**ECF 4.01 Errata 02 Candidates**

The following item are provided for considerations are for ECF 4.01 Errata 02:

1. Payment Receipt Message
2. NIEM Entity References
3. Content ID & BinaryLocationURI
4. UBL 2.1

**1. Payment Receipt Message**

Which message should be included in the NotifyFilingReviewComplete exchange?

1. PaymentMessage, or
2. PaymentReceiptMessage

The ECF specification (Errata 01), section C.1.1 shows that the parameter is the PaymentMessage.

However, the Web Services wsdl specifies the PaymentReceiptMessage.

 <xsd:complexType name="NotifyFilingReviewCompleteRequestMessageType">

 <xsd:annotation>

 <xsd:documentation>Multi-part message type (required for conformance with WS-I Basic Profile 1.1</xsd:documentation>

 </xsd:annotation>

 <xsd:complexContent>

 <xsd:extension base="ecf:ElectronicFilingMessageType">

 <xsd:sequence>

 <xsd:element ref="reviewcb:ReviewFilingCallbackMessage"/>

 <xsd:element ref="receipt:PaymentReceiptMessage"/>

 </xsd:sequence>

 </xsd:extension>

 </xsd:complexContent>

</xsd:complexType>

Additionally, the ECF specification, section 3.2.8 NotifyFilingReviewComplete, is unclear when it requires that “a receipt for the payment MUST be included in the operation.”

What is meant by “receipt”?

Is this intended to refer to the PaymentReceiptMessage, or is it suggesting that receipt information must be included in the PaymentMessage?

It is presumed that it is actually PaymentReceiptMessage that should be specified in the NotifyFilingReviewComplete, otherwise it appears that there would be no use for PaymentReceiptMessage.

**2. NIEM Entity References**

This ‘error’ exists in one of the ECF example messages; specifically ECF-4.0-CoreFilingMessage-Criminal2.xml.

Here is an extract from this XML:

 <criminal:CriminalCase xmlns:criminal="urn:oasis:names:tc:legalxml-courtfiling:schema:xsd:CriminalCase-4.0" xsi:schemaLocation="urn:oasis:names:tc:legalxml-courtfiling:schema:xsd:CriminalCase-4.0 ../xsd/casetype/ECF-4.0-CriminalCase.xsd">

 <nc:CaseTitleText>State vs. John Doe </nc:CaseTitleText>

 <nc:CaseTrackingID>123456ABC</nc:CaseTrackingID>

 <j:CaseAugmentation>

 <j:CaseCourt>

 <nc:OrganizationIdentification>

 <nc:IdentificationID>1</nc:IdentificationID>

 </nc:OrganizationIdentification>

 <j:CourtName>Municipal Court</j:CourtName>

 </j:CaseCourt>

 <j:CaseRespondentParty>

 <nc:EntityPersonReference s:ref="Person1"/>

 </j:CaseRespondentParty>

 </j:CaseAugmentation>

 <ecf:CaseAugmentation>

 <ecf:CaseParticipant s:id="Person1">

 <ecf:EntityPerson>

 <nc:PersonBirthDate>

 <nc:Date>1983-01-01</nc:Date>

 </nc:PersonBirthDate>

Observe that in this example, the s:ref fro nc:EntityPersonReference points to the s:id for ecf:CaseParticipant. This is incorrect per the NIEM NDR, instead the s:id should be on ecf:EntityPerson.

I slightly modified the example XML so that I could present the problem to the NIEM help desk. This correspondence is below:

Which is correct?

<j:CaseRespondentParty>

 <nc:EntityPersonReference structures:ref=”Person1”/>

</j:CaseRespondentParty>

<j:CaseDefendantParty structures:id=”Person1”>

 <nc:EntityPerson>

 <nc:PersonName>

 <nc:PersonFullName>Jack Smith</nc:PersonFullName>

 </nc:PersonName>

 </nc:EntityPerson>

</j:DefendantParty>

Or

<j:CaseRespondentParty>

 <nc:EntityPersonReference structures:ref=”Person1”/>

</j:CaseRespondentParty>

<j:CaseDefendantParty>

 <nc:EntityPerson structures:id=”Person1”>

 <nc:PersonName>

 <nc:PersonFullName>Jack Smith</nc:PersonFullName>

 </nc:PersonName>

 </nc:EntityPerson>

</j:DefendantParty>

The difference is where the structures:ref attribute in nc:EntityPersonReferece ‘points’. In the first example, it ‘points’ to j:CaseDefendantParty, and in the second example it ‘points’ to nc:EntityPerson.

It would appear that section 7.6.2 Reference Elements from the NIEM Naming and Design Rules v1.3 is relevant. In particular, [Rule 7-60] appears applicable. This rule states: ‘Within the schema, if both elements *NCName* and *NCName*Reference exist then the appinfo:ReferenceTarget of any *NCName*Reference element MUST be the type *NCName*.’

The above issue was submitted to the NIEM Help Desk on Friday, Nov. 7, 2014. The following response was received Tue. Nov. 11, 2014:

Subject

---------------------------------------------------------------

NIEM Web site - NIEM Entity References

Discussion Thread

---------------------------------------------------------------

Response Via Email(Cuong To) - 11/11/2014 03:31 PM

Dear Gary,

Thank you for contacting the NISS or NIEM Help Desk.

Here is the response from our expert:

--------------

The second example is correct, based on Rules 7-60 and 8-5:

Within an element instance, given that a reference element is restricted to a target type T, any attribute structures:ref MUST reference an element that has a type definition of type T, or that is derived from type T.

Since the nc:EntityPersonReference reference element (with the structures:ref attribute) is restricted to the target type of nc:EntityPerson, which is nc:PersonType, the target element (with the structures:id attribute) must be of type nc:PersonType.

The element nc:EntityPerson is of type nc:PersonType; the element j:CaseDefendantParty is not of type nc:PersonType (it is of type nc:EntityType).

--------------

If you have any additional questions or concerns, please feel free to contact us via web at <http://it.ojp.gov/niss/helpdesk/> (<http://it.ojp.gov/niss/helpdesk/>) or by calling the toll free number (877) 333-5111 or local number (703) 726-1919 or by emailing us at NISShelp@ijis.org (mailto:NISShelp@ijis.org) .

Sincerely,

Cuong To

NISS Help Desk

Cuong To refers to NDR rules 7-60 and 8-5. Rule 7-60 is stated in the problem description presented to the NIEM Help Desk. Rule 8-5 is stated below:

[Rule 8-5] Within an element instance, given that a reference element is restricted to a target type T, any attribute structures:ref MUST reference an element that has a type definition of T or is derived from T.

**3. Content ID & BinaryLocationURI**

The correct way in which binary attachments are handled using MTOM may not be correctly represented in the ECF specification, in particular Figure 2, shown below:



The above diagram, suggests that the MIME attachment Content-ID: is A1, however, in actual practice it would be <A1>, as in:

Content\_ID: <A1>

Additionally, the value of the BinaryLocationURI (defined as: “a URL or file reference of a binary object”) should not contain the two forward slash characters; ii should only contain “cid:A1”.

Additionally, since the ECF specification states:

### Attachment [2.3.2]

An attachment is a series of bytes in the message stream transmitted between MDEs that constitutes, in whole or in part, an electronic document whose conventional equivalent would be a document on paper. The contents are preceded by one or more “headers” that uniquely identify the attachment (using a content identifier) and specify the format or type of the attachment. Note that the contents of an attachment can be binary octets (the “raw” binary data of the document), binary data encoded in text (e.g., via base-64 or some other algorithm), XML text or plain text.

Attachments appear in the message stream after the messages. The order of attachments within the message stream is not important and cannot be treated as significant. In particular, this means that the series of bytes representing the content of a lead document need not appear before the attachments representing the content of documents supporting that lead document.

Then since message sequence cannot be treated as significant, then the AttachmentSequenceNumbers show in Figure 2 should not be present (the actual element name is AttachmentSequenceId and not AttachmentSequenceNumber).

There may be a more difficult issue to address regarding the use of MTOM and an XOP processor. ECF Errata 01 deprecated the use of embedded attachments:

**Errata 01**

## Use of embedded attachments is deprecated [2.4]

In Section 2.3.1 Messages of the specification document, replace:

*The actual binary encoded electronic document MAY be either included in one or more attachments to the message or embedded in the message using the following structure:*

with

*The actual binary encoded electronic document SHOULD be included in one or more attachments to the message or MAY be embedded in the message using the following structure:*

As revised, section 2.3.1 appears as:

**Messages [2.3.1]**

* The message includes the document’s metadata, for example, its title, type, identifier, parent document identifier and document sequence number. Each document structure may reference one or more attachments, including attachment identifiers and sequence numbers. When included in attachments, a logical document MAY be split into several physical parts if necessary to satisfy a court requirement regarding maximum document size. ~~The actual binary encoded electronic document MAY be either included in one or more attachments to the message or embedded in the message using the following structure:~~ The actual binary encoded electronic document SHOULD be included in one or more attachments to the message or MAY be embedded in the message using the following structure:

<FilingLeadDocument> (or <FilingConnectedDocument>)

<ecf:DocumentRendition>

<DocumentRenditionMetadata>

<DocumentAttachment>

<BinaryBase64Object>2345klj345h…<BinaryBase64Object>

</DocumentAttachment>

</DocumentRenditionMetadata>

</ecf:DocumentRendition

</FilingLeadDocument> (or </FilingConnectedDocument>)

When using the WebServices SIP, section 1.2.7 requires the use of MTOM and XOP. XOP works by first locating XML embedded Base64 content, removing it to a MIME part in SOAP< and replacing the XML element content (i.e. BinaryBase64Object) with an xop:Include. An example of this is shown below:

MIME-Version: 1.0

Content-Type: Multipart/Related;boundary=MIME\_boundary;

 type="application/xop+xml";

 start="<mymessage.xml@example.org>";

 start-info="text/xml"

 Content-Description: A SOAP message with my pic and sig in it

 SOAPAction: "http://example.org/action/data"

 --MIME\_boundary

 Content-Type: application/xop+xml;

 charset=UTF-8;

 type="text/xml"

 Content-Transfer-Encoding: 8bit

 Content-ID: <mymessage.xml@example.org>

 <s11:Envelope

 xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'

 xmlns:xmime='http://www.w3.org/2005/05/xmlmime'>

 <s11:Body>

 <types:ReviewFilingRequest>

 …

 <CoreFilingMessage>

 <FilingLeadDocument>

 <DocumentRendition>

 <DocumentRenditionMetadata>

 <DocumentAttachment>

 <BinaryBase64Object><xop:Include xmlns:xop=http://www.w3.org/2004/08/xop/include href="cid:64449861-d66c-4540-86bc-4486bfa796cc">

 </BinaryBase64Object>

 <nc:BinaryLocationURI>

 cid:64449861-d66c-4540-86bc-4486bfa796cc

 </nc:BinaryLocationURI>

 </nc:BinaryLocationURI>

 </DocumentAttachment>

 </DocumentRenditionMetadata>

 </DocumentRendition>

 </FilingLeadDocument>

 …

 </CoreFilingMessage>

 <PaymentMessage>

 …

 </PaymentMessage>

 </types:ReviewFilingRequest>

 </s11:Body>

 </s11:Envelope>

 --MIME\_boundary

 Content-Type: image/png

 Content-Transfer-Encoding: binary

 Content-ID: <64449861-d66c-4540-86bc-4486bfa796cc>

 // binary octets for signature

 --MIME\_boundary—

Prior to XOP processing, the BinaryBase64Object element would have contained the Base64 content of the document rendition.

The problem seems to be the nc:BinaryLocationURI element content which should ideally point to the binary content, and as shown in the example above, contain the Content-ID value. Unless there is some way to specify Content\_ID values prior to XOP processing, how can the nc:BinaryLocationURI element be populated during XOP processing?

If this is not possible, then the BinaryLocationURI elements should also be removed from Figure 2.

**4. UBL 2.1**

Since UBL 2.1 was not yet an OASIS standard when ECF 4.01 was approved as an OASIS standard, then some namespace adjustments are now required. The following changes are required to schema:

Edit the **ECF-4.0-PaymentMessage.xsd** file; modify the schemaLocation for UBL. Change it from:

<xsd:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2" schemaLocation="http://docs.oasis-open.org/ubl/prd2-UBL-2.1/xsd/common/UBL-CommonAggregateComponents-2.1.xsd"/>

Change it to:

<xsd:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2" schemaLocation="http://docs.oasis-open.org/ubl/os-UBL-2.1/xsd/common/UBL-CommonAggregateComponents-2.1.xsd"/>

For the **ECF-4.0-FeesCalculationResponseMessage.xsd**, change it from:

<xsd:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2" schemaLocation="http://docs.oasis-open.org/ubl/prd2-UBL-2.1/xsd/common/UBL-CommonAggregateComponents-2.1.xsd"/>

To:

<xsd:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2" schemaLocation="http://docs.oasis-open.org/ubl/os-UBL-2.1/xsd/common/UBL-CommonAggregateComponents-2.1.xsd"/>

For the **ECF-4.0-PaymentReceiptMessage.xsd**, change it from:

 <xsd:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2" schemaLocation="http://docs.oasis-open.org/ubl/prd2-UBL-2.1/xsd/common/UBL-CommonAggregateComponents-2.1.xsd"/>

To:

<xsd:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2" schemaLocation="http://docs.oasis-open.org/ubl/os-UBL-2.1/xsd/common/UBL-CommonAggregateComponents-2.1.xsd"/>