**ECF v4.1 WD08 Feedback**

This document provides feedback for ECF 4.1 WD08 and ECF 4.1 WS SSIP WD05.

The WD08 core specification addresses nearly all of the prior feedback items (thank you Jim). However, a couple of items appear to have been overlooked or not fully addressed, and at least one new minor item has been noted.

1. Log #14 – in ‘Related Work’ (pg. 1), should ECF v4.01 Errata 01 be listed? It was decided to ask TC Admin. Errata 2 is listed but Errata 01 is not. Did OASIS TC Admin recommend including one Errata but not the other?

2. Section 1.2, Backwards Compatibility (Log #15). Revised to say “ECF 4.1 is not guaranteed to be backward compatible with previous version including ECF 4.0 and ECF 4.01 …”.

The full statement is: “This specification does not assume that prior specifications will be deprecated. However, ECF 4.1 is not guaranteed to be backward-compatible with previous versions including ECF 4.0 and 4.01 and applications using the ECF 3.0, 3.01 and 3.1 specifications will certainly not interoperate successfully with applications using these specifications.”

This appears to be subject to multiple interpretations. Perhaps this can be improved as:

This specification does not assume that prior specifications will be deprecated.  However, ECF 4.1 is not guaranteed to be backward-compatible with previous versions including ECF 4.0 and 4.01, both based on NIEM 2.x.  Applications based on ECF versions which themselves are based on NIEM versions other than NIEM 2.x (such as ECF 3.0, 3.01 and 3.1 specifications) will certainly not interoperate successfully with applications using this specification.  This fact is indicated by the assignment of a new major and minor version number to the specifications.

Furthermore, TC Admin advises:

Describe precisely what are the non backward compatible features, and on the public relation side, explain in an FAQ why it wasn’t possible to preserve compatibility. A section describing the “diff” from V(n) to V(n+1) will greatly help implementers of the new version to understand what they can and cannot reuse from their implementation of the previous version. It may help also to specify how features of a previous version map to – have been replaced by – similar features in the new version. Now, some subset of features may still be backward compatible if not all features, and these should be identified. That will greatly help users who may actually be only concerned with these features, as well as developers who will try to architect a multi-version implementation. Describe the expected effect of using the “wrong” version when the new one (or the old one in the forward compatibility case) is handled. You might be able to define a related (and restricted) conformance profile for which backward compatibility is supported. The whole point is to honestly help users understand what can and cannot be reused from a previous implementation or environment.

It is not clear whether the above OASIS recommendation should be included within the specification document or should be provided in a separate Project Note. If provided as a separate document, its publication should be timely with the specification publication.

Recommendation: The TC should provide a Project Note document addressing compatibility between ECF 4.01 and ECF 4.1 as recommended by OASIS TC Admin. This document should be provided timely with the ECF v4.1 core specification and ECF WS-SIP v4.1 specifications.

3. In prior feedback, it was noted that wrappers.xsd was not listed in section 4.2 ECF 4.1 Common Schemas. It was agreed (log #17) to include wrappers.xsd in a new section (name to be determined).

I cannot identify any change that meets this expectation. As such, the new wrappers.xsd is still absent from section 4 ECF 4.1 Schemas.

4. Asynchronous response optionality – Log #23. It was agreed that the core specification would address this having been previously discussed by the TC. I am not finding this in the revised specification.

5. Section 3.2.8 NotifyFilingReviewComplete includes the requirement to return a SHA-256 document hash. After Phil raised his questions, and following TC discussion on May 1, 2023, I had expected further clarification of this requirement. Recall that it merely states that the SHA 256 document hash is “intended to protect document integrity”.

As discussed at the 5-1-2023 TC meeting, this is vague and has been interpreted in different ways.

One explanation of protecting document integrity as that since with some message transport protocols, the document rendition content (e.g., binary content) may be carried separate from the XML (e.g., as an attachment as described in section 2.3.2), and the SHA 256 hash provides a means to verify that the separate binary content arrives without alteration.

One TC member interpreted this SHA 256 hash requirement as a means to permit the filer to be assured that the CRMDE received the original submitted document rendition without alteration. In this usage, the CRMDE would hash the document rendition received from the filer and provide the hash in the NFRC for use by the filer as verification.

Most TC members understood that the SHA 256 hash requirement intended for the hash to be of the document rendition ‘filed’ in the CRMDE (e.g., file stamped rendition, etc.) and not as provided by the filer in the FAMDE.

The meeting minutes for issue #27 record: Resolved by clarifying that the hashes are for the document as accepted by the clerk.

It was my expectation that this issue would be clarified in the revised working draft specification.

Both RecordDocketingCallbackMessage and ReviewFilingCallbackMessage include the DocumentHashText element which provides the following element documentation:

A hash of the document as it appears in the court record. This attribute will be populated by either the clerk review process or the court record system. If the latter, then it will be absent in the RecordDocketingMessage. It will also be absent in callbacks for rejected documents.

Are element documentation/comments binding (e.g., normative)?

I think not (e.g., are illustrative, not normative).

How should the SHA 256 hash requirement be specified in the core specification?

The requirement should:

* Spell out exactly which state of the filing document or rendition is to be hashed – e.g., as provided by the filer or as retained (e.g., ‘filed’) by the CRMDE and returned by GetDocument (e.g., as it appears in the court record, e.g., as accepted by the clerk in clerk review).
* Clarify the purpose for the hash (e.g., “protect document integrity by …”).
* Consider providing requirements or guidance on the generation of the SHA-256 hash and its use.
* Correct the informative DocumentHashText element documentation, if necessary, due to any core specification edits.
* Since the DocumentHashText element appears in multiple messages (i.e., RecordDocketingMessage, RecordDocketingCallbackMessage, and ReviewFilingCallbackMessage), then any requirements or informative usage information should be provided in the core specification in a manner that is clearly applicable to any or all messages in which this element appears. Currently, the SHA-256 hash requirement is only addressed for NotifyFilingReviewComplete.

The ECF TC should consider why “protecting document integrity” is not important in a ReviewFilingRequest and other operations such as GetDocument.

6. In Appendix A, A.2 ‘Package Structure’, the directory path ‘ecf-4.1/’ was removed. This leaves an awkward sentence, i.e., “Unzipping this archive creates a directory named containing this specification document and a number of subdirectories.”

Recommend rewording as: “Unzipping this archive creates a directory containing this specification document and a number of subdirectories.”

7. WS-SIP WD05 – New Namespaces. The previous 34 ECF 4.1 namespaces have been removed from ‘Declared XML namespaces’ but the four new WS-SIP v4.1 namespaces are not listed:

urn:oasis:names:tc:legalxml-courtfiling:schema:wsdl:CourtRecordMDE-4.1

urn:oasis:names:tc:legalxml-courtfiling:schema:wsdl:FilingAssemblyMDE-4.1

urn:oasis:names:tc:legalxml-courtfiling:schema:wsdl:FilingReviewMDE-4.1

urn:oasis:names:tc:legalxml-courtfiling:schema:wsdl:ServiceMDE-4.1

Is it correct to not include the four WSSIP namespaces in the ‘Declared XML namespaces:’ listing?

Since these above 4 namespaces are unique to WS-SIP v4.1, and are defined in the specification WSDL, then in my opinion, these 4 namespaces should be listed.

8. Cardinality revisions for RecordDocketingCallbackMessage within NotifyDocketingComplete (log #31) and for ReviewFilingCallbackMessage in NotifyFilingReviewCompleteRequest (Log #32) were raised in feedback. There were several things that required consideration to reach the cardinality consideration objective. These discussion items were listed and enumerated in the feedback document.

The TC decided to adopt most of the recommendations by agreeing that the operation signatures should be defined in the core specification rather than in SIP specifications and that Appendix C would be made normative and moved into a numbered core specification section (in WD08, this is now section 5 ‘MDE Operations’).

However, I do not recall that the TC ever actually discussed the cardinality issues raised in the feedback. As such, these cardinality considerations appear to still be pending.

Note: in the new Section 5 ‘MDE Operations’ the callback message cardinalities are as shown below for the two operations:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Called By** | **Output** | **Parameters** |
| NotifyDocketingComplete | Court Docketing MDE | xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage (1,1) | xsd/message/ECF-4.1-RecordDocketingCallbackMessage.xsd : RecordDocketingCallbackMessage (1,1) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Called By** | **Output** | **Parameters** |
| NotifyFilingReviewComplete | Filing Review MDE | xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage (1,1) | xsd/message/ECF-4.1-ReviewFilingCallbackMessage.xsd : ReviewFilingCallbackMessage (1,1) |
| xsd/message/ECF-4.1-PaymentReceiptMessage.xsd : PaymentReceiptMessage (1,1) |

In the original feedback the following was requested:

c. Allow multiple RecordDocketingCallbackMessage elements within a NotifyDocketingComplete operation invocation request.

c.1 If NotifyDocketingComplete is not revised to permit multiple RecordDocketingCallbackMessages, then revise section 3.2.7 ‘NotifyDocketingComplete’ to require a callback message for each FilingLeadDocument that has been provided on the RecordDocketing request. These multiple NotifyDocketingComplete requests, responding to a single RecordDocketing request, need not all be provided in quick succession (e.g., time gaps, short or long, between callback messages are permitted).

And

Allow multiple ReviewFilingCallbackMessage elements within a NotifyFilingReviewCompleteRequest. Note that if NotiyDocketingComplete is revised to allow multiple RecordDocketingCallbackMessage elements (see #20 2.c above) then this revision is essential.

c.1 If NotifyFilingReviewCompleteRequest is not revised to permit multiple ReviewFilingCallbackMessages, then revise section 3.2.8 ‘NotifyFilingReviewComplete’ to require a callback message for each LeadDocument that has been provided on the ReviewFiling request. These multiple NotifyFilingReviewCompleteRequests, responding to a single ReviewFiling request, need not all be provided in quick succession (e.g., time gaps, short or long, between callback messages are permitted).

When considering these suggested cardinalities, the TC may want to consider that the suggested changes would better align ECF 4.1 to ECF 5 which currently allows unbounded callback messages within NotifyDocketingCompleteRequest and NotifyFilingReviewCompleteRequest.

9. Bulk and Batch filing

The core specification recognizes that ‘batch’ filing ‘is a thing’ (e.g., see 2.2.2 ‘Machine Readable Court Policy “Whether the court accepts multiple (batch) filings”. No definition for ‘batch’ filing is provided, but it suggests that is involves multiple filings (e.g., ReviewFilingRequests). It does not suggest that these multiple filings are for the same case or multiple different cases.

The Web Services SIP specification (wd05) announces “this version adds support for bulk filings.” No definition for ‘bulk’ filing is provided nor does the specification provide information about how ‘bulk’ filing is supported by the specification.

When discussed by the TC, it became clear that some members consider ‘bulk’ filing and ‘batch’ filing to be two different concepts.

One concept (I think it was ‘batch’) consisted of multiple normal ECF filing transactions (e.g., RvFRs) submitted continuously in rapid succession.

The other concept (‘bulk’?) involved a single transaction that consisted of multiple filings (e.g., a single operation invocation providing multiple RvFRs). In this single operation invocation, a single PaymentMessage may apply to more than one CoreFilingMessage.

This ‘bulk’ filing concept does not appear to be supported either by the core specification or the Web Service v4.1 specification.

It may be useful to note that neither ‘batch’ nor ‘bulk’ filing are mentioned within the ECF 5.01 core specification, but the term ‘bulk’ filing is still used in the ECF Wed Services SIP v5.01 (wd03), i.e., “This version adds support for bulk filings.”

Recommendations:

a. Define the meaning of the terms ‘batch filing’ and ‘bulk filing’ and include these in the glossary in both the core specification and the Web Services SIP specification.

a.1 When considering the definition for ‘batch filing’, take note of the ECF v5.01 ReviewFilingRequestType which consists of one or more FilingMessages and an optional PaymentMessage.

a.2 When defining the structure for ‘batch’ and/or ‘bulk’ filings, consider whether the core specification should support operation invocations in which the invocation parameters may be provided in multiplicity within the operation signature (e.g., more than one set of parameters within a single invocation).

For example, ECF 4.1 WD08 provides the following for the ReviewFiling operation:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Called By** | **Output** | **Parameters** |
| ReviewFiling | Filing Assembly MDE | xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage (1,1) | xsd/message/ECF-4.1-CoreFilingMessage.xsd : CoreFilingMessage (1,1) |
| xsd/message/ECF-4.1-PaymentMessage.xsd : PaymentMessage (0,1) |

This is understood as the operation signature for the ReviewFiling operation consists of one and only one CoreFilingMessage and an optional PaymentMessage.

The CoreFilingMessage and PaymentMessage comprise a parameter-set. Incidentally, this is the same XML schema definition for ReviewFilingRequest within the ECF 4.1 wrappers.xsd.

As such, and to permit operation invocation consisting of one or more of the above parameter-sets, then the following may be necessary or more appropriate:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Called By** | **Output** | **Parameters** |
| ReviewFiling | Filing Assembly MDE | xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage (1,1) | xsd/wrappers.xsd : ReviewFilingRequest (1, unbounded) |

a.3 Consider the implications of ‘batch’ and ‘bulk’ filing definitions on operation invocations other than just ReviewFiling, such as RecordFiling, NotifyDocketingComplete, NotifyFilingReviewComplete, ServeFiling, GetFeesCalculation, GetFilingList and GetFilingStatus.

b. Modify specification documents and artifacts as necessary to properly describe and support definitions for ‘batch’ and ‘bulk’ filings.

 b.1 For the Web Services SIP v4.1, this may mean to remove the statement that “This version adds support for bulk filings.” If it is appropriate to retain this declaration (as is or modified as ““This version provides support for bulk filings”) then include information within the Web Service SIP specification as to how bulk filing is supported within the specification.

10. Element description for CoreFilingMessage (log #29) – not revised.

It was greed that the element documentation within schema for CoreFilingMessage would be revised. The suggested revised description provided in the feedback document is:

The structure of a Filing will be documented in this section. This describes the proposed or actual filing transaction between the Filing Assembly MDE and the Filing Review MDE. This information may become part of the Record Docketing between the Filing Review MDE and the Court Record MDE but does not necessarily describe the information that may be actually stored in the Court Record.