ISO 20022
Supplementary Data
Frequently Asked Questions
Version 1.5

Date: May 2013

This FAQ for Supplementary Data was developed by the ISO 20022 Technical Support Group.
Table of 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? .........................................................3
     2.1.3 Where can I find the Supplementary Data component? ........................................4
     2.1.4 How does the validation work? ..............................................................................5
     2.1.5 How do I make an ISO 20022 MessageDefinition extensible? ............................6
     2.1.6 What is the ISO process to follow if I want to create an ISO 20022 Extension
          MessageDefinition? .................................................................................................6
     2.1.7 How do I model an Extension MessageDefinition? ...............................................6
     2.1.8 What if a community needs to extend a MessageDefinition in multiple locations, with
          different extension components? .............................................................................6
     2.1.9 Where do I describe which components are extended? ........................................7
     2.1.10 Some messages use a different extension component. Why is that? .....................8
     2.1.11 What is the format for the namespace? ................................................................9
  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 ................................................................10
     2.3.1 How and where do I declare the different namespaces? .........................................10
     2.3.2 How do I construct a PlaceAndName? ..................................................................11
     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? ............13
     2.3.5 How do I extend a repetitive MessageElement? ..................................................13
A.1. TRANSACTION BASED EXTENSION IN PAYMENTS CARDS ........................................16
Revision Record .....................................................................................................................20
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?

When available, the Supplementary Data component is put at the end of the portion of the Message Definition that is extendable. This usually means one can find a Supplementary Data component at the end of the Message Definition, to allow for any part of the Message to be extensible.

In some cases, the Message Definition may include a 'local' Supplementary Data component, i.e. not at the end of the Message Definition, in case of 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).

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.
Example

You can find a complete business example in Appendix A.

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), 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) 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).

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.

![Message instance with two extensions](image)

**Figure 1 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

![Diagram of Extension1 component using Text](image)

**Figure 2 the Extension1 component using Text**

```xml
<Doc xmlns:urn:iso:std:iso20022:tech:xsd:seev.039.001.01>
   <CorpActnNtfctr>
      <CorpActnOptnDtls>
         <CshMvmntDtls>
            Information that is extended
            <CshMvmntDtls/>
         </CorpActnOptnDtls>
      </CorpActnOptnDtls>
   </CorpActnNtfctr>

   <Xtn>
      <PlcAndNm>CorpActnNtfctr/CorpActnOptnDtls/CshMvmntDtls</PlcAndNm>
      <Txt>Here is the extension information</Txt>
      <Xtn/>
   </Xtn>
</Doc>
```

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

Example

```xml
<Xtn>
   <PlcAndNm>CorpActnNtfctr/CorpActnOptnDtls/CshMvmntDtls</PlcAndNm>
   <Txt>part 1 of the extension</Txt>
</Xtn>
<Xtn>
   <PlcAndNm>CorpActnNtfctr/CorpActnOptnDtls/CshMvmntDtls</PlcAndNm>
   <Txt>part 2 of the extension</Txt>
</Xtn>
```
2.1.11 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: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.\(^1\)

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?

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

2. As a best practice, declare all used namespaces (so also the namespace(s) used in the extension) at the root element of the document.

\(^1\) Note that there might be some resource effort required to implement this approach
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.
   This means that all XPaths in PlcAndNm must resolve the XPath starting from <Document>.

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

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

If you want to refer to ME <b>
<PlaceAndName>b</PlaceAndName>

If you want to refer to ME <d>
<PlaceAndName>c/d</PlaceAndName>
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.

![Figure 3 the SupplementaryData component](image)

```xml
<core:Document xmlns:core="urn:iso:std:iso20022:tech:xsd:seev.039.001.03"
               xmlns:ext="urn:iso:std:iso20022:tech:xsd:supl.001.001.01">
  <core:CorpActnNtfctn>
    <core:CorpActnOprtnDtls>
      <core:CshMvmtDtls>
        <Information that is extended />
        </core:CshMvmtDtls>
    </core:CorpActnOprtnDtls>
  </core:CorpActnNtfctn>

  <core:SplmtryData core:CorpActnNtfctn/core:CorpActnOprtnDtls/core:CshMvmtDtls(core:PlcAndNm core:Envelope
)
  <ext:Document>
    <ext:MyNewElement>abc</ext:MyNewElement>
    <ext:MyNewElement2>def</ext:MyNewElement2>
  </ext:Document>

  </core:Envelope>
  </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:iso20022:tech:xsd:supl.001.001.01)
2.3.4 How does an ISO 20022 Message that uses several extensions look like?

Example of an ISO 20022 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:

![Message instance with two extensions]

- **Extension for Message Component 1**
  - Supplementary Data
    - PlaceAndName
    - Extension

- **Extension for Message Component 3**
  - Supplementary Data
    - PlaceAndName
    - Extension

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.

*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:Emvlp>
    <ext:Document>
      <ext:CshMvmntDtls>
        <ext:MyExtensionData>This extends the second occurrence</ext:MyExtensionData>
      </ext:CshMvmntDtls>
    </ext:Document>
  </core:Emvlp>
<core:SplmtryData>
```
If you need to extend several CashMovementDetails elements, SupplementaryData would look like this:

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

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

**Important notes**

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

A.1. TRANSACTION BASED EXTENSION IN PAYMENTS CARDS

This example explains how adding the SupplementaryData component together with the transaction instead of 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 4 Logical structure of a camt.053.001.03
Figure 5 Message instance containing two transactions with extensions

The processing of such messages requires the individual transactions to be isolated and regrouped into new messages, both by the receiving Clearing House as well as the Creditors Bank, as shown in the following processing steps:

1) Individual Customers send their payments to their bank. The bank resorts/regroups and sends the re-bulked messages to a CSM (Clearing and Settlement Mechanism)
2) The CSM in its turn regroups the messages and sends them out to the receiving banks

3) The Creditors bank regroups the incoming messages and delivers to the individual customers
null
## Revision Record

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Author</th>
<th>Description</th>
<th>Sections affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.x</td>
<td>Oct 2011</td>
<td>SWIFT</td>
<td>draft</td>
<td>All</td>
</tr>
<tr>
<td>0.6</td>
<td>Dec 2011</td>
<td>RA</td>
<td>draft</td>
<td>All</td>
</tr>
<tr>
<td>0.8</td>
<td>Dec 2011</td>
<td>RA</td>
<td>draft</td>
<td>All</td>
</tr>
<tr>
<td>1.0</td>
<td>May 2012</td>
<td>TSG</td>
<td>includes TSG comments</td>
<td>All</td>
</tr>
<tr>
<td>1.1</td>
<td>June 2012</td>
<td>TSG</td>
<td>adding a transaction based example in the extension</td>
<td>Appendix A</td>
</tr>
<tr>
<td>1.2</td>
<td>June 2012</td>
<td>TSG</td>
<td>removing &quot;.xsd&quot; in some urn + typo's</td>
<td>1.3, 2.1.3, 2.1.9, 2.3.1, 2.3.3, 2.3.5,</td>
</tr>
<tr>
<td>1.3 and 1.4</td>
<td>September 2012</td>
<td>TSG</td>
<td>Cosmetic changes + removal of references to PlaceAndName in Extension components.</td>
<td>All</td>
</tr>
<tr>
<td>1.5</td>
<td>May 2013</td>
<td>TSG</td>
<td>Added last paragraph about use of ISO 20022 XML syntax</td>
<td>1.1</td>
</tr>
</tbody>
</table>

**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.