**ECF5 Spec Feedback and Considerations – 19**

This document contains additional questions and commentary resulting from a review at the Electronic Court Filing Version 5.0 Working Draft 24, unless otherwise noted.

1. **Participant structures:ref attribute usage**

Feedback document 15, item 2 continues the discussion of normative party to attorney representation mark-up. On page 8, the document provides:

When identifying the party represented by the attorney, there are two methods supported. One and only one of these two methods MUST be used. The two methods are:

1. The party is identified using the structures:ref attribute on the substituted entity type element. This is the preferred and recommended method.
2. The party entity details are elaborated within the substituted entity type element (e.g. EntityItem, EntityOrganization or EntityPerson).

In response, Jim Cabral provided:

The schema alone cannot express this constraint but I do not think we need this rule. Let’s discuss with the TC.

When discussed with the TC’s work team during the 9-5-2017 conference call a preference for making structures:ref the recommended approach seem to be prevailing. Jim Cabral seemed reluctant to make structures:ref the recommended method in the limited circumstance of ecf:CaseRepresentedParty and seemed to be hopeful that the rule could be broadly and generally applied. Jim Cabral provided a list of elements to consider whether or not structures:ref should be mandatory in each ‘role’ (as Jim Cabral refers to these elements). Jim Price and Gary Graham accepted responsibility for review the elements on the list and making a recommendation regard making the use of structures:ref mandatory.

The elements provided are:

* appellate:AppellateCaseParty
* biom:BiometricCapturer
* caselistrequest:CaseParticipant
* ecf:AffectedParty
* ecf:CaseParty
* ecf:CaseRepresentedParty
* ecf:DocumentFiler
* ecf:DocumentReviewer
* j:CaseOtherEntity
* j:CaseOfficial
* nc:BinaryCapturer
* nc:ContactEntity
* nc:DocumentSubmitter
* nc:Employer
* nc:ObligationEntity
* nc:ObligationRecipient
* serviceinformationresponse:ServiceRecipient

Below are the two different CaseRepresentedParty optioins identified in the feedback 15 document:

Option 1 – using structures:ref:

<j:CaseOfficial>

<nc:RoleOfPerson structures:id="Person3">

<nc:PersonName>

<nc:PersonFullName>Jane Doe, JD</nc:PersonFullName>

</nc:PersonName>

<ecf:PersonAugmentation>

<ecf:ParticipantID>

<nc:IdentificationID>10</nc:IdentificationID>

</ecf:ParticipantID>

</ecf:PersonAugmentation>

</nc:RoleOfPerson>

<j:JudicialOfficialBarMembership>

<j:JudicialOfficialBarIdentification>

<nc:IdentificationID>100</nc:IdentificationID>

</j:JudicialOfficialBarIdentification>

</j:JudicialOfficialBarMembership>

<ecf:CaseOfficialAugmentation>

<ecf:CaseRepresentedParty>

<nc:EntityPerson structures:ref="Person1" xsi:nil="true"/>

</ecf:CaseRepresentedParty>

</ecf:CaseOfficialAugmentation>

</j:CaseOfficial>

</j:CaseAugmentation>

<ecf:CaseAugmentation>

<ecf:CaseCategoryCode>Estate</ecf:CaseCategoryCode>

<ecf:CaseParty>

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

<nc:PersonName>

<nc:PersonGivenName>John</nc:PersonGivenName>

<nc:PersonMiddleName>W.</nc:PersonMiddleName>

<nc:PersonSurName>Doe</nc:PersonSurName>

</nc:PersonName>

Option 2 – in place elaboration:

<j:CaseOfficial>

<nc:RoleOfPerson structures:id="Person3">

<nc:PersonName>

<nc:PersonFullName>Jane Doe, JD</nc:PersonFullName>

</nc:PersonName>

<ecf:PersonAugmentation>

<ecf:ParticipantID>

<nc:IdentificationID>10</nc:IdentificationID>

</ecf:ParticipantID>

</ecf:PersonAugmentation>

</nc:RoleOfPerson>

<j:JudicialOfficialBarMembership>

<j:JudicialOfficialBarIdentification>

<nc:IdentificationID>100</nc:IdentificationID>

</j:JudicialOfficialBarIdentification>

</j:JudicialOfficialBarMembership>

<ecf:CaseOfficialAugmentation>

<ecf:CaseRepresentedParty>

<nc:EntityPerson structures:id="Person10">

<nc:PersonName>

<nc:PersonGivenName>John</nc:PersonGivenName>

<nc:PersonMiddleName>W.</nc:PersonMiddleName>

<nc:PersonSurName>Doe</nc:PersonSurName>

</nc:PersonName>

</nc:EntityPerson>

</ecf:CaseRepresentedParty>

</ecf:CaseOfficialAugmentation>

</j:CaseOfficial>

In both options, the person John W. Doe has been specified as a case party. In option 1 this is done using the ecf:CaseParty element. In option 2, by use of the ecf:CaseRepresenetedParty element.

First, it must be understood that any reference using structures:ref must point to a like-kind element with a matching structures:id. Also, since entity type elements (e.g. nc:EnitityPerson, nc:EntityOrganization, ecf:EntityItem) cannot exist alone and can only exist within a role element, it must be apparent that some ‘role’ elements must use structures:id and not stuctures:ref.

Also note that in ECF 4.01, attorney to party representation was indicated using the element ecf:CaseRepresentedPartyReference which is clearly a reference element.

After careful consideration, the only role element in the list provided (shown above) that makes sense to either be restricted to the use of structures:ref or to be recommended as to use structures:ref is ecf:CaseRepresentedParty. We further recommend that if the ecf:CaseRepresentedParty element is restricted to be used only for reference (e.g. as a reference element and not a role element), it should be renamed as ecf:CaseRepresentedPartyReference and should not provide the structures:id attribute.

|  |  |
| --- | --- |
| **Role Element** | **Restricted to structures:ref** |
| appellate:AppellateCaseParty | No |
| biom:BiometricCapturer | No |
| caselistrequest:CaseParticipant | No |
| ecf:AffectedParty | No |
| ecf:CaseParty | No |
| ecf:CaseRepresentedParty | Yes |
| ecf:DocumentFiler | No |
| ecf:DocumentReviewer | No |
| j:CaseOtherEntity | No |
| j:CaseOfficial | No |
| nc:BinaryCapturer | No |
| nc:ContactEntity | No |
| nc:DocumentSubmitter | No |
| nc:Employer | No |
| nc:ObligationEntity | No |
| nc:ObligationRecipient | No |
| serviceinformationresponse:ServiceRecipient | No |

1. **PaymentMessage**

In section 6.4.1 it states that documents for transactions such as the payment of a criminal fine are excluded. Why?

Does “MAY NOT” mean “MUST NOT” or does “MAY NOT” mean possibly not?

Why wouldn’t this just be a policy choice that courts could make individually or by implementation as best suits their needs and capabilities?

1. **NotifyFilingReviewComplete for Cancellation**

In ‘ECF5 Spec Consideration 11.docx’, item 5, ‘Cancel Filing Message’, the question as to what the appropriate message is when a clerk has agreed to cancel a filing where cancellation has been requested with a CnacelFilingMessage (cancel.xsd).

The response was:

As a minimum, the filer will receive the result in a NotifyFilingReviewComplete message. WD16 clarifies Section 6.1.10 NotifyFlingReviewComplete to include “if the clerk cancels” as a trigger. Let’s discuss with the TC whether we need a separate CancelFilingResponse message.

Note: there is not currently any example cancel response message (there is an example cancel request message, i.e. cancel.xml)

So what should an example reviewfilingcallback message look like for an accepted and approved cancellation request?

Here are some things to consider:

* It seems essential that the filing identifier be included (i.e. nc:DocumentIdentiifcation/nc:IdentificationID).
* It seems appropriate and necessary that ecf:FilingStatus/ecf:FilingStatusCode should be “cancelled’.
* Is it necessary or even useful to include reviewed lead and connected document information? Note that ecf:ReviewedLeadDocument is required.
* It may be useful to include ecf:DocumentAugmentation/ecf:AffectedParty. If so, who would identify the affected parties, the clerk or the cancellation requester?
* If affected parties are identified, or any other information which is contained within ecf:DocumentAugmentation, is it necessary to include an ecf:RegisterActionDescriptionCode?
* Is it necessary to include the document filer? And who would be the document filer for the cancellation response? Is this the clerk or the cancellation requester? Note that ecf:DocumentFiler is a mandatory element.
* Although identifying the clerk who approved the cancellation request should be optional (at the court’s discretion), if the clerk is identified, wouldn’t this be in nc:DocumentSubmitter?
* It seems appropriate that ecf:FilingCompletionDate must be absent.
* It seems right, at least as a sanity check, to include the court and case number (i.e. j:CaseAugmentation/j:CaseCourt and nc:CaseTrackingID).

In my view, reviewed lead and connected document information should not be required in a cancellation approval response message.

1. **FilingStatus in reviewfilingcallback**

The ecf:FilingStatus element in the reviewfilingcallback example is empty.

What is this trying to demonstrate?

1. **Clarification and Typo**

After considering the document reference lifecycle, it occurred to me that the statement, shown below, from section 6.2.5 Document identifiers, may benefit from some additional clarification.

ecf:ReviewedLeadDocument MUST reference filing:FilingLeadDocument and ecf:ReviewedConnectedDocument MUST reference filing:FilingConnectedDocument using nc:DocumentAssociation/nc:PrimaryDocument, and ecf:DocumentRelatedCode with a value or “reviewed”.

Even though it is only the RecordDocketingMessage that contains both filing document and reviewed document elements, it may help with reader understanding to add this additional clarification as in:

In the RecordDocketingMessage, ecf:ReviewedLeadDocument MUST reference filing:FilingLeadDocument and ecf:ReviewedConnectedDocument MUST reference filing:FilingConnectedDocument using nc:DocumentAssociation/nc:PrimaryDocument, and ecf:DocumentRelatedCode with a value of “reviewed”.

Even if this clarification is not agreed to, at least correct the typo, replacing ‘or’ with ‘of’ as in ‘with a value of “reviewed”.’

1. **Participants vs Parties**

In section 3.2.1, the term ‘parties’ should be revised to ‘participants’, i.e.:

This process describes the sequence of operations in a basic filing and service cycle from Filing Preparation to Docketing. This process involves the following parties:

Should be revised to:

This process describes the sequence of operations in a basic filing and service cycle from Filing Preparation to Docketing. This process involves the following participants:

Also, in the same section, the following should also be revised:

At any point during or after the ReviewFiling operationa party MAY access information through the following operations:

To be:

At any point during or after the ReviewFiling operation a participant MAY access information through the following operations:

And:

At any point during or after the ReviewFiling operation and before the RecordDocketing operation a party MAY request cancellation of the filing through the following operation:

Should be revised to:

At any point during or after the ReviewFiling operation and before the RecordDocketing operation a participant MAY request cancellation of the filing through the following operation:

And finally:

At any point after the NotifyFilingReviewComplete operation, if the case is accessible, a party MAY access information through the following operations:

Should be revised to:

At any point after the NotifyFilingReviewComplete operation, if the case is accessible, a participant MAY access information through the following operations:

1. **Document Hash**

The document hash element is at the wrong level, it should instead be called ecf:DocumentRenditionHash and reside within ecf:DocumentRendition.

Consider that there may be multiple renditions for a document, such as an un-redacted rendition and a redacted rendition, or an English language rendition and a French language rendition, or a un-file stamped rendition and a file stamped rendition, etc.

Each rendition will require its own individual hash as the different renditions will not produce the same hash value.

1. **Filed Documents**

Section 6.1.10 NotifyFilingReviewComplete includes:

The operation MAY return the filed documents or links to the documents, but MUST include the **[FIPS 180-2]** SHA 256 document hash, a condensed representation of a document intended to protect document integrity.

By my understanding, a document is only considered to be ‘filed’ when it has been ‘docketed’. As such, ‘filed documents’ are only available to the FRMDE for inclusion in the NotifyFilingReviewComplete message when they have been provided by the CRMDE in the NotifyDocketingComplete message.

As such, should not the same statement be included in section 6.1.9 NotifyDocketingComplete?

Additionally, my interpretation of the 6.1.10 extract above is that inclusion of filed documents or links is optional, but inclusion of the hash is not optional. Even if the filed document or links are not included, the hash must still be included.

If this interpretation is wrong, then please help me understand why.

If this understanding is not wrong, then please note that ecf:DocumentHash is optional in both docketcallback.xsd and reviewfilingcallback.xsd.

Or is the intent that inclusion of the filed documents or links is optional, but if either the flied document or link is included, then the hash must also be included?

1. **AffectedParty**

The element ecf:AffectedPartyID in ecf:DocumentAugmentation was replaced with ecf:AffectedParty.

The old element, ecf:AffectedPartyID was an nc:IdentificationType, but the new element, ecf:AffectedParty is an nc:EntityType.

The element definition was not adjusted when the element name was changed. Here is a suggested definition revision:

Old and current definition:

“Identifier recognized by the court as being unique within this case, and used to identify a party other than the filer who is affected by the document.”

Suggested rewrite:

A participant entity who is affected by the document.

Should the name of this entity be changed to ecf:AffectedParticipant or is ecf:AffectedParty only to be used for case parties (e.g. litigants)?

1. **Frivolous and Nonsensical Parent-Child References**

The following is perfectly valid in ECF5:

<filing:FilingLeadDocument structures:ref="Document1" xsi:nil="true"/>

<filing:FilingLeadDocument structures:ref="Document1" xsi:nil="true"/>

<filing:FilingLeadDocument structures:ref="Document1" xsi:nil="true"/>

<filing:FilingLeadDocument structures:ref="Document1" xsi:nil="true"/>

<filing:FilingLeadDocument structures:id="Document1" structures:metadata="Document1Metadata">

<nc:DocumentCategoryText>Apperance</nc:DocumentCategoryText>

<nc:DocumentSoftwareName>Microsoft Word</nc:DocumentSoftwareName>

<nc:DocumentDescriptionText>Appearance</nc:DocumentDescriptionText>

…

I do not know what this may be intended to mean, or why anyone would deliberately do this. I cannot image any useful purpose, but it is legal, well formed, and valid per schema. Unless someone can come up with some reasonable justification, I feel the specification should forbid this and other similarly frivolous markup.

Is this already prohibited as a circular reference as described or suggested in section 6.3 Reference Rules?

I’m not sure that it is prohibited using circularity as described in section 6.3. And I have to admit that I always thought about a circular reference as involving two or more differently named elements.

The current circularity statement is below:

circular references, in which a reference element references other reference elements which ultimately refer back to the original reference element (e.g. through a chain of references), are NOT permitted.

And what about the following?

<ecf:ReviewedConnectedDocument structures:ref="Document1" xsi:nil="true"/>

<ecf:ReviewedLeadDocument structures:id="Document1" structures:metadata="Document1Metadata">

<nc:DocumentCategoryText>Apperance</nc:DocumentCategoryText>

<nc:DocumentSoftwareName>Microsoft Word</nc:DocumentSoftwareName>

<nc:DocumentDescriptionText>Appearance</nc:DocumentDescriptionText>

This too is both well formed and valid per schema. But what does this mean? Is this a way of saying that there is a connected document that is identically the same as a lead document? Or is this too a circular reference?

Perhaps we need a new rule:

Elements which have a parent to child relationship, whether that relationship is established either logically or structurally, must not participate in any element reference that contradicts the parent to child relationship.

By structural parent child relationship, I am referring to circumstances, such as element nesting, where one element is a sub-element of another element.

An example of a structural parent-child relationship is as established with j:CaseLineageCase when used in the context of an appellate case. The case at the appellate court exists due to an appeal filed in a prior case at a lower court. The lower court case is the predecessor case, and the appellate case is the descendent case. As such, the lower court is in the role of parent, and the appellate case is in the role of child.

Another example is ecf:ChildDocket in ecf:CourtEventAugmentation. In this arrangement, j:CaseCourtEvent is the parent, and ecf:ChildDocket is the child event.

A logical parent-child relationship is established by other means such as by definition. For example, a FilingConnectedDocument must be the child of some parent FilingLeadDocument; this is by definition and not by XML structure.

So the following are examples of violations of this new proposed rule:

Example 1:

<j:CaseCourtEvent structures:id=”Event1” >

…

<ecf:CourtEventAugmentation>

<ecf:ChildDocket structures:ref=”Event1” xsi:nil=”true”/>

Example 2:

<filing:FilingLeadDocument structures:id=”Document1”>

…

<ecf:DocumentAugmentation>

<nc:DocumentAssociation>

<nc:PrimaryDocument structures:ref=”Document1” xsi:nil=”true”/>

<ecf:DocumentAssociationAugmentation>

<ecf:DocumentRelatedCode>parent</ecf:DocumentRelatedCode>

Example 3:

<ecf:ReviewedLeadDocument structures:id=’Document1”>

…

</ecf:ReviewedLeadDocument>

<ecf:ReviewedLeadDocument structures:ref=”Document1” xsi:nil=”true”/>

Example 4:

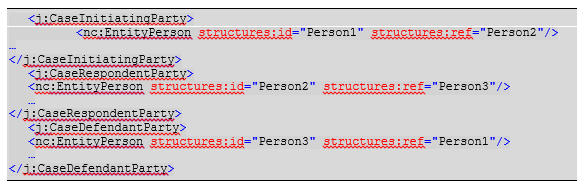
<filing:FilingConnectedDocument structures:id=”Document1>

…

</filing:FilingConnectedDocument>

<filing:FilingLeadDocument structures:ref=”Document1” xsi:nil=”true”/>

By the way, can we fix the indentation in the second (circular reference) example in section 6.3; it looks like this currently:



I think it would be easier to read and understand if it looked like:

<j:CaseInitiatingParty>

<nc:EntityPerson structures:id="Person1" structures:ref="Person2"/>

…

</j:CaseInitiatingParty>

<j:CaseRespondentParty>

<nc:EntityPerson structures:id="Person2" structures:ref="Person3"/>

…

</j:CaseRespondentParty>

<j:CaseDefendantParty>

<nc:EntityPerson structures:id="Person3" structures:ref="Person1"/>

…

</j:CaseDefendantParty>

1. **Reference Rules, section 6.3**

In the second paragraph in section 6.3 there is a reference to the nc:RoleOfPerson element in the prior section.

In an earlier working draft, there had been a prior section on Content References, which included an example that contained the nc:RoleOfPerson element. This example has been moved into section 6.3.1 Attorney to Party References.

Perhaps an appropriate correction would be:

In the example in section 6.3.1, the nc:RoleOfPerson element is a reference element.

1. **cyfs Parent and Child**

I’m not suggesting that there is any error or correction needed in cyfs:ParentChildAssociation, I am just asking for an explanation if this is possible.

The outer element cyfs:ParentChildAssociation contains sub-elements cfs:Child and cyfs:Parent. So far so good, this makes complete sense.

However, cyfs:Parent is an nc:PersonType, but cyfs:Child is a cyfs:ChildType which contains nc:RoleOfPerson which is an nc:PersonType. Why?

1. **appellate:AppellateCaseParty**

This element only permits nc:EntityPerson and nc:EntityOrganization. Element ecf:EnityItem must also be allowed.

Also, the definition for this element is incorrect, it begins with “An identifier assigned to a party …”. This is not an identifier, it is a party.

But why does it need to be a ‘special’ party (i.e. appellate:AppellateCaseParty), why can’t it just be a regular party (i.e. ecf:CaseParty)?

1. **xxx**