



**papiNet**

**papiNet Envelope – Version 1.0**

**The papiNet Standard**

**Documentation**

**Global Standard for the Paper and Forest  
Products Supply Chain**

**Build Env\_V1R00\_20101115**

**Date 2010-12-16**

**Interoperability and the papiNet  
Envelope**

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

### **Copyright**

Copyright 2000 – 2010 papiNet G.I.E ("papiNet") and International Digital Enterprise Alliance, Inc. ("IDEAlliance") collectively "Copyright Owner". All rights reserved by the Copyright Owner under the laws of the United States, Belgium, the European Economic Community, and all states, domestic and foreign. This document may be downloaded and copied provided that all copies retain and display the copyright and any other proprietary notices contained in this document. This document may not be sold, modified, edited, or taken out of context such that it creates a false or misleading statement or impression as to the purpose or use of the papiNet specification, which is an open standard. Use of this Standard, in accord with the foregoing limited permission, shall not create for the user any rights in or to the copyright, which rights are exclusively reserved to the Copyright Owner.

papiNet, IDEAlliance, and the members of all papiNet Groups (collectively and individually, "Presenters") make no representations or warranties, express or implied, including, but not limited to, warranties of merchantability, fitness for a particular purpose, title, or non-infringement. The presenters do not make any representation or warranty that the contents of this document are free from error, suitable for any purpose of any user, or that implementation of such contents will not infringe any third party patents, copyrights, trademarks or other rights. By making use of this document, the user assumes all risks and waives all claims against Presenters.

In no event shall Presenters be liable to user (or other person) for direct, indirect, special or consequential damages arising from or related to any use of this document, including, without limitation, lost profits, business interruption, loss of programs, or other data on your information handling system even if Presenters are expressly advised of the possibility of such damages.

---

### **Use of Documents in papiNet Implementations**

Documents may be used as templates for a papiNet implementation. The Presenters grant the right to modify and edit them to fit an actual implementation project provided all copies display the copyright and any other proprietary notices contained in this document. Such modified documents must not be distributed beyond the trading partners implementing or maintaining a papiNet connection.

# papiNet Envelope Version 1.0

## The papiNet Standard

### Table of Contents

Copyright .....	2
Use of Documents in papiNet Implementations .....	2
Table of Contents .....	3
Envelope Documentation .....	4
Envelope Introduction .....	4
Interoperability Vision .....	4
Aspects of Interoperability .....	4
The Need for Coordination between Trading Partners .....	6
Selecting the Proper Environment .....	6
papiNet XML Business Document Creation.....	8
Creating Well-formed papiNet XML.....	8
Sample XML - Document in Envelope .....	9
papiNet Envelope Root Element.....	12
papiNetEnvelope .....	12
Primary Elements .....	13
PayloadInfo .....	13
Payload.....	14
Envelope Use Cases.....	15
XML Transmission Use Cases.....	15
Envelope Parsing Usage Scenario.....	18
papiNetEnvelope Supporting Elements.....	22
Attachment .....	22
AttachmentIdentifier [attribute].....	22
Build [attribute] .....	22
BusinessDocument .....	22
BusinessReceiver.....	22
BusinessSender.....	23
Date .....	23
Day .....	24
Document .....	24
DocumentDate .....	25
DocumentHistoryNumber.....	25
DocumentName [attribute] .....	25
DocumentNumber .....	27
DocumentType [attribute] .....	27
ID .....	32
Month.....	32
NumberOfAttachments .....	32
OtherDocumentName [attribute] .....	32
PartyIdentifier.....	33
PartyIdentifierType [attribute].....	33
Schema .....	34
SequenceNumber .....	34
Signature .....	34
TestFlag [attribute].....	34
Time.....	35
TimeStamp.....	35
TransmissionInfo.....	35
TransmissionReceiver.....	36
TransmissionSender.....	36
Version [attribute] .....	37
Year .....	37

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

### **Envelope Documentation**

---

#### **Envelope Introduction**

The papiNet envelope is a light-weight wrapper that provides a common starting point for message services wishing to parse a papiNet message. The papiNet envelope does not replace the message service envelope but works in concert with all message service envelopes to expose business routing and identification information.

---

#### **Interoperability Vision**

The papiNet Standards Group has the vision of enterprises of any size and in any geographical location meeting and conducting the business of paper, printing, and publishing with each other through the exchange of XML based e-business documents. The intent is to define a neutral method (one that is open and non proprietary) for exchanging these electronic business documents. In addition to being neutral the exchange process has to guarantee safe, secure delivery.

This interoperability vision can be summarized in the following way:

- Participants in the messaging transfer process should be able to choose the technology they desire to use to communicate messages independent of other participants in the communications network.
- Errors in transmission to the destination must be communicated.
- Security must be assured:
  - Privacy - Protect against information being disclosed or revealed to any entity not authorized to have that information by permitting the use of encryption techniques.
  - Authentication - Authenticate the claimed identity of the originator.
  - Authorization - Protect against the threat that unknown entities enter into the system and ensures that an entity performs only authorized actions within the system.
  - Integrity - Protect against the threat that the value of a data item might be changed en route.
  - Non-repudiation - Protect against one party to a transaction or communication later falsely denying that the transaction or communication occurred.

---

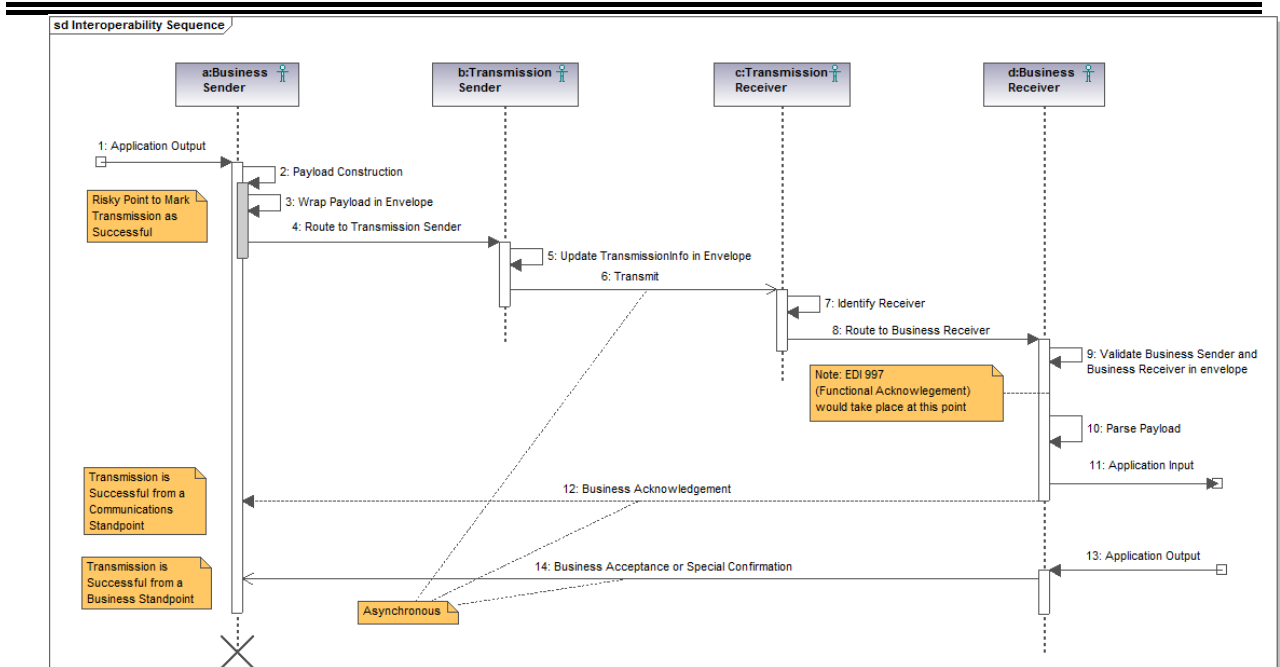
#### **Aspects of Interoperability**

When we talk about e-business document communication there are many aspects to consider. The process starts with an individual reviewing the output from a business application and initiating some sort of action that can be communicated using a papiNet e-business document.

The papiNet e-business document is sent from your location to the recipient's location over the internet via one of several possible transfer methods. The payload is wrapped in an electronic envelope that provides identification and supports security features.

# papiNet Envelope Version 1.0

## The papiNet Standard



### The papiNet Payload

In the diagram above the papiNet e-business document, the papiNet payload, is created in Step 2, read in Step 10, and processed in Step 11. The standardized format of the papiNet payload provides a common view of the business document across trading partner boundaries.

### The papiNet Envelope

The papiNet Standards Committee has found that there is a consistent group of information that is beneficial for XML processing programs to have readily available. Grouping this information into an envelope that contains the papiNet payload means that this information is always in the same location and provides a hook for the XML processing programs to use when processing a transmission.

### Message Service and Transport Protocol Envelopes

We'll cover these two items together because, in many cases, the transport protocol envelope and the message service envelope are so closely related that it is hard to distinguish them.

In the papiNet world we deal with the HTTP, FTP, and SMTP transport protocols. From an allegorical standpoint you could think of the transport protocols in the following way:

- SMTP – as secure as your postal mail and with the same issue of junk mail.
- FTP – a package delivery service that delivers the package to your front door step but does not wait for someone to provide a signature of receipt. Sometimes they ring the doorbell. You can secure your front door step but you cannot ensure that a receipt is provided.
- HTTP/HTTPS – (think of HTTPS as an armoured car delivery) Depending on the nature of the message service you use HTTP can be implemented so that it provides delivery similar to a personal high-security courier who personally delivers the message to you, provides the security cipher, and guarantees that the message has not been tampered with and is from the party indicated. You can also run HTTP/HTTPS in such a way that it provides the same lower level of capability as with SMTP or FTP.

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

In the HTTP/HTTPS transport protocol environment there are several message protocols that you need to be aware of they are SOAP, AS2, and SOAP-ebXML. The level of functionality provided by the HTTP/HTTPS requires more coordination between the sender and recipient than is required by SMTP and FTP.

---

### **The Need for Coordination between Trading Partners**

When two trading partners decide to exchange electronic business documents they must agree on the parameters for the exchange. Be aware that all communication and all messages do not need to be sent in the same way between all trading partners. You could be receiving messages in one form from a trading partner and sending them in another form. However, both of you must at least agree that you will be “looking out” for messages of a certain type.

Most companies like to reduce the complexity of their infrastructure by reducing the number of communications approaches they have to support. There are two competing factors that prevent a single uniform approach for all companies.

- First, there is the drive to the “lowest common denominator” which would be the solution that is easiest and cheapest to implement. Depending on who you are this could be email, ftp, phone, or fax.
- Secondly, there is the drive to the “highest level of security” which would be the HTTP/HTTPS with AS2, SOAP, or ebXML.

So, coordination is required between two trading partners. The first time you set-up your environment for one of the more robust solutions there will be some initial steps that you need to perform. These will be mostly one-time steps that will be applicable for subsequent implementations.

---

### **Selecting the Proper Environment**

The number one or two question that we are continually asked is, “What message service should I use when communicating with my trading partners?” The answer depends on many different factors.

In order to achieve the security considerations mentioned at the beginning of this document our first recommendation is always going to be one of the HTTP oriented protocols. However, if security is less of a concern then other choices may be possible.

- Are you a large company, a medium sized company, or a small company?

Security considerations aside, a small company may feel very comfortable in using SMTP (e-mail) to send papiNet messages. In general, most medium and large companies use one of the HTTP oriented protocols. FTP is used by some medium and large companies as it provides a bit more reliability and security than e-mail however; it is not as robust as the HTTP oriented protocols.

- What are your trading partners doing? What are your customers requiring?

This point gets to the coordination that is required between the sender and receiver and the amount of investment that is required to implement any solution. Obviously you do not want to implement a solution that does not meet your

---

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

security requirements but all the approaches require a degree of coordination.

- How sensitive is the information you are communicating?

While all the approaches provide a certain degree of security SMTP provides the lowest and HTTPS with AS2, SOAP, or ebXML provides the highest. Of course the coordination requirements are more extensive with AS2, SOAP, or ebXML.

# papiNet Envelope Version 1.0

## The papiNet Standard

---

### papiNet XML Business Document Creation

---

#### Creating Well-formed papiNet XML

This document reviews a typical XML document, the sample XML that immediately follows. You might want to place this discussion next to the sample XML so that you can follow the discussion line-by-line. If you have any questions about this discussion you may address them to the papiNet Technical Director (<mailto:acolman@idealliance.org>).

##### Line 1:

The required XML element with attributes for version and encoding is the first line of the XML document. There is currently only a single version of xml and while there are several encoding approaches UTF-8 is the approach that is prescribed by the Internet Engineering Task Force (IETF). UTF-8 can handle every character, pictograph languages may take up more space but they can be represented.

##### Line 2:

The root element of the papiNet envelope (papiNetEnvelope) is shown on line 2. If you scan down to line 44 you will find a second "root element". The second root element is the root element of the papiNet payload, the business document, which is, in this case, "DeliveryMessage". These root elements share many characteristics, we'll cover them when we discuss lines 3, 4, 5, and 6. A similar discussion will take place with the DeliveryMessage root element. Note that line 2 starts with "<" and that a closing ">" does not show until line 6. Line 2 is an element and the entries on line 3, 4, 5, and 6 are attributes.

##### Line 3:

Line 3 indicates that a namespace (without a prefix) is being defined. By looking at the attribute value you should be able to determine that this is the namespace for the papiNet envelope.

##### Line 4:

Another namespace, this one with the prefix of "xsi" is being defined.

##### Line 5 & 6:

The namespace of "xsi" is used along with one of its attributes to communicate the location for the schema. The value of the xsi:schemaLocation (what's between the quotation marks) is entries, the first shows on line 5 and the second on line 6. In programming terminology this would be a name-value pair. The entry on line 5 is the name, which in this case is a namespace, and is the namespace that we defined on line 3. The entry on line 6 is the value, which in this case is a location on the internet, and is the location where papiNet stores the schema for the papiNetEnvelope. Unfortunately, to the uninitiated, line 5 & 6 are extremely confusing because while it is easy to understand that the schema location (the portion on line 6) would be a location on the internet it is very confusing as to why a namespace would look like a location on the internet. The reason why many namespaces look like web-addresses is because web-addresses are associated with domain names and domain names are a very easy way to ensure uniqueness and control (there are other ways, as well).

While this example shows the use of the xsi:schemaLocation attribute you can still choose to use a schema stored at a different location. This is common practice in a production environment where you want to use a locally stored version (refer to Use Case Section). Regardless, it is strongly recommended by



# papiNet Envelope Version 1.0

## The papiNet Standard

---

papiNet to include a schemaLocation in your XML document.

### **Line 7 through 43 and 94 through 97:**

The papiNetEnvelope structure. The closing elements of the envelope are shown on lines 83, 84, and 85, thus the envelope "wraps around" the papiNet payload (the BusinessDocument or, in this case, a DeliveryMessage).

### **Line 44 through 94:**

The papiNet payload, the BusinessDocument section, or in this case a DeliveryMessage.

### **Line 45 through 48:**

The namespace and schemaLocation for the DeliveryMessage is shown in these lines.

Line 45 indicates that this DeliveryMessage lies in the v2r40 namespace which is managed by papiNet. The manner in which this namespace is applied to the document means that this namespace definition applies to those items between lines 44 and 82 (replacing the envelope namespace). The reason this happens is because the namespace is specified on an element.

---

### **Sample XML - Document in Envelope**

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <papiNetEnvelope
3.     xmlns="http://www.papinet.org/env/v1r00"
4.     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5.     xsi:schemaLocation="http://www.papinet.org/env/v1r00
6.         http://www.papinet.org/env/v1r00/papiNetEnvelope.xsd">
7.   <PayloadInfo TestFlag="false">
8.     <TransmissionInfo>
9.       <SequenceNumber>0</SequenceNumber>
10.      <ID>123</ID>
11.      <TimeStamp>2006-02-03T15:00:00</TimeStamp>
12.      <TransmissionSender>
13.        <PartyIdentifier PartyIdentifierType="papiNetGlobalPartyIdentifier">
14.          1.3.6.1.4.1.13099
15.        </PartyIdentifier>
16.      </TransmissionSender>
17.      <TransmissionReceiver>
18.        <PartyIdentifier PartyIdentifierType="DunsNumber">
19.          14-315-8939</PartyIdentifier>
20.      </TransmissionReceiver>
21.    </TransmissionInfo>
22.    <BusinessSender>
23.      <PartyIdentifier PartyIdentifierType="papiNetGlobalPartyIdentifier">
24.        1.3.6.1.4.1.14100</PartyIdentifier>
25.    </BusinessSender>
26.    <BusinessReceiver>
27.      <PartyIdentifier PartyIdentifierType="DunsNumber">
28.        15-415-9939</PartyIdentifier>
29.    </BusinessReceiver>
30.  </PayloadInfo>
31. </papiNetEnvelope>
```

# papiNet Envelope Version 1.0

## The papiNet Standard

---

```
28.      <DocumentNumber>001002456</DocumentNumber>
29.      <DocumentDate>
30.          <Date>
31.              <Year>2008</Year>
32.              <Month>03</Month>
33.              <Day>26</Day>
34.          </Date>
35.      </DocumentDate>
36.  </Document>
37.  <NumberOfAttachments>0</NumberOfAttachments>
38.  <Schema Version="V2R40">
39.      DeliveryMessageV2R40.xsd
40.  </Schema>
41. </PayloadInfo>
42. <Payload>
43.     <BusinessDocument>
44.         <DeliveryMessage
45.             xmlns=http://www.papinet.org/v2r40
46.             xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
47.             xsi:schemaLocation="http://www.papinet.org/v2r40
48.                 http://www.papinet.org/v2r40/DeliveryMessageV2R40.xsd"
49.             DeliveryMessageStatusType="Original"
50.             DeliveryMessageType="DeliveryMessage">
51.         <DeliveryMessageHeader>
52.             <DeliveryMessageNumber>001002456</DeliveryMessageNumber>
53.             <DeliveryMessageDate>
54.                 <Date>
55.                     <Year>2008</Year>
56.                     <Month>03</Month>
57.                     <Day>06</Day>
58.                 </Date>
59.             </DeliveryMessageDate>
60.             <ShipToCharacteristics>
61.                 <ShipToParty
62.                     PartyType="Buyer">
63.                     <NameAddress>
64.                         <Name1>myBuyerName</Name1>
65.                     </NameAddress>
66.                 </ShipToParty>
67.             </ShipToCharacteristics>
68.             <DeliveryLeg>
69.                 <DeliveryLegSequenceNumber>
70.                     1</DeliveryLegSequenceNumber>
71.                 <DeliveryOrigin>
72.                     <LocationParty
73.                         PartyType="Supplier">
74.                         <NameAddress>
75.                             <Name1>mySupplierName</Name1>
```

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

```
75.         </NameAddress>
76.         </LocationParty>
77.         </DeliveryOrigin>
78.         </DeliveryLeg>
79.     </DeliveryMessageHeader>
80.     <DeliveryMessageLineItem>
81.         <DeliveryMessageLineItemNumber>
82.             1</DeliveryMessageLineItemNumber>
83.         <Product>
84.             <ProductIdentifier
85.                 Agency="Buyer"
86.                 ProductIdentifierType="SKU">
87.                     1234567890</ProductIdentifier>
88.             </Product>
89.             <Quantity
90.                 QuantityType="Count">
91.                     <Value
92.                         UOM="Reel">5</Value>
93.                     </Quantity>
94.             </DeliveryMessageLineItem>
95.     </DeliveryMessage>
96. </BusinessDocument>
97. </Payload>
98. </papiNetEnvelope>
```

# papiNet Envelope Version 1.0

## The papiNet Standard

---

### papiNet Envelope Root Element

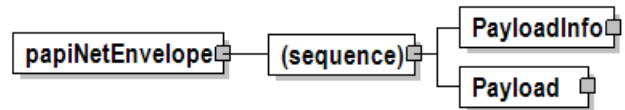
---

#### papiNetEnvelope

The root element of the papiNet Envelope.

##### (sequence)

*The sequence of items below is mandatory. A single instance is required.*



##### **PayloadInfo**

*PayloadInfo is mandatory. A single instance is required.*

A grouping element for items describing the e-business document and parameters related to its communication.

##### **Payload**

*Payload is mandatory. A single instance is required.*

A grouping element for payload information of the papiNet e-business document envelope.

# papiNet Envelope Version 1.0

## The papiNet Standard

### Primary Elements

#### PayloadInfo

A grouping element for items describing the e-business document and parameters related to its communication.

##### TestFlag [attribute]

*TestFlag is mandatory. A single instance is required.*

If "true" or "1" then this is a test transmission. If "false" or "0" then this transmission is for production use.

##### (sequence)

*The sequence of items below is mandatory. A single instance is required.*

##### TransmissionInfo

*TransmissionInfo is mandatory. One instance is required, multiple instances might exist.*

Information about the transmission. Each participant in the transmission process appends a new TransmissionInfo element indicating that they are the TransmissionSender and to who, the TransmissionReceiver, the XML message is being transmitted to.

##### BusinessSender

*BusinessSender is mandatory. A single instance is required.*

The party who issued the papiNet e-document.

##### BusinessReceiver

*BusinessReceiver is mandatory. A single instance is required.*

The business entity for whom the business document is intended, the destination of the document.

- The entity interested in the content. If the receiver party has outsourced the message service to a third party the ReceiverParty is the intended party for the e-document and not the party performing the receiving service of the electronic message.

##### Document

*Document is mandatory. A single instance is required.*

A group item containing information that describes the document contained in the envelope payload.

##### NumberOfAttachments

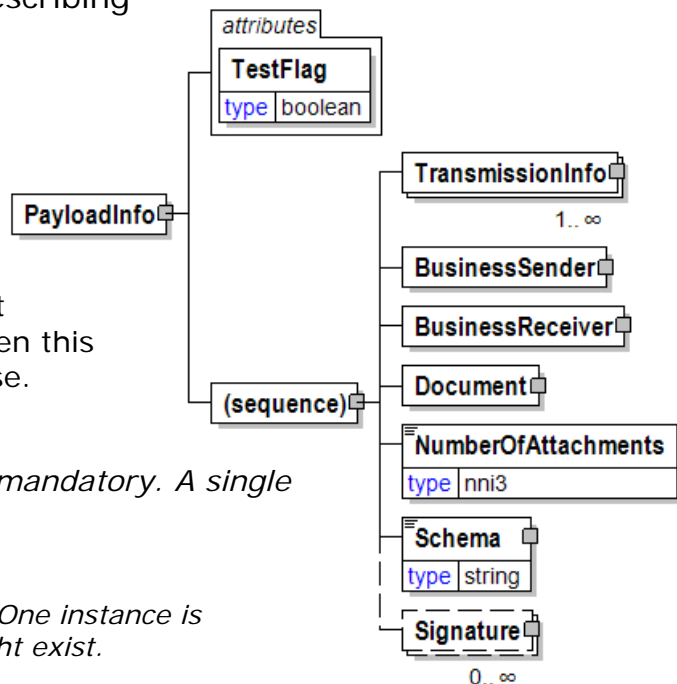
*NumberOfAttachments is mandatory. A single instance is required.*

The number of Attachment element instances in the payload.

##### Schema

*Schema is mandatory. A single instance is required.*

The name of the schema to use for validation. This can be a base URI or a



# papiNet Envelope Version 1.0

## The papiNet Standard

---

complete pointer to the schema to use.

### Signature

*Signature is optional. Multiple instances might exist.*

A signature for the BusinessDocument or an Attachment included in the Payload that conforms to the W3C XML Digital Signature specification.

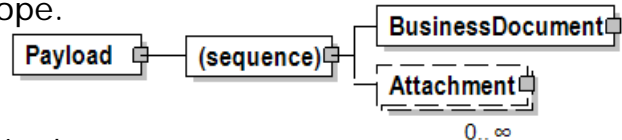
---

## Payload

A grouping element for payload information of the papiNet e-business document envelope.

### (sequence)

*The sequence of items below is mandatory. A single instance is required.*



### BusinessDocument

*BusinessDocument is mandatory. A single instance is required.*

This element contains the actual papiNet e-business document payload content.

### Attachment

*Attachment is optional. Multiple instances might exist.*

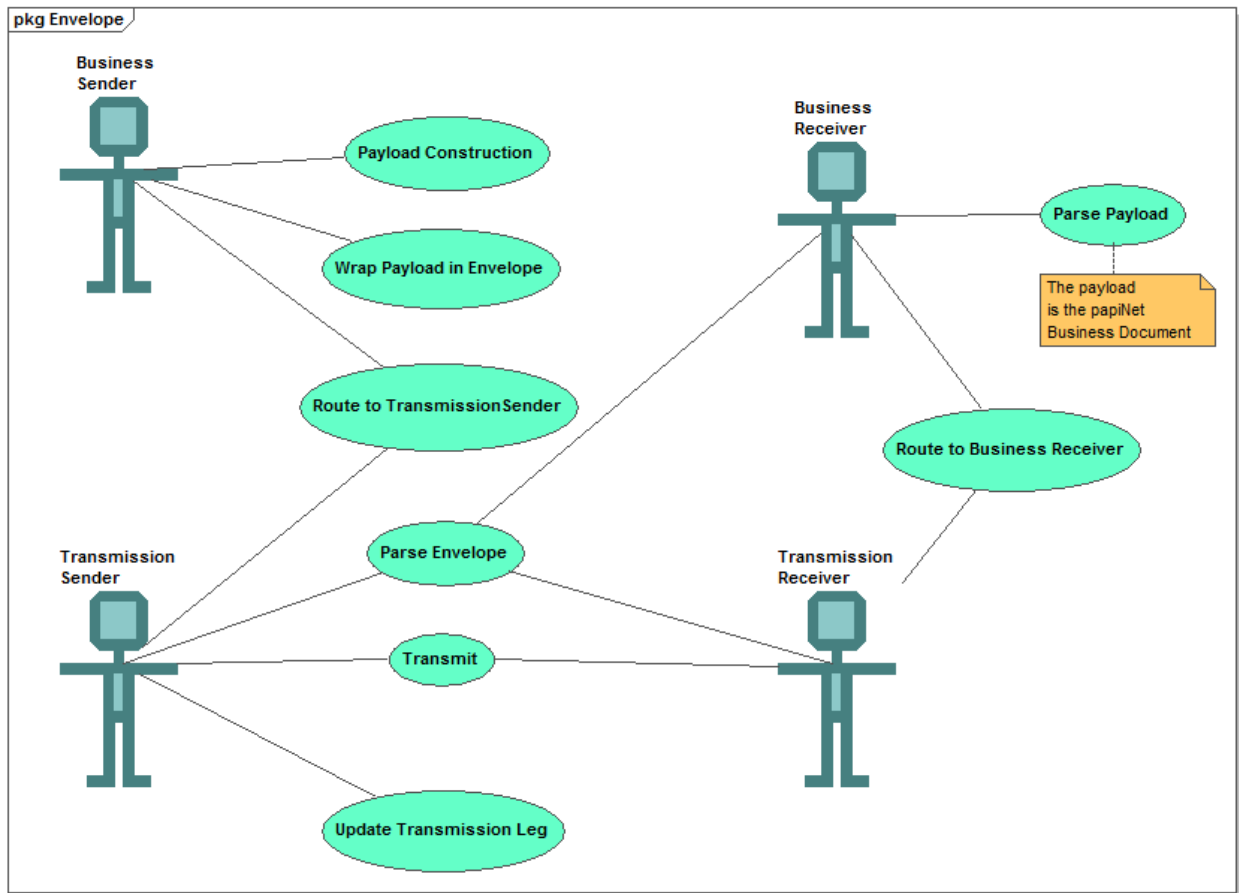
An attachment to the e-business document in Base-64-encoded binary format.

# papiNet Envelope Version 1.0

## The papiNet Standard

### Envelope Use Cases

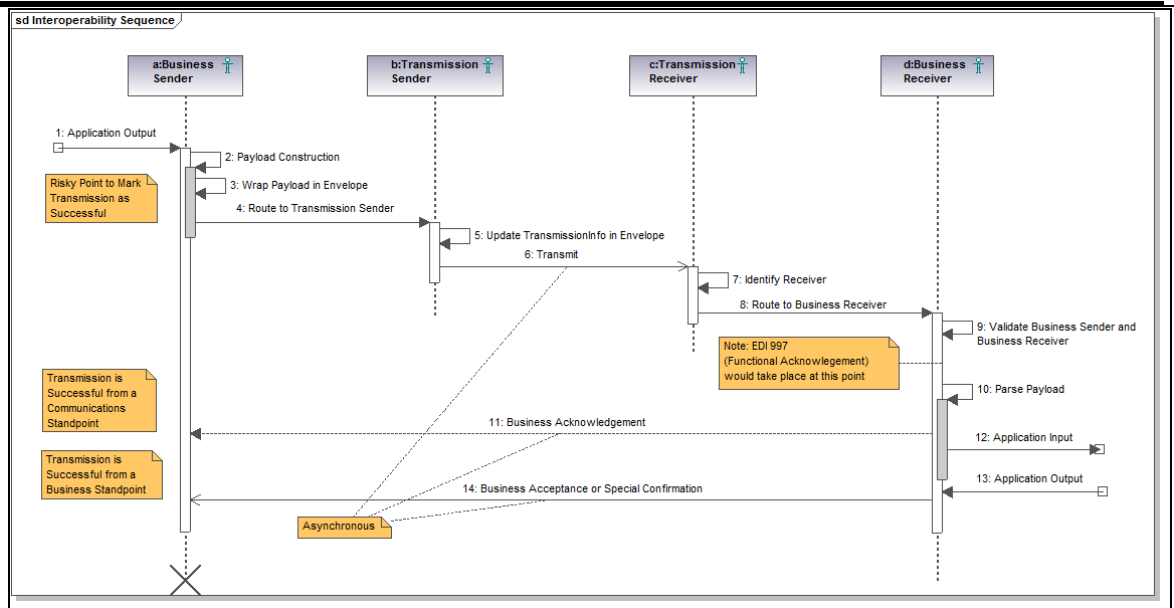
#### XML Transmission Use Cases



<b>Goal</b>	Receipt of papiNet payload by BusinessReceiver
<b>Actors</b>	BusinessSender, BusinessReceiver, TransmissionSender, TransmissionReceiver.
<b>System</b>	We are dealing with a message service and its associated components.
<b>Discussion</b>	

# papiNet Envelope Version 1.0

## The papiNet Standard



### Discussion

The Business Sender is responsible for constructing the papiNet payload, wrapping it in the papiNet envelope, and routing the information to Transmission Sender.

- The Business Sender and Transmission Sender can be the same entity.
- The creation of the digital signature by the Business Sender and the insertion into the papiNet Envelope means that the Business Sender is responsible for the creation of the papiNet envelope.

The Transmission Sender will parse the papiNet envelope to identify the Business Receiver and determine the appropriate transmission route, update the envelope with a new TransmissionInfo element instance, and then transmit.

The Transmission Receiver will parse the papiNet envelope to identify the Business Receiver and either route directly to the Business Receiver or the Transmission Receiver will take the role of Transmission Sender if another intermediate transmission leg is required.

The Business Receiver will parse the papiNet envelope to validate the Business Receiver and Business Sender identification, signatures, and schema information to use when parsing the payload and attachments.

- In some environments it may be possible to



## **papiNet Envelope Version 1.0**

### **The papiNet Standard**

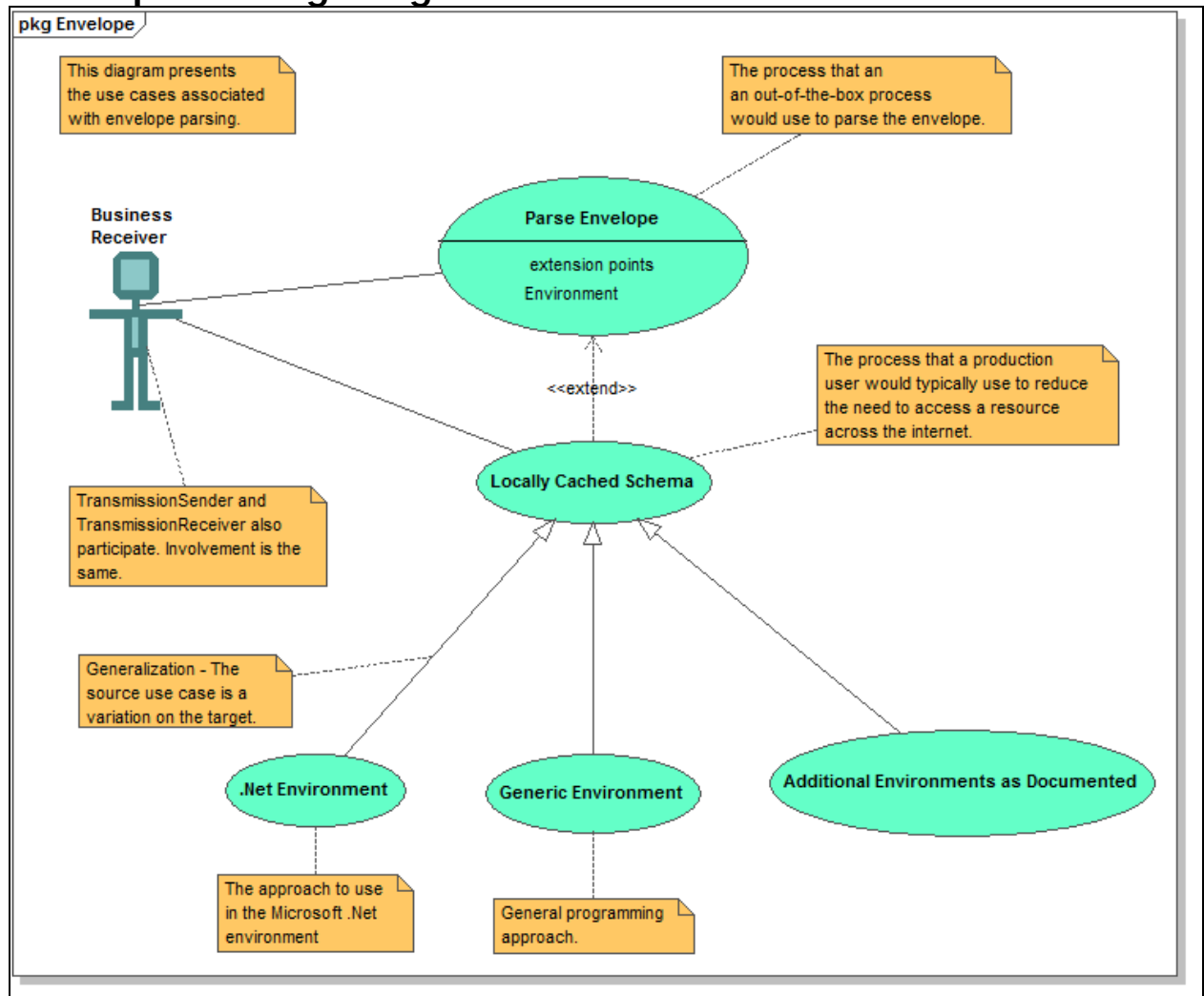
---

	separate the Business Document, Attachments, and Signatures from the papiNetEnvelope wrapper. However, it is expected that, in order to assure the validity of all information, in many environments the Business Receiver will process the XML information from within the papiNetEnvelope.
--	--

# papiNet Envelope Version 1.0

## The papiNet Standard

### Envelope Parsing Usage Scenario



### Parse Envelope

<b>Goal</b>	Read the papiNetEnvelope for information associated routing and payload processing
<b>Actors</b>	BusinessReceiver, TransmissionSender, TransmissionReceiver. <ul style="list-style-type: none"> <li>For clarity purposes only BusinessReceiver is shown.</li> </ul>
<b>System</b>	In an environment where routing is managed outside of a formal message service (e-mail or sneaker-net) the system would typically be the XML parser that is within the Actors browser.  Note: In environments where the routing process is automated the system events take place within the message service (MS) or a service that is tightly coupled with the MS.
<b>Discussion</b>	The parser will use the what it is defined to use to

# papiNet Envelope Version 1.0

## The papiNet Standard

	locate the schema for validation. This could be the xs:schemaLocation or hints provided by the namespace.
--	---

### Locally Cached Schema

<b>Goal</b>	Permit the Actors to use schema that are locally stored instead of using schema that are indicated in the papiNet envelope. The schema indicated in the papiNet envelope would typically be stored at a location on the internet that would result in processing.
<b>Actors</b>	Refer to "Parse Envelope" use case.
<b>System</b>	In environments where the routing process is automated the system events take place within the message service (MS) or a service that is tightly coupled with the MS.
<b>Discussion</b>	<p>papiNet examples indicate a schemaLocation. We do this to permit validation of the XML information in low-tech environments. You are not required to use schemaLocation information, even if it is included. There are several reasons why you would disregard the schemaLocation. The reason that we will be using for demonstrative purposes is the desire to use a locally stored schema, to improve the responsiveness of your production environment.</p> <p>Depending on your programming environment the technique for managing locally cached schema will be different. Refer to the specific use case for your environment for recommendations. If your environment is not listed contact the papiNet Technical Director (acolman@idealliance.org) and he will work with you to determine the approach.</p>

### Locally Cached Schema in a .Net Environment

<b>Goal</b>	Refer to "Locally Cached Schema" use case.
<b>Actors</b>	Refer to "Parse Envelope" use case.
<b>System</b>	Refer to "Locally Cached Schema" use case.
<b>Discussion</b>	<p>If you are using the .Net Framework version 1.0 then refer to System.Xml.XmlSchemaCollection for instructions.</p> <p>System.Xml.Schema.XmlSchemaSet is the</p>

# papiNet Envelope Version 1.0

## The papiNet Standard

	<p>approach recommended in .Net Framework version 2.0 as follows:</p> <ul style="list-style-type: none"> <li>External namespaces or locations referenced in include, import, and redefine elements of a schema are resolved with respect to the base URI of the schema that includes or imports them. For example, if the base URI of the including or importing schema is empty or a null reference ("Nothing" in Visual Basic), the external locations are resolved with respect to the current directory. The XmlResolver class is used to resolve external schemas by default. To disable resolution of include, import, and redefine elements of a schema, set the XmlResolver property to a null reference ("Nothing" in Visual Basic).</li> </ul> <p>System.Xml.Schema.XmlSchemaSet is applied to a System.Xml.XmlReader using the XmlReaderSettings object:</p> <ul style="list-style-type: none"> <li>The ProcessInlineSchema and ProcessSchemaLocation validation flags of an XmlReaderSettings object are not set by default. When these flags are set, the XmlResolver of the XmlReaderSettings object is used to resolve schema locations encountered in the instance document in the XmlReader. If the XmlResolver object is a null reference ("Nothing" in Visual Basic), schema locations are not resolved even if the ProcessInlineSchema and ProcessSchemaLocation validation flags are set.</li> </ul>
<b>References</b>	<p>XmlSchemaSet:</p> <ul style="list-style-type: none"> <li><a href="http://msdn2.microsoft.com/en-us/library/system.xml.schema.xmlschemaset.aspx">http://msdn2.microsoft.com/en-us/library/system.xml.schema.xmlschemaset.aspx</a></li> </ul> <p>XmlReader:</p> <ul style="list-style-type: none"> <li><a href="http://msdn2.microsoft.com/en-us/library/system.xml.xmlreader.aspx">http://msdn2.microsoft.com/en-us/library/system.xml.xmlreader.aspx</a></li> </ul> <p>XmlReaderSettings:</p> <ul style="list-style-type: none"> <li><a href="http://msdn2.microsoft.com/en-us/library/system.xml.xmlreadersettings.aspx">http://msdn2.microsoft.com/en-us/library/system.xml.xmlreadersettings.aspx</a></li> </ul>

# papiNet Envelope Version 1.0

## The papiNet Standard

---

---

### Locally Cached Schema in a Generic Environment

<b>Goal</b>	Refer to "Locally Cached Schema" use case.
<b>Actors</b>	Refer to "Parse Envelope" use case.
<b>System</b>	Refer to "Locally Cached Schema" use case.
<b>Discussion</b>	<p>Generally speaking when you want to use a local version of a schema instead of the schema referenced in the communicated XML information you will follow the follow steps:</p> <ol style="list-style-type: none"><li>1. Determine if there is functionality within your environment to manage this particular scenario. Refer to the use case extensions for examples.</li><li>2. Perform a non-validating read of the papiNet envelope to identify the /papiNetEnvelope/PayloadInfo/Schema element content along with its attributesUse the above information to populate the validating xml reader.</li></ol>

# papiNet Envelope Version 1.0

## The papiNet Standard

### papiNetEnvelope Supporting Elements

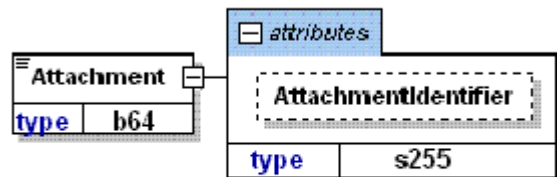
#### Attachment

An attachment to the e-business document in Base-64-encoded binary format.

##### AttachmentIdentifier [attribute]

*AttachmentIdentifier is optional. A single instance might exist.*

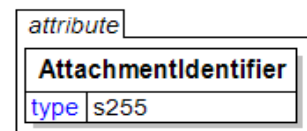
The identifier for the attachment. Permits referencing the attachment from within the BusinessDocument payload or for signature purposes.



The AttachmentIdentifier links to any type of content for inclusion

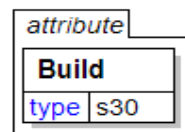
#### AttachmentIdentifier [attribute]

The identifier for the attachment. Permits referencing the attachment from within the BusinessDocument payload or for signature purposes.



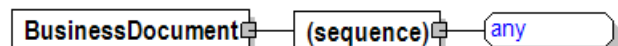
#### Build [attribute]

The build number of the schema. The build number is comprised of the version number associated with a date. The format is vNrNN\_yyyymmdd.



#### BusinessDocument

This element contains the actual papiNet e-business document payload content.



##### (sequence)

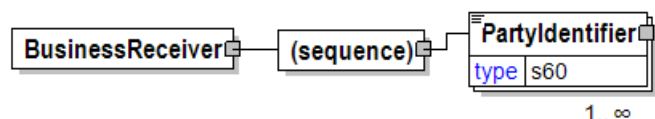
*The sequence of items below is mandatory. A single instance is required.*

##### [Any Type]

Any type of content is permitted at this point.

#### BusinessReceiver

The business entity for whom the business document is intended, the destination of the document.



- The entity interested in the content. If the receiver party has outsourced the message service to a third party the ReceiverParty is the intended party for the e-document and not the party performing the receiving service of the electronic message.

# papiNet Envelope Version 1.0

## The papiNet Standard

---

### (sequence)

*The sequence of items below is mandatory. A single instance is required.*

#### PartyIdentifier

*PartyIdentifier is mandatory. One instance is required, multiple instances might exist.*

A unique identifier of a specific party. This element contains an attribute PartyIdentifierType that indicates the type of party.

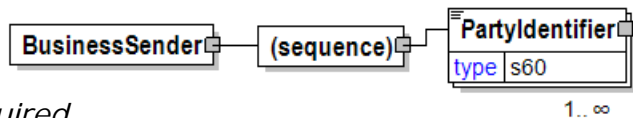
---

## BusinessSender

The party who issued the papiNet e-document.

### (sequence)

*The sequence of items below is mandatory. A single instance is required.*



#### PartyIdentifier

*PartyIdentifier is mandatory. One instance is required, multiple instances might exist.*

A unique identifier of a specific party. This element contains an attribute PartyIdentifierType that indicates the type of party.

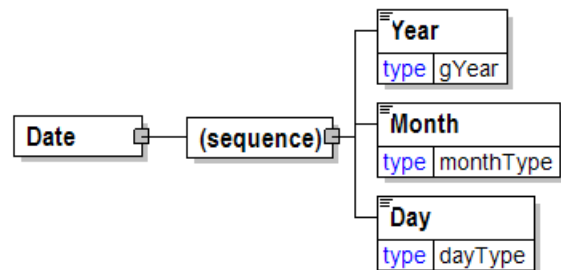
---

## Date

A group element that contains the specification of Year, Month, and Day.

### (sequence)

*The sequence of items below is mandatory. A single instance is required.*



#### Year

*Year is mandatory. A single instance is required.*

Identifies the year number when communicating a calendar date.

#### Month

*Month is mandatory. A single instance is required.*

Identifies the month number when communicating a calendar date.

#### Day

*Day is mandatory. A single instance is required.*

Identifies the day of the month when communicating a calendar date. Day is an integer between 1 and 31.

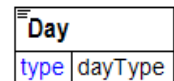
# papiNet Envelope Version 1.0

## The papiNet Standard

---

### Day

Identifies the day of the month when communicating a calendar date. Day is an integer between 1 and 31.



### Document

A group item containing information that describes the document contained in the envelope payload.

#### DocumentName [attribute]

*DocumentName is mandatory. A single instance is required.*

Defines the document name for documents defined by papiNet.

Refer to DocumentName definition for any enumerations.

#### DocumentType [attribute]

*DocumentType is optional. A single instance might exist.*

Identifies the type of a document.

Refer to DocumentType definition for any enumerations.

#### OtherDocumentName [attribute]

*OtherDocumentName is optional. A single instance might exist.*

Defines the document name for documents not defined by papiNet. Documents not defined by papiNet are contained in the papiNetEnvelope when attribute DocumentName has the value "Other".

#### (sequence)

*The sequence of items below is mandatory. A single instance is required.*

#### DocumentNumber

*DocumentNumber is mandatory. A single instance is required.*

The unique identifier of the document included in the BusinessDocument element of the papiNetEnvelope Payload.

#### DocumentDate

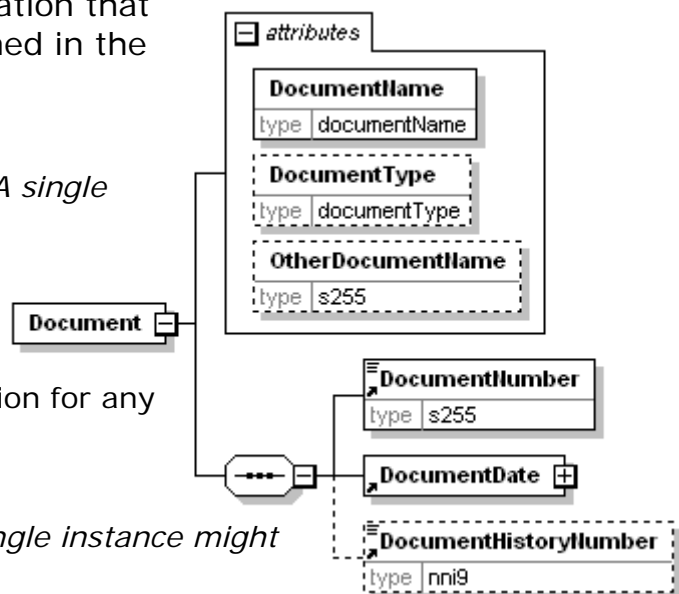
*DocumentDate is mandatory. A single instance is required.*

The Date and optionally Time the document that is contained in the BusinessDocument element of the papiNetEnvelope to was issued.

#### DocumentHistoryNumber

*DocumentHistoryNumber is optional. A single instance might exist.*

A sequential number that keeps track of the version of the e-document being sent by the document originator. The envelope should always contain info for the e-document sent.





# papiNet Envelope Version 1.0

## The papiNet Standard

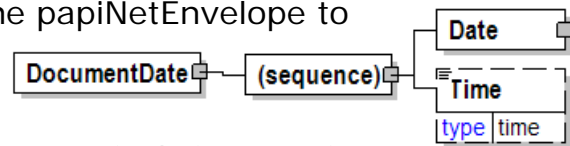
---

### DocumentDate

The Date and optionally Time the document that is contained in the BusinessDocument element of the papiNetEnvelope to was issued.

#### (sequence)

*The sequence of items below is mandatory. A single instance is required.*



#### Date

*Date is mandatory. A single instance is required.*

A group element that contains the specification of Year, Month, and Day.

#### Time

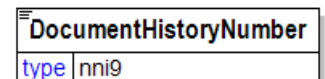
*Time is optional. A single instance might exist.*

Times are treated in a standard XML fashion. The standard XML approach to time is hh:mm:ss, hh:mm:ssZ, or hh:mm:ss±hh:mm.

- hh:mm:ss±hh:mm is used to represent a time with a time zone. The time zone is communicated as the number of hours offset from the Universal Time Coordinate (UTC), a.k.a. Greenwich Mean Time.
- hh:mm:ssZ indicates the time at Zulu (another name for UTC).
- hh:mm:ss or hh:mm indicates in the XML standard the time at UTC. PapiNet recommends that timezone should always be specified on Time in the papiNet envelope.

### DocumentHistoryNumber

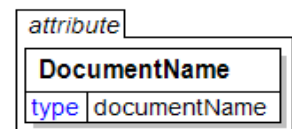
A sequential number that keeps track of the version of the e-document being sent by the document originator. The envelope should always contain info for the e-document sent.



### DocumentName [attribute]

Defines the document name for documents defined by papiNet.

*This item is restricted to the following list.*



**Availability**

**BookSpecification**

**BusinessAcceptance**

**BusinessAcknowledgement**

**Calendar**

**CallOff**

**CallOffWood**

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

**Complaint**  
**ComplaintResponse**  
**CreditDebitNote**  
**DeliveryInstruction**  
**DeliveryMessage**  
**DeliveryMessageBook**  
**DeliveryMessageWood**  
**GoodsReceipt**  
**InfoRequest**  
**InventoryChange**  
**InventoryDispositionInstructions**  
**InventoryStatus**  
**Invoice**  
**InvoiceWood**  
**LoadAvailable**  
**LoadTender**  
**LoadTenderResponse**  
**MeasuringTicket**  
**OrderConfirmation**  
**OrderConfirmationWood**  
**OrderStatus**  
**PackingList**  
**Planning**  
**ProductAttributes**  
**ProductPerformance**  
**ProductQuality**  
**PurchaseOrder**  
**PurchaseOrderWood**  
**RFQ**  
**RFQResponse**  
**ScaleTicket**  
**ShipmentStatus**  
**ShippingInstructions**  
**Statement**  
**Usage**  
**Other**

# papiNet Envelope Version 1.0

## The papiNet Standard

---

### DocumentNumber

The unique identifier of the document included in the BusinessDocument element of the papiNetEnvelope Payload.

DocumentNumber	
type	s255

---

### DocumentType [attribute]

Identifies the type of a document.

*This item is restricted to the following list.*

attribute	
DocumentType	
type	documentType

#### Agreement

A legally binding exchange of promises between parties that the law will enforce.

#### Approximate

An invoice that is created to record ownership transfer without the material physically being moved. Typically what occurs is that the buyer has agreed to take ownership of the item at a certain point in time but has not arranged for transportation. Approximate invoices are invoices that are claiming payment.

#### AvailabilityStatus

#### BlanketOrder

An order that communicates the same information as a StandardOrder but also indicates that the order may be completed over a period of time or, until a certain volume is achieved or, until a certain currency amount is spent.

#### Booking

An indication that time for a service is required. For example a booking that is part of a detail transport plan.

#### BookingConfirmation

A communication affirming the placement of the booking with a particular party.

#### CalibrationCheckLog

A MeasuringTicketType that contains a check log with information about calibration of measuring equipment.

#### CallOff

Defines a call-off sent by the buyer to the seller.

#### CallOffConfirmation

Defines a call-off confirmation sent in response to a call-off.

#### Claim

A "Claim" type is used when there is a customer reaction that includes a financial compensation.

#### Complaint

#### ConfirmingOrder

A confirming order is used to electronically confirm an order that has not been

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

placed electronically by the customer that is, it was placed using another mean such as fax, phone or email.

### **ConsumptionOrder**

A purchase order that is based upon consumption for a particular period of time or instance.

### **CreditNote**

Credits that are provided for amount reductions and complaints. A CreditNode will normally have a reference to an earlier issued Invoice.

### **DebitNote**

A DebitNote claims payment for additional charges. A DebitNote normally has a reference to an earlier issued Invoice.

### **DeliveryInstruction**

A Delivery Instruction can specify many delivery events. Each event can contain quantity for one or many loads. A sequence in a Delivery Instruction specifies all details for one event. The receiver of a Delivery Instruction can generate many Loading Instructions from one Delivery Instruction.

### **DeliveryMessage**

A delivery message type that contains optional routing information, quantities at the DeliveryMessageLineItem level, and details at the DeliveryMessageLineItemDetail level. The seller uses a DeliveryMessage to provide delivery details to the ship-to party and tracking information.

### **Duplicate**

To be removed in a future version. When an invoice is reissued as a duplicate the invoice type should be the same as on the original invoice. Use attribute Reissued = Yes to indicate a duplicate.

### **Feedback**

A "Feedback" type is used when there is a customer reaction that doesn't include any financial compensation.

### **InitialShipmentAdvice**

A delivery message type that contains detailed routing information, quantities at the DeliveryMessageLineItem level and optionally details at the DeliveryMessageLineItemDetail level. The seller uses the InitialShipmentAdvice to provide preliminary notification of shipment routing and quantities (can be used as an advanced notice of delivery). The quantities indicated may not necessarily be delivered—for example, if there is damage in transit or the delivery is rerouted to another destination.

### **InventoryChange**

### **InventoryStatus**

### **Invoice**

A regular invoice (referred to as an invoice) is claims payment for products, services, and charges supplied under conditions agreed to between seller and buyer. The invoice communicates, among other things, payment terms, payment method, and the payment amount required for one or more deliveries.

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

### **InvoiceSpecification**

A MeasuringTicketType that contains information about measured products including prices and amounts. This MeasuringTicketType is normally used as a detailed specification of billable or invoiced products.

### **LoadAvailable**

### **LoadAvailableConfirmation**

### **LoadedSpecification**

A Delivery Message type that contains a specification of loaded goods on a transport unit (e.g. container, rail wagon). A loading specification (LoadedSpecification) can contain goods belonging to many suppliers and buyers and is normally referring to a Loading Instruction.

### **LoadingInstruction**

A Loading Instruction specifies loading of transport units. A Forwarder receiving a Delivery Instruction from a supplier can convert it into many Loading Instructions, which are sent to a Warehouse Operator. A sequence in a Loading Instruction specifies all details how to load one transport unit. A Loading Instruction can include goods from many Delivery Instructions and also goods belonging to many suppliers.

### **LoadingOrder**

Specifies delivery schedule, the transport booking requirements and shipping instructions, as well as the internal work order for the shipment

### **LoadingSchedule**

The schedule for transport vehicle loading.

### **LoadTender**

More than one carrier has been informed about the upcoming transport requirement (the load).

### **LoadTenderResponse**

Indicates the ability of the respondent to fulfill the load tender request.

### **MeasuringTicket**

A MeasuringTicketType that contains detailed information about measured products and items. This information can be used as a base for calculation of billable products.

### **OfficeSchedule**

The schedule for office personnel.

### **OrderStatus**

### **PackingSpecification**

A Packing Specification is a business transaction specifying the complete consignment being shipped from a consignor to a consignee. The specification describes in detail what is being shipped and how it is packed and marked.

### **PreBooking**

A requirement for a transport vehicle or unit reservation that has risen from a rough transport plan.

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

### **PreBookingConfirmation**

A communication of the requirement for a transport vehicle or unit reservation that has risen from a rough transport plan.

### **PrePayment**

A prepayment invoice is used as a notification of amounts to be paid in advance. It is not a legal invoice claiming payment for provided products or services.

### **PriceList**

Used when sending a pricelist

### **ProductCrossReference**

Used in the exchange of ProductIdentifier cross-references for the product codes used by two trading partners

### **ProductionTicket**

A MeasuringTicketType that contains information about produced products and items. A ProductionTicket can also contain measuring information for produced items.

### **ProductProperties**

Used when sending a list of ProductAttributes

### **ProForma**

A pro-forma invoice is not a legal invoice claiming payment. A pro-forma invoice is a document/message serving as a preliminary invoice, containing on the whole the same information as the final invoice, but not actually claiming payment. Pro-forma invoices are typically used in a cross-border shipment for customs purposes. Can travel with the product or be sent to a broker.

### **ReleaseOrder**

A release order is used to release a product to be made. This type of order refers to a previous order or agreement, such as a reservation order, a contract, or a blanket order. A release number version which is an incremented number, is used in this type.

### **ReservationOrder**

A reservation order is used to book a paper supplier's capacity for specific quantities of specific products, but it is not yet a firm commitment to buy. Consequently, all line items of a reservation order have a ProductionStatusType of NotFree. To release one or more purchase order line items of a reservation order, a buyer can use a standard order or a release order.

### **SampleMeasuringTicket**

A MeasuringTicketType that contains detailed information about measured samples taken from e.g. a delivery.

### **SelfBillingStatement**

A SelfBillingStatement is an invoice that is prepared by the buyer of the product (or the consumer of the service) for the provider of the product or service. The preparer will pay the invoice.

# **papiNet Envelope Version 1.0**

## **The papiNet Standard**

---

Typically, logistics cost-charges are the services that are communicated. They are incurred during a certain period of time between the party who provides the logistics services (typically the party with a LogisticsRole of "LogisticsProvider") and the requestor of such services (typically the party with a LogisticsRole of "Consignor").

Raw materials, maintenance, repair, and operating supplies are examples of products that can be entered on a self billing statement for products. This type of invoice is also referred to as an invoice for "evaluated receipts" because it developed from the receiving department's evaluation of their receipts and from that creating the invoice.

The SelfBillingStatement process exists when there is a great deal of trust between the trading partners that permits this cost savings business process.

### **ShipmentAdvice**

A Delivery Message type that contains a specification of goods, that are dispatched and will be delivered to a warehouse. A ShipmentAdvice can refer to a Delivery Instruction Sequence or a CallOff, but is normally also used for notifying a receiving warehouse operator of shipments to the warehouse.

### **ShippingInstructionsConfirmation**

Confirmation of the Shipping Instruction.

### **ShippingInstructionsDistribution**

Shipping Instruction serves as a billable order for shipping product that was produced under a previous manufacturing order.

### **ShippingInstructionsPackaging**

To be deprecated in a future version. Guidance will be given to trading partners not to use this enumeration.

### **ShippingInstructionsRouting**

Shipping Instructions issued in conjunction with a Purchase Order for shipping the product.

### **SpecOrder**

Used when the BookSpecification message is intended for use with the Purchase Order transaction.

### **StandardOrder**

A standard order is a regular one-time order. It defines quantities for one or more products to be purchased and may also define a delivery location and date. A standard order may reference a previous agreement (for example, a quotation, a contract, or a reservation order) for information.

### **SpecRFQ**

Used when the BookSpecification message is intended for use with the Request for Quote (Estimate) transaction.

### **StandardRFQ**

### **SupplierSpec**

Will be removed in a future version. Recommended not to be used.  
(Old definition: Used when the BookSpecification message is intended for use

# papiNet Envelope Version 1.0

## The papiNet Standard

---

by the Supplier only and is not generated by, or transmitted to the Buyer).

### **SupportSchedule**

The support schedule for transport operations.

### **ThirdPartyShipmentAdvice**

Notification of a shipment from a warehouse to a party other than the party that ordered the product from the supplier.

### **TrialOrder**

A trial order is used whenever a seller has agreed to have a buyer sample a product. The seller needs to process this order differently from other orders, because a trial order may include a product sold at a discounted rate or even provided for free. A trial order typically is for small quantities.

### **UnloadingSchedule**

The schedule for transport vehicle unloading.

### **Waybill**

The forwarding agreement or carrying agreement that is used as a receipt for cargo and as a contract of carriage.

---

## **ID**

The unique identifier for the transmission. You can develop this identifier independently from other systems or you can use a similar identifier to one that you are currently using. For example, you could use something similar to the ISA control number in X12.

ID
type string

---

## **Month**

Identifies the month number when communicating a calendar date.

Month
type monthType

---

## **NumberOfAttachments**

The number of Attachment element instances in the payload.

NumberOfAttachments
type nni3

---

## **OtherDocumentName [attribute]**

Defines the document name for documents not defined by papiNet. Documents not defined by papiNet are contained in the papiNetEnvelope when attribute DocumentName has the value "Other".

attributes	
OtherDocumentName	
type	s255



# papiNet Envelope Version 1.0

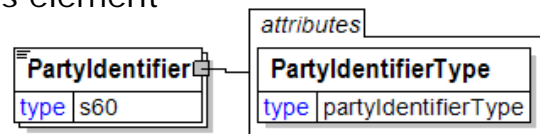
## The papiNet Standard

---

### PartyIdentifier

A unique identifier of a specific party. This element contains an attribute

PartyIdentifierType that indicates the type of party.



#### **PartyIdentifierType [attribute]**

*PartyIdentifierType is mandatory. A single instance is required.*

Provides a contextual definition for the party identifier value. This party identifier enables the trading partners to use ID codes for the different organisation versus business entities involved in the transaction.

Refer to PartyIdentifierType definition for any enumerations.

### PartyIdentifierType [attribute]

Provides a contextual definition for the party identifier value. This party identifier enables the trading partners to use ID codes for the different organisation versus business entities involved in the transaction.



*This item is restricted to the following list.*

#### **DunsNumber**

A 9-digit identifier maintained by Dun & Bradstreet to uniquely identify a commercial enterprise.

#### **Duns4Number**

Similar to the DunsNumber that defines a parent company/organization. The additional 4 digits enable the definition/identification of a sub-entity, such as a mill/plant of this parent company/organisation.

#### **GlobalLocationNumber**

The globally unique EAN.UCC System identification number for legal entities, functional entities and physical locations. The Global Location Number uses the EAN/UCC-13 Data Structure. Details about the structure and usage are available at the <http://www.gs1.org> website.

#### **papiNetGlobalPartyIdentifier**

A readily available and free identifier supported by IANA (the Internet Assigned Numbers Authority). Private Enterprise Numbers that conforms to proper URI naming conventions are available, free-of-charge from the Internet Assigned Numbers Authority (IANA). The application form is available at the <http://www.iana.org/cgi-bin/enterprise.pl>

website. papiNet will be developing a broader recommendation for party naming in the future however, obtaining an IANA Private Enterprise Number (PEN) will permit globally unique trading partner identification.

#### **StandardAddressNumber**

# papiNet Envelope Version 1.0

## The papiNet Standard

---

A 7-digit code to uniquely identify an organisation and its location. The number consists of 6 digits with an 11-modulus check digit as the seventh digit. ISAN is the same with the ISO Country code as a prefix. This identifier is solely used by the book industry.

---

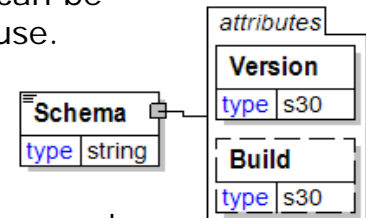
### Schema

The name of the schema to use for validation. This can be a base URI or a complete pointer to the schema to use.

#### Version [attribute]

*Version is mandatory. A single instance is required.*

The version identifier for the schema. For papiNet schema the version is at the end of the schema file name and follows the format of VnRnn. The version can also be found in the version attribute of the root element as the beginning of the build number.



#### Build [attribute]

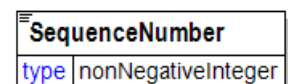
*Build is optional. A single instance might exist.*

The build number of the schema. The build number is comprised of the version number associated with a date. The format is vNrNN\_yyyymmdd.

---

### SequenceNumber

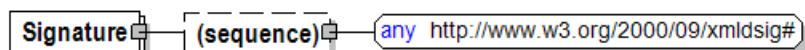
The sequence of the item. Provides a mechanism for sorting.



---

### Signature

A signature for the BusinessDocument or an Attachment included in the Payload that conforms to the W3C XML Digital Signature specification.



#### (sequence)

*The sequence of items below is optional. A single instance might exist.*

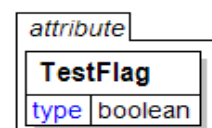
#### [Any Type]

Any type of content is permitted at this point.

---

### TestFlag [attribute]

If "true" or "1" then this is a test transmission. If "false" or "0" then this transmission is for production use.



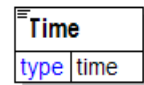
# papiNet Envelope Version 1.0

## The papiNet Standard

---

### Time

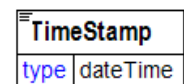
Times are treated in a standard XML fashion. The standard XML approach to time is hh:mm:ss, hh:mm:ssZ, or hh:mm:ss±hh:mm.



- hh:mm:ss±hh:mm is used to represent a time with a time zone. The time zone is communicated as the number of hours offset from the Universal Time Coordinate (UTC), a.k.a. Greenwich Mean Time.
- hh:mm:ssZ indicates the time at Zulu (another name for UTC).
- hh:mm:ss or hh:mm indicates in the XML standard the time at UTC. PapiNet recommends that timezone should always be specified on Time in the papiNet envelope.

### TimeStamp

The date and time e-business document, the payload, was prepared for transmission. The format and interpretation of Timestamp is the standard W3C xs:dateTime format.



### TransmissionInfo

Information about the transmission. Each participant in the transmission process appends a new TransmissionInfo element indicating that they are the TransmissionSender and to who, the TransmissionReceiver, the XML message is being transmitted to.

#### (sequence)

*The sequence of items*

*below is mandatory. A single instance is required.*

#### SequenceNumber

*SequenceNumber is mandatory. A single instance is required.*

The sequence of the item. Provides a mechanism for sorting.

#### ID

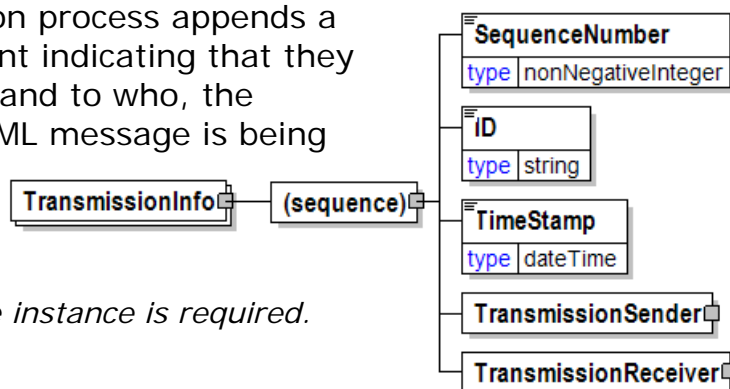
*ID is mandatory. A single instance is required.*

The unique identifier for the transmission. You can develop this identifier independently from other systems or you can use a similar identifier to one that you are currently using. For example, you could use something similar to the ISA control number in X12.

#### TimeStamp

*TimeStamp is mandatory. A single instance is required.*

The date and time e-business document, the payload, was prepared for



# papiNet Envelope Version 1.0

## The papiNet Standard

---

transmission. The format and interpretation of Timestamp is the standard W3C xs:dateTime format.

### TransmissionSender

*TransmissionSender is mandatory. A single instance is required.*

The party responsible for transmitting the papiNet e-document. This can be the same as the party issuing the e-document or any handling service for the BusinessSender such as a portal, a hub, etc. depending on the set-up of communication services.

### TransmissionReceiver

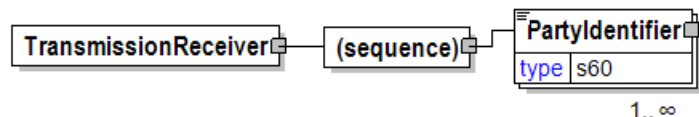
*TransmissionReceiver is mandatory. A single instance is required.*

The party responsible for the transmission receipt of the papiNet e-document. This can be the party who will eventually process the papiNet for the e-document contents or any handling service BusinessReceiver such as a portal, a hub etc. depending on the set-up of communication services.

---

## TransmissionReceiver

The party responsible for the transmission receipt of the papiNet e-document. This can be the party who will eventually process the papiNet for the e-document contents or any handling service BusinessReceiver such as a portal, a hub etc. depending on the set-up of communication services.



### (sequence)

*The sequence of items below is mandatory. A single instance is required.*

#### PartyIdentifier

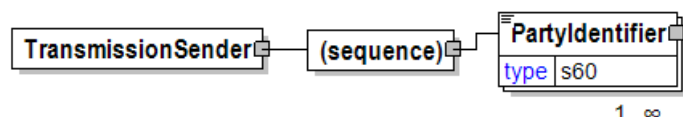
*PartyIdentifier is mandatory. One instance is required, multiple instances might exist.*

A unique identifier of a specific party. This element contains an attribute PartyIdentifierType that indicates the type of party.

---

## TransmissionSender

The party responsible for transmitting the papiNet e-document. This can be the same as the party issuing the e-document or any handling service for the BusinessSender such as a portal, a hub, etc. depending on the set-up of communication services.



### (sequence)

*The sequence of items below is mandatory. A single instance is required.*

#### PartyIdentifier

*PartyIdentifier is mandatory. One instance is required, multiple instances might exist.*

# papiNet Envelope Version 1.0

## The papiNet Standard

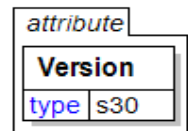
---

A unique identifier of a specific party. This element contains an attribute PartyIdentifierType that indicates the type of party.

---

### Version [attribute]

The version identifier for the schema. For papiNet schema the version is at the end of the schema file name and follows the format of VnRnn. The version can also be found in the version attribute of the root element as the beginning of the build number.



---

### Year

Identifies the year number when communicating a calendar date.

