



# Peppol

The future is open

## Migration guide for Service Providers Peppol Service Metadata Publisher (SMP) specification from v1.1.0 to v1.2.0

Version: January 24<sup>th</sup>, 2022

**OpenPeppol AISBL**  
Rond-point Schuman 6, box 5  
1040 Brussels Belgium

[info@peppol.eu](mailto:info@peppol.eu)  
[www.peppol.eu](http://www.peppol.eu)  
Last updated: 24.01.2022

## Table of Contents

|     |  |   |
|-----|--|---|
| 1   | Introduction .....   | 3 |
| 1.1 | Audience .....   | 3 |
| 2   | Changes between the versions .....   | 3 |
| 2.1 | Updated the references .....   | 3 |
| 2.2 | Explicitly allowing Content-Type “application/xml” as it is equivalent to “text/xml” ..... | 4 |
| 2.3 | Change “is not” to “MUST NOT” in chapter 5.5 .....   | 4 |
| 2.4 | Replaced the references to the BusDox Common Definition document .....                     | 5 |
| 2.5 | Added clarifications on ServiceActivationDate and ServiceExpirationDate ...                | 6 |
| 2.6 | Linking peppol-smp-types-v1.xsd in the Appendix .....                                      | 7 |
| 2.7 | Fixed a typo in the name of the transformation .....                                       | 8 |
| 2.8 | Changed the Canonicalization Algorithm from “Exclusive” to “Inclusive” .....               | 8 |
| 3   | Migration Period .....   | 9 |

## 1 Introduction

OpenPeppol released the new version 1.2.0 of the “Service Metadata Publishing (SMP)” document. The previously active version was 1.1.0, published in August 2012.

The official source of the document is at <https://docs.peppol.eu/edelivery/>

This document highlights the changes between these two versions and points to the affected Peppol components.

### 1.1 Audience

This document describes changes to a Peppol policy and guidelines for use of identifiers within the Peppol network. The intended audience for this document are organizations wishing to be Peppol enabled for exchanging electronic documents, and/or their ICT-suppliers. More specifically it is addressed towards the following roles:

- ▶ ICT Architects
- ▶ ICT Developers
- ▶ Business Experts

## 2 Changes between the versions

The Revision History of version 1.2.0 list the following changes compared to version 1.1.0:

- 1) Updated the references
- 2) Improved layout
- 3) Explicitly allowing Content-Type “application/xml” as it is equivalent to “text/xml” (chapter 5.1)
- 4) Removing the requirement that the encoding attribute value is case sensitive (chapter 5.2)
- 5) Change “is not” to “MUST NOT” in chapter 5.5
- 6) Replaced the references to the BusDox Common Definition document (BDEN-CEDF)
- 7) Added clarifications on ServiceActivationDate and ServiceExpirationDate
- 8) Linking peppol-smp-types-v1.xsd in the Appendix
- 9) Fixed a typo in the name of the transformation
- 10) Changed the Canonicalization Algorithm from “Exclusive” to “Inclusive”

Change 2) has no impact to users.

### 2.1 Updated the references

The following changes were done to the normative references:

- ▶ The link of the [XML-DSIG] entry was changed from <http://www.w3.org/TR/xmlsig-core/> to <https://www.w3.org/TR/xmlsig-core1/> to ensure that we're using the major version 1 only, in case there will ever be a version 2.
- ▶ The entry [BDEN-CDEF] was removed, because the document is no longer referenced. See chapter 2.4 for details.
- ▶ Added entry [PFUOI4] referencing the “Peppol Policy for use of Identifiers” as a new normative reference. This was only missing because the last update of the SMP specification was too long ago.

These changes are considered to have no impact, as they are purely clarifications without any intended side effects.

## 2.2 Explicitly allowing Content-Type “application/xml” as it is equivalent to “text/xml”

Version 1.1.0, chapter 5.1:

A service implementing the REST binding **MUST** set the HTTP “content-type” header, and give it a value of “text/xml”.

Version 1.2.0, chapter 5.1:

A service implementing the REST binding **MUST** set the HTTP Content-Type header, and give it a value of `text/xml` **or** `application/xml`.

The intention of this change is to allow different Content-Types in the result, as the new “application/xml” value is much more common than the old-style “text/xml”. RFC 7303<sup>1</sup> “XML Media Types” defines “text/xml” to be an alias to “application/xml” hence they are equivalent.

Effected components:

- ▶ SMP servers may now use “application/xml” as the “Content-Type” for the predefined queries.
- ▶ SMP clients must eventually be altered to not enforce the “text/xml” “Content-Type” but also allow the “application/xml” “Content-Type”. This kind of check is deemed to be rare on the sender side, hence this change is considered to have a low impact on Service Providers.

## 2.3 Change “is not” to “MUST NOT” in chapter 5.5

Version 1.1.0, chapter 5.5:

---

<sup>1</sup> See <https://www.rfc-editor.org/rfc/rfc7303>

At the transport level, the service is not secured.

Version 1.2.0, chapter 5.5:

At the transport level, the service **MUST NOT** be secured.

The intention of this change is to enforce RFC 2119 terminology.

This has no impact on Service Providers.

## 2.4 Replaced the references to the BusDox Common Definition document

The “BusDox Common Definition” document was a relict from the original PEPPOL project specifications. It has been superseded by the “Peppol Policy for use of Identifiers” document.

The usages of the [BDEN-CDEF] in version 1.1.0 were:

- ▶ Chapter 1.4 “Terminology”
  - The respective terminology was copied into v1.2.0.
- ▶ Chapter 1.5 “Namespaces”
  - The relevant namespaces were copied into v1.2.0.
- ▶ Chapter 4.2, field “ParticipantIdentifier”
  - This reference was changed to point to the “Peppol Policy for use of Identifiers” instead.
- ▶ Chapter 4.3, mentioning the “percent encoding of URLs”
  - The explicit bibliographic reference was removed, as it was used in the context of an example only.
- ▶ Chapter 4.3, fields “ServiceInformation/ParticipantIdentifier”, “ServiceInformation/DocumentIdentifier”, “ServiceInformation/ProcessList” and “/ProcessList/Process/ProcessIdentifier”
  - These references were changed to point to the “Peppol Policy for use of Identifiers” instead.
- ▶ Chapter 5.3, mentioning the “percent encoding of URLs”
  - This section was converted to chapter 5.3.1 in v1.2.0. This reference was changed to RFC 3986<sup>2</sup> “Uniform Resource Identifier (URI): Generic Syntax” instead.
- ▶ Chapter 5.3, section “Using identifiers in the REST Resource URLs”
  - This section was converted to chapter 5.3.2 in v1.2.0. This reference was changed to point to the “Peppol Policy for use of Identifiers” instead.

---

<sup>2</sup> See <https://www.rfc-editor.org/rfc/rfc3986>

These changes are considered to be fully transparent to Service Providers and are mainly closing the gap between the specification and reality.

This impact is meant to have no impact on Service Providers.

Note: this exercise was also performed for the Peppol SML specification, therefore rendering the “BusDox Common Definition” document unused in the Peppol eDelivery network.

## 2.5 Added clarifications on ServiceActivationDate and ServiceExpirationDate

This change affects the interpretation of two elements of the “Endpoint” data type.

Version 1.1.0, chapter 4.3:

|   |  |
|---|--|
| ServiceInformation/<br>ProcessList/./Endpoint/<br>ServiceActivationDate | Activation date of the service. Senders should ignore services that are not yet activated. Format of ServiceActivationDate date is xs:dateTime |
| /ProcessList/./Endpoint/<br>ServiceExpirationDate                       | Expiration date of the service. Senders should ignore services that are expired. Format of ServiceExpirationDate date is xs:dateTime.          |

Version 1.2.0, chapter 4.3:

|                                |  |
|--------------------------------|--|
| Endpoint/ServiceActivationDate | Activation date of the service. Senders <b>MUST</b> ignore services that are not yet activated. <b>A missing activation date MUST be interpreted as “valid since forever”.</b> Format of ServiceActivationDate is xs:dateTime. |
| Endpoint/ServiceExpirationDate | Expiration date of the service. Senders <b>MUST</b> ignore services that are expired. <b>A missing expiration date MUST be interpreted as “valid until eternity”.</b> Format of ServiceExpirationDate is xs:dateTime.          |

This change is intended to clarify the usage of the “ServiceActivationDate” and the “ServiceExpirationDate” element. With the old version of the specification, it was unclear how to interpret these fields and it led to a couple of Service Desk questions.

This change impacts the following components:

- ▶ SMP server implementations need to follow these new rules and hence may ban values like “01-01-1900” for an “early service activation date” and “31-12-

9999” for a late expiration. If the service activation date is not in the future, it is recommended to not use this field at all. If the service expiration date is undefined, it is recommended to not use this field at all.

- ▶ SMP client implementations are now forced to interpret these fields accordingly
  - The need to reject endpoints that are not yet active (ServiceActivationDate is provided AND ServiceActivationDate is after now).
  - The need to reject endpoints that are no longer active (ServiceExpirationDate is provided and before now).
  - In all other cases, the endpoint needs to be accepted.
  - Note: the values of “ServiceActivationDate” and “ServiceExpirationDate” that exactly match the current date and time (up to second precision) are considered to be “valid endpoint” timestamps.

Depending on the current interpretation of SMP server and SMP client, these potential scenarios may exist:

|  | Receiving SMP endpoint has no dates | Receiving SMP endpoint has dates specified that include the current date | Receiving SMP endpoint has dates specified that exclude the current date |
|--|-------------------------------------|--|--|
| Sending AP was interpreting the fields     | No change                           | No change  | No change  |
| Sending AP was NOT interpreting the fields | No change                           | No change  | Document exchange will not take place                                    |

Only in the case where an SMP endpoint has a defined date range that excludes the current date and the SMP lookup client was previously not interpreting these two fields an action is required. In all other cases, the existing scenarios will continue to work without any change.

## 2.6 Linking peppol-smp-types-v1.xsd in the Appendix

Version 1.1.0 contains the XML Schema for the SMP specific elements as Appendix A.

Version 1.2.0 still contains the XML Schema as Appendix A, but marks this section as “non-normative”. The normative version of the XML Schema is now to be distributed together with the SMP specification. The reason for this is, that the XML format can be sensitive to whitespaces, and copying from a PDF document does not necessarily preserve the whitespaces as intended.

This impact is meant to have no impact on Service Providers.

## 2.7 Fixed a typo in the name of the transformation

Version 1.1.0, chapter 5.5:

```
http://www.w3.org/2000/09/xmldsig#envelopedsignature
```

Version 1.2.0, chapter 5.5.1:

```
http://www.w3.org/2000/09/xmldsig#enveloped-signature
```

This change closes the gap between reality and specification. The algorithm name in the old specification is wrong, and there exists no algorithm with this name. All the implementations known use the correct algorithm name. Hence this change is considered to have no impact on Service Providers.

## 2.8 Changed the Canonicalization Algorithm from “Exclusive” to “Inclusive”

Version 1.1.0, chapter 5.5:

```
The canonicalization algorithm MUST be http://www.w3.org/2001/10/xml-exc-c14n#
```

Version 1.2.0, chapter 5.5.1:

```
The canonicalization algorithm MUST be http://www.w3.org/TR/2001/REC-xml-c14n-20010315
```

Version 1.1.0 was using “Exclusive XML Canonicalization 1.0 (omit comments)” whereas version 1.2.0 is now using “Canonical XML 1.0 (omit comments)”<sup>3</sup>.

The original algorithm choice taken back in the PEPPOL project was poor, because the interoperability of the “Exclusive XML Canonicalization” is low. By switching to the “Inclusive” canonicalization algorithm we are

- ▶ Improving interoperability between heterogenous implementations
- ▶ Align ourselves with the algorithm choice of the OASIS BDXR SMP v1 specification

This change impacts the following components:

---

<sup>3</sup> See <https://lists.w3.org/Archives/Public/w3c-ietf-xmldsig/2002JanMar/0161.html> for a detailed explanation on the differences. Further limitations are described at <https://www.w3.org/TR/xml-exc-c14n/#sec-Limitations>



- ▶ SMP server implementations need to change the Canonicalization Algorithm when creating the digital signature on SMP responses.

In the wild, both canonicalization algorithms were already used and no complaint was ever filed about some SMP server using the wrong canonicalization algorithm. Our assumption is, that implementations verifying digital signatures can handle both algorithms. Hence, this change is considered to only have low impact on Service providers.

### 3 Migration Period

The new Peppol SMP specification comes into effect on 10 December 2021. The migration period for Service Providers is at least 3 months, meaning all changes need to be in effect on 1 May 2022.

All SMP lookup clients (as used in Access Points) have to make sure, they honour the “ServiceActivationDate” and “ServiceExpirationDate” as described in chapter 2.5.

All SMP server operators should check the “ServiceActivationDate” and “ServiceExpirationDate” values of all their existing endpoints, to ensure its validity is correct as described in chapter 2.5.