

**ISO 20022  
Supplementary  
Data Frequently Asked  
Questions Version 1.7**

Date: Dec 2018

This FAQ for Supplementary Data was developed by the ISO 20022 Technical Support Group.

## Contents

1	INTRODUCTION .....	3
1.1	Purpose and use of this document .....	3
1.2	Intended audience .....	3
1.3	Terminology .....	3
2	Frequently Asked Questions .....	3
2.1	Question during development of an Extension MessageDefinition .....	3
2.1.1	Introduction .....	3
2.1.2	What is the SupplementaryData Component? .....	4
2.1.3	Where can I find the Supplementary Data component? .....	4
2.1.4	How does the validation work?.....	6
2.1.5	How do I make an ISO 20022 MessageDefinition extensible?.....	7
2.1.6	What is the ISO process to follow if I want to create an ISO 20022 Extension MessageDefinition? .....	7
2.1.7	How do I model an Extension MessageDefinition? .....	7
2.1.8	What if a community needs to extend a MessageDefinition in multiple locations, with different extension components?.....	8
2.1.9	Where do I describe which components are extended? .....	9
2.1.10	Some messages use a different extension component. Why is that?.....	9
2.1.11	Some Extension Components also contain a PlaceAndName. Why is that? .....	10
2.1.12	What is the format for the namespace? .....	10
2.2	Questions during implementation of an Extension MessageDefinition .....	10
2.2.1	Do I need to implement all Extension MessageDefinitions for a certain ISO 20022 MessageDefinition? .....	10
2.2.2	Am I protected from receiving extensions I don't support? .....	10
2.3	Questions when using an Extension Message .....	11
2.3.1	How and where do I declare the different namespaces? .....	11
2.3.2	How do I construct a PlaceAndName? .....	12
2.3.3	How does an ISO 20022 Message that uses an extension look like? .....	12
2.3.4	How does an ISO 20022 Message that uses several extensions look like? .....	14
2.3.5	How do I extend a repetitive MessageElement? .....	14

# 1 INTRODUCTION

## 1.1 Purpose and use of this document

This document clarifies, through a number of questions and answers, how to use the ISO 20022

Supplementary Data extension mechanism in ISO 20022 messages.

Additional documents, published by individual user communities, may be available that discuss the implementation of message extensions in a more specific context.

This document should serve as the general basis for the more specific community implementation guides that are developed.

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 depending on the demand for this in the ISO 20022 community.

## 1.2 Intended audience

- Submitters/developers of an Extension MessageDefinition
- Users/implementers of an extension

## 1.3 Terminology

### **Extension MessageDefinition**

formal description of an extension to a message

### **Extension Message**

valid instance of an Extension MessageDefinition

# 2 Frequently Asked Questions

## 2.1 Question during development of an Extension MessageDefinition

### 2.1.1 Introduction

Creating an Extension MessageDefinition is very similar to adding data to a MessageDefinition during a normal maintenance cycle. However, instead of adding this information into the ISO 20022

MessageDefinition (which is what normally happens during a maintenance), it is moved into another so-called Extension MessageDefinition, following some specific modeling rules. Furthermore, the Extension MessageDefinition can be created outside of a maintenance cycle, since it doesn't require any change to the MessageDefinition.

## 2.1.2 What is the SupplementaryData Component?

The SupplementaryData component is a technical container that allows adding information to a MessageDefinition in a controlled manner.

The presence of the SupplementaryData component in an ISO 20022 MessageDefinition is a pre-condition for that MessageDefinition to be extensible. So, without this component, a MessageDefinition cannot be extended.

The SupplementaryData component is composed of 2 parts:

- A (optional) part that refers to the MessageElement that is actually being extended
- A part that contains the actual extension data.

## 2.1.3 Where can I find the Supplementary Data component?

The SupplementaryData component is put at the end of the portion of the MessageDefinition that is extendable.

So when the SupplementaryData component is at the end of the message, then any element in the MessageDefinition may be extended.

When the SupplementaryData component is at the end of a MessageComponent (so not at the end of the message), then only that MessageComponent may be extended.

Such extension is also called a local extension. A MessageDefinition may contain more than one local extension.

A local extension may be useful to simplify it in repetitive messages (bulk messages, statements, mass transaction messages, etc.) where information in a repeating sequence may need to be extended. With an extension mechanism at the end of the message only, a message instance would be composed of two repeating parts: one part containing the repeating sequence from the core message and one part containing all the extensions (one for each occurrence in the core message).

## Message instance with extension outside repeating sequence

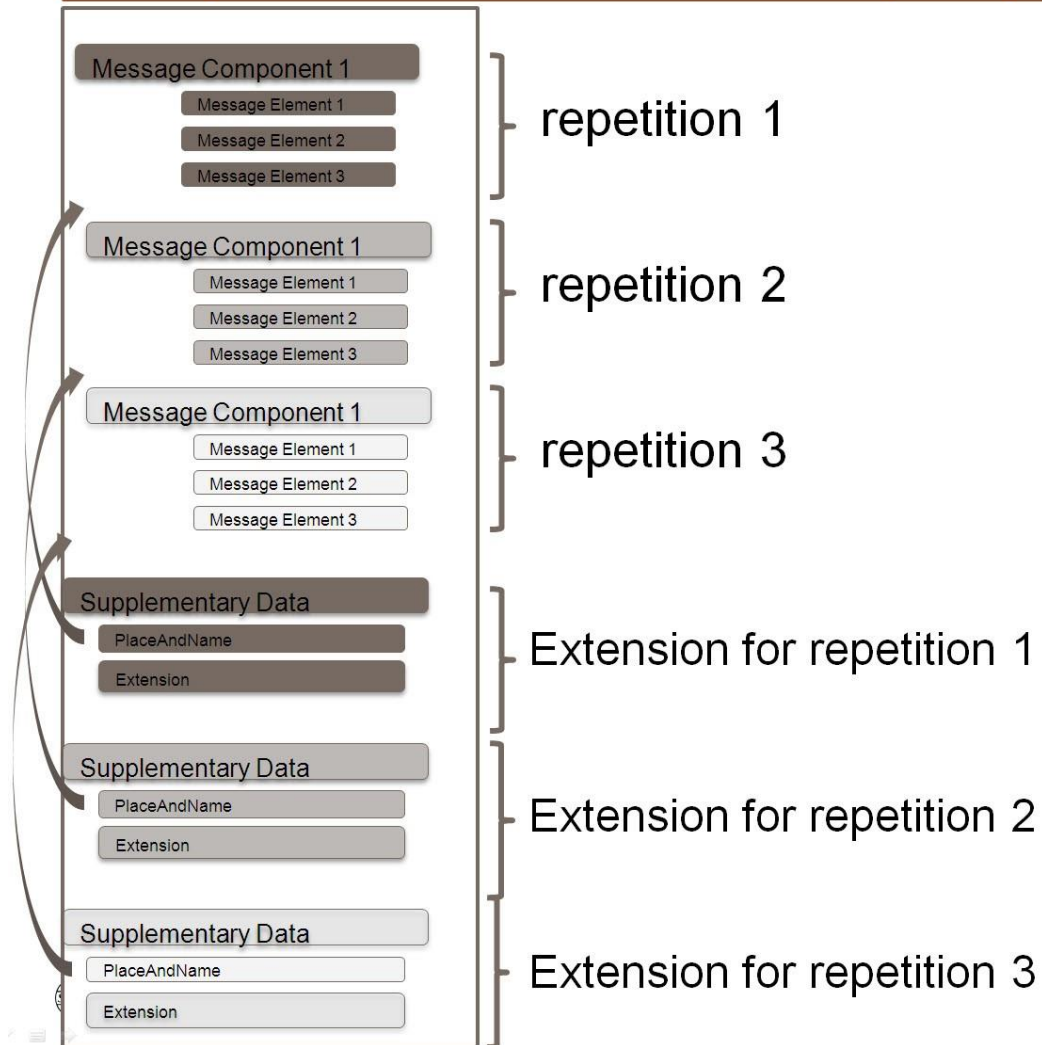


Figure 1 Repetition without local extension

Processing such a message might prove difficult as each repetition has to be reassembled by suffixing it with its corresponding extension.

A better alternative (see below depiction) is to (also) include the extension mechanism at the end of the repetitive sequence. In this way, a message instance is easier to process as each extension is joined together with the right repetition.

## Message instance with extension inside repeating sequence

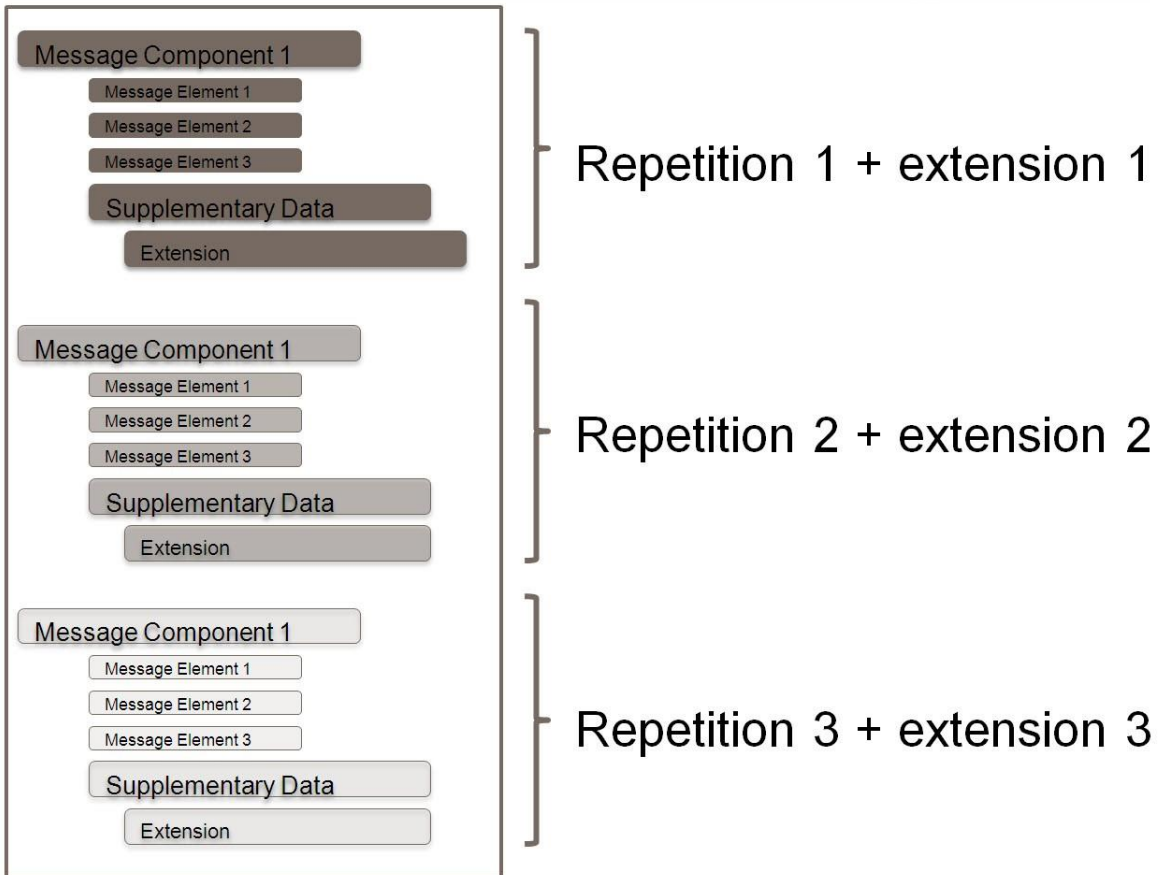


Figure 2 Repetition with local extension

### 2.1.4 How does the validation work?

A receiver will try to validate the content of SupplementaryData on a can-do basis: it will validate the extension if it has implemented the XML Schema that defines the content of the extension. Otherwise it will ignore the extension.

This mechanism is very useful in case there are parties who do not use/support the extension.

However, whenever there may be parties which do not use/support the extension, all content in the extension must have 'can ignore' semantics, i.e. it must be possible to process the message correctly when the extension is completely ignored. If the extension has 'must understand' semantics, i.e. the message cannot be processed correctly without processing the extension, then the extension should only be used within communities where all participants use/support the extension.

#### Example

Party A and Party B have an agreement to exchange an ISO 20022 Message with a specific extension. Party A generates the Extension Message and Party B uses the Extension Message. This means both parties have implemented the Extension MessageDefinition. If the message has to transit through Party C, who does not need to process the Extension Message but agreed to pass it as is to Party B, Party C will not need to implement the Extension MessageDefinition. Party C will process the core ISO 20022

Message, and just transmit the Message along with the Extension Message to Party B.

However, if the Extension Message must be forwarded by Party C to Party B using another MessageDefinition which also supports the Extension MessageDefinition (for example, in a scenario using a 'pain' message, followed by a 'pacs' message, followed by a 'camt' message), then Party C will need to move the Extension Message in the next Message in the chain and adapt the location element to refer to the message element that is extended in this new Message (see 2.3.2).

### 2.1.5 How do I make an ISO 20022 MessageDefinition extensible?

In order for an ISO 20022 MessageDefinition to be extensible, it must contain at least one SupplementaryData component. A series of ISO 20022 Securities MessageDefinitions and most Payments MessageDefinitions already include this component.

If the MessageDefinition does not contain a SupplementaryData component, you may introduce a request to add such a component to the MessageDefinition. If accepted, your request will require the maintenance of that MessageDefinition.

The [ISO 20022 maintenance process](#) is described on the ISO20022.org website.

### 2.1.6 What is the ISO process to follow if I want to create an ISO 20022 Extension MessageDefinition?

A prerequisite is that the ISO 20022 MessageDefinition can support extensions. In other words, it must have a SupplementaryData component.

The conditions and process for creating an ISO 20022 Extension MessageDefinition for use in a SupplementaryData component is described on the ISO20022.org website ([http://www.iso20022.org/supplementary\\_data.page](http://www.iso20022.org/supplementary_data.page)), but the main steps involve

- The organisation that wants to develop the Extension MessageDefinition submits a Change Request (CR) to the RA which forwards it to the appropriate SEG
- If the SEG approves the CR, the submitter may model and submit an Extension MessageDefinition to the RA. How to model an ISO 20022 compliant Extension MessageDefinition is described in chapter 2.1.7 below
- The RA will subsequently publish the Extension MessageDefinition on the ISO20022.org website ([http://www.iso20022.org/supplementary\\_data.page](http://www.iso20022.org/supplementary_data.page)) when it complies with the ISO 20022 modeling rules for Extension MessageDefinitions.

### 2.1.7 How do I model an Extension MessageDefinition?

Creating extensions to a MessageDefinition is not very different from maintaining that MessageDefinition.

The main difference is that the newly added components do not end up in the core MessageDefinition, but are grouped together in a separate Extension MessageDefinition. In this way, the core ISO 20022 MessageDefinition is not affected.

The ISO 20022 RA has, together with the ISO 20022 Technical Support Group, developed a number of modeling rules and guidelines specifically for the design of ISO 20022 Extension MessageDefinitions. You can find these rules on the iso20022.org website

([http://www.iso20022.org/documents/general/ISO20022\\_MasterRules.ZIP](http://www.iso20022.org/documents/general/ISO20022_MasterRules.ZIP)).

### 2.1.8 What if a community needs to extend a MessageDefinition in multiple locations, with different extension components?

Best practice is to define a single Extension MessageDefinition including all extension components. In such case, the SupplementaryData component is repeated for each extension location that is extended.

Below example shows an ISO 20022 MessageDefinition that is extended in two different locations, by two different Extension MessageComponents.

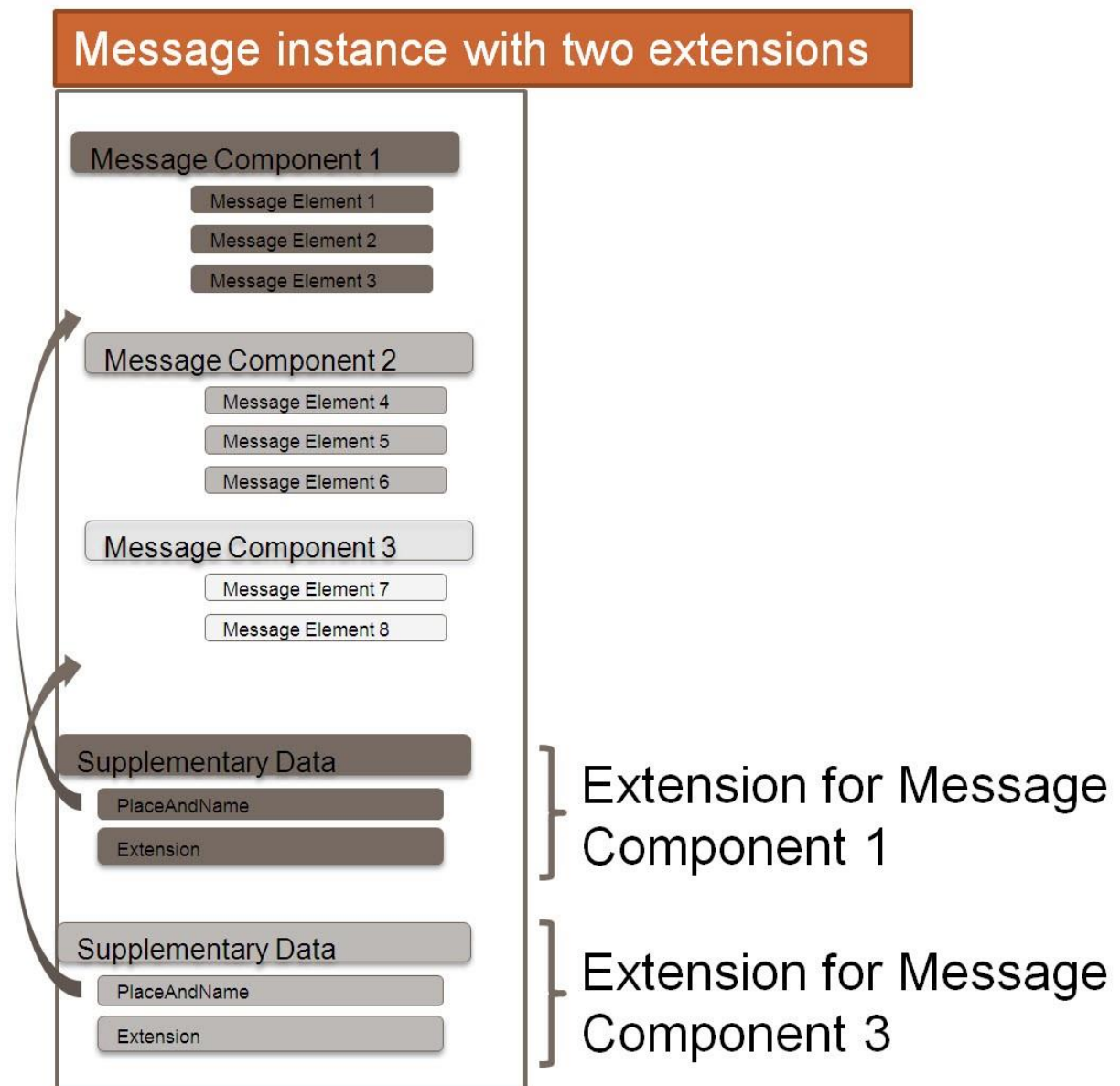


Figure 3 Extension MessageDefinition with two extensions



## 2.1.9 Where do I describe which components are extended?

The Message Definition Report of the Extension MessageDefinition must describe the location of the MessageElement(s) that is (are) extended in the core message.

## 2.1.10 Some messages use a different extension component. Why is that?

Some MessageDefinitions are still supplied with earlier versions of the SupplementaryData component: Extension1 and Extension2. These will be upgraded with the SupplementaryData component at the occasion of a forthcoming maintenance. Use of these extension mechanisms is subject to explicit authorization of the ISO 20022 Registration Authority.

Extension2 behaves exactly like SupplementaryData while Extension1 is not making use of an envelope to wrap an XML structure. Instead, Extension1 uses a Text field to contain the extension information.

This type of extension mechanism is of course limited in functionality as it cannot be used to specify a structured extension. It has also a length limitation.

### Example

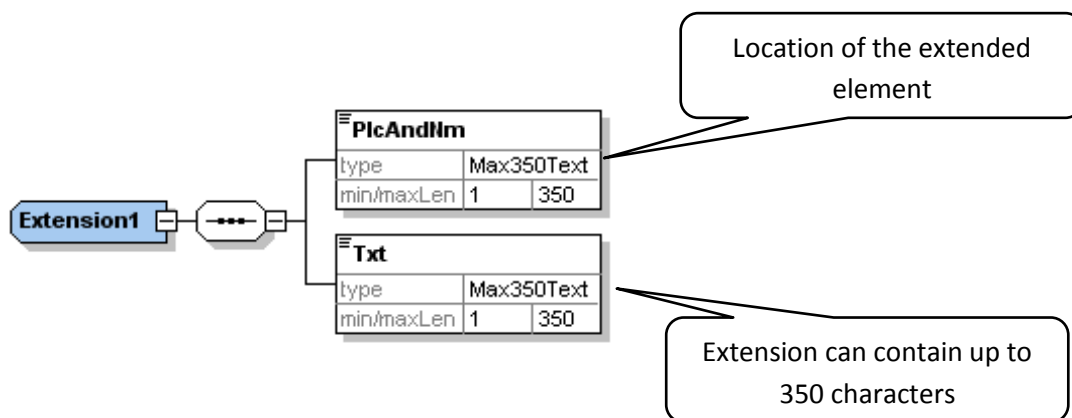


Figure 4 the Extension 1 component using Text

```
<Document xmlns = "urn:iso:std:iso:iso20022:tech:xsd:seev.039.001.01" >
<CorpActnNtfctn>
  <CorpActnOptnDtls>
    <CshMvmntDtls>Information that is extended</CshMvmntDtls>
  </CorpActnOptnDtls>
</CorpActnNtfctn>

<Xtsn>
  <PlcAndNm>CorpActnNtfctn/CorpActnOptnDtls/CshMvmntDtls</PlcAndNm>
  <Txt>Here is the extension information</Txt>
</Xtsn>
</Document>
```

If Extension1 is not large enough, attach several occurrences of this component to the same MessageElement and use the same path.

### Example

```
<Xtsn>
  <PlcAndNm>CorpActnNtfctn/CorpActnOptnDtIs/CshMvmntDtIs</PlcAndNm>
  <Txt>part 1 of the extension</Txt>
</Xtsn>
<Xtsn>
  <PlcAndNm>CorpActnNtfctn/CorpActnOptnDtIs/CshMvmntDtIs</PlcAndNm>
  <Txt>part 2 of the extension</Txt>
</Xtsn>
```

#### 2.1.11 Some Extension Components also contain a PlaceAndName. Why is that?

In exceptional cases, when an extension message contains many extension components extending many different locations, it may be more efficient to put the PlaceAndName element inside the extension component. It would avoid having to repeat the entire Extension Message for each extension component occurrence. In that case, it is allowed to use the PlaceAndName elements that are inside the extension components, and not use the PlaceAndName element from the SupplementaryData element.

#### 2.1.12 What is the format for the namespace?

The MessageIdentifier for Extension MessageDefinitions uses 'supl' as BusinessArea. ISO 20022 Extension MessageDefinitions adopt the same structure as other ISO 20022 MessageDefinitions.

Example: urn:iso:std:iso:iso20022:tech:xsd:supl.001.001.01

## 2.2 Questions during implementation of an Extension MessageDefinition

### 2.2.1 Do I need to implement all Extension MessageDefinitions for a certain ISO 20022 MessageDefinition?

NO. When a SEG approves the usage of one or more Extension MessageDefinitions for a (set of) MessageDefinition(s), it doesn't mean all users must implement these extensions.

It is up to a user community (e.g. a market infrastructure with its users, a closed user group) to decide if, when and which of these extensions they are going to implement.

MessageDefinitions that are provisioned with a SupplementaryData component are designed in such a way that users are not impacted when an Extension MessageDefinition is added. The use, processing and transmission of SupplementaryData extensions must be pre-agreed by all parties involved. Users are hence only impacted when they join the group of users (e.g. a Market Infrastructure, a community) that uses the specific extension(s).

### 2.2.2 Am I protected from receiving extensions I don't support?

A user may normally receive extended messages only if they have joined a user community that has decided to use extensions.

If a user receives an ISO 20022 Message that uses an extension it is not aware of, then it can simply ignore the extension portion and process the core ISO 20022 Message as if it didn't have an extension<sup>1</sup>.

---

<sup>1</sup> Note that there might be some resource effort required to implement this approach

A service provider could protect its service users from receiving unwanted extensions or ensure that only the right extensions are used.

## 2.3 Questions when using an Extension Message

### 2.3.1 How and where do I declare the different namespaces?

All namespaces must be qualified. In other words all XML elements in the XML document must use a namespace prefix

As a best practice, declare the namespace used in the extension at a common parent element between the extension and the extended element. In case the extension is at the end of the Message, then the common parent element is always the "Document" element of the core Message.

**Technical**

```
<core:Document
xmlns:core = "urn:iso:std:iso:20022:tech:xsd:seev.031.001.01"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xmlns:ext = "urn:iso:std:iso:iso20022:tech:xsd:supl.001.001.01">
<core:CorpActnNtfctn>
  --core data
</core:CorpActnNtfctn>

<core:SplmtryData>
  <core:PlcAndNm>core:CorpActnNtfctn/core:CorpActnOptnDtls[1]/core:CshMvmntDtls[2]</core:PlcAndNm>
  <core:Envlp>
  <ext:Document>
  <ext:CashMvmntDtls>
  <ext:MyExtensionData>This extends the second occurrence</ext:MyExtensionData>
  </ext:CashMvmntDtls>
  </ext:Document>
</core:Envlp>
</core:SplmtryData>
</core:Document>
```

Good

```
<Document
xmlns = "urn:iso:std:iso:20022:tech:xsd:seev.031.001.01"
xmlns = "http://www.w3.org/2001/XMLSchema-instance"
>
<CorpActnNtfctn>
  --core data
</CorpActnNtfctn>

<SplmtryData>
  <PlcAndNm>core:CorpActnNtfctn/core:CorpActnOptnDtls[1]/core:CshMvmntDtls[2]</PlcAndNm>
  <Envlp>
  <Document xmlns = "urn:iso:std:iso:iso20022:tech:xsd:supl.001.001.01">
  <CashMvmntDtls>
  <MyExtensionData>This extends the second occurrence</MyExtensionData>
  </CashMvmntDtls>
  </Document>
  </Envlp>
</SplmtryData>
</Document>
```

Bad

### 2.3.2 How do I construct a PlaceAndName?

When populating the location element, following rules must be followed:

1. When sending a Message this location element must contain the XPath of the extended MessageElement.
2. The XPath must be a valid XPath 2.0 expression.
3. A PlcAndNm must point to at least one valid location.
4. The root element of the XPath must be the element containing the namespace declaration of the extension. In case of a Message with the supplementary data component at the end, this will be the <Document> element.  
This means that where the Supplementary Data component is at the end of the Message, all XPaths in PlcAndNm must resolve the XPath starting from <Document>. Where the Supplementary Data component is local, all Xpaths in PlcAndNm must resolve the XPath starting from the common ancestor between the extension and the extended element.

#### Note

The PlcAndName element is necessary to remove ambiguity. When there is no ambiguity as to where the extension is for (this may for instance happen with a local extension), the PlcAndName may be omitted.

#### Example

Suppose a message with three MessageElements <a>, <b> and <c>. MessageElement <c> has two MessageElements <d> and <e>.

```
<Document>
  <a>
  <b>
  <c>
    <d>
    <e>
```

In case the Supplementary Data component is at the end of the Message:

If you want to refer to ME <b>

```
<PlaceAndName>b</PlaceAndName>
```

If you want to refer to ME <d>

```
<PlaceAndName>c/d</PlaceAndName>
```

In case the Supplementary data is local to <d>:

There is no need to mention a PlcAndName as it can only extend <d>.

### 2.3.3 How does an ISO 20022 Message that uses an extension look like?

The SupplementaryData component contains two pieces of information:

1. The location in the core message of the element that must be extended
2. The extension data

The structured extension makes use of another XML Schema (the Extension MessageDefinition Schema) to specify the extension.

**Technical**

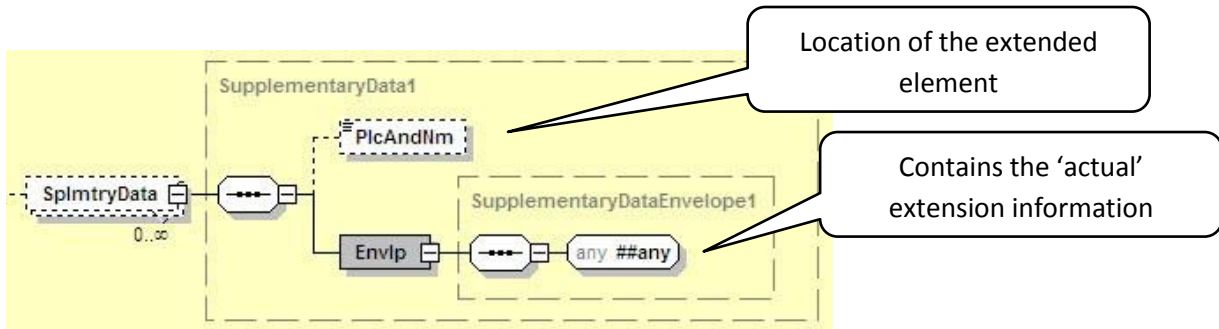


Figure 5 The supplementaryData component

```

<core:Document xmlns:core="urn:iso:std:iso:iso20022:tech:xsd:seev.031.001.03"
xmlns:ext="urn:iso:std:iso:iso20022:tech:xsd:supl.001.001.01">
<core:CorpActnNtfctn>
  <core:CorpActnOptnDtIs>
  <core:CshMvmntDtIs>
    Information that is extended
  </core:CshMvmntDtIs>
  </core:CorpActnOptnDtIs>
</core:CorpActnNtfctn>

<core:SplmtryData>
  <core:PlcAndNm>core:CorpActnNtfctn/core:CorpActnOptnDtIs/core:CshMvmntDtIs</core:PlcAndNm>
  <core:Envlp>
  <ext:Document>
  <ext:MyNewElement>abc</ext:MyNewElement>
  <ext:MyNewElement2>def</ext:MyNewElement2>
  </ext:Document>
  </core:Envlp>
</core:SplmtryData>
</core:Document>

```

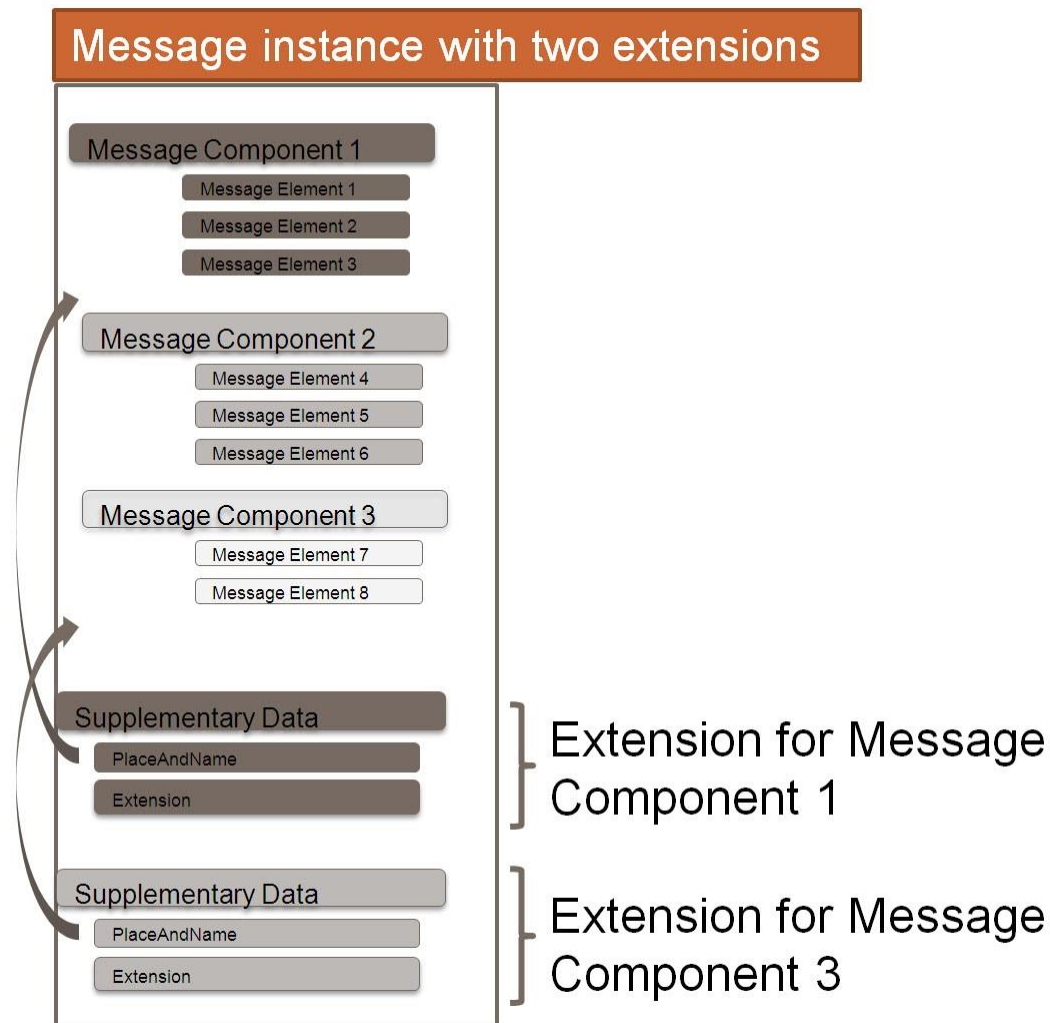
NOTES

- The extension itself is inside SplmtryData
- The PlcAndNm element points to the element of the core message that is extended, using a standard XML convention called XPath).
- The identification of the XML Schema that allows the receiver to know how to 'read' the extension information, is located after the Document element (in this example urn:iso:std:iso:iso20022:tech:xsd:supl.001.001.01)

**Technical**

### 2.3.4 How does an ISO 2022 Message that uses several extensions look like?

Example of an ISO 2022 Message that is extended at two locations by two different extensions components, each pointing to the location in the core message they are extending, using the PlaceAndName:



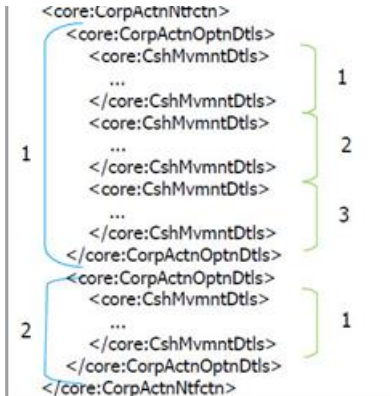
### 2.3.5 How do I extend a repetitive MessageElement?

The PlaceAndName element points to the location of the MessageElement that is extended. It is possible to specify the exact occurrence that is extended.

# Technical

The XPath syntax must be used to specify the XML path of the MessageElement that is extended. XPath notation allows specifying a specific occurrence of an element as a number between square brackets, after the name of the repeating element

## Example



In the above example, there are four CashMovementDetails elements in the Message (three in the first CorporateActionOptionDetails and one in the second CorporateActionOptionDetails). Below snippet specifies that the extension data in the Envelope applies to the second CashMovementDetails element in the first CorporateActionOptionDetails element:

```
<core:SplmtryData>
  <core:PlcAndNm>core:CorpActnNtfctn/core:CorpActnOptnDtls[1]/core:CshMvmntDtls[2]</core:PlcAndNm>
  <core:Envlp>
    <ext:Document>
      <ext:CashMvmntDtls>
        <ext:MyExtensionData>This extends the second occurrence</ext:MyExtensionData>
      </ext:CashMvmntDtls>
    </ext:Document>
  </core:Envlp>
</core:SplmtryData>
```

If you need to extend several CashMovementDetails elements, SupplementaryData would look like this:

```
<core:SplmtryData>
  <core:PlcAndNm>CorpActnNtfctn/CorpActnOptnDtls[1]/CshMvmntDtls[2]</core:PlcAndNm>
  <core:Envlp>
    <ext:Document">
      <ext:CashMvmntDtls>
        <ext:MyExtensionData>extends the second occurrence of CshMvmntDtls</ext:MyExtensionData>
      </ext:CashMvmntDtls>
    </ext:Document>
  </core:Envlp>
</core:SplmtryData>

<core:SplmtryData>
  <core:PlcAndNm>CorpActnNtfctn/CorpActnOptnDtls[1]/CshMvmntDtls[3]</core:PlcAndNm>
  <core:Envlp>
    <ext:Document>
      <ext:CashMvmntDtls>
        <ext:MyExtensionData>extends the third occurrence of CshMvmntDtls</ext:MyExtensionData>
      </ext:CashMvmntDtls>
    </ext:Document>
  </core:Envlp>
</core:SplmtryData>
```

First extension

Second extension

#### NOTES

- Without the index, this same extension would be applied to all 4 locations
- Index starts from [1], so not from [0]
- In some MessageDefinitions a SupplementaryData component may be available at the level of the repetitive sequence



# Appendix A Example

---

This example explains how adding the SupplementaryData component together with the transaction instead of putting it at the end of the message, would improve processing of that message.

Suppose a transaction based message such as the BankToCustomerStatement message (camt.053.001.03) that currently contains SupplementaryData only at the end of the message.

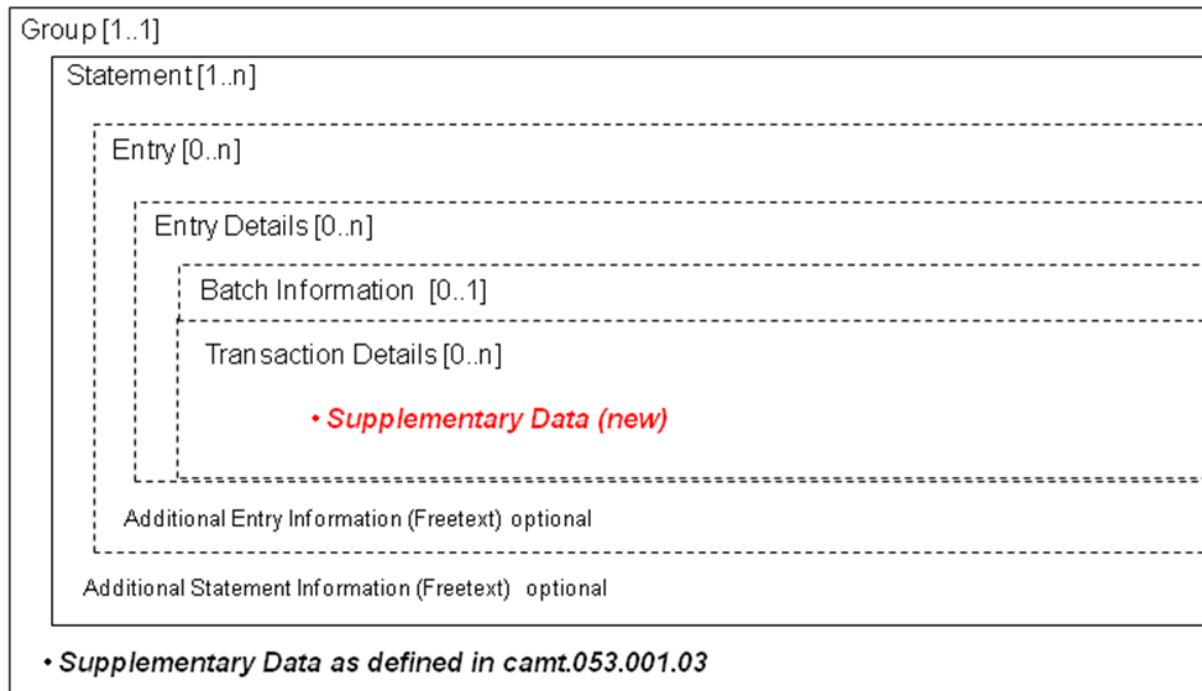


Figure 6 Logical structure of a camt.053.001.03

```

<BkToCstmStm t>
  <GrpHdr>
  ...
  </GrpHdr>
  <Stm t>
  ...
  <Ntry>
  <Amt Ccy="EUR">100000.50</Amt>
  ...
  <NtryDtls>
    <TxDtls>
      ...
      <SpIntryData>
        ... Supplementary Data Transaction 1
      </SpIntryData>
    </TxDtls>
  </NtryDtls>
</Ntry>
  <Ntry>
  <Amt Ccy="EUR">200000.50</Amt>
  ...
  <NtryDtls>
    <TxDtls>
      ...
      <SpIntryData>
        ... Supplementary Data Transaction 2
      </SpIntryData>
    </TxDtls>
  </NtryDtls>
</Ntry>
</Stm t>
</BkToCstmStm t>

```

Figure 7 Message instance containing two transactions with local extensions

## Revision Record

Revision	Date	Author	Description	Sections affected
0.x	Oct 2011	SWIFT	draft	All
0.6	Dec 2011	RA	draft	All
0.8	Dec 2011	RA	draft	All
1.0	May 2012	TSG	includes TSG comments	All
1.1	June 2012	TSG	adding a transaction based example in the extension	Appendix A
1.2	June 2012	TSG	removing ".xsd" in some urn + typo's	1.3, 2.1.3, 2.1.9, 2.3.1, 2.3.3, 2.3.5,
1.3 and 1.4	September 2012	TSG	Cosmetic changes + removal of references to PlaceAndName in Extension components.	All
1.5	May 2013	TSG	Added last paragraph about use of ISO 20022 XML syntax	1.1
1.6	Aug 2017	TSG	Clarified use of XPath and use of PlaceAndName + cosmetic changes	2.1.11 and 2.3.2
1.7	Dec 2018	RA	Update MX identifier in example	2.3.1 and 2.3.3

## 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.