**ECF5 Spec Feedback and Considerations**

Responses from Jim Cabral in Red

This document contains questions and commentary resulting from a “first look” at the Electronic Court Filing Version 5.0 Working Draft 06 document dated March 9, 2017. In many respects, this is a document of “first impressions”, at least from the perspective of a person who is well familiar with ECF 4. This document is not intended to identify all difference with ECF 4, or even to compare and contrast, although some comparisons have been made. For the most part, this document was reviewed without diving into the included technical information (e.g. schema, etc.). Once the technical information has been explored, some questions may be answered. Since however, some readers of this document may never venture into the technical details, this ‘top only’ read may be the full depth of specification understanding that some readers will achieve. As such it should provide a thorough description of the specification as well as an entry point into further detailed explorations.

1. Should the definition of ‘Filer’ in be expanded?

The existing definition is “an attorney or pro se (self-represented) litigant acting as an individual who assembles and submits one or more filings (combinations of data and documents).”

What about other types of filers? E.g. judges, guardians, etc.

Agreed – I added judicial officials. Guardians are either represented by an attorney or pro se litigants.

1. (Informative) Scope

Consider adding:

* Transmission of data modified (e.g. corrected) in the clerk review operation in addition to the unmodified data originally provided by the filer.

Added

1. In section 2.1.1 National Information Exchange Model (NIEM) – I suggest a minor rewording.

Current wording: “The [NIEM] is an XML standard designed specifically for justice information exchanges, …”

Suggested rewording: “The [NIEM] is an XML standard that includes specifications designed specifically for justice information exchanges …”

Changed this to The **[NIEM]** is a framework that enables efficient cross-domain information exchanges

1. Section 3.1 Major Design Elements

The third paragraph following the bullets, begins with “Each of the operations supported by an MDE accepts one or more messages as input and returns an immediate, synchronous response message to the calling MDE.”

This should be reworded since not all SIPs may support or require a synchronous response message. For example, the IBM MQ SIP which uses MQ as a transport is guaranteed delivery. As such, a response message to confirm receipt of the request message is not required.

I change this to “typically returns an immediate, synchronous response message”. Note that, the response is rarely just an acknowledgement. Even with reliable transport, many operations return important information in the synchronous response.

1. Section 3.2.1 The Filing and Service Process

In the second paragraph, it states “The operations in bold are required and MUST occur in every successful filing as long as sending and receiving MDEs are implemented.” (underline added) What is the significance or intent of the underlined language? Does this suggest that not all MDEs may be implemented, and if not implemented, then the operation is not mandatory? For example, if the FRMDE is not implemented, then the ReviewFiling operation is not mandatory in the “Filing and Service Process”. Note however that in section 3.1 Major Design Elements it states that “a complete ECF 5.0 system MUST include at least one each of the Filing Assembly, Filing Review and Court Record MDEs”.

Your interpretation is correct and the language is consistent. As some MDEs are not required (e.g. Service MDE, Court Scheduling MDE), those operations are not required.

Regarding the CancelFiling operation, is states: “At any point during or after the ReviewFiling operation and before the RecordFiling operation, if the filing is accessible, a party MAY request …” (underline added). This same “if the filing is accessible” qualifier is also used for DocumentStampInformation. What does it mean for a filing to be accessible? To whom or what must it be accessible? Under what circumstances or conditions would a filing not be accessible? How is it accessed?

Removed “if the filing is accessible” from each of these occurrences.

1. Section 5.2.2 Court-Specific Code Lists

I am having some trouble understanding what is being said here beneath the table. It states:

The “default values” in the above table column indicates whether the **[Genericode]** code list provided in the specification includes non-normative default values. If a court does not reference a court-specific code list for any of the following XML elements in court policy, then any value MUST be considered acceptable for the corresponding XML element.

However no XML elements appear to be listed. Immediately following the above paragraph, there is:

The default values are provided in the Genericode file referenced in the “Genericode code list” column, not in the written specification

Court policy MUST include a policyresponse:RuntimePolicy/policyresponse:CodeListExtension element for each court-specific code list. The latest version and URL of all court-specific code lists MUST be defined using the policyresponse:ExtensionCanonicalVersionURI and policyreponse:ExtensionLocationURI elements, respectively.

Is this ‘CodeListExtension’ element the only element in the list referenced in the first paragraph, or is the list absent and the second paragraph is unrelated to the prior paragraph?

Implementations are required to reference court-specific code lists in Court Policy (the response to a GetPolicy operation). This paragraph defines the required elements in Court policy that reference each court-specific code list.

1. Section 6.2.8 Filer and Party Identifiers

This section only addresses persons and organization. Property can also be a party (e.g. in rem property) and should also be addressed.

Revised as “person, organizations and property”

1. Section 3.2.1 The Filing and Service Process

Why is the ‘DocumentStampInformation’ the only process that does not begin with a verb? Shouldn’t this be named “GetDocumentStampInformation”?

DocumentStampInformation is an asynchronous operation, so it should not begin with “Get”.

1. If it isn’t already, it needs to be clear that 3.2.1 **The Filing and Service Process** is separate and independent of the 3.2.2 **The Scheduling Process**.

e.g. the operations required for 3.2.1 are different than the operations required for 3.2.2. This relates to compliance, conformance, and completeness.

Added – “This process and operations are separate and independent of the Filing and Service Process.” to 3.2.2 The Scheduling Process section.

1. Section 3.2.2 The Scheduling Process – It states “the operations in bold are required and MUST occur in every successful filing as long as the sending and receiving MDEs are implemented.” (underline added)

This is similar to #5 above for section 3.2.1 The Filing and Service Process. Again, since section 3.1 Major Design Elements it states that “a complete ECF 5.0 system MUST include at least one each of the Filing Assembly, Filing Review and Court Record MDEs” is this qualifier only provided for incomplete ECF 5.0 systems?

Also, it is not clear what a “complete’ (or ‘compliant’) ECF implementation is now that the Scheduling Process has be included. Previously, an ECF implementation was ‘compliant’ if it implemented the FAMDE, FRMDE, and the CRMDE and provided the ReviewFiling, RecordFiling, NotifyDocketingComplete, and NotifyFilingReviewComplete operations. With ECF 5.0, can an implementation be ‘complete’ (or ‘compliant’) if the Scheduling Process is not supported?

Revised to “The operations in bold are required and MUST occur in every successful filing as long as a Court Scheduling MDE is implemented.“

1. Section 3.2.2 The Scheduling Process – similar to the Filing and Service Process, for the Scheduling Process it states “At any point during or after the ReviewFiling operation …” (underline add). I think this must be a mistake since the ReviewFiling operation does not occur in the Scheduling Process. Perhaps this should be the ReserveCourtDate operation instead.

The statement referred to above continues with “… if the filing is accessible, a party MAY access information through the following operations: …”. It then lists just the GetCourtSchedule operation.

Does this mean that a party cannot use the ‘GetCourtSchedule’ operation until the party has first employed the ReserveCourtDate operation?

Revised to “At any point during the Scheduling Process , a party MAY access information through the following operation:”

1. Section 3.2.2 The Scheduling Process – can the court initiate this conversation? E.g. can the court begin this conversation with NotifyCourtDate()? If so, then in this scenario, ReserveCourtDate() may need to be optional.

Very good point – I made ReserveCourtDate and AllocateCourtDate optional.

1. Section 5 Court Policy – the paragraph in ECF4 relating to versioning of policy is absent from the ECF5 specification. Why has this been removed?

It isn’t clear to me that paragraph was adding any value.

Furthermore, one problem often discussed regarding machine readable policy (MRP) relates to the burden of requesting and receiving this MRP; e.g. (1) in a multiple court implementation, a request for MRP returns all MRP for all courts, (2) the size of MRP (either for a single court or multiple courts, and (3) the frequency of change/requests.

Regarding the last item, it had been suggested that when requesting MRP, the requester should be able to provide the version number of the MRP in the requester’s (i.e. FAMDE’s) possession. The CourtPolicyMDE (CPMDE) could respond with a small response message when the FAMDE has current MRP and thus avoid returning all MRP when the FAMDE already has the current version of MRP.

I don’t think we want to define 2 different messages (one when the version is current and one when it is not) which would be a non-standard message exchange pattern and would likely be difficult to implement in some messaging frameworks (e.g. web services).

1. Section 6.1.16 ReserveCourtDate

It specifies that the FAMDE may “request one or more court dates” using ReserveCourtDate (presumably all in the same ReserveCourtDate transaction. If multiple dates are requested in a single ReserveCourtDate operation, then even if the Court Record MDE (CRMDE) ‘accepts’ (or approves, grants, etc.), then the other dates will have been ‘rejected’. However, the specification says “if the initial date(s) requested are rejected, as described below in the NotifyCourtDate operation, …”. The NotifyCourtDate operation states that “a Court Scheduling MDE MUST invoke the NotifyCourtDate operation on the Filing Assembly MDE that invoked a ReserveCourtDate operation to accept or reject the date(s) requested in the ReserveCourtDate operation.”

So if the FAMDE requests 3 dates, and one is accepted and 2 are rejected, then is the result for each of the three requested dates communicated in a single NotifyCourtDate operation, or is this multiple operations?

Revised to “A Court Scheduling MDE MUST invoke the NotifyCourtDate operation on the Filing Assembly MDE that invoked a ReserveCourtDate operation to either accept one of the dates or reject all the date(s) requested in the ReserveCourtDate operation. “

1. Section 6.2 Identifier Rules

I think it would be useful to map the terminology used in this section to the XML elements that contain these identifiers, (perhaps also with examples or illustrations). This appears to be done for 6.2.8 Filer and Party Identifiers.

Other Examples:

**Case Identifiers**

Case identifiers (case numbers, e.g. nc:CaseTrackingID) are assigned …

**Filing Identifiers**

Filing Identifiers MUST be unique within a court and will be generated by the court in response to a ReviewFiling operation. For the cbrn:MessageStatus (note: formerly known as MessageReceiptMessage) the filing identifier is carried in the ?????? element (Note: formerly in the nc:DocumentIdentification\nc:IdentificationID element).

Added non-normative examples of each identifier to the specification.

Also note that there is no schema appears to have been included in the ecf-v5.0-wd06.zip file for MessageStatus:



The schema is provided in schema/niem/domains/cbrn/3.2/cbrn.xsd

In FilingMessage (formerly known as CoreFilingMessage), filing identifiers are carried in the nc:DocumentIdentification\nc:DocumentID element.

1. While on the topic of mapping, there also appears to be a need to map messages (e.g. filing:FilingMessage) to schema (e.g. filing.xsd). This does not appear to be a straightforward as in prior versions of ECF.

Note: there is a message to example mapping in Appendix E. How about a table like this that maps the message to its schema.

This mapping is provided in the table in 4.1 Messages. Each schema links to the appropriate schema.

1. Section 6.2.8 Filer and Party Identifiers

Why were the additional identifier elements mentioned in ECF 4(i.e. nc:PersonOtherIdentification, nc:OrganizationOtherIdentification, ecf:FilingPartyID, and ecf:FilingAttorneyID), removed from the narrative?

ECF 5 introduces a new element, ecf:FilerIdentification, that replaces all of these.

1. Section 6.2.8 Filer and Party Identifiers

The statement at the end of the last paragraph should be modified to say “The attorney elements for a self-represented litigant SHOULD NOT include a bar number, unless of course the self-represented litigant is also an attorney.” This is not infrequent in matters of attorney discipline where the attorney who is undergoing a disciplinary proceeding is self-represented (not always though, sometimes they will retain counsel).

ECF seems to suggest that self-represented litigants should be expressed “using both attorney and party elements”. This is stated as “MAY” and not “MUST” and as such may be construed as an ECF ‘best practice’ recommendation.

Revised to “Attorney elements MAY reference the parties they represent with party identifiers. Self-represented litigants that are also an attorney MAY be represented using both attorney and party elements for the same individual, with a reference from the attorney element to the party element. Otherwise, the attorney elements for a self-represented litigant SHOULD NOT include a bar number.”

However, in Arizona, we follow a different approach as we do not concur with the practice of using attorney elements to represent self-represented litigants who are not attorneys.

For the benefit of those who may want to know self-represented litigants are expressed in AZ in ECF 4.01, I am providing the following:

**Self-Represented Litigants**

Parties who are self-represented litigants (aka pro se or pro per) must be explicitly declared as such in the XML. The absence of an Attorney-to-Party relationship does not denote that a party is self-represented, nor does the presence of Attorney-to-Party relationship indicate that the part is not self-represented. The attorney representing a party can be identified in the XML through: (1) ecf:CaseRepresentedPartyReference in ecf:CaseOfficial within j:CaseAugmentation; or (2) as aoc:AttorneyReference within aoc:LitigantGroup. So, if a party identified in j:CaseAugmentation does not have an ecf:CaseRepresentedPartyReference, does this make the party a self-represented litigant? The answer is no. This situation can arise for varied reasons: (1) in a case initiation RFR, typically the opposing parties are identified, but the attorneys for the opposing parties are not specified.  This should not be interpreted that the opposing parties are self-represented; (2) self-represented litigants can have attorneys (e.g., advisory counsel).

Currently ECF does not provide an adequate means to identify self-represented litigants. (see ECF section 3.3.1.7 for additional information)  Additionally, there does not appear to be any specific element within LegalXML for this purpose. A search of NIEM yielded:

**j:CaseDefendantSelfRepresentationIndicator** (niem-xsd:Boolean)

True if a defendant waived the right to have a defense attorney and is representing him or herself; false otherwise.

This NIEM element would be right on the mark if it were it not case party type role specific (i.e. defendant) and there are no other similar party role type specific elements in NIEM (e.g. CasePlaintiffRepresentationIndicator, CaseAppellantSelfRepresentationIndicator, etc.). Therefore, the AOC standard is to use a new to 2.0 extension element that does not contain a party role type in its name and can therefore be used for any and all self-represented litigants regardless of party role type. This new element is:

**aoc:CasePartySelfRepresentationIndicator**  (niem-xsd:boolean)

True if a case party waived the right to be represented by an attorney and is representing him or herself; false otherwise.

Example: Within the Case element (e.g., aoc:AppellateCase):

 <j:CaseAugmentation>

 <j:CaseInitiatingParty>

 <nc:EntityPersonReference s:ref="icc0aaa97-d68c-4dec-8590-0e83232a7453"/>

 </j:CaseInitiatingParty>

 </j:CaseAugmentation>

 <aoc:CaseAugmentation>

 <aoc:CaseParticipant>

 <ecf:EntityPerson s:id="icc0aaa97-d68c-4dec-8590-0e83232a7453">

 <nc:PersonName>

 <nc:PersonGivenName>Donna</nc:PersonGivenName>

 <nc:PersonSurName>Jackson</nc:PersonSurName>

 <nc:PersonFullName>Donna Jackson</nc:PersonFullName>

 </nc:PersonName>

 </ecf:EntityPerson>

 <ecf:CaseParticipantRoleCode>Respondent</ecf:CaseParticipantRoleCode>

 <aoc:CasePartySelfRepresentationIndicator>true</aoc:CasePartySelfRepresentationIndicator>

 </aoc:CaseParticipant>

 </aoc:CaseAugmentation>

1. Section 6.3.3 docket:RecordDocketingMessage states: “the court record system SHOULD retain all complete message transmissions, including any message envelopes and headers defined by the service interaction profile, for evidentiary purposes.”

This seems like a good place to include some mention of the new docket:CorrectedFiling in docket:RecordDocketingMessage and how and when to use it. Some mention in 6.1.7 RecordDocketing may also be useful.

Added “If the clerk made any modifications to the original filing, the modified filing SHOULD be included in the docket:CorrectedFiling element.”

1. I notice that the docket:RecordDocketingMessage.xsd has docket:CorrectedFiling as minOccurs = 1 (i.e. mandatory). I thought we had agreed that this would be optional – not all courts will permit clerk review corrections/modifications.

Modified the multiplicity to 0,1

1. Section 6.3.1 filing:filingMessage – the second paragraph includes: “ If a filing:filingMessage includes multiple renditions of the same document, the nc:BinaryDescriptionText element SHOULD be used to determine how to process multiple renditions of the same document.”

This seems like a good place to include some additional guidance and perhaps an example or two of how this is intended to be done.

Added an example.

1. Why does ecf:CaseFilingType only include j:CaseCourt and not nc:Case?

Without including nc:Case shouldn’t it be called CourtFilingType and not CaseFilingType?

NIEM includes an element nc:CaseFiling based on nc:DocumentType. ecf:CaseFiling is the ECF extension of that element.

1. Section 6.4.1 Appellate Rules – bullet 9 – I believe the terminology “FilingCase object” need to be updated to just “Case object”.

Revised to “If the ROA transaction is a case initiating filing in the destination court, then the nc:Case object MUST be present and the nc:CaseTrackingID MUST be absent.”

Also, bullet 10 – probably needs updating since it is referring to “case type-specific and court-specific extensions”. How to adjust this language is a little unclear. In ECF 4.01 section 2.4.1 Case-Type and Court Extensions addressed this topic. In ECF 5, it appears that this is now superseded by augmentations. There is a section in ECF 5 which addresses case type augmentations (i.e. section 4.2) and court augmentations is addressed in section 5.2.1.

Revised to “Each predecessor case identified in the target case’s case lineage may include case type and court-specific augmentations. The case type and the case type augmentations for each predecessor case MUST be consistent throughout the case lineage.”

1. Conformant, Compliant, or Complete?

Section 9 addresses **Conformance** and states: *An implementation conforms with the Electronic Court Filing Version 5.0 if the implementation meets the requirements in Sections 1, 3, 4 and 5 including conformance with the XSD schemas and* **[Genericode]** *code lists referenced in Sections 4 and 5.* (underline added)

Section 1 Introduction states:

In order to be compliant, an implementation of the ECF specification MUST implement the core specification and at least one service interaction profile and one document signature profile. (underline added)

Section 3 Service Model, uses the term compliant as in:

An ECF 5.0-compliant implementation may implement one or more of the MDEs defined in the specification but a complete ECF 5.0 system MUST include at least one each of the Filing Assembly, Filing Review and Court Record MDEs. (underline added)

And again:

In order to be compliant with ECF 5.0, an MDE must support all messages required for that MDE. (underline added)

Also, section 5 Human-Readable Court Policy states:

To be compliant with the ECF 5.0 specification, each court MUST publish a human-readable court policy that MUST include each of the following: (underline added)

These terms (i.e. conformant, compliant, complete) may need to be sorted out. Definitions of each perhaps should also be included, and we may want to include some type of conformance/compliance/completeness checklist in this section, perhaps similar to one that was created/drafted for ECF 4.

Globally replaced “compliant” with “conformant”

1. With the introduction of MessageWrappers and document/literal style, is the definition for ‘message’ provided in section 4.1 Messages still correct?

This definition is: “A message is an XML document that is a well-formed XML data structure with a single root element that is valid as defined by a normative XML schema provided with the specification.”

It seems that with a message wrapper such as RecordDocketingRequest, which binds docketing:RecordDocketingMessage with payment:PaymentMessage (optionally), the RecordDocketingRequest is passed from the FRMDE to the CRMDE is the message.

Consider the circumstance with there is no PaymentMessage; the FRMDE would still send RecordDocketingRequest and not RecordDocketingMessage.

Revised to “A message is an XML document that is a well-formed XML data structure with a root element that is valid as defined by a normative XML schema provided with the specification. “

1. Section 4.3 Code Lists

This section states: “The allowable values for the following XML elements are normative for all ECF 5.0 implementations and are defined in ECF **[Genericode]** code lists or NIEM or UBL XML schema.” (underline added)

Since these are normative, it is important to be able to locate these code lists. The first 4 in the table are relatively easy to fide (they are in the schema folder), however, those identified as [NIEM] XML schema are not so easy to locate. Where are these lists? Where are the [UBL] XML schema code lists?

Added links to the schemas.

1. Section 5.2 Machine-Readable Court Policy

Item 7 on the enumerated list has been revised from ECF4 (note this is item 6 in ECF4):

ECF5: “Whether the court accepts multiple lead documents in a single filing (batch filing).”

ECF4: “Whether the court accepts multiple (batch) filings.”

The revision seems to imply that the definition of a “batch filing” is a filing with multiple lead documents. This is not correct. Although a “batch filing” may have multiple lead documents, it is my understanding that a ‘batch filing’ is a whole bunch of RvFRs in a collection or continuous stream. However, since there is no MessageWrapper for a collection of RvFRs, then it must be a quick series of RecordFilingRequests, one after the other in rapid succession.

Revised to “Whether the court accepts multiple lead documents in a single filing.”