**ECF5 Civil Complaint Use Case Examples Review Consolidated**

A series of meetings were held by Webex to review the Civil Complaint Use Case examples. This meeting series commenced on July 25, 2018 and culminated on Sept. 7, 2018. Specific issues or concerns were identified in advance (see ‘LegalXML ECF 5 Civil Complaint Examples.docx) and participants were also encouraged to raise other issues. Individual meeting notes documents were created following each meeting. This document consolidates the notes from all meetings into a single document.

See civil-complaint.zip for wd34 for example XML files reviewed.

Items reviewed:

1. Civil-complaint-000-FilingMessage-02.xml

Discussed possible confusion by the inclusion of this example. Clarified that this example is not in any exchange context (like that of all other examples included in the example folder). Generally agreed that examples provided with the specification should all be in an exchange context (i.e. ’wrapped’ using wrapper.xsd). Identified that volunteers will be needed to rework existing examples contained in the ‘examples’ folder.

**Recommendation:** Remove Civil-complaint-000-FilingMessage-02.xml from the use case collection.

**Status:**

The Civil-Complaint-000-FilingMessage-02.xml example has been removed from the use case collection.

1. Use of elements nc:BinaryFormatText and cbrn:MultimediaDataMIMEKindText.

What is the difference between nc:BinaryFormatText and cbrn:MultimediaDataMIMEKindText?

Which should be used, or should both be used?

cbrn:MultimediaDataMIMEKindText is MIME type (<http://www.iana.org/assignments/media-types/index.html>). Stems from ECF 4 nc:BinaryCategoryText

nc:BinaryFormatText has ECF provided code list (i.e. BinaryFormatText.gc) which is a subset of all MIME types. Stems from ECF 4 nc:BinaryFormatStandardName.

**Recommendation:** Consider removing cbrn:MultimediaDataMIMEKindText.

**Recommendation:** No change to examples which only include nc:BinaryFormatText containing MIME type.

Note: section 5.2.2 includes:

“The acceptable values for nc:BinaryFormatText, defined in the BinaryFormatText.gc code list whether court-provided or specification-provided, MUST conform with [IANA Media Types] but MAY not be a superset of the specification-provided code list.”

1. Civil-complaint-001-FilingMessage-02.xml, line 47:

Todd found this comment line confusing. The distinction between message ID and filing ID was discussed along with which MDE issues which in the ReviewFilingRequest context.

**Recommendation**: Agreed to try to rewrite comment to be more understandable.

 **Status:**

Relocated line 47 which appeared just before the messageID to line 54 which is just after messsageID. Added additional commentary providing additional clarity on the [ECF] filing identifier (lines 55 to 58).

1. DocumentInformationCutoffDate in Civil-complaint-001-FilingMessage-02.xml (lines 84 to 90)

Discussed purpose and need for nc:DocumentInformationCutoffDate. Agreed that in the ReviewFilingRequest, this element is not easily distinguished from nc:DocumentPostDate and any distinction is not a practical distinction.

Possible ECF 4 legacy element.

**Recommendation**: As best practice, recommend not using nc:DocumentInformationCutOffDate in ReviewFilingRequest. Modify example to include comments advising not to use (e.g. deprecated in ReviewFilingRequest).

**Status**:

Added additional commentary comparing and contrasting nc:DocumentInformationCutOffDate with nc:DocumentPostDate and specifying the nc:DocumentCutOffDate is a legacy element and has been deprecated (lines 92 to 96). Commented out nc:DocumentCutOffDate (lines 97 to 101).

Note: nc:DocumentCutOffDate appears in the following examples from within the examples folder: appellate.xml, AttyDiscipline-RvFR.xml, civil.xml, citation.xml, civil-complaint.xml (this example should be deleted since it is superseded with the civil-complaint use case set), criminal.xml, domestic.xml, feesrequest.xml, juvenile.xml, and serveprocess.xml).

1. Separation of message metadata from query parameters in Get-Request messages (e.g. GetFilingSttausRequest, GetCaseRequest, GetFilingListRequest, GetDocumentRequest).

Discussed the value in organizing the Get-Requests to separate message metadata from query paraments/criteria. Agreed that this would be beneficial.

Discussed how/degree of separation; e.g. full elements such as nc:Case as parameters, vs more specific elements, e.g. j:CaseNumberText, or some middle ground.

In the end agreed to just look to reorganize messages for separation and not necessarily look to expand parameters or change type of parameters.

**Recommendation**: Gary to provide Jim Cabral with specific considerations from the examples.

**Status**:

Specific element considerations provided. Most of these were incorporated into query parameter container elements introduced in WD35; j:CaseCourt was not included in any of the query criteria container elements.

1. Filer vs Submitter

Agreed that in the Get-Request messages, nc:DocumentSubmitter should be used and not ecf:DocumentFiler.

**Recommendation:** No changes necessary.

1. Civil-Complaint-004-GetFilingStatusResponse-02.xml, line 68 ecf:MatchingFiling

It was suggested that the message structure for GetFilingStatusResponse is really that of a GetDocumentStatusResponse and is not properly structured for filing statuses.

A single filing may contain multiple documents.

Two possible corrections were considered:

* Restructure GetFilingStatusResponse to permit the return of the status for each and every document within the filing, and possibly the overall status of the filing.

Or,

* Rename the message to GetDocumentStatusResponse, retaining its ability to only report status for a single document, and also rename its corresponding request message to GetDocumentStatusRequest.

Also discussed, but with no resolution or recommendation at this time, is what and how responses should be returned when the request contains query parameters that result in multiple filings (or perhaps documents), such as when querying by j:CaseNumberText. When there are multiple response messages for a request, does this result in a chain of response messages?

**Recommendation**: Redesign MatchingFiling

**Status**:

This subject was discussed at the Aug. ECF TC conference call (8-14-2018). Upon review of the current filingstatusresponse.xml example, Jim Cabral suggested that it may be wrong (e.g. is clearly returning information about a single document, and not a filing). The question regarding what elements need to be returned for a filing was not fully answered. Some suggested a subset of all filing elements. Should document information be returned as well for the documents within the filing? There was general agreement to include document information in a GetFilingListResponse. Again, some suggested that it could be a subset of all document information. Todd thinks that payment information should also be included.

Since at this time (8-24-2018) the meeting minutes for the 8-14-2018 TC conference all have not yet been released, and since the next anticipated WD has not yet been distributed, the details regarding what may or may not have been agreed to and what may or may not have been understood are not determinable.

1. Civil-Complaint-005-RecordDocketingRequest-02.xml

Considered best practices for addressing ecf:ReviewedDocument, per 6.4.3. Reviewed the requirements expressed in 6.4.3.

Introduced four alternatives (A – D) to be reviewed more completely.

**Recommendation**: A set of principles should be developed to guide best practice consideration.

1. Reviewed representations of CaseAugmentation elements in docket:CorrectedCase (i.e. ecf:CaseAugmentation, j:CaseAugmentation, and civil:CaseAugmentation) per requirements of 6.4.3.

Suggested and discussed principles for representing original case information from within docket:CorrectedCase. Principles should make the methods more of a science and less of an art.

**Recommendation**: Revise schema so that the xsi:nil attribute is provided for all CaseAugmentation elements, including:

* ecf:CaseAugmentation
* j:CaseAugmentation
* appellate:CaseAugmentation
* bankruptcy:CaseAugmentation
* citation:CaseAugmentation
* civil:CaseAugmentation
* criminal:CaseAugmentation
* domestic:CaseAugmentation
* juvenile:CaseAugmentation

**Recommendation:** Principles for the inclusion of xsi:nil in other container elements should be considered.

**Status**:

Also see Feedback 24, #2 which addresses the xsi:nil consideration for ecf:DocumentRendition. Made recommendation to provide xsi:nil on all CaseAugmentation at 8-14-2018 ECF TC Conference call. J. Cabral said that would be ‘easy’ which I understood as an agreement to do so.

1. Civil-Complaint-005A-RecordDocketingRequest-02.xml

Began review of Alternative A (the first of 4 alternatives to consider).

Addressed the value used for structures:id for element ecf:ReviewedDocument (line 238). Since the exhibit document was generally referenced as ‘Document2’, having structures:id of ‘ReviewedDocument3’ does not facilitate understanding.

**Recommendation:** Modify example to use a more meaningful value for structures:id.

**Status**:

This has been corrected so that the structures:id value is now ‘ReviewedDocument2”.

1. Harmonize values used for structures:id.

This suggestion was provided when reviewing the ecf:DocumentAugmentation element in Alternative A on line 347 which contained a structures:id=”SubmittedComplaintAugmentation”.

**Recommendation**: labels used in examples should be clear and facilitate reader understanding.

1. The name and schema definitions for elements ecf:LeadDocumentReview and ecf:ConnectedDocumentReview were discussed. It was suggested that the element names are confusing and/or are not suggestive of their purpose and that the schema definitions are not clear.

**Recommendation**: Jim Price to further consider and provide specific recommendations.

1. Discussed the possible value of broader application of container elements, such as having one or more container elements for all current elements within nc:DocumentType which come before nc:DocumentAugmentationPoint.

The benefit would be that concise representation using structures:ref would be more broadly available and would not be limited to existing container elements such as ecf:CaseAugmentation, etc.

 **Recommendation**: none at this time.

1. Civil-Complaint-005A-RecordDocketingRequest-02.xml

Continued review and consideration of best practices for addressing ecf:ReviewedDocument, per 6.4.3.

Focus was on 4 alternatives (A – D) provided within the example, for how to best handle the addition of a stamped complaint document rendition. Alternatives A & D use separate ecf:DocumentAugmentation elements for each rendition, whereas alternatives B & C employ a single ecf:DocumentAugmentation, but include separate ecf:DocumentRendition elements within the single ecf:DocumentAugmentation.

**Recommendation**: A slightly revised version of Alternative A was selected by the task team as the best practice preference recommendation.

Alternative A was selected because it employed concise representation for the original rendition’s ecf:DocumentAugmentation (which was preferred over alternative D which employed verbose copy), and because the use of multiple separate ecf:DocumentAugmentation elements provides each rendition with its own separate set of additional descriptive elements within the augmentation (e.g. ecf:DocumentFiler, ecf:RedactionRequiredIndicator, ecf:RegisterActionDescriptionCode, ecf:SpecialHandlinginstructions). Alternative A permits each rendition to have individual values for these supplemental elements; e.g. the RegisterActionDescriptionCode for the original rendition can have a different value than that of the stamped rendition.

**Recommendation**: It was also agreed that these document rendition supplemental augmentation elements (e.g. ecf:DocumentFiler, ecf:RedactionRequiredIndicator, ecf:RegisterActionDescriptionCode, etc.) must be repeated in the additional rendition’s ecf:DocumentAugmentation to be applicable to the additional rendition. Absence of the element and/or element value must be understood as the element/value is not applicable and not that the element/value is unchanged from that of the original rendition.

The use case example will be modified to provide this best practice recommendation. Explanatory comments will also be included. These comments should address three scenarios for ecf:DocumentAugmentation child elements other than ecf:DocumentRendition, such as ecf:DocumentFiler, ecf:RegisterActionDescriptionCode, etc. These 3 scenarios are (as expressed for ecf:DocumentFiler):

* The stamped rendition has the same filer as the original rendition.
* The stamped rendition has a different filer than the original.
* The stamped rendition has no filer.

**Status:**

The Civil-Complaint-005-RecordDocketingRequest-02.xml has been modified to only include the Alternative A example with explanatory comments.

1. Other questions addressed:
* Is a redacted version of a document, such as a complaint, a different document or a different rendition of a document?

**Status**: question still pending.

* Is it common for courts to consider that a stamped version of a document has a different document type code (e.g. RegisterActionDescriptionCode) than the unstamped version? How about an issue stamped summons vs. an unstamped summons?

**Status**: questions still pending.

* Should different renditions of the same document be permitted to have different document titles or is a single document title sufficient for all renditions of the same document?

**Status**: question still pending.

1. Civil-Complaint-005-RecordDocketingRequest-02.xml

Considered cardinality for ecf:ReviewedDocument in ecf:LeadDocumentReview and ecf:ConnectedDocumentReview.

When there have been no changes to document information during clerk review, is it necessary to provide ecf:ReviewedDocument to reference the document within filing:FilingMessage, as in:

<ecf:ReviewedDocument structures:ref="Document2" xsi:nil="true"/>

Or, if the element were optional, could the absence of ecf:ReviewedDocument convey the same understanding?

**Recommendation:** After consideration, the task team recommends keeping ecf:ReviewedDocument as a mandatory element.

1. Civil-Complaint-007-NotifyDocketingCompleteRequest-02.xml

Considered ecf:DocumentFiler (line 101). Should the document filer information from the filing:FilingMessage within ReviewFilingRequest, and provided to the CRMDE in the RecordDocketingRequest, be returned in ecf:DocumentFiler?

**Recommendation:** Although not required by specification or by cardinality, returning this information is recommended. The task team further recommends explicitly explaining the recommended approach within applicable examples.

In general, information that is to be provided in the next message downstream (e.g. NotifyFilingReviewComplete), that is not introduced in the MDE, is obtained from the prior message (e.g. NotifyDocketingCompleteRequest). Although this principle does not preclude MDEs from retaining information for subsequent reference during or after the e-filing transaction, it does foster and facilitate a stateless operation.

1. Various use cases for the addition, modification, or removal of document and document renditions were considered. Scenarios are:
2. Newly added documents – documents not originally provided by the filer/submitter may be added into the transaction either in Clerk Review (e.g. a receipt) or in docketing (e.g. a notice).
3. Replacement – e.g. circumstances where one document or document rendition is substituted by a different document or rendition.
4. Removal of a document – do this by rejection in clerk review.

**Recommendation:** For newly added documents:

In both RecordDocketingRequest and NotifyDocketingCompleteRequest, within elements ecf:LeadDocumentReview or ecf:ConnectedDocumentReview:

 ecf:Document is absent.

 ecf:DocumentReviewDisposition/ecf:DocumentReviewStatus/nc:StatusText is ‘accepted’.

 ecf:ReviewedDocument contains the complete document elaboration.

1. Purpose and need for ecf:Document in NotifyDocketingCompleteRequest

For a NotifyDocketingCompleteRequest, it was noted that ecf:Document within ecf:LeadDocumentReview and ecf:ConnectedDocumentReview should **NEVER BE USED !!!!**

Unfortunately, this element is present in NotifyDocketingCompleteRequest and therefore may be used (and still be valid per schema), and even worse, may be misused (and still be valid per schema).

**Recommendation**: either remove ecf:Document from ecf:LeadDocumentReview and ecf:ConnectedDocumentReview in NotifyDocketingCompleteRequest and NotifyFilingReviewCompleteRequest (preferred), or in the alternative, add language and clarification in the specification document (and element description) that makes it clear that ecf:Document MUST NOT be used in NotifyDocketingCompleteRequest or NotifyFilingReviewCompleteRequest.

1. How can newly added documents be identified as newly added (e.g. court added) documents?

Although newly added documents could be discerned in the RecordDocketingRequest by the absence of ecf:Document, this same consideration cannot be applied to the NotifyDocketingCompleteRequest since the ecf:Document element must never be used in this message. As such, a new means, such as a Court-Added-Document-Indicator element is needed.

**Recommendation**: Add a new element to the NotifyDocketingCompleteRequest and NotifyFilingReviewCompleteRequest that can be used to signal that a document (either lead or connected) has been added by the court (either in clerk review or during docketing). Specifying the MDE within which the newly added document was added is not necessary.

1. What to do if a lead document is revised to be a connected document in clerk review, or vice versa?

A scenario was raised in which a lead document ‘A’ has two connected documents, ‘B’ and ‘C’. In clerk review, the reviewing clerk determines that document ‘B’ should in fact be the lead document, and document ‘A’ should be a connected document to document ‘B’.

How would this be handled in the RecordDocketingRequest and NotifyDocketingCompleteRequest?

This will be considered for a future meeting.

**Status:** Pending, not yet considered.

1. FilingStatus usage within NotifyDocketingCompleteRequest was debated.

A submission (a.k.a. a filing) will have an overall filing status at the conclusion of clerk review (e.g. from FilingStatusCode.gc; either ‘accepted’, ‘cancelled’, ‘partially-accepted’, ‘received’, ‘rejected’, or ‘issued’). This clerk review filing status should be carried in the NotifyDocketingCompleteRequest (e.g. in ecf:FilingStatus).

However, the results of docketing can be important also. In some rarer circumstances, the docketing operation may not be successful. There are too many possibilities for an unsuccessful result to address each individually, save to say that some errant results may not be correctable by the court.

Typically, the CRMDE is the system of record for the court. When docketing cannot/does not occur, then the filing has not yet been made a part of the official court record, even if the submission/documents were evaluated in clerk review as being acceptable. If, and once the court determines that docketing cannot and will not occur, then the docketing operation is concluded and the NotifyDocketingCompleteRequest, which should identify this unsatisfactory result, must be sent.

**Recommendation**: Provide two separate elements for the status of a filing for both clerk review status and docketing (or transaction) status in NotifyDocketingCompleteRequest and NotifyFilingReviewCompleteRequest. The terminology used for status related to docketing (or transaction) should be appropriate, such as ‘successful’, or ‘error’, and not ‘accepted’ or ‘rejected’.

1. ‘Outbound’ message Document Status

When reviewing the use of the ecf:FilingStatusCode element within the NotifyDocketingCompleteRequest (above), it was also suggested that this same two-element status designation capability (e.g. clerk review status and docketing status) should be considered for the reviewed documents.

**Recommendation**: Provide the ability to designate both a clerk review status as well as a docketing status for each reviewed document (i.e. ecf:LeadDocumentReview and ecf:ConnectedDocumentReview) within the NotifyDocketingCompleteRequest and the NotifyFilingReviewCompleteRequest.

1. Civil-Complaint-007-NotifyDocketingCompleteRequest-02.xml

Considered ecf:RegisterActionDesriptionCode use and purpose (lines 242, 365, 453).

This element is defined as “the docket code used by the court for the type of document submitted. Allowable values set forth in Court Policy.”

Discussed whether the purpose for this element changes from its use/purpose in the RecordFilingRequest and RecordDocketingRequets (e.g. ‘inbound’ messages) when used in the NotifyDocketingCompleteRequest and NotifyFilingCompleteRequest (e.g. ’outbound’ messages). In the ‘inbound’ messages, ecf:RegisterActionDescriptionCode identifies the type of document. In the ‘outbound’ messages, it identifies a ‘docket code’ – which may or may not be the same as a document type.

It was agreed that expressing the ‘type of document’ was essential for both inbound as well as outbound messages. For those implementations or circumstances where the ‘docket code’ applied to a document is not identical to the document’s document type code, there should be the ability to provide both a document type code as well as a docket code.

**Recommendation**: The preferred approach is to provide a document type code element with an element name and definition that is well suited to its purpose (e.g. ecf:DocumentTypeCode). This document type code element should be available for each document, on both ‘inbound’ as well as ‘outbound’ messages. Additionally, provide an element to use for docket code, with name and definition appropriate to its usage. At a minimum, this docket code should be available on outbound messages for each document.

1. DocketingResult

Some documents do not get docketed; e.g. a Summons is issued, but not docketed in many courts. It should be clear in ‘outbound’ messages, which documents were docketed and which were not.

**Recommendation**: Consider adding an element to indicate if docketing successfully occurred for each document.

1. DocumentSubmitter usage within NotifyDocketingCompleteRequest was considered.

It was noted that there is but one nc:DocumentSubmitter element for the NotifyFilingReviewComplete message. In the unusual circumstance where the clerk reviewer person (e.g. the nc:DocumentSubmitter for the RecordDocketingRequest) is not the same as the docketing person, then who should be recorded as the nc:DocumentSubmitter, or should multiple elements be provided?

**Recommendation**: Continue to only provide a single nc:DocumentSubmitter element. Courts will decide whether to use it, and if used, who (e.g. the reviewer or the docketer) should be provided.

1. Civil-complaint-011-GetCaseRequest-02.xml

Reviewed the GetCaseRequest. Discussed use of various caserequest:CaseQueryCriteria elements. Considered cardinality of ecf:CourtEventTypeCode which is mandatory. Compared this to caserequest:DocketEntryTypeCode which is optional. Both elements are provided for the purpose of “filter criterion…”.

**Recommendation**: The element ecf:CourtEventTypeCode should be optional in the GetCaseRequest.

1. Civil-complaint-011-GetCaseRequest-02.xml

Reviewed caserequest:DocketEntryTypeCodeFilter (“Filter criterion indicating that only docket entries of a specified type are being requested.”) How the element should be used when there are multiple Docket Codes requested was discussed. At present, this element has maxOccurs of 1.

It was agreed that a comma delimited (or any other delimiter) list should **not** be used. Also, when multiple codes are provided, there are other complexities, such as the use of ‘AND’ and ‘OR’, parenthetic groupings, etc. There was no interest in defining a filter language.

**Recommendation**: Modify the maxOccurs to unbounded for caserequest:DocketEntryTypeCodeFilter. Add clarification that when multiple docket codes are provided, these should be interpreted as OR conditions.

1. The element ecf:CourtEventTypeCode (“Filter criterion indicating that only calendar entries of a specified type are being requested.”) is similar in use and purpose to caserequest:DocketEntryTypeCodeFilter.

Whether the same treatment should be applied was considered.

**Recommendation**: Modify the maxOccurs to unbounded for ecf:CourtEventTypeCode. Add clarification that when multiple docket codes are provided, these should be interpreted as OR conditions.

1. Civil-complaint-012-GetCaseResponse-02.xml

The best practice for the use of ecf:ConnectedDocument when the response contains both filing lead and filing connected documents was evaluated. Two options were presented for consideration.

Option A – This option is presented in the example. Option A uses an ecf:ConnectedDocument element for each document, then associates the child (e.g. filing connected document) to its parent (e.g. filing lead document) using nc:DocumentAssociation, with nc:PrimaryDocument identifying the parent document and ecf:DocumentRelatedCode of ‘parent’.

Option B – In this option ecf:ConnectedDocument would only be used for the child document. The parent document would be elaborated within nc:DocumentAssociation, using the nc:PrimaryDocument element.

**Recommendation**: Option A is the preferred approach. The examples will illustrate Option A with clarifying/explanatory comments. Option B will be discouraged, but not prohibited.

1. Civil-complaint-013-GetFilingListRequest-02.xml

Considered the use of the (ECF) filing identifier as a query parameter when specifying a single filing or a set of specific filings. It was agreed that it would be reasonable to request filings using their (ECF) filing identifier. When doing so, the filing identifier would be a query parameter.

**Recommendation**: Add nc:DocumentIdentification into filinglistrequest:FilingListQueryCriteria.

1. Use of nc:DocumentSubmitter in filinglistrequest:FilingListQueryCriteria

The purpose for nc:DocumentSubmitter in filinglistrequest:FilingListQueryCriteria was discussed. It was agreed that one may want to get a list of filings submitted by an individual person or entity, perhaps restricted within a date range (using nc:DateRange). It was also agreed that one may want to make this request within the context of a single case, or indifferent to case.

**Recommendation**: Change the cardinality for j:CaseNumberText to make this element optional.

1. j:CaseCourt as a query parameter

Whether j:CaseCourt is a query parameter and/or message metadata was kicked around. This element is contained in the base set and can have a dual understanding as both a query parameter and message metadata.

**Recommendation**: Leave j:CaseCourt where it is.

1. Civil-complaint-014-GetFilingListResponse-02.xml

The issue with ecf:MatchingFiling had been discussed at the 8-14-2018 ECF TC Conference call where it had been agreed that it needed to be restructured, along the lines of an NotifyFilingReviewComplete (NFRC), but perhaps only providing a subset of the elements.

The major elements within the NFRC were reviewed; this include: ecf:FilingCompletionDate, ecf:FilingStatus, ecf:LeadDocumentReview, ecf:ConnectedDocumentReview, and nc:Case.

Todd suggested that there should not be any requirement to provide document binaries (e.g. therefore nc:Attachment may be absent or optional), however providing document binaries in the response should not be prohibited.

Also see #7 above.

**Recommendation**: Jim Cabral will create examples with NFRC-like MatchingFilings for review.

**Additional note**: My notes from the 8-14-2018 conference call also included discussion on whether FilingStatus should also be included for documents and not just for the filing. Those who offered an opinion agreed, except for Tyler (Phillip) who expressed ‘no opinion’, since Tyler deals with the filing as a whole and not individual documents. There was also conversation as to whether MatchingFiling should also include payment information. Todd said it should.

1. GetDocumentRequest

Reviewed the use of nc:DocumentFileControlID as a query parameter used to specify a specific document rendition (e.g. the file stamped complaint rendition) in a GetDocumentRequest. Verified that this element is contained within the NFRC.

Reviewed specification sections 6.2.6 and 6.1.15.

**Recommendation**: The element nc:DocumentFileControlID is not only the recommended element to use to specify the requested document, but is also the mandatory element per 6.1.15.

1. Use of CaseNumberText in GetDocumentRequest

In documentrequest:GetDocumentRequestMessage, j:CaseNumberText is a required element. However, nc:DocumentFileControlID provides a unique reference to a document within a Court Record MDE (see 6.2.6). As such, there is ***no need*** to specify a case number to uniquely specify a document.

**Recommendation**: Make j:CaseNumberText optional in documentrequest:GetDocumentRequestMessage.

1. Query Criteria element in GetDocumentRequestMessage

Since ncDocumentFileControlID is a mandatory query criteria element for a document request, and since this element uniquely identifies a document, no other document sub-elements are needed.

**Recommendation**: Further modify the query criteria element within GetDocumentRequestMessage, removing nc:Document, and replacing with nc:DocumentFileControlID (mandatory).

1. Section 6.1.15

The language in 6.1.15 is unclear and confusing.

**Recommendation**: Replace the current language in 6.1.15 with the language provided below:

6.1.15 **GetDocument**

The GetDocument operation MAY be invoked by an MDE. If this operation is enabled by court policy, then when a Filing Assembly MDE invokes the GetDocument query operation on the Court Record MDE to retrieve a particular document, the query MUST provide the document file control identifier (nc:DocumentFileControlID)and the Court Identifier (j:CaseCourt). The case number (j:CaseNumberText) and/or case tracking ID (ecf:CaseTrackingID) MAY be provided as well. The Court Record MDE will respond synchronously with the requested document or instructions on how to access it or a status message explaining why the document cannot be provided.

1. GetDocumentResponse

When using query criteria such as nc:DocumentFileControlID in a GetDocumentRequest, at most a single document will be returned in a GetDocumentResponse. However, if less specific criteria is permitted in the request (e.g. nc:DocumentSubmitter, or nc:DocumentStatus, or nc:DocumentTitleText, etc.), then multiple documents may need to be returned.

GetDocumentResponse only supports the return of a single document.

Agreed that mandatory use of nc:DocumentFileControlID in the GetDocumentRequest precludes any possibility of multiple documents as a result.

**Recommendation**: Require the mandatory use of nc:DocumentFileControlID in GetDocumentRequest (as expressed in the rewrite of 6.1.15 above).

1. GetDocumentResponse

Reviewed examples Civil-complaint-017-GetDocumentResponse-02.xml and Civil-complaint-018-GetDocumentResponse-02.xml, which provide (e.g. return) the complaint and connected exhibit documents, respectively.

Reviewed the use of nc:DocumentAssociation in Civil-complaint-018-GetDocumentResponse-02.xml as a way to identify that the returned document is a child document of some other parent document. Agreed that the value for nc:DocumentFileControlID for the parent document (e.g. from NFRC) should be used in nc:DocumentIdentification as the identification value for specifying the associated parent document.

Agreed that labeling this document identification using ‘DocumentRenditionFileControlID’ for nc:IdentificationCategoryDescriptionText was appropriate. Agreed that these rules should be included in the specification document.

**Recommendation:** Gary to draft specification language.

1. Considered the potential for allowing both a lead document and all its connected documents to be returned in a single GetDocumentResponse. Concluded that a GetDocumentResponse should only return a single document.

**Recommendation**: No change.

1. Noted that nc:PrimaryDocument in nc:DocumentAssociation could be used to return an additional document (e.g. parent document) in a GetDocumentResponse. Agreed that this should not be done; agreed that it was appropriate to include document elements such as nc:DocumentDescriptionText or nc:DcoumentTitleText, but document binaries (e.g. nc:Attachment) should not be included.

**Recommendation**: Verify that the specification currently prohibits more than one document in the GetDocumentResponse. If not currently prohibited by specification, then include specification language which prohibits the return of multiple documents.

1. Civil-complaint-019-ReviewFilingRequest-02.xml

Reviewed the use of j:CaseCourtEvent to include additional details for evidence of service, esp. date served, and defendant served.

Todd expressed that this was a valid use case, and some courts also want cost of service to be included.

Todd felt that the use of j:CaseCourtEvent (“a court occurrence related to a case.”) was inappropriate, in that j:CaseCourtEvent should only be used for events that occur within the court.

Todd suggested that a more appropriate means of providing additional information for a filing would be through the use of an augmentation (e.g. a document augmentation).

It was agreed that a library of document augmentations for each type of document (e.g. proof of service, complaint, petition for review, etc.) would result in a large library of augmentations. An alternative would be to provide a generic abstracted augmentation data structure.

There appears to be three options for this use case example:

* Continue to employ j:CaseCourtEvent
* Remove j:CaseCourtEvent and only illustrate filing the evidence of service document and not include the additional data.
* Modify the example to use a non-ecf provided document augmentation for the supplemental data. Jim Cabral says that we cannot provide an example that includes non-ecf provided schema.

**Recommendation:** Gary will further consider how best to represent this example.

1. Whether the cardinality for ecf:CourtEventActor within j:CaseCourtEvent/ecf:CourtEventAugmentation should be revised from at most 1 to unbounded was considered. Although, in the circumstance of the service of a summons, only one served actor is appropriate, in general events can include multiple actors.

**Recommendation**: Change cardinality to permit multiple ecf:CourtEvetActor elements.

1. Whether ecf:ServiceStatusCode should be used in a ReviewFilingResponse to a ReviewFilingRequest for a service related filing (e.g. evidence of service) was considered. It was agreed that this was appropriate.
2. Message ID

Whether a message identifier must be provided for cbrn:MessageStatus and other synchronous message was considered.

Specification section 6.2.5 provides: “All synchronous and asynchronous response messages MUST, in addition to any message identifiers for the response message itself, include the message identifier for the original message it is responding to.”

**Recommendation**: It was agreed that the use case example message status messages should include a message identifier.