to be employed.

The guidelines are pure technical guidelines, which only discuss the formal aspects of the construction of the signature. All other aspects of the signature, such as, who can sign and such as the message scenarios, which require signing, are not part of this annex.

4.2 Introduction

A digital signature is a mathematical scheme for verifying the authenticity of digital messages. A valid digital signature, where the prerequisites are satisfied, gives a recipient sufficient confidence that the message was created by a known sender (authentication) and that the message was not altered in transit (integrity).

The optional Signature element of the BAH can be employed to contain the digital signature of the Business Entity authorised to sign the Business Message, i.e. the ISO 20022 Business Application Header and the ISO 20022 Message.

There is no support for signatures covering only a part of the business message.

The Signature must be structured as per the W3C XML Signatures specification and any additional constraints stated for the service and business area within which this BusinessMessage is exchanged. This requires that the creator of the Signature and all verifiers of that Signature agree how the Signature is structured.

4.2.1 Current Signature Guidelines

The *Business Application Header Message Usage Guidelines* specifies: "Signature contains the digital signature of the person authorised to sign this BusinessMessage (based on W3C's XML Signature standard)".

This means that:

- The Signature element conforms to the W3C Signature Syntax and Processing Recommendation
- The BusinessMessage is the ISO 20022 Business Application Header and the ISO 20022 Message
- The person authorised to sign is not further specified; this depends on the particular context in which the message is sent.

As a reminder, the terminology of BusinessMessage, ISO 20022 Business Application Header and ISO 20022 Message is shown in the following picture

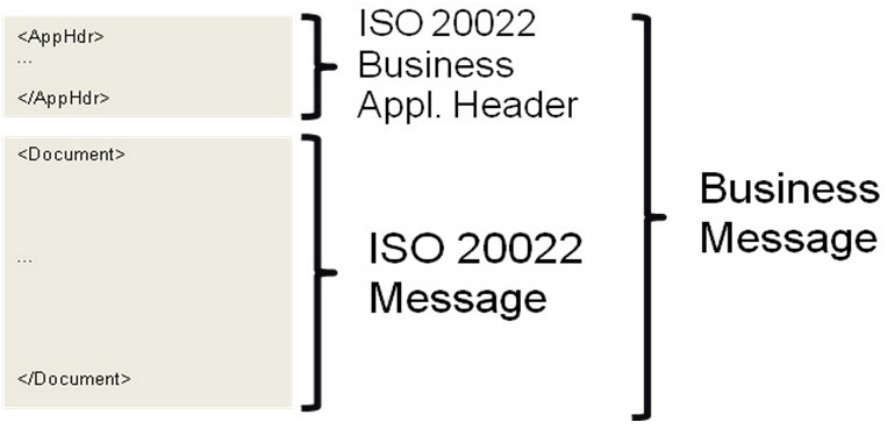


Figure 1 - Business Message

4.2.2 Short introduction to W3C Signatures

This short introduction provides an overview of signing and verifying based on the *W3C Signature Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008*.

The following diagram shows the process followed to sign and verify an XML digital signature.

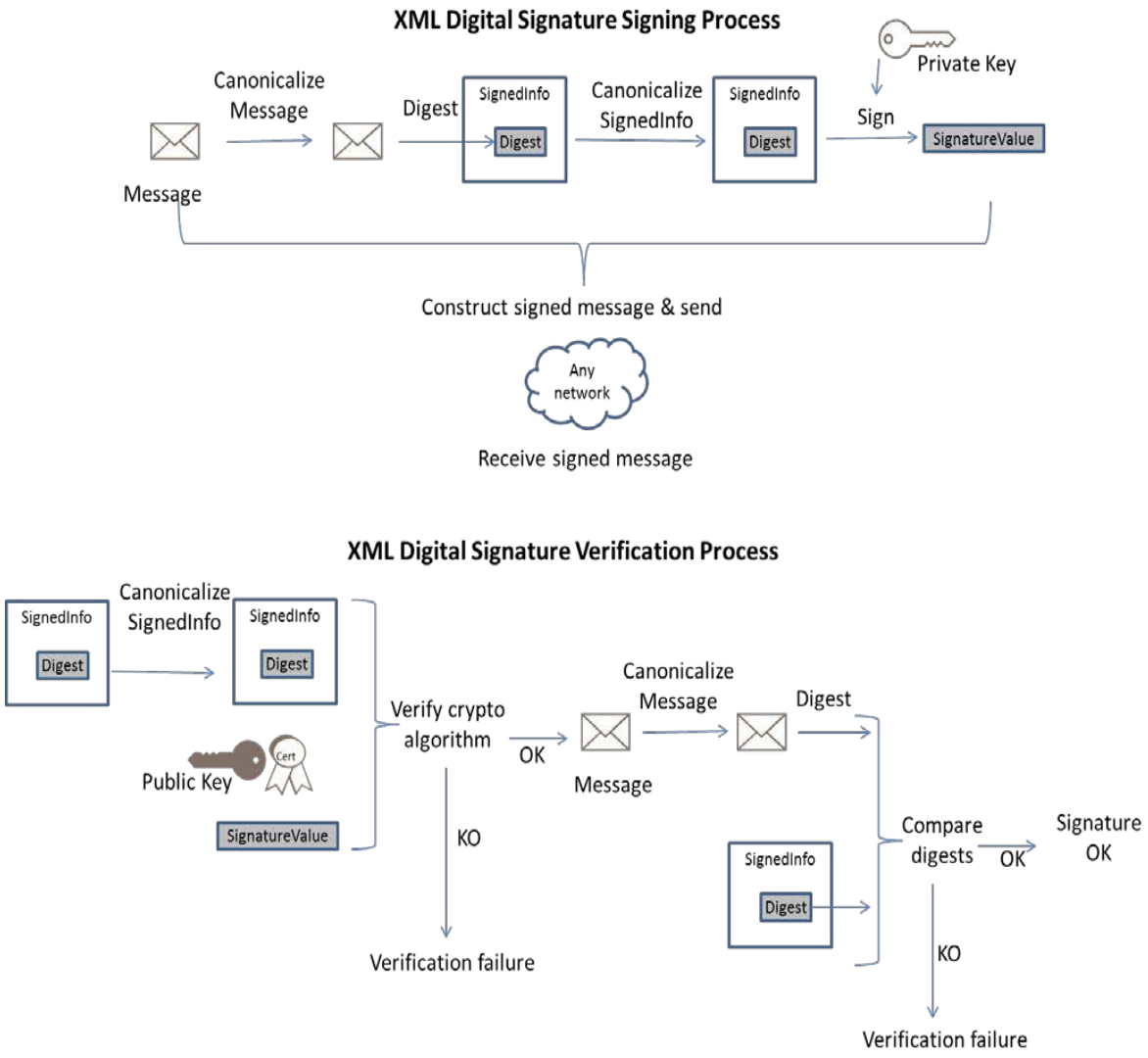


Figure 2 Signing and verifying process

The message that is to be signed is made available. The first step in the signing process is to canonicalize the message. This is a transformation that constructs the canonical form of the XML that is being signed. By doing so, the input of the digest algorithm is robust enough to be independent on the parsing technology used.

The calculated digest is placed into the DigestValue (a sub-element of SignedInfo). The exact mapping is made clear in the picture below. The SignedInfo element itself is canonicalized, and as last step the SignedInfo itself is signed using the signing algorithm. The result is placed in SignatureValue. The diagram shows a private key used for this purpose.

The message and its signature are then sent to the receiver. The verification of the Signature follows the reverse process. First, the SignatureValue is verified to be correct. The input to the verification algorithm is the canonicalized SignedInfo, the certificate containing the certified public key and the SignatureValue containing the result of the signing operation. In case the verification succeeds, the SignedInfo is authentic and signed by the signer. The verification process continues and checks if the digest within the SignedInfo matches the Message that is covered by the digest. This is done by canonicalizing the Message and then digest the canonicalized Message. The verification is ok if the calculated digest is equal to the digest within the SignedInfo.

The Signature elements used are illustrated in the following picture:

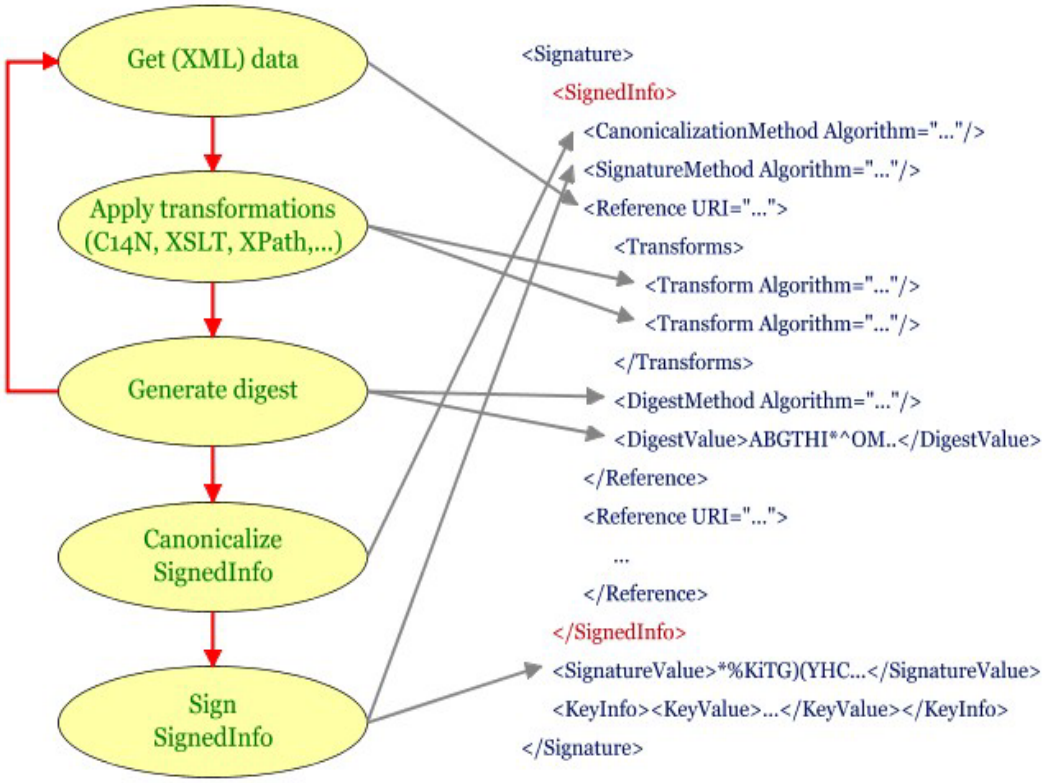


Figure 3 - XML Signature elements

As shown, the SignedInfo contains one or more Reference elements. Each Reference identifies an XML resource that is to be signed. The example in the picture shows a KeyInfo element with as content a KeyValue. Using a Public Key Infrastructure

4.2.3 Scope of this documents Signature Guidelines

This document does not introduce any change in business requirements related to the signature within the Business Application Header.

This document proposes the approach to implement the current signature guidelines. In particular, it addresses the following

- The structure of the Signature

- Algorithms used
- KeyInfo
- Structure of Reference elements

- The process used for signing and verifying

4.3 Structure of the Signature

4.2.4 Overview

This section describes the structure of the Signature signing the Business Message. The elements used within the Signature of the BAH are discussed. An example of a Signature used in the BAH can be found in Example on section Example.

The next section describes the processing of signing and verifying the Signature of a Business Message.

4.2.5 Algorithms used

Security algorithms are selected to be widely available but yet strong enough to offer sufficient security. It is possible that algorithms need to be updated to a stronger version, for example SHA3 for the digest algorithm. Applications should take this into account.

The following section shows an illustrative selection of algorithms:

- The signing algorithm is RSA based on the SHA-256 digest on the information signed
- The digest algorithm is SHA-256
- The canonicalization algorithm is the Exclusive XML Canonicalization. This algorithm is applied prior to digesting the XML. Using the canonicalization allows applications to use XML processing and parsing on the XML data without breaking the signature. The Exclusive Canonicalization algorithm is chosen to avoid any impact of enveloping elements when sending or receiving Business Messages.

This is shown in the following table:

Purpose	Element	Algorithm attribute value
PKI sign algorithm	SignMethod	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Digest algorithm	DigestMethod	http://www.w3.org/2001/04/xmlenc#sha256
Canonicalization algorithm	CanonicalizationMethod	http://www.w3.org/2001/10/xml-exc-c14n#

4.2.6 KeyInfo

The `KeyInfo` must contain the X509v3 certificate containing the validation key. Optionally, it can also contain the certificates that are part of the certification chain. This depends on the allowed PKI infrastructures used within the business context. For instance, when between sender and receiver always the same PKI infrastructure is used, then it can be better to use the CA root certificate from the environment than to check that the correct CA root certificate is used within the `KeyInfo`.

NOTE It is recommended not to use any other elements within the `KeyInfo`.

The `KeyInfo` itself is signed and must contain the `Id` attribute. This attribute should contain a unique value.

The KeyInfo looks like:

```
<ds:KeyInfo Id="Unique-id-to-KeyInfo">
 <ds:X509Data>
  <ds:X509Certificate>MIID1DCCArygAwIB...    </ds:X509Certificate>
 </ds:X509Data>
</ds:KeyInfo>
```

The X509Certificate contains the certificate that is used to verify the Signature. In case the full certification chain is to be present at least the CA root certificate is added as shown in the following example. The order of the different certificates is not relevant.

```
<ds:KeyInfo Id="Unique-id-to-KeyInfo">
 <ds:X509Data>
  <ds:X509Certificate>MIID1DCCArygAwIB...    </ds:X509Certificate>
  <ds:X509Certificate>MIIDkDCCAnigUoSI...    </ds:X509Certificate>
 </ds:X509Data>
</ds:KeyInfo>
```

NOTE Inclusion of the entire certification chain is recommended. Only when signatures must be performed using certified keys by a specific PKI infrastructure, only the signing certificate is sufficient.

4.2.7 SignedInfo

The `SignedInfo` contains the 3 `Reference` elements.

- `Reference` digesting the ISO 20022 Message
- Reference digesting the BAH
- Reference digesting the KeyInfo

The order of the `Reference` elements in the `SignedInfo` is not relevant.

4.2.8 Reference of ISO 20022 Message

The `Reference` is identified by the fact that the `URI` attribute is absent. The absence of the `URI` attribute indicates that the logic for de-referencing this reference is part of the application logic. The processing of such `Reference` is further described in section Process for signing and verifying.

NOTE Although the Business Message, as illustrated in Figure 1, consists of the BAH and the ISO 20022 Message, it is not possible to refer easily from the BAH to the ISO 20022 Message. Therefore, the option to use the absent `URI` attribute was selected, since this is possible on one `Reference` element within the `SignedInfo`.

The `Reference` looks like

```
<ds:Reference>
 <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
 </ds:Transforms>
 <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
 <ds:DigestValue>QYYVi9JdlsOxjplrW3vIjT8cWYyzYD4ZnnNJ9SH+dvQ=</ds:DigestValue>
</ds:Reference>
```

4.2.9 Reference of BAH

The `Reference` of the BAH is identified by the empty `URI` attribute `URI=""`.

Such empty URI attribute URI="" refers to the XML element in which the Signature is placed. This means that the signing and verifying of the Signature within the BAH happens within the context of the BAH.

The complete BAH is signed except the `Signature` itself. This is indicated in the first `Transform` algorithm within the `Transforms`. The second algorithm contains the Exclusive XML canonicalization.

```
<Reference URI="">
 <Transforms>
  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
  <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
 </Transforms>
 <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
 <DigestValue>+OS/MM1NBTiOZWWpvzOWkfRjyP2/F1lg9P+zvC+Gulk=</DigestValue>
</Reference>
```

4.2.10 Reference of KeyInfo

The reference to the `KeyInfo` requires that the `KeyInfo` can be identified with an `Id` attribute. To ensure that such an attribute can be handled correctly when the `Signature` is enveloped within other XML elements, that attribute should contain a unique value within the XML document enclosing the Signature.

NOTE The `Signature` is XAdES-BES compliant by adding the `Reference` within the `SignedInfo`.

```
<ds:Reference URI="#Unique-id-to-KeyInfo">
 <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
 </ds:Transforms>
 <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
 <ds:DigestValue>QYYVi9JdlsOxjplrW3vIjT8cWYyzYD4ZnnNJ9SH+dvQ=</ds:DigestValue>
</ds:Reference>
```

The optional attribute `Type="http://www.w3.org/2000/09/xmldsig#KeyInfo"` can be present on the `Reference` element referring to the `KeyInfo`. There is no obligation to further process this attribute.

```
<ds:Reference URI="#Unique-id-to-KeyInfo"
Type="http://www.w3.org/2000/09/xmldsig#KeyInfo">
 <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
 </ds:Transforms>
 <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
 <ds:DigestValue>QYYVi9JdlsOxjplrW3vIjT8cWYyzYD4ZnnNJ9SH+dvQ=</ds:DigestValue>
</ds:Reference>
```

4.4 Process for signing and verifying

4.2.11 Signing is performed as last step in generating the Business Message

When an application prepares a Business Message several steps may be needed. Indeed, some applications require an approval process where the message may be amended.

In some cases, some elements of the BAH may be created after the ISO 20022 Message has been created.

In any case, the last step in the creation of the Business Message is calculating the SignatureValue within the Signature. The creation of the signature can be implemented within the flow of sending the Business Message as long as the process as described in this section is applied.

4.2.12 Two node-sets

As illustrated in Figure 1, the Business Message consists of two XML documents. The way in which those two documents are created, sent, received and processed depends on the infrastructure used by the sender and receiver. It is therefore important that the process in creating the Signature is independent of the infrastructure used.

This is achieved by performing the Signature creation and verification using a well-defined XML context. Two such contexts are used:

- The context of the BAH element (AppHdr)
- The context of the ISO 20022 Message element (for instance Document)

4.2.13 Signing a BAH

The process of signing is illustrated in the following flowchart. The start is a Business Message composed of a BAH and an ISO 20022 Message. The Signature is created, similar to the example in next section. The DigestValue and SignatureValue have to be created. It is worth noting that the digest of the BAH is computed without the signature element. This process is described in the flowchart.

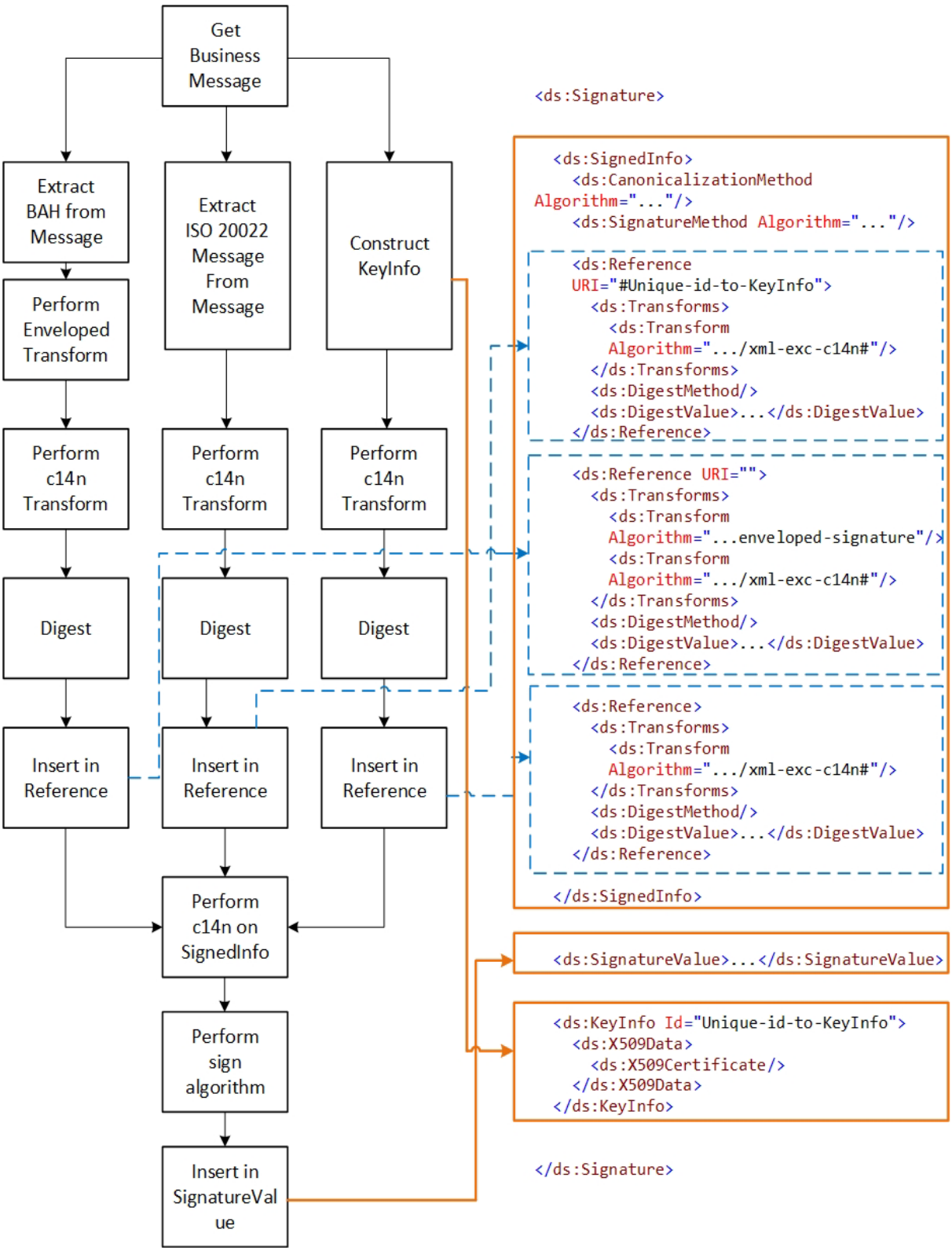


Figure 4 - Flowchart signing a BAH

Verifying a BAH Signature

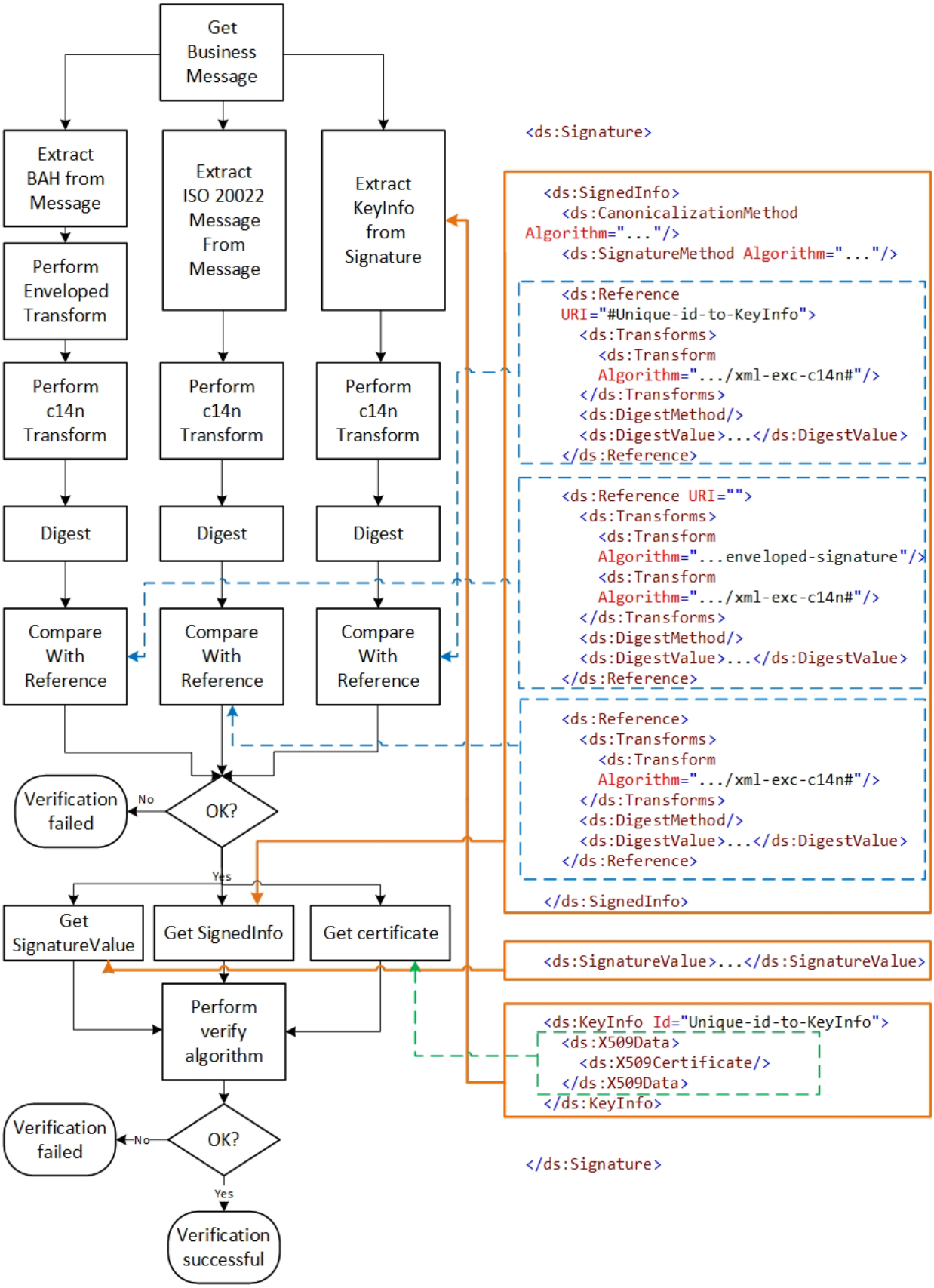


Figure 5 - Verifying a BAH signature

4.5 Example

An example of a `Signature` within the BAH that can be verified is as follows:

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
 <ds:SignedInfo>
  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
  <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
  <ds:Reference URI="#Unique-id-to-KeyInfo">
   <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
   </ds:Transforms>
   <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
   <ds:DigestValue> QYYVi9JdlsOxjplrW3vIjT8cWYyzYD4ZnnNJ9SH+dvQ=
</ds:DigestValue>
  </ds:Reference>
  <ds:Reference URI="">
   <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
   </ds:Transforms>
   <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
   <ds:DigestValue>
+OS/MM1NBTiOZWWpvzOWkfRjyP2/F1lg9P+zvC+Gulk=
</ds:DigestValue>
  </ds:Reference>
  <ds:Reference>
   <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
   </ds:Transforms>
   <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
   <ds:DigestValue> QYYVi9JdlsOxjplrW3vIjT8cWYyzYD4ZnnNJ9SH+dvQ=
</ds:DigestValue>
  </ds:Reference>
 </ds:SignedInfo>
 <ds:SignatureValue> IKDs7kwX14CxK3AZ1ojJLr35A4eU/98tp/10KFQTtPOwR5WCKyx
4I05ZV1IljOpEgpkt6xejXhshaEnNBD5B5PII1VN6mviJJjU/njGikNeXzi1Djei2dPEap
nPX1f26UnQcgYTAqaSVwAnIR7L8/W2UeT8J9z8Rd1OebYV5xE8jVehbgMcAmJwv2rC/c2d
UkUe2/eBU0APyWGCgKawxGGAPLP3AS4+Mp0ODKlVp08rUzVOF+pFF/1dBknlK/v0dWkDdj
YvwFRvZhHXue/PYvMNtQBytMUUdB1MiQrNX0jSCE6Y2nljhTXdcrb2lgfgwCclB6xArBRV
WfMa0kQVQ4Q==
</ds:SignatureValue>
 <ds:KeyInfo Id="Unique-id-to-KeyInfo">
  <ds:X509Data>
   <ds:X509Certificate>MIID1DCCArygAwIB </ds:X509Certificate>
  </ds:X509Data>
 </ds:KeyInfo>
</ds:Signature>
```

Revision Record

Revision	Date	Author	Description	Sections affected
1.0	15/04/2010	ISO 20022 RMG/TSG	Initial version	All
1.1	30/06/2010	TSG secretariat	corrected typos in some examples using T2S and added this revision record	2.1.1.1 and 2.1.6.1
1.2	20/10/2010	TSG secretariat	Corrected typos, clarified scenario 2, updated the TOC. Replaced MessageInstanceIdentifier by BusinessMessageIdentifier in chapter 3 Added new scenario 2.10	2.1.2, 3.x, 2.10, TOC
1.4	20/4/2011	TSG secretariat	corrected typos, added 1.7, corrected scenario 2.4	1.7, 2.4
1.5		TSG secretariat	Corrected typo; Added an XML Schema that formally describes an ISO 200222 Business Message.	2.3 Diagram 1.7
1.6	8/5/2013	TSG secretariat	Added ANNEX A - Signature Guidelines	4
1.7	23/5/2013	TSG secretariat	Added last paragraph about use of ISO 20022 XML syntax	1.1
1.8	8/4/2016	TSG	Added new section on BAH deployment options	2.2 and 2.3
1.9	15/10/2019	TSG	Updated for BAH V2	All
2.0	22/10/2020	TSG	Updated for BAH v2 Clarification of text, diagrams and formatting.	All
3.0	03/11/2021	RA	Updated for BAH v3	All

Disclaimer:

Although the Registration Authority has used all reasonable efforts to ensure accuracy of the contents of the iso20022.org website and the information published thereon, the Registration Authority assumes no liability whatsoever for any inadvertent errors or omissions that may appear thereon. Moreover, the information is provided on an "as is" basis. The Registration Authority disclaims all warranties and conditions, either express or implied, including but not limited to implied warranties of merchantability, title, non-infringement and fitness for a particular purpose. The Registration Authority shall not be liable for any direct, indirect, special or consequential damages arising out of the use of the information published on the iso20022.org website, even if the Registration Authority has been advised of the possibility of such damages.