**ECF5 Spec Feedback and Considerations – 12**

This document contains additional questions and commentary resulting from a review at the Electronic Court Filing Version 5.0 Working Draft 16.

Jim Cabral’s responses in red.

1. **ReviewFilingRequest Example**

The new civil-ReviewFilingRequest-01.xml example is not valid. The error reported by XMLSpy is:

Element <ecf:CaseParticipantRoleCode> is not allowed at this location under element <ecf:CaseOfficialAugmentation> ‘AttorneyID’ is expected at this location.

Since the civil-ReviewFilingRequest-01.xml example was valid when I submitted it, and since ecf:FilingAttorneyID was renamed to ecf:AttorneyID and made mandatory in wd16, I suspect this may have something to do with this error.

This indeed was the cause. The ‘civil-ReviewFilingRequest-01.xml’ example has been revised to include ecf:AttorneyID.

Replaced the sample with this version.

1. **Attorney ID in Docket example**

I also modified the docket.xml example now that ecf:AttorneyID is available.

At line 306, I changed the ecf:FilingAttorneyID/nc:IdentificationID value to 10 from 100. I then removed line 307 which contained the following comment:

<!-- modified from 13 to 100, the attorney's Bar Number since in ECF5 there is no Attorney ID-->

I removed this comment because now ECF5 does have an Attorney ID.

The same was done at lines 346 and 347.

Replaced docket.xml with this version.

1. **Entity Item Augmentation**

As of wd16, EntityItem does not have an augmentation element (e.g. ItemAugmentation), as EnityPerson and EntityOrganization do. Without any augmentation element, EntityItem cannot have ecf:CaseParicipantRoleCode or ecf:FilingPartyID elements.

Added ecf:ItemAugmentation which includes ecf:CaseParticipantRoleCode.

1. **Associating Reviewed Document to Filing Documents**

The ‘ECF Spec Feedback and Considerations – 11’ document, item 7, notes:

However, since both Lead and Reviewed Documents can have more than one nc:DocumentIdentification/nc:IdentificationID, the specification must go further and detail how the appropriate nc:DocumentIdentification/nc:IdentificationID is known (such as with a specific nc:IdentificationCategory element value, e.g. ‘ECFDocumentID’).

You asked for clarification with:

The changes to 6.2.4 already state that DocumentIdentification must be used. It isn’t clear to me how common the problem you identified would occur or that adding a nc:IdentificationCategory of “ECFDocumentID” solves the problem you identified.

Let me try to explain.

First look back at ‘ECF Spec Feedback and Considerations – 5’, item 11 DocumentIdentification cardinality.

In this discussion, three example usages of Documentidentification were provided. All three usages may occur within a single RvFR and RFR in Arizona. As a result, the cardinality of nc:DocumentIdentification was modified to allow many.

So even though the cardinality change was done to allow more than one nc:DocumentIdentification for the submittal, this cardinality is also present for documents (e.g. filing:FilingLeadDocument).

So since it is possible for a document to have more than one nc:DocumentIdentification, then one of these must be the ‘Document Identifier’, while the others (for whatever purpose) are not the ‘Document Identifier’ (unless you contend that both/all are ‘Document Identifiers’ per section 6.2.4 Document Identifiers).

So since the cardinality allows for multiple nc:DocumentIdentifier elements, additional information would need be applied to select the appropriate element from the set, such as a name/value pair approach as illustrated in item 11, in ‘ECF Spec Feedback and Considerations – 5’. This additional information would not be necessary in implementations that never use more than one Documentidentifier.

As you said, this isn’t necessary in implementations that never use more than one DocumentIdentifier. In implementations that do use have more than one, I suggest they can define an extension for nc:IdentificationCategory.

1. **Cancel Filing Response**

This continues the discussion from ‘ECF Spec Feedback and Considerations – 11’, item 5.

You suggest discussing the need for a CancelResponseFiling with the TC and I agree.

Your response in the document referred to above, and specification edits made in section 6.1.10 NotifyFilingReviewComplete, seem to suggest that you are thinking that this CancelFilingResponse message may not be necessary because the NotifyFilingReviewComplete operation would be used instead when the clerk agrees to cancel.

I see that there is already a ‘cancelled’ code in the FilingStatusCode.gc file.

Assuming that this is your intention, and although this does not seem unreasonable, I would like to see an example docketcallback:NotifyDocketingCompleteMessage for this cancellation circumstance.

It does seem that in the circumstance of a cancelation, the information returned could be minimized (e.g. Filing Identifier, Filing Status Code, Date/Time). It does not seem that ecf:ReviewLeadDocument, ecf:ReviewedConnectedDocument, or even nc:Case would be necessary. Note that currently, ecf:ReviewedLeadDocument and nc:Case are currently mandatory.

Additionally, there does not seem to be any worthwhile reason to return [FIPS 180-2] SHA 256 document hash values for documents, as required in section 6.1.10, when the submission has been cancelled.

If the filing submission included a payment, but the submission is canceled, what should be done?

Is the impact of cancellation on payments and fees also beyond the scope of this specification as is the case with authentication of requests and service?

I provided a new example called docketcallback-cancelled.xml. In my opinion, the impact on payments/fees is beyond the scope of the specification.

1. **NotifyFilingReviewComplete Invocation**

Section 6.1.10 NotifyFilingReviewComplete states: ‘If the clerk cancels or rejects a filing or a Filing Review MDE receives a NotifyDocketingComplete operation, the Filing Review MDE MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE that invoked the ReviewFiling operation as a callback message to indicate whether the filing was accepted and docketed by the clerk and court record system.”

In Arizona, we have a rule/practice that documents for rejected submissions must be retained by the system of record (CRMDE) in the event that these documents need to be referenced in the future for any action, cause, or reason (subject to record retention schedules). These documents are not ‘docketed’.

To accomplish this, we may be playing fast and loose with the requirement that “the Filing Review MDE MUST invoke the NotifyFilingReviewComplete operation”. It is our practice to send an RFR to the CRMDE even upon clerk review rejection. The CRMDE will archive the rejected documents per business requirements, then produce an NDC which is returned to the FRMDE NotifyDocketingComplete operation, thereby invoking the NotifyFilingReviewComplete operation in the FAMDE.

Now let me also point out that in section 6.1.9 NotifyDocketingComplete, it states (highlighting added): “the Court Record MDE MUST invoke the NotifyDocketingComplete operation on the Filing Review MDE that invoked a RecordFiling operation as a callback message to indicate whether the filing was accepted or rejected by the court record system.” I do recognize that in this context it is a CRMDE rejection (which may not be the same as an FRMDE/Clerk Review rejection).

Understandably, not all courts will have the same business practice regarding individually rejected documents or all documents for fully rejected submissions, or courts with similar practices may archive the rejected documents from within the FRMDE.

So I am not sure what to suggest. We would like to have our cake and eat it too, in the sense that we need to maintain this business practice, but also want to be ECF conformant.

Perhaps it is enough to revise the specification in 6.1.10 to say:

If the clerk cancels or rejects a filing or a Filing Review MDE receives a NotifyDocketingComplete operation, the Filing Review MDE MUST cause the invocation of the NotifyFilingReviewComplete operation on the Filing Assembly MDE that invoked the ReviewFiling operation as a callback message to indicate whether the filing was accepted and docketed by the clerk and court record system.

Another alternative wording is:

If the clerk cancels a filing or a Filing Review MDE receives a NotifyDocketingComplete operation, the Filing Review MDE MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE that invoked the ReviewFiling operation as a callback message to indicate whether the filing was accepted and docketed by the clerk and court record system. If the clerk rejects a filing, then the Filing Review MDE MUST either invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE, or invoke the RecordDocketing operation on the Court Record MDE.

Updated 6.1.10 with your first suggestion.

1. **CaseOfficialRoleCode**

CaseOfficialRoleCode.gc is still provided. Where is this used?

It was left over from previous drafts and has been removed.

1. **CaseJudge Represented Party**

Why is ecf:CaseRepresentedParty available in j:CaseJudge through CaseOfficialAugmentation?

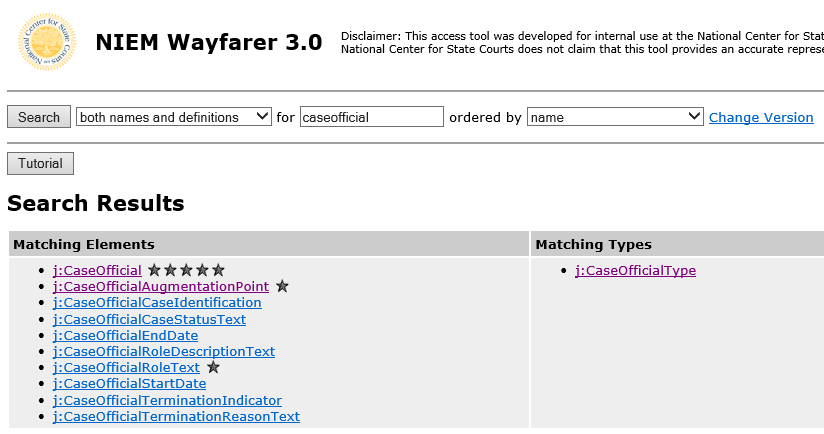
How would this be used?

It is included in ecf:CaseOfficialAugmentation which is inherited by j:JudicialOfficial.

1. **CaseOfficial**

In section 5.2.1 Court-Specific Augmentations, page 29, in the augmentation point table, the fourth row, in the first column (ECF augmentable element), shows ‘cyfs:CaseOfficial’. Should this be ‘j:CaseOfficial’ instead?

NIEM Wayfarer 3.0 only lists j:CaseOfficial and does not include cyfs:CaseOfficial:



Corrected the table.

1. **AppellateCaseParty**

EntityItem needs to be added to AppellateCaseParty.

ecf:EntityItem is already allowed under AppellateCaseParty.

1. **Cancel Filing Error**

In ‘ECF5 Spec Feedback and Consideration – 10’, item 1, the following questions were asked:

I presume that Court Policy should declare whether or not cancellation is supported. If not supported, and a cancellation request is received by the FRMDE anyway, what should be done? Return an error?

Your answer was: Yes.

ErrorCodeText.gc was discussed in the May 9, 2017 TC conference call. The minutes reflect:

The consensus is to have a few general error types, with the implementers to supplement with their own. Jim Cabral to factor discussion into another pass.

The current ErrorCodeText.gc contains:

1. No Error
2. Invalid per schema or code list
3. Invalid per ECF specification
4. Invalid per service interaction profile.

Perhaps we should also have a standard error code for ‘Invalid message per policy.’

At the risk of being nit-picky, the first letter ‘i’ in the word ‘invalid’ in the definition for Error Code 3 should be capitalized for consistency (as shown above)

Fixed the capitalization and added “4 Invalid per policy”

1. **RserveCourtDateMessage Example InValid**

The ReserveCourtDateMessage example (reservedate.xml) is not valid.

XMLSpy reports the following error:

Unable to load a schema with target namespace ‘https:\\docs.oasis-open.org/legalxml-courtfiling/ns/v5.0/reservedate’ from ‘c:\Users\JamesCabral\OneDrive\xml\ecf5\reservedate.xsd’

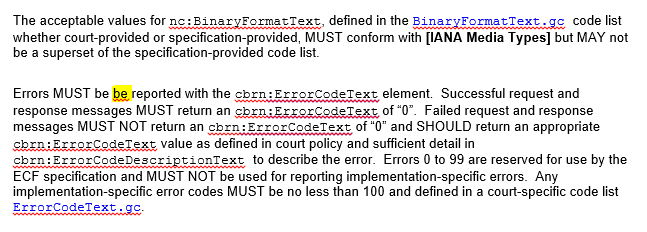
To correct, modify the xsi:schemaLocation in the root element, changing it to:

xsi:schemaLocation="https://docs.oasis-open.org/legalxml-courtfiling/ns/v5.0/reservedate ../schema/reservedate.xsd"

Replaced the hard coded path with a relative path as suggested.

1. **Typo**

In section 5.2.2 Court Specific Code Lists, the word ‘be’ is repeated:



Fixed the typo.

1. **xxx**