CPA Creation Toolkit – 4.0
Simple Message Format Specification 2.0

October 2010 (English version)
Version: 1.0

Table of Contents

This Page Intentionally Left Blank

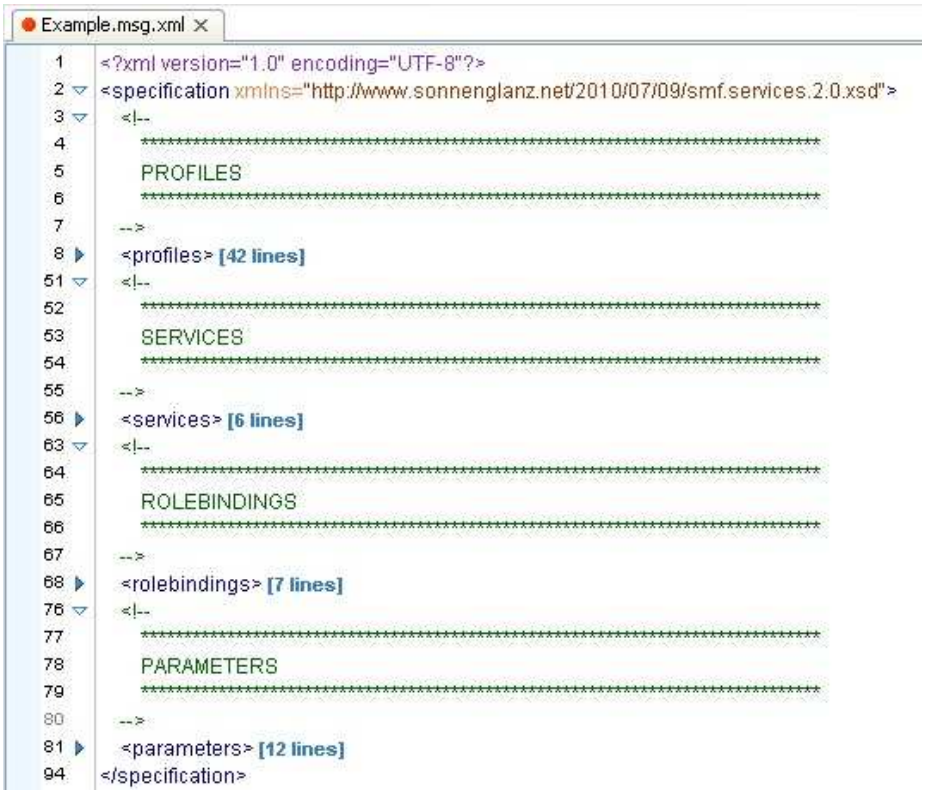
1 Introduction

The “CPA Creation Toolkit” simplifies the creation of CPA’s (Collaboration Protocol Agreement, as part of the ebXML standard). The initial step is the creation of a service specification. This document explains the format of such a service specification.

A service specification describes the messages and services, together with the roles for the participants. The data is stored in an xml format, called SMF: **Simple Message Format**. The SMF consists of four elements:

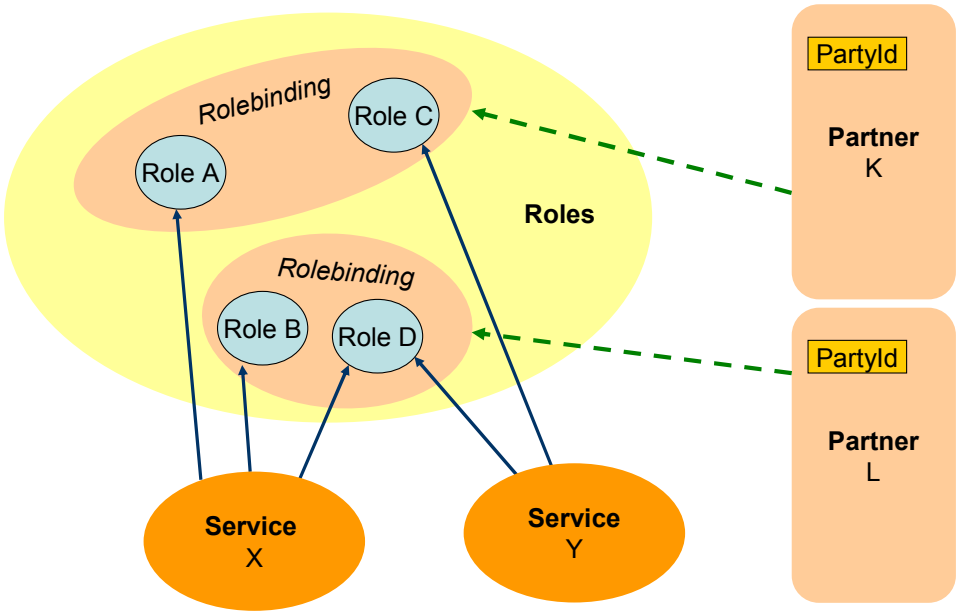
- profiles
  The definition of a set of message exchange characteristics; it is referenced by name.
- services
  The definition of the services and the messages that can be exchanged by those services. Messages are exchanged between roles. A specification can contain one or more service definition.
- rolebindings
  The definition of a set of roles (as defined within the services); it is referenced by name. Each set has to be assigned (in the end) to a collaborating partner. A partner can participate in different services, with different rolebindings as well.
- parameters
  The definition of a set of parameters for which a partner has to provide their values. Different groups of parameters can be defined.

The xml structure of the SMF is relatively simple. Each part will be explained in more detail in the next chapters. The xml structure is as follows:



Example

The figure below is a visual representation of a specification with two services (X and Y). There are two Rolebinding groups: one for role A en C, and one for role B and D. Partner K participates in each service, but with a different role: role A in service X and role C in service Y.



End Example

1.1 XML Validation

Please validate an SMF file by using a good XML editor and by using the provided XSD's (they are available on-line):

- http://www.sonnenglanz.net/2010/07/09/smf.services.2.0.xsd
  for the validation of the service specification (SMF);
- http://www.sonnenglanz.net/2010/07/09/smf.parameters.2.0.xsd
  for the validation of the parameter values that have to be provided by a partner;

2 Profile and Service Specification

2.1 The <profiles> element

The <profiles> element defines a number of message exchange characteristics. The profiles are used (referenced) by the messages in the <message> elements.

The <profiles> (plural) element consists of one or more <profile> (singular) elements. Each <profile> element has a number of attributes. Apart from the name attrbitute, the attributes are explained in more detail in Chapter “The <message> element”.

- name (required): the name of the profile by which it is referenced in a <message> element (by means of the profile attribute);
- transport (optional): the exchange method, with or without acknowledgements;
- security (optional): the security method (transport security and/or payload security);
- retryCount (optional) in case of transport="reliable": the number of attempts to send a message after the first attempt has failed;
- retryInterval (optional) in case of transport="reliable": the time between two attempts of sending a message;

It will be clear that if a profile is defined, besides the name attribute, at least one or more other attribute has to be used.

2.2 The <services> element (plural)

The main element <services> (plural) is a sequence of one or more <service> (singular) elements. A number of defaults values can be specified for the <services> (plural) element. The default values are related to the transport and security attributes. The meaning of these attributes is explained in more detail in Chapter "The <message> element". The four attributes are given the prefix 'default':

- defaultTransport (optional) with a default value of defaultTransport="besteffort" (messages are not acknowledged);
- defaultSecurity (optional) with a default value of defaultSecurity="none";
- defaultRetryCount (optional) in case of transport="reliable", with a default value of defaultTransport="8";
- defaultRetryInterval (optional) in case of transport="reliable", with a default value of defaultSecurity="PT3H" (three hours);

These default values specify the message exchange characteristics and are used if and only if:

- no similar attribute has been used by the <message> element, or
- a referenced <profile> element does not specify one of these similar attributes.

2.3 Acknowledgements and the defaultDeliveryChannel

Regarding the use of **acknowledgments** (and the corresponding defaultDeliveryChannel in the CPA) some choices have to be made: within a single CPA these acknowledgements can only be exchanged by one single type of channel, the defaultDeliveryChannel, with a single setting for the **security** attribute.

Therefore, acknowledgments will be delivered based on the defaultSecurity attribute.

If a number of messages have to be defined using *different values for the default security attribute*, different service specifications have to be created!

Version 4.0 of the CPA Creation program does not support the creation of different CPA's for different default delivery channels, based on different values for the defaultSecurity attribute. Different specification files (SMF) have to be created by hand.

NB. The ebMS 2.0 standard does not allow the transfer of acknowledgements by means of acknowledgements (which is logical, of course). See page 42 of the [ebMS 2.0] specification:

"Note: *Acknowledgment Messages* are never acknowledged."

If it would use acknowledgements, an infinite number of acknowledgments is required…

2.4 The <service> element (singular)

The <service> element specifies all messages exchanged between two partners, based on their role. Each service element has three required attributes and four optional attributes:

- service (**required**): the unique name of the service within the process for which the message exchange is defined.
- serviceId (**required**): the unique identifier of the service, used within the service specification (SMF) only.
- serviceType (optional): the type of the service (an uri).
- process (**required**): the name of the process for which the message exchange is defined.
- processHref (optional): a reference (HREF) to a process description (human readable).
- processUuid (optional): the UUID of the process description.
- processVersion (optional): the version of the process description.

Example

```
<service      serviceId="s1"
              process="Update"
              service=" update:1:2"
              serviceType="urn:sonnenglanz:services">
```

End Example

2.5 The <message> element

A message within a service is described by its own <message> element. Each <message> element has a "from" attribute and a "to" attribute. The value of these attributes resembles the role of the partner within this service specification (within the <service> element). Optionally, the characteristics of the message exchange can be specified. This is done by means of the (optional) attributes security, transport, retryCount, retryInterval and profile.

- security (optional): indicates the level of security. The default value is "none" (no transport or payload security) unless stated otherwise by the defaultSecurity attribute in the <services> element. Possible values are:
  - transport for two-sided ssl, using client & server certificates (default no transport security) at HTTPS level;
  - sign for signing the payload (default no signing);
  - payload for signing and encrypting the payload (default no payload security);
  - transport-and-sign for a combination of transport and payload security;
  - transport-and-payload for a combination of transport and payload security;

- transport (optional): indicates the method of exchange. The default is besteffort, unless stated otherwise in the defaultTransport attribute in the <services> element. Possible values are:
  - besteffort for the "fire-and-forget" exchange pattern;
  - reliable for the acknowledged message exchange pattern;

- retryCount (optional): indicates the number of attempts in case a message exchange fails. The attribute is only useful in case of transport="reliable"! An integer value has to be used. The default value is '8'.

- retryInterval (optional): indicates the time interval between two successive attempts of a message exchange. The attribute is only useful in case of transport="reliable"! Values have to be specified according to the 'duration' format as defined in "**XML Schema[XMLSCHEMA-2]**". The default value is three hours: 'PT3H'. By default, the total interval during which the exchange is performed is (8+1) x 3 hours = 27 hours.

- profile (optional): the name of the profile as defined in the <profiles> element.

A service specification can define several <message> elements with different exchange characteristics for transport, security, retryCount or retryInterval.

Example

A <services> element defines defaultSecurity=“payload”. The message definition …

```
<message from="A" to="B">A</message>
```

… is the same as:

```
<message from="A" to="B" transport="besteffort" security="payload">A</message>
```

Another example:

```
<services defaultTransport ="besteffort">
        <service  serviceId="s1" service="VerwijsIndex:Actualiseren:0:8"
                  servicetype="urn:ebv:services" process="Verwijsindex Actualiseren">
              <message  from="Ketenpartner" to="VerwijsIndex"
                        transport ="reliable" security="payload"
                        retryCount="12"  retryInterval="PT3H"
                        >InschrijvenPersoonsverwijzing</message>
              <message  from="Ketenpartner" to="VerwijsIndex"
                        transport ="reliable" security="payload"
                        retryCount="5" retryInterval="PT8H"
                        >UitschrijvenPersoonsverwijzing</message>
        </service>
        <service  serviceId="s2" service="VerwijsIndex:ZoekenPersoon:0:8"
                  servicetype="urn:ebv:services" process="Verwijsindex ZoekenPersoon">
              <message  from="Ketenpartner" to="VerwijsIndex"
                        >ZoekenPersoon</message>
              <message  from="VerwijsIndex" to="Ketenpartner"
                        security="payload"
                        >PersoonsgegevensGevonden</message>
              <message  from="VerwijsIndex" to="Ketenpartner"
                        >NietsGevonden</message>
        </service>
</services>
```

End Example

2.6 Rules to determine the attribute value in a <message> element

The attributes within a <message> element are defined by means of:
(i) the implicit default values,
(ii) a default value in the <services> element,
(iii) a value in the <profile> element or
(iv) a value in the <message> element,
Therefore, it can be difficult to determine the actual value of the attribute.

Below a set of rules is give by which the actual value of the attributes is derived. It relates to (of course) the attributes transport, security, retryCount or retryInterval.

1. Is the attribute specified in the <message> element?
   If so, use that value.
2. If the above does not apply: is the attribute defined by the referenced profile in the <message> element?
   If so, use the value of the specified profile.
3. If the above does not apply: is the attribute defined by the default attribute in the <services> (plural) element?
   If so, use that default value.
4. If nothing has been specified for the attribute, use the appropriate default value, as shown below:
   transport = "besteffort"
   security = "none"
   retryCount = "8"
   retryInterval = "PT3H" (three hours)

Summary: the attribute value is determined in the order of occurrence:
1e : the <message> element;
2e : the referenced <profile> element;
3e : the default value in the <services> element;
4e : the implied value (in case nothing has been defined).

2.7 Persist Duration

Based on the retryCount en retryInterval the **Persist Duration** is derived. A minimum of 7 days is used. On top of that, the time interval is added as defined by the retryCount en retryInterval.

> **Note:** the fact that the 'persist duration' is defined within the CPA does not mean that the ebMS adapter will adhere to that value. Please consult the documentation of the ebMS adapter, regarding the use of that value within a CPA.

3 Parameters specification

Each partner who wants to collaborate in a particular service, has to provide a number of technical details, such as the URL of the ebMS endpoint and the identification of the partner (the partyId). The <parameters> element specifies what types of information have to be provided by the partner. The actual values have to be provided by a separate file, referred to as the **parameters.xml** file (see Appendix 6).

3.1 The <parameters> element

The <parameters> (plural) element consists of one or more <group> elements. Each <group> element has an attribute parameterId to uniquely identify that group. Within a <group> element, one ore more <parameter> (singular) elements are defined.

For every type of data within the <group> element, it is indicated whether it is required or not. It is indicated by the attribute required and can have the value “true” or “false”.

If a certain type of data is not required (required=“false”), a default value must be supplied!

Note. Data which is not specified by means of the <parameter> (singular) element cannot be provided (or used) at a later stage, even though the partner provides the values in a file (the **parameters.xml** file) at a later stage.

Note: the **parameters.xml** is a separate file and is not part of the service specification (SMF) file. This allows re-use of the service specification as well as re-use of the parameters (if the partner wants to collaborate in another service).

Example

```
<parameters>
        <group parameterId="p1">
                <parameter name="PartyName" required="true"/>
                <parameter name="PartyRef" required="true"/>
                <parameter name="PartyIdList" required="true"/>
                <parameter name="PartyIdType" required="false">urn:xyz:private</parameter>
                <parameter name="ConditionalEndpointUri" required="true"/>
        </group>
</parameters>
```

End Example

The Appendix ‘List of Available Parameters’ provides a complete overview of all possible parameters.

3.2 RetryCount and RetryInterval

Regarding the aspects of Reliable Messaging, each partner can specify its own values for a RetryCount and RetryInterval parameter.

If there are several partners within a service specification, it could be necessary to have different values for each RetryCount and RetryInterval between different partners. To make this possible, *to a certain extend* (it is limited!), the attribute '**bothsides**' is introduced. If one of the partners specifies a value together with the attribute value bothsides="true", the value will be used for the *collaborating partner* as well.

Example

In this example there are three partners. One of the partners functions as a central hub (P1): the other partners (P2 and P3) communicate via the central partner (P). The RetryCount between partner P1 and P2 must be set to 10, while the RetryCount between partner P1 en P3 must be set to 20. The example below shows the correct specification. Note that *P1* ***does not*** *specify a value* for the RetryCount: during the creation of the CPA the value of the collaborating partner is used. Note that if P2 and P3 would communicate directly, it is not known what the actual value will be. The use of the attribute '**bothsides**' is therefore limited and must be applied with care.

```
<parameters>
        <group parameterId="P1">
                <parameter name="PartyName" required="true"/>
                <parameter name="PartyId" required="true"/>
        </group>
        <group parameterId="P2">
                <parameter name="PartyName" required="true"/>
                <parameter name="PartyId" required="true"/>
                <parameter name="RetryCount" required="false"
                        bothsides="true">10</parameter>
        </group>
        <group parameterId="P3">
                <parameter name="PartyName" required="true"/>
                <parameter name="PartyId" required="true"/>
                <parameter name="RetryCount" required="false"
                        bothsides="true">20</parameter>
        </group>
</parameters>
```

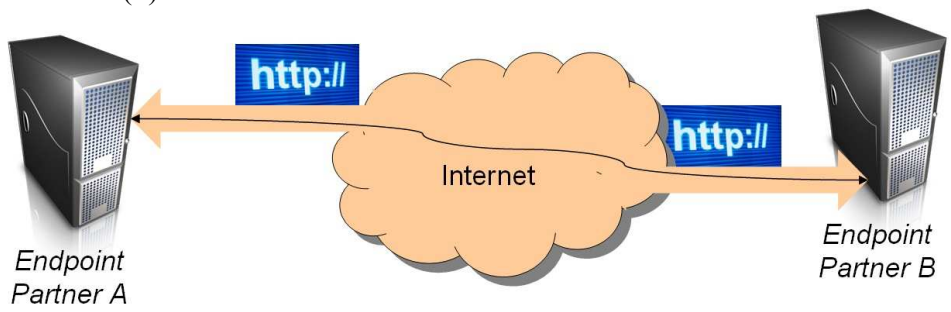
End Example

3.3 Special Case: Connection Type

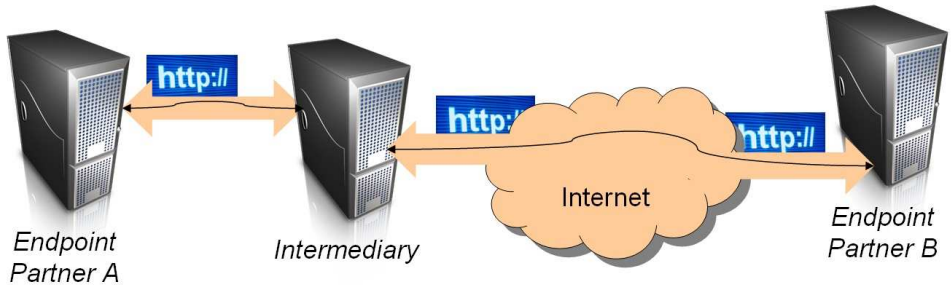
There is a special type of parameter: ConditionalEndpointUri

This parameter is used for a special type of connection. There are two types:

- **Direct**: a Point-to-Point connection between the endpoints; they can 'ping' each other over HTTP(S).



- **Intermediary**: a Gateway and Router is in between the endpoints.



The Intermediary routes ebMS packets (based on the partyId) to the Endpoint that is behind the Gateway. So, the Intermediary is between the endpoints; the endpoints can only 'see' the Intermediary. The Intermediary has the following characteristics:

  - Endpoint A uses the URL of the Intermediary to send ebMS packets to Partner B (a kind of 'next hop').
  - Endpoint B uses the URL of the Intermediary to send ebMS packets to Partner A (a kind of 'next hop').
  - The Intermediary:
    - knows the partyId and the URL of Endpoint A;
    - knows the partyId and the URL of Endpoint B;
    - routes ebMS packets based on the partyId;
  - No CPA's needed between:
    - Partner A and the Intermediary;
    - Partner B and the Intermediary;

When a partner provides information for the parameter ConditionalEndpointUri, two occurrences of the elements <condition> must be provided. **See Appendix 6**! To provide the URL for the direct connection, use:

```
<condition name="local" switch="false">URL</condition>
```

To specify ***a default URL*** for the Intermediary connection, use:

```
<condition name="default" switch="true">URL-of-the-Intermediary</condition>
```

4 Role Binding specification

A partner can fulfil different roles within one and the same service. On the other hand, a partner could also fulfil different roles within different services. Which role a partner will play is defined in the <binding> element within the <rolebindings> structure. Because there can be several services, it has to be specified in which service the particular role is used.

4.1 The <rolebindings> element

The <rolebindings> element contains (at least) two (or more) <binding> elements. Each <binding> element has two required attributes:

- name (**required**): the name of the binding. This name will be presented to the user during the creation of the CPA (by means of the CPA Creation Toolkit).
- parameterId (**required**): a reference to the group of parameters that has to be supplied by a partner (see previous chapter).

A <binding> can contain one or more role-names. These role-names must have been used in the (referenced) service in the <message> elements (the "to" and "from" attribute values). A <role> element has two required attributes:

- serviceId (**required**): a reference to the service element.
- name (**required**): the name of the role within the referenced service.

Example

Two bindings are provided: one with the name "CentralOffice" and one with the name "CollaboratingPartner". The parameters that have to be provided are specified in the <parameters> section, referenced by "p1", not shown in this example.

```
<rolebindings>
    <binding name="CentralOffice" parameterId="p1">
      <role serviceId="s1" name="Answer"/>
      <role serviceId="s2" name="Answer "/>
    </binding>
    <binding name="CollaboratingPartner" parameterId="p1">
      <role serviceId="s1" name="Supply"/>
      <role serviceId="s2" name="Request"/>
    </binding>
  </rolebindings>
```

End Example

Although not shown in the example above, a binding can be defined in which different roles are used within the same service (bound to the same partner). To illustrate this, the following example is given.

Example

```
<binding name="CentralOffice" parameterId="p1">
        <role serviceId="s1" name="Seller"/>
        <role serviceId="s1" name="Buyer"/>
</binding>
```

It will be clear that this partner will not sell goods to itself: consequently, partners could ‘change roles’ within this service. If two other partners would have been defined, the service could be a supply-chain with three partners.

End Example

5 Appendix – List of Available Parameters

The parameters are grouped into ‘required parameters’ and ‘optional parameters’.

5.1 Required Parameters

Parameter **PartyName**
Meaning The name of the partner used in the collaboration.
Default Not applicable.
Occurrence Required.
Example <parameter name="PartyName" required="true"/>

Parameter **PartyId**
Meaning The logical identifier of the partner.
Default Not applicable. Specify the proper PartyId type!
Occurrence Either a PartyId or a PartyIdList has to be defined (not both).
Example <parameter name="PartyId" required="true"/>

Parameter **PartyIdList**
Meaning A list of logical identifiers of a partner (aliases). These identifiers are bound to the same PartyName.
Default *See PartyId.*
Occurrence Either a PartyId or a PartyIdList has to be defined (not both).
Example
```
<parameter name="PartyIdList" required="true"/>
or
<parameter name="PartyIdList"" required="false">
    <value>01</value>
    <value>02</value>
    <value>03</value>
</parameter>
```

Parameter **PartyIdType**
Meaning The type of the PartyId (or PartyIdList element) of the logical identifier.
Default Not applicable.
Occurrence Required.
Example <parameter name="PartyIdType" required="false">urn:epv:sysda</parameter>

Parameter **EndpointUriConditional**
Meaning The public domain HTTP(S) transport url of the partner and the url of the ebMS adapter itself within the infrastructure of the organisation.

Note: read Chapter 5.3 “Use of HTTP Endpoint URL”.

Default Not applicable.
Occurrence Required if EndpointUri is not used.
Example <parameter name="ConditionalEndpointUri" required="true"/>

5.2 Optional Parameters

Parameter **PartyRef**
Meaning A reference to a web page or site with information about the service. The reference is intended to inform human beings.
Default Empty value.
Occurrence Optional.
Example <parameter name="PartyRef" required="true"/>

Parameter **EndpointUri**
Meaning The public domain HTTP(S) transport url of the ebMS adapter of the organisation.

Note: read Chapter 5.3 "Use of HTTP Endpoint URL".

Default Not applicable.
Occurrence Optional if and only if **EndpointUriConditional** is used; otherwise required.
Example <parameter name="EndpointUri" required="true"/>

Parameter **RetryCount**
Meaning The number of attempts to re-start a message exchange after a previous attempt failed.
Default 8 Retries.
Occurrence Optional. Must be an integer value, larger than zero.
Optie As an extra option it can be indicated if the value should be applied to both partners, by using the attribute bothsides="true" (default "false").
Example <parameter name="Retries" required="false">8</parameter>

Parameter **RetryInterval**
Meaning The time interval between two attempts of a message exchange.
Default PTH3 (three hours).
Occurrence Optional.
Possible values: see 'duration' in "**XML Schema[XMLSCHEMA-2]**".
Option As an extra option it can be indicated if the value should be applied to both partners, by using the attribute bothsides="true" (default "false").
Example <parameter name="RetryInterval" required="false">PT3H</parameter>

Parameter **PersistDuration**
Meaning The total time during which the messages are stored by the ebMS adapter. It is a minimum period.
Default A default of 7 days, increased by the time period due to the RetryCount and RetryInterval.
Occurrence Optional.
Possible values: see 'duration' in "**XML Schema[XMLSCHEMA-2]**".
Example <parameter name="PersistDuration" required="false">PT7D</parameter>

Parameter **ClientCert**
Meaning The public certificate of the HTTPS client-certificate of the organisation.
Default Not applicable.
Occurrence Optional. The value must be presented as a KeyInfo structure.
Example <parameter name="ClientCert" required="true"/>

Parameter	**ServerCert**
Meaning	The public certificate of the HTTPS server-certificate of the ebMS adapter of the organisation.
Default	Not applicable.
Occurrence	Optional. The value must be presented as a KeyInfo structure.
Example	<parameter name="ServerCert" required="true"/>

Parameter	**SigningCert**
Meaning	The public certificate of the payload signing-certificate of the organisation.
Default	Not applicable.
Occurrence	Optional. The value must be presented as a KeyInfo structure.
Example	<parameter name="SigningCert" required="true"/>

Parameter	**EncryptionCert**
Meaning	The public certificate of the payload encryption-certificate of the organisation.
Default	Not applicable.
Occurrence	Optional. The value must be presented as a KeyInfo structure.
Example	<parameter name="EncryptionCert" required="true"/>

5.3 Use of HTTP Endpoint URL

If one has to use HTTP as well as HTTPS (secure transport) and only one URL is provided for the ebMS endpoint, the following applies:

- In case the default ports are used (port 80 for HTTP and port 443 for HTTPS), a single URL suffices. Do not provide a port number in the URL! During the CPA creation the correct value for the transport is derived, based on the specified security attributes (within the SMF file) and the provided URL (by the partner).
- In case the defaults ports are NOT used, the following applies:
    - The difference between the HTTP port and the HTTPS port value MUST BE 363 (= 443-80).
        - In case the provided URL uses ‘http’, the port value is incremented by 363 if ‘https’ must be used in the CPA.
        - In case the provided URL uses ‘https’, the port value is decremented by 363 if ‘http’ (without the ‘s’) must be used in the CPA.

      Example:
      The URL uses HTTP and port 4080. In that case, the HTTPS port becomes 4443 (= 4080 + (363) = 4443).

If *a value other than* ***363 is required*** for the difference between port ‘http’ and ‘https’, change the default value in the property file of the CPA Creation Toolkit (part 10 of the property file; see *CPA Creatie Toolkit Installation Manual* or the appendix of the *CPA Creation User Manual*).

6 Appendix – Providing Parameter Values

A service specification defines which parameters have to be provided by a partner in other to create a CPA. This chapter shows how the parameter information itself has to be specified. The structure of the file is according to the XML Schema Definition, found at:

- http://www.sonnenglanz.net/2010/07/09/smf.parameters.2.0.xsd

The values are provided by a separate file, referred to as the **parameters.xml** file.

The file consists of the element <parameters> (plural) containing one or more <parameter> (singular) elements. Each <parameter> element has exactly one attribute name, which represents the name of the parameter (as found in the specification).

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<parameters>
  <parameter name="PartyName">Sonnenglanz Consulting BV</parameter>
  <parameter name="PartyRef">http://www.sonnenglanz.net</parameter>
  <parameter name="PartyIdList">
    <value>IA0112</value>
  </parameter>
  <parameter name="PartyIdType">urn:sonnenglanz:nl</parameter>
  <parameter name="ConditionalEndpointUri">
    <condition name="local" switch="false"
              >http://ebms.sonnenglanz.nl:4080/exchange/IA0112</condition>
    <condition name="default"
              switch="true">http://10.19.3.21:4080/exchange/intermediary</condition>
  </parameter>
</parameters>
```

Note that the value for the parameter ConditionalEndpointUri contains two <condition> elements. One <condition> element specifies the URL for the direct (point-to-point) connection between the ebMS adapters, and one <condition> element specifies the URL of the Intermediary (see Chapter 3.3). In this case, the Intermediary is used as a default.

Note that the <condition> elements can ONLY be provided in the **parameters.xml** file! They cannot be specified in the service specification.

End Example