



Electronic Court Filing Version 4.01 **Plus** **Errata 01**

OASIS Standard **incorporating Public Review Draft 01** **of Errata 01**

~~23 May 2013~~ **13 May 2014**

Specification URIs

This version:

<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/ecf-v4.01-spec-errata01-csprd01-complete.doc> (Authoritative)
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/ecf-v4.01-spec-errata01-csprd01-complete.html>
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/ecf-v4.01-spec-errata01-csprd01-complete.pdf>

Previous version:

<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/os/ecf-v4.01-spec-os.doc> (Authoritative)
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/os/ecf-v4.01-spec-os.html>
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/os/ecf-v4.01-spec-os.pdf>

Latest version:

<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/ecf-v4.01-spec-errata01-complete.doc> (Authoritative)
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/ecf-v4.01-spec-errata01-complete.html>
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/ecf-v4.01-spec-errata01-complete.pdf>

Technical Committee:

OASIS LegalXML Electronic Court Filing TC

Chairs:

James Cabral (jcabral@mtgmc.com), MTG Management Consultants
Jim Harris (jharris@ncsc.org), National Center for State Courts

Editors:

Adam Angione (aangione@courthousenews.com), Courthouse News Service
James Cabral (jcabral@mtgmc.com), MTG Management Consultants

Additional artifacts:

This prose specification is one component of a Work Product which also includes:

- *Electronic Court Filing Version 4.01 Errata 01*. Edited by James Cabral. 13 May 2014. OASIS Committee Specification Draft 01. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/ecf-v4.01-spec-errata01-csprd01.html>.

- XML schemas:
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/xsd/>
- XML sample messages:
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/xml/>
- Model:
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/model/>
- Generic code lists:
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/gc/>

Related work:

This specification replaces or supersedes:

- *OASIS LegalXML Electronic Court Filing Version 3.0*. 15 November 2005.
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v3.0/>
- *OASIS Electronic Court Filing Version 4.0*. 21 September 2008.
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.0/ecf-v4.0-spec/>

This specification is related to:

- National Information Exchange Model 2.0. <http://niem.gov/>

Declared XML namespaces:

urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:AppInfo-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:AppellateCase-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:BankruptcyCase-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseListQueryMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseListResponseMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseQueryMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseResponseMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CitationCase-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CivilCase-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CommonTypes-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CoreFilingMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CourtPolicyQueryMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CourtPolicyResponseMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CriminalCase-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:DocumentQueryMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:DocumentResponseMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:DomesticCase-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FeesCalculationQueryMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FeesCalculationResponseMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FilingListQueryMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FilingListResponseMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FilingStatusQueryMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FilingStatusResponseMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:JuvenileCase-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:MessageReceiptMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:PaymentMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:PaymentReceiptMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:RecordDocketingCallbackMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:RecordDocketingMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:ReviewFilingCallbackMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:ServiceInformationQueryMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:ServiceInformationResponseMessage-4.0
 urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:ServiceReceiptMessage-4.0

Abstract:

This document defines the LegalXML Electronic Court Filing 4.01 (ECF 4.0) specification, which consists of a set of non-proprietary XML and Web services specifications, along with clarifying explanations and amendments to those specifications, that have been added for the purpose of promoting interoperability among electronic court filing vendors and systems. ECF Version 4.01 is a maintenance release to address several minor schema and definition issues identified by implementers of the ECF 4.0 specification.

Status:

This document was last revised or approved by the members of OASIS on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at <https://www.oasis-open.org/committees/legalxml-courtfilling/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<https://www.oasis-open.org/committees/legalxml-courtfilling/ipr.php>).

Citation format:

When referencing this specification the following citation format should be used:

[ECF v4.01]

Electronic Court Filing Version 4.01 Plus Errata 01. 13 May 2014. OASIS Standard incorporating Public Review Draft 01 of Errata 01. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/csprd01/ecf-v4.01-spec-errata01-csprd01-complete.html>. Latest version: <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/ecf-v4.01-spec-errata01-complete.html>.

Notices

Copyright © OASIS Open 2014. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the “OASIS IPR Policy”). The full [Policy](#) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an “AS IS” basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS’ procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name “OASIS” is a trademark of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark> for above guidance.

Table of Contents

| | | |
|--------|--|----|
| 1 | Introduction..... | 8 |
| 1.1 | Scope..... | 8 |
| 1.2 | Relationship to Prior Specifications..... | 9 |
| 1.3 | ECF Version 4.01..... | 9 |
| 1.3.1 | National Information Exchange Model (NIEM)..... | 10 |
| 1.3.2 | OASIS Universal Business Language..... | 10 |
| 1.3.3 | W3C XML-Signature Syntax and Processing..... | 10 |
| 1.3.4 | OASIS Reference Model for Service Oriented Architecture..... | 11 |
| 1.3.5 | OASIS Code List Representation (Genericode)..... | 11 |
| 1.4 | Terms and Definitions..... | 11 |
| 1.5 | Symbols and Abbreviations..... | 12 |
| 1.6 | Normative References..... | 13 |
| 1.7 | Non-Normative References..... | 14 |
| 2 | ECF 4.0 Architecture..... | 15 |
| 2.1 | Core vs. Profiles..... | 15 |
| 2.2 | Major Design Elements..... | 15 |
| 2.3 | Information Model..... | 16 |
| 2.3.1 | Messages..... | 16 |
| 2.3.2 | Attachment..... | 17 |
| 2.3.3 | Sample Message Streams..... | 17 |
| 2.4 | Court Policy..... | 20 |
| 2.4.1 | Human-Readable Court Policy..... | 20 |
| 2.4.2 | Machine-Readable Court Policy..... | 20 |
| 2.4.3 | Case-Type and Court Extensions..... | 21 |
| 2.4.4 | Court-Specific Code Lists..... | 21 |
| 2.4.5 | Court-Specific Constraint Schemas..... | 22 |
| 3 | ECF 4.0 Process Model..... | 24 |
| 3.1 | The Filing-Preparation-to-Docketing Process Model..... | 24 |
| 3.2 | Business Rules..... | 26 |
| 3.2.1 | GetPolicy..... | 26 |
| 3.2.2 | GetServiceInformation..... | 26 |
| 3.2.3 | GetFeesCalculation..... | 26 |
| 3.2.4 | ReviewFiling..... | 26 |
| 3.2.5 | ServeFiling..... | 27 |
| 3.2.6 | RecordFiling..... | 27 |
| 3.2.7 | NotifyDocketingComplete..... | 27 |
| 3.2.8 | NotifyFilingReviewComplete..... | 27 |
| 3.2.9 | GetFilingList..... | 27 |
| 3.2.10 | GetFilingStatus..... | 28 |
| 3.2.11 | GetCaseList..... | 28 |
| 3.2.12 | GetCase..... | 28 |
| 3.2.13 | GetDocument..... | 28 |

| | | |
|--|--|----|
| 3.3 | Message Business Rules | 28 |
| 3.3.1 | Identifiers | 28 |
| 3.3.2 | Code Lists..... | 29 |
| 3.3.3 | Message-Specific Business Rules | 31 |
| 3.4 | Filing the Record on Appeal | 31 |
| 4 | ECF 4.0 Schemas | 34 |
| 4.1 | ECF 4.0 Case Type Schemas | 34 |
| 4.2 | ECF 4.0 Common Schemas..... | 34 |
| 4.3 | ECF 4.0 Constraint and Subset Schemas..... | 35 |
| 4.4 | ECF 4.0 Message Schemas..... | 35 |
| 5 | Service Interaction Profiles..... | 37 |
| 5.1 | Service Interaction Profile Requirements | 37 |
| 5.2 | Service Interaction Profile Approval and Revision Processes..... | 38 |
| 5.3 | Supported Service Interaction Profiles | 39 |
| 6 | Document Signature Profiles..... | 40 |
| 6.1 | Document Signature Profile Requirements | 40 |
| 6.2 | Document Signature Profile Approval and Revision Processes..... | 40 |
| 6.3 | Supported Document Signature Profiles | 41 |
| 7 | Conformance..... | 42 |
| Appendix A. (Informative) Release Notes | | 43 |
| A.1 | Availability..... | 43 |
| A.2 | Package Structure..... | 43 |
| A.3 | Recursive Structures | 43 |
| A.4 | Date and Time Formats..... | 43 |
| A.5 | Known Errata..... | 44 |
| Appendix B. (Informative) ECF 4.0 Development Approach and Artifacts | | 45 |
| B.1 | Principles | 45 |
| B.2 | Approach | 45 |
| B.3 | ECF 4.0 Exchange Content Models | 45 |
| B.4 | Spreadsheet Models | 47 |
| Appendix C. (Informative) MDE Operations..... | | 48 |
| C.1 | Filing Assembly MDE | 48 |
| C.1.1 | Provided Operations | 48 |
| C.1.2 | Consumed Operations | 48 |
| C.2 | Filing Review MDE | 49 |
| C.2.1 | Provided Operations | 49 |
| C.2.2 | Consumed Operations | 49 |
| C.3 | Court Record MDE | 50 |
| C.3.1 | Provided Operations | 50 |
| C.3.2 | Consumed Operations | 50 |
| C.4 | Legal Service MDE | 50 |
| C.4.1 | Provided Operations | 51 |
| C.4.2 | Consumed Operations | 51 |
| Appendix D. (Informative) Example Instances..... | | 52 |
| Appendix E. (Informative) Ongoing Work Items..... | | 54 |

Appendix F. (Informative) Acknowledgments 55
Appendix G. (Informative) Revision History 56

1 Introduction

This document is a specification developed by the OASIS LegalXML Electronic Court Filing Technical Committee. It defines a technical architecture and a set of components, operations and message structures for an electronic court filing system, and sets forth rules governing its implementation.

1.1 Scope

This specification describes the technical architecture and the functional features needed to accomplish a successful electronic court filing system, and defines both the normative (required) and non-normative (optional) business processes it supports. The non-functional requirements associated with electronic filing transactions, as well as the actions and services needed to accomplish the transactions, such as network and security infrastructures, are defined in related specifications, namely:

- Service interaction profile specifications that define communications infrastructures, within which electronic filing transactions can take place
- Document signature profile specifications that define mechanisms for stating or ensuring that a person signed a particular document

This specification supports the following automated information exchanges:

- Transmission of documents in electronic form from law firms and from other persons and organizations to a court for entry (“official filing”) into the court’s official case records
- Recording of documents in electronic form from members of the court and court administrators into the court’s official case records
- Transmission of data needed to complete (or demonstrate the previous completion of) financial transactions involving filing fees or the payment of any other court fees, fines and financial obligations
- Transmission of the metadata needed to initiate a new case record in a court’s automated case management system (CMS) when the document being transmitted is one that commences a new case in that court
- Transmission of the metadata needed to create an entry that records (indexes) a filed document in a court’s electronic listing of cases and their contents (variously called a “docket” or “register of actions”)
- Transmission of the metadata needed to update the information recorded about a case that is maintained in a court’s CMS
- Messages returned to the sender that confirm a court’s receipt of the sender’s filing message
- Messages notifying the sender of events such as the entry of the document(s) submitted by the sender into the court record (or an error message stating that the document[s] could not be accepted for filing and stating the reason[s] why)
- Queries to the court seeking information about data and documents held within the court’s official electronic records and the return of information in response to those queries
- Queries from filers for the court rules and requirements for electronic filing
- Queries by filers seeking from the court record system the names and addresses of parties in a case who must be served and whether by traditional or electronic means
- Transmission of copies of documents submitted for filing to the other parties in a case who are registered to receive service electronically

In addition to filing of court case documents, this specification supports “secondary service” – the delivery of copies of filed documents to persons who have already been made parties to a case. This

45 specification does NOT support “primary service,” which entails the service of summonses, subpoenas,
46 warrants and other documents that establish court jurisdiction over persons, making them parties to a
47 case. Therefore, this specification does NOT support the following automated information exchanges:

- 48 • A query by a filer seeking from the court record system the names and addresses of parties in a new
49 case who must be served to establish court jurisdiction over them in the new case
- 50 • Transmission of copies of or links to documents submitted for filing to any party in a new case or any
51 newly added parties in an existing case

52

53 This specification defines a set of core structures that are common to most types of court filings and
54 defines specific structures that apply to filing documents in the following types of court cases:

- 55 • Appellate
- 56 • Bankruptcy
- 57 • Civil (including general civil, mental health, probate and small claims)
- 58 • Criminal (both felony and misdemeanor)
- 59 • Domestic relations (including divorce, separation, child custody and child support, domestic violence
60 and parentage, i.e., maternity or paternity)
- 61 • Juvenile (both delinquency and dependency)
- 62 • Violations (including traffic, ordinances and parking)

63

64 Although ECF 4.01 does not define data structure elements specific to other case types (e.g.,
65 administrative tribunals), the basic structure will support other types of court filings and is extensible
66 through court-specific and case-type-specific extensions.

67 1.2 Relationship to Prior Specifications

68 Electronic Court Filing 4.0 superseded the LegalXML Electronic Court Filing 3.0, 3.01 and 3.1
69 specifications developed by the predecessor organizations to the OASIS Electronic Court Filing Technical
70 Committee. Those specifications were prepared for and approved by the COSCA/NACM Joint
71 Technology Committee as proposed standards.

72 Relative to the ECF 3.0, 3.01 and 3.1 specifications, the ECF 4.0 and 4.01 specifications provide a
73 number of enhancements including:

- 74 • Leveraging of the National Information Exchange Model (**[NIEM]**), a national standard for information
75 sharing
- 76 • Leveraging of the updates to the OASIS Universal Business Language (**[UBL]**), for describing
77 payments
- 78 • The inclusion of the data elements needed for appellate cases

79

80 This specification does not assume that prior specifications will be deprecated. However, ECF 4.0 is not
81 backward-compatible and applications using the ECF 3.0, 3.01 and 3.1 specifications will not interoperate
82 successfully with applications using these specifications. This fact is indicated by the assignment of a
83 new major version number to the ECF 4.0 and 4.01 specifications.

84 1.3 ECF Version 4.01

85 ECF 4.01 is a maintenance release to address several minor schema and definition issues identified by
86 implementers of the ECF 4.0 specification. All references in this document to ECF 4.0 apply to ECF 4.01
87 as well. Relationship to other XML Specifications

88 The ECF specification incorporates other existing, non-proprietary XML specifications wherever possible.
89 In particular, the specification has dependencies on the **[NIEM]**, the **[UBL]** data library and the World

90 Wide Web Consortium (W3C) XML Digital Signatures specification. The terminology used in this
91 specification to describe the components of the ECF technical architecture conforms to the OASIS
92 Reference Model for Service Oriented Architecture.

93 It is recommended that implementations cache external schemas locally to improve performance and
94 reliability. (The alternative would be to rely on the external schemas as they are, in someone else's
95 control, and assume they will not be changed or become hard to access due to Internet or network
96 problems.) The copies of external schemas that are cached in this way should be updated and refreshed
97 often to ensure changes will be quickly learned and addressed.

98 **1.3.1 National Information Exchange Model (NIEM)**

99 **[NIEM]** conformance, as defined by the NIEM Implementation Guidelines (**[NIEM Guide]**), is a core
100 objective of this specification. The **[NIEM]** is an XML standard designed specifically for justice information
101 exchanges, providing law enforcement, public safety agencies, prosecutors, public defenders and the
102 judicial branch with a tool to effectively share data and information in a timely manner. The **[NIEM]**
103 provides a library of reusable components that can be combined to automate justice information
104 exchanges. The **[NIEM]** removes the burden from agencies to independently create exchange
105 standards. Because of its extensibility, there is more flexibility to deal with unique agency requirements
106 and changes. Through the use of a common vocabulary that is understood system to system, **[NIEM]**
107 enables access from multiple sources and reuse in multiple applications. The use of **[NIEM]** element
108 names does not require any change in local legal terminology. XML tag names are invisible to the user of
109 an application employing them.

110 The **[NIEM]** is most useful for describing common objects such as persons and locations, and criminal
111 justice-specific processes such as arrest, booking, jail and prosecution. The **[NIEM]** is not as well
112 developed for describing non-criminal information exchanges and processes. ECF 4.0 uses the **[NIEM]**
113 version 2.0 where the structures and definitions correspond to the requirements of ECF 4.0. The
114 development process, including the **[NIEM]** modeling process, is described in Appendix B.

115 **1.3.2 OASIS Universal Business Language**

116 **[UBL]** is an OASIS Standard that provides a single ubiquitous language for business communication, and
117 takes into account the requirements common to all enterprises. **[UBL]** provides a shared library of
118 reusable components, essential to interoperability that can be combined to create electronic business
119 schemas. Without a common set of base components, each document format would risk redefining
120 addresses, locations and other basic information in incompatible ways.¹

121 ECF 4.0 employs the following structures in the **[UBL]** to describe filing payments and payment receipts:

122 <AllowanceCharge>

123 Information about a charge or discount price component.

124 <Address>

125 Information about a structured address.

126 <Payment>

127 Information directly relating to a specific payment.

128 **1.3.3 W3C XML-Signature Syntax and Processing**

129 The W3C XML Signature Syntax and Processing (**[XMLSIG]**) specification describes a mechanism for
130 signing electronic documents. This mechanism allows recipients of electronic documents to identify the

¹ <http://www.oasis-open.org/committees/download.php/1023/UBL%3A%20The%20Next%20Step%20for%20Global%20E-Commerce>

131 sender and be assured of the validity of the electronically transmitted data. **[XMLSIG]** defines standard
132 means for specifying information content that is to be digitally signed.²
133 ECF 4.0 employs the **[XMLSIG]** specification to describe digital signatures applied to the entire ECF 4.0
134 message transmission in order to provide authentication, encryption and message integrity. **[XMLSIG]** is
135 also used in the ECF 4.0 XML Document Signature Profile.

136 **1.3.4 OASIS Reference Model for Service Oriented Architecture**

137 The **[SOA-RM]** is a framework for understanding significant entities, and the relationships between those
138 entities, within a service-oriented architecture. ECF 4.0 describes such an architecture and includes
139 terminology that conforms to the **[SOA-RM]**.

140 **1.3.5 OASIS Code List Representation (Genericode)**

141 The OASIS Code List Representation format, **[Genericode]**, is a model and XML schema that can be
142 used to encode a broad range of code list information. The XML format is designed to support
143 interchange or distribution of machine-readable code list information between systems. All ECF 4.0 code
144 lists that are not defined in the NIEM are provided in **[Genericode]** 1.0 format.

145

146 **1.4 Terms and Definitions**

147 The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD
148 NOT”, “RECOMMENDED”, “MAY” and “OPTIONAL” in this document are to be interpreted as described
149 in [RFC2119].

150

151 This section defines key terms used in this specification.

152

153 **Attachment**

154 See definition in Section 2.3.2.

155 **Callback message**

156 A message transmission returned by some operations some time after the operation was invoked
157 (asynchronously).

158 **Document**

159 An electronic equivalent of a document that would otherwise be filed on paper in a traditional,
160 non-electronic fashion.

161 **Document hash**

162 A condensed representation of a document intended to protect document integrity, calculated
163 according to the FIPS 180-2 SHA 256 algorithm.

164 **Docketing**

165 The process invoked when a court receives a pleading, order or notice, with no errors in
166 transmission or in presentation of required content, and records it as a part of the official record.

167 **Filer**

168 An attorney or a *pro se* (self-represented) litigant acting as an individual who assembles and
169 submits one or more filings (combinations of data and documents).

² <http://xml.coverpages.org/xmlSig.html>

- 170 **Filing**
171 An electronic document (with any associated data, attachments and the like) that has been
172 assembled for the purpose of being filed into a specified court case.
- 173 **Hub Service MDE**
174 A centralized Service MDE capable of receiving a single set of service notifications for all parties
175 registered for electronic service in a case and transmitting the service notifications to the Service
176 MDEs registered to each party in the case.
- 177 **Major Design Element (MDE)**
178 A logical grouping of operations representing a significant business process supported by ECF
179 4.0. Each MDE operation receives one or more messages, returning a synchronous response
180 message (a reaction to a message received) and, optionally, returning an asynchronous (later)
181 response message to the originating message sender.
- 182 **Message**
183 See definition in Section 2.3.1.
- 184 **Message Transmission**
185 The sending of one or more messages and associated attachments to an MDE. Each
186 transmission must invoke or respond to an operation on the receiving MDE, as defined in the
187 ECF 4.0 specification.
- 188 **Operation (or MDE Operation)**
189 A function provided by an MDE upon receipt of one or more messages. The function provided by
190 the operation represents a significant step in the court filing business process. A sender invokes
191 an operation on an MDE by transmitting a request with an operation identifier and a set of
192 messages.
- 193 **Operation signature**
194 A definition of the input message and synchronous response message associated with an
195 operation. Each message is given a name and a type by the operation. The type is defined by a
196 single one of the message structures defined in the ECF 4.0 specification.
- 197 **Synchronous response**
198 A message transmission returned immediately (synchronously) as the result of an operation.
199 Every operation has a synchronous response.

200 **1.5 Symbols and Abbreviations**

201 This section defines key symbols and abbreviations used in this specification.

202

203 **ECF 4.0**

204 Electronic Court Filing 4.0

205 **IEPD**

206 Information Exchange Package Documentation

207 **MDE**

208 Major Design Element

209 **NIEM**

210 National Information Exchange Model

211 **OASIS**

212 Organization for the Advancement of Structured Information Standards

| | |
|-----|--|
| 213 | XML |
| 214 | eXtensible Markup Language |
| 215 | W3C |
| 216 | World Wide Web Consortium |
| 217 | WS-I |
| 218 | Web Services Interoperability Organization |
| 219 | |

220 1.6 Normative References

| | |
|-----|---|
| 221 | [FIPS 180-2] |
| 222 | <i>Secure Hash Standard</i> , http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf , National Institute for Standards and Technology, August 2002. |
| 223 | |
| 224 | |
| 225 | [Genericode] |
| 226 | A. B. Coates, <i>Code List Representation (Genericode) 1.0</i> , http://docs.oasis-open.org/codelist/ns/genericode/1.0/ , OASIS Committee Specification, December 28, 2007 |
| 227 | |
| 228 | |
| 229 | [NIEM] |
| 230 | <i>National Information Exchange Model 2.0</i> , http://niem.gov , US DOJ and DHS, 2007. |
| 231 | |
| 232 | [NIEM Guide] |
| 233 | <i>NIEM Implementation Guidelines</i> , http://www.niem.gov/implementationguide.php , US DOJ and DHS, 2007. |
| 234 | |
| 235 | [NIEM Techniques] |
| 236 | <i>Techniques for Building and Extending NIEM</i> , |
| 237 | http://www.niem.gov/topicIndex.php?topic=techPDF , Georgia Tech Research Institute, August 2007. |
| 238 | |
| 239 | [Namespaces] |
| 240 | T. Bray, <i>Namespaces in XML</i> , http://www.w3.org/TR/1999/REC-xml-names-19990114 , January 14, 1999. |
| 241 | |
| 242 | [RFC2046] |
| 243 | N. Freed, <i>Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types</i> , |
| 244 | http://www.ietf.org/rfc/rfc2046.txt , IETF RFC 2046, November 1996. |
| 245 | [RFC2119] |
| 246 | S. Bradner, <i>Key words for use in RFCs to Indicate Requirement Levels</i> , |
| 247 | http://www.ietf.org/rfc/rfc2119.txt , IETF RFC 2119, March 1997. |
| 248 | [RFC4122] |
| 249 | Leach, et al., <i>A Universally Unique Identifier (UUID) URN Namespace</i> , |
| 250 | http://www.ietf.org/rfc/rfc4122.txt , IETF RFC 4112, July 2005. |
| 251 | [Schema Part 1] |
| 252 | H. S. Thompson, D. Beech. M. Maloney, N. Mendelsohn, <i>XML Schema Part 1: Structures Second Edition</i> , http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/ , W3C Recommendation, October 28, 2004. |
| 253 | |
| 254 | |
| 255 | [Schema Part 2] |
| 256 | P. Biron, A. Malhotra, <i>XML Schema Part 2: Datatypes Second Edition</i> , |
| 257 | http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/ , W3C |
| 258 | Recommendation, October 28, 2004 |

259 **[UBL]** *Universal Business Language Version 2.1 30 May 2011*. 30 May 2011.
 260 Committee Specification Draft 02 / Public Review Draft 02. [http://docs.oasis-](http://docs.oasis-open.org/ubl/prd2-UBL-2.1/UBL-2.1.html)
 261 [open.org/ubl/prd2-UBL-2.1/UBL-2.1.html](http://docs.oasis-open.org/ubl/prd2-UBL-2.1/UBL-2.1.html)
 262 J. Bozak, T. McGrath, G. K. Holman (editors), *Universal Business Language 2.0*,
 263 , OASIS Standard, December 12, 2006.

264 **[XML 1.0]**
 265 T. Bray, *Extensible Markup Language (XML) 1.0 (Third Edition)*,
 266 <http://www.w3.org/TR/REC-xml/REC-XML-20040204>, W3C Recommendation,
 267 February 4, 2004.

268 **[XMLENC]**
 269 D. Eastlake, J. Reagle, *XML Encryption Syntax and Processing*,
 270 <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>, W3C
 271 Recommendation, December 2002.

272 **[XMLSIG]**
 273 D. Eastlake, J. Reagle, D. Solo, *XML-Signature Syntax and Processing*,
 274 <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>, W3C
 275 Recommendation, February 2002.

276 **1.7 Non-Normative References**

277 **[Court Document]**
 278 *OASIS LegalXML Court Document Subcommittee, Charter, July 2006*,
 279 [http://www.oasis-open.org/committees/download.php/19120/DocumentSC-](http://www.oasis-open.org/committees/download.php/19120/DocumentSC-Charter-Rev6-1.txt)
 280 [Charter-Rev6-1.txt](http://www.oasis-open.org/committees/download.php/19120/DocumentSC-Charter-Rev6-1.txt).

281 **[NIEM MNDR]**
 282 W. Roberts, S. Liebeskind, M. Kindl *National Information Exchange Model*
 283 *Naming and Design Rules Draft 1.2*,
 284 <http://www.niem.gov/topicIndex.php?topic=file-NDR-lineNum>, August 7, 2007.

285 **[Juvenile XML]**
 286 S. Rondendell, et. al., *Juvenile Justice XML Report*,
 287 [http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml](http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml%5ffinal%5freport%5f20050630.pdf)
 288 [%5ffinal%5freport%5f20050630.pdf](http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml%5ffinal%5freport%5f20050630.pdf), IJIS Institute, July 2005.

289 **[NIEM]**
 290 *NIEM Concept of Operations*, <http://www.niem.gov>, DOJ/DHS, October 7, 2005.

291 **[NCSC Guide]**
 292 *State Court Guide to Statistical Reporting*,
 293 [http://www.ncsconline.org/D_Research/csp/2003_Files/CompleteGuide11_02_04](http://www.ncsconline.org/D_Research/csp/2003_Files/CompleteGuide11_02_04.pdf)
 294 [.pdf](http://www.ncsconline.org/D_Research/csp/2003_Files/CompleteGuide11_02_04.pdf), National Center for State Courts, November 2004.

295 **[Rap Sheet]**
 296 *Interstate Criminal History Transmission Specification XML Version 3.00*,
 297 http://www.search.org/files/pdf/CH_transmission_spec.pdf, Joint Task Force on
 298 Rap Sheet Standardization, February 2005.

299 **[SOA-RM]**
 300 MacKenzie, et al., *Reference Model for Service Oriented Architecture 1.0*,
 301 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm, OASIS
 302 Public Review Draft 1.0, February 10, 2006.

303 **[Traffic IEPD]**
 304 *Traffic Citation IEPD*,
 305 [http://www.ncsconline.org/d_tech/gjxdm/iepd/citation/Traffic_Citation_08_08_200](http://www.ncsconline.org/d_tech/gjxdm/iepd/citation/Traffic_Citation_08_08_2005.zip)
 306 [5.zip](http://www.ncsconline.org/d_tech/gjxdm/iepd/citation/Traffic_Citation_08_08_2005.zip), National Center for State Courts, August 8, 2005.

307 2 ECF 4.0 Architecture

308
309 The ECF 4.0 architecture consists of four Major Design Elements (MDEs), which support operations and
310 messages. An MDE is a logical grouping of operations, such as the operations involved in creating a
311 filing or the operations involved in receiving and recording a filing, that is, incorporating the constituent
312 documents into a court document management system. A message is the data exchanged between
313 MDEs in the form of an XML document that may include one or more additional binary attachments.
314 These messages contain the information to be filed with the court. This section describes the ECF 4.0
315 architecture including the MDEs, the operations and the messages.

316 2.1 Core vs. Profiles

317 The ECF 4.0 architecture can be divided into three principal elements:

- 318 • **Core Specification** – This core specification defines the MDEs and the operations and messages
319 that are exchanged between MDEs.
- 320 • **Service Interaction Profiles** – Service interaction profiles are specifications that describe
321 communication infrastructures that deliver messages between MDEs.
- 322 • **Document Signature Profiles** – Document signature profiles are specifications that describe
323 mechanisms for signing electronic documents.

324 In order to be compliant, an implementation of the ECF specification **MUST** implement the core
325 specification and at least one service interaction profile and one document signature profile.

326 The MDEs and messages that make up the core specification are discussed in Sections 2.2 and 2.3
327 below, respectively. Service interaction profiles are discussed in Section 5 below. Document signature
328 profiles are discussed in Section 6 below.

329 2.2 Major Design Elements

330 ECF 4.0 defines four MDEs. They are:

- 331 • **Filing Assembly MDE** – enables a filer to create a filing message for submission to a court, and for
332 service on other parties in the case, returning a response from the court to the filer.
- 333 • **Filing Review MDE** – enables a court to receive and review a filing message and prepare the
334 contents for recording in its case management and document management systems, sending a
335 response concerning the filing to the Filing Assembly MDE. The Filing Review MDE also enables
336 filers to obtain court-specific policies regarding electronic filing and to check on the status of a filing.
- 337 • **Court Record MDE** – enables a court to record electronic documents and docket entries in its case
338 management and document management systems and returns the results to the Filing Review MDE.
339 The Court Record MDE also enables filers to obtain service information for all parties in a case, to
340 obtain information about cases maintained in the court’s docket, register of actions and calendars,
341 and to access documents maintained in the court’s electronic records.
- 342 • **Legal Service MDE** – enables a party to receive service electronically **FROM** other parties in the
343 case. Note that service **TO** other parties in the case is performed by the Filing Assembly MDE.

344 The MDEs defined in the ECF 4.0 specifications are meant only to define the “interface” to each
345 operation; the specification is not intended to define how operations must be implemented. This strategy
346 allows MDE implementations to interoperate while leaving room for vendors and courts to have differing
347 implementations (e.g., an implementation that supports a particular CMS).

348 An ECF 4.0-compliant implementation may implement one or more of the MDEs defined in the
349 specification but a complete ECF 4.0 system **MUST** include at least one each of the Filing Assembly,
350 Filing Review and Court Record MDEs. For instance, a court may decide to provide certain MDEs and

351 allow private providers to furnish the remaining MDEs. When multiple MDEs are implemented by a single
352 court, vendor or application, the application MUST maintain the ECF 4.0 specified operations between
353 each MDE so that other applications will be able to interoperate with it.

354 Each of the operations supported by an MDE accepts one or more messages as input and returns an
355 immediate, synchronous response message to the calling MDE. For some operations, the MDE will also
356 return an asynchronous (callback) message at a later time that reports the result of a business process
357 implemented within the MDE. In order to be compliant with ECF 4.0, an MDE must support all messages
358 required for that MDE. However, in an ECF 4.0 system that does not support electronic service, the
359 operations associated with the Legal Service MDE are not required.

360 An MDE defines an information model and behavior model of a service as described in the [SOA-RM].
361 One must remember that “service” in the service oriented architecture sense is not the same as the
362 business function of “service of filing” used throughout in this document.

363 2.3 Information Model

364 The ECF information model describes the messages that may be exchanged between MDEs. All ECF
365 4.0 operations use the same core message stream structure, which is implemented in the service
366 interaction profiles. Each ECF core message stream is a stream of bytes that contains at least one
367 message and may also contain attachments.

368 2.3.1 Messages

369 A message is an XML document that is a well-formed XML data structure with a single root element that
370 is transmitted between MDEs and is valid as defined by one of the defined message structure schemas in
371 the ECF 4.0 specification. A message may be related to one or more attachments. A message contains
372 the following information:

- 373 • Message information about the filing and court case, such as identifiers for the sender and receiver,
374 the sending and receiving MDEs, and the submission date and time, typically a composition of:
 - 375 – A core message which includes basic information common to all courts and case types and
376 Information about each of the documents associated with the message
 - 377 – Case-type-specific extensions that includes information appropriate only for a particular type of
378 filing
 - 379 – Court-specific extensions that includes information appropriate only for cases in a particular court
- 380 • Information about each of the documents associated with the message. A document in this sense is
381 the electronic representation of what would be recognized as a “document” if it were a single, whole,
382 physical paper object. This includes both a lead document, one that will be placed on the court’s
383 register of actions (docketed, indexed) and any supporting document(s), which are present to
384 supplement the lead document in some way. The message includes the document’s metadata, for
385 example, its title, type, identifier, parent document identifier and document sequence number. Each
386 document structure may reference one or more attachments, including attachment identifiers and
387 sequence numbers. When included in attachments, a logical document MAY be split into several
388 physical parts if necessary to satisfy a court requirement regarding maximum document size. ~~The~~
389 ~~actual binary encoded electronic document MAY be either included in one or more attachments to the~~
390 ~~message or embedded in the message using the following structure:~~ **The actual binary encoded**
391 **electronic document SHOULD be included in one or more attachments to the message or MAY be**
392 **embedded in the message using the following structure:**

```
393     <FilingLeadDocument> (or <FilingConnectedDocument>)  
394     <ecf:DocumentRendition>  
395         <DocumentRenditionMetadata>  
396         <DocumentAttachment>  
397         <BinaryBase64Object>2345klj345h...<BinaryBase64Objec  
398         t>
```


399 </DocumentAttachment>
400 </DocumentRenditionMetadata>
401 </ecf:DocumentRendition
402 </FilingLeadDocument> (or </FilingConnectedDocument>)

403
404 Elements defined by this specification, whether in core messages, case type-specific extensions or court-
405 specific extensions, are intended to be useful to an automated case management system for the
406 purposes of partially or fully automating case workflow after filing (e.g., filing review, noticing, docketing,
407 judicial assignment, calendaring, standardized forms receipt and generation, fee processing) or
408 ascertaining the adequacy or appropriateness of the filing (e.g., fee or fine calculation, jurisdiction).
409 Elements defined by this specification are not intended to fully populate the automated case management
410 system with all data contained within filed documents. That is, these elements should be useful as “filing
411 metadata” about the case, the filing transaction, parties or documents. These elements may also be “filing
412 data”, or the contents of the filings. For instance, information found on a filing cover sheet can generally
413 be considered filing metadata, even if the information is also repeated in the document(s) being filed.

414
415 The scope of the ECF core messages and extensions is limited by the following criteria:
416 • Elements in the ECF core messages should be applicable to most courts and case types
417 • Elements in the ECF case-type-specific extensions should only be applicable to one of the seven
418 case types defined in National Center for State Courts (NCSC) statistical standards
419 • Elements in locally-defined court-specific extensions should only be applicable to a particular court or
420 court system but not to courts in general
421 All “filing data” elements should be described in the filed documents, whose structure is outside the scope
422 of the ECF specification.

423 **2.3.2 Attachment**

424 An attachment is a series of bytes in the message stream transmitted between MDEs that constitutes, in
425 whole or in part, an electronic document whose conventional equivalent would be a document on paper.
426 The contents are preceded by one or more “headers” that uniquely identify the attachment (using a
427 content identifier) and specify the format or type of the attachment. Note that the contents of an
428 attachment can be binary octets (the “raw” binary data of the document), binary data encoded in text
429 (e.g., via base-64 or some other algorithm), XML text or plain text.

430 Attachments appear in the message stream after the messages. The order of attachments within the
431 message stream is not important and cannot be treated as significant. In particular, this means that the
432 series of bytes representing the content of a lead document need not appear before the attachments
433 representing the content of documents supporting that lead document.

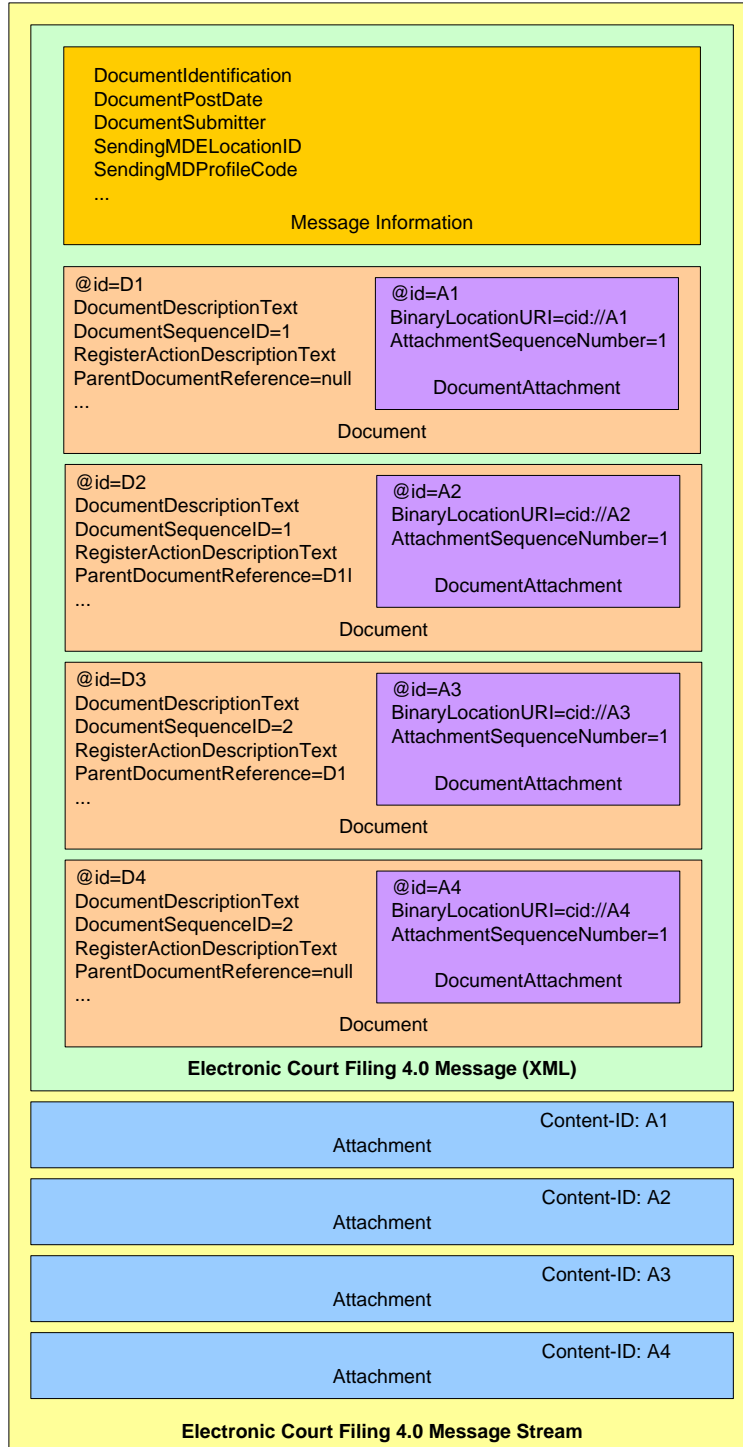
434 **2.3.3 Sample Message Streams**

435 The following conceptual diagrams illustrate the containment structures involved in the message stream.

436
437

438
439
440
441
442

Figure 1 illustrates a message stream involving two lead documents, the first of which has two supporting documents. The second lead document has no supporting documents. Each document is associated with a single attachment.

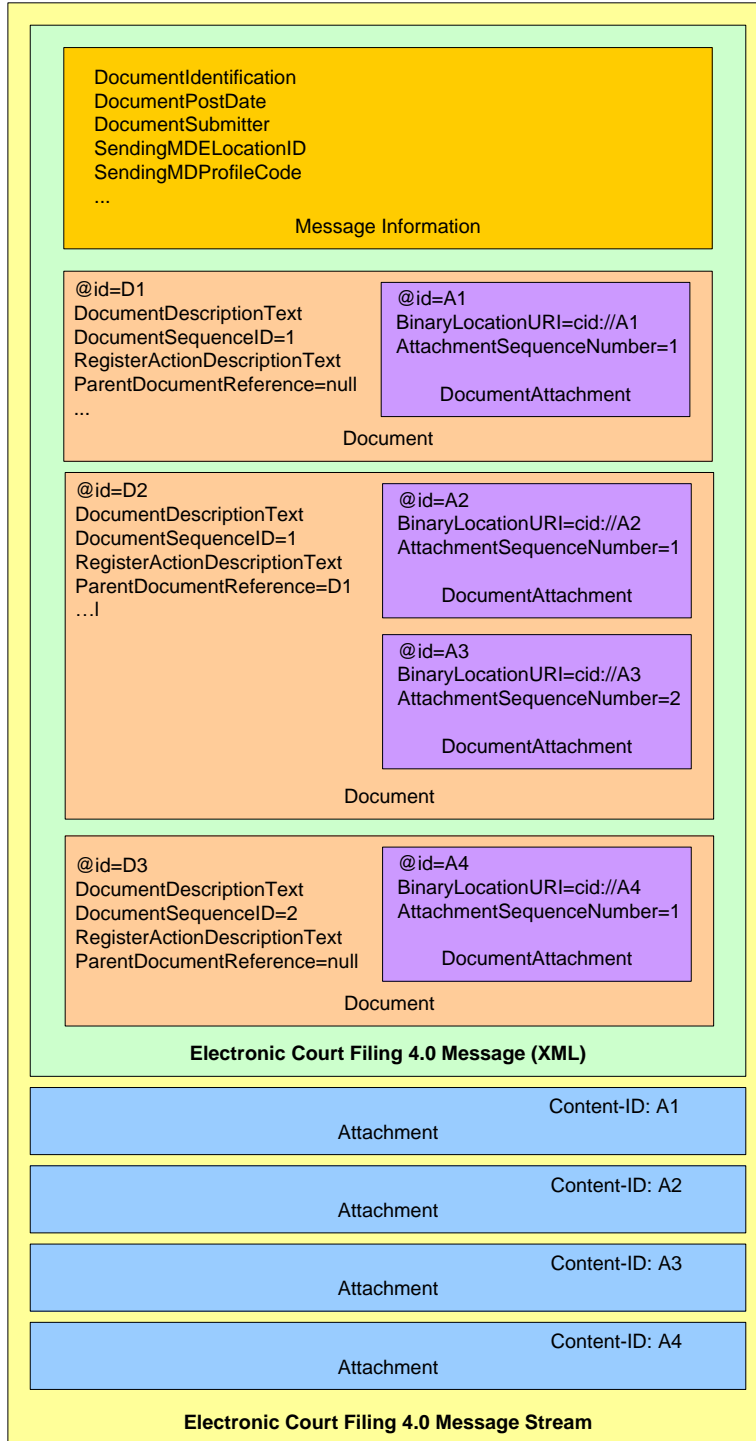


443
444
445

Figure 1. Simple Message Stream

446
 447
 448
 449
 450
 451

Figure 2 illustrates a message stream involving two lead documents, the first of which has a single supporting document. The second lead document has no supporting documents. The supporting document associated with the first lead document is split into two pieces, each treated as an attachment, presumably due to limits set by the court on size. Each lead document is associated with a single attachment, and the one supporting document is associated with two attachments.



452
 453
 454

Figure 2. Message Stream with a Document in Multiple Attachments

455 2.4 Court Policy

456 A court's customary practices may influence many aspects of its ECF 4.0 implementation, and those local
457 practices and variations are expressed through the "court policy" component of e-filing, which includes:

- 458 • **Human-readable court policy** – a textual document publishing the court's rules and requirements for
459 electronic filing.
- 460 • **Machine-readable court policy** – an ECF 4.0 message that describes the features of the ECF 4.0
461 implementation supported by this specification, the court's code lists and any other information a
462 Filing Assembly MDE would need to know in order to successfully submit an electronic filing into that
463 court.

464 The court **MUST** have only one active, authoritative version of its policies at a given time; both the
465 human-readable and the machine-readable statements of those policies **MUST** have the same release
466 dates for the court.

467 The court's human-readable and machine-readable court policies **MUST** each have a version numbering
468 method associated with it. The court's versioning process **SHOULD** comply with the following rules: 1)
469 Versions are denoted using a standard triplet of integers: MAJOR.MINOR.PATCH; 2) Different MAJOR
470 versions are to be considered incompatible, large-scale upgrades of the Policy; 3) Different MINOR
471 versions are to be considered to retain source and binary compatibility with earlier minor versions, and
472 changes in the PATCH level are perfectly compatible, forward and backward. It is important to note that a
473 policy that has not reached version 1.0.0 is not subject to the guidelines described in this document.
474 Before a 1.0 release is achieved (i.e., any version numbered 0.x.y), court policy can be changed freely
475 without regard to the restrictions on compatibility between versions.

476 Court policy is not directly equivalent to "service policy" in the **[SOA-RM]**. However, thinking about court
477 policy from a policy assertion, policy owner and policy enforcement framework as described in the **[SOA-**
478 **RM]** is helpful. Note that "court policy" refers to a set of constituent rules and requirements, while the
479 **[SOA-RM]** looks at each individual item as a "service policy." In all cases the policy owner is the court
480 where the document is to be filed. Also note that none of the elements of court policy rise to the level of a
481 "service contract" as defined by the **[SOA-RM]**.

482 2.4.1 Human-Readable Court Policy

483 To be compliant with the ECF 4.0 specification, each court **MUST** publish a human-readable court policy
484 that **MUST** include each of the following:

- 485 1. The unique court identifier
- 486 2. The location of the machine-readable court policy
- 487 3. A definition of what constitutes a "lead document" in the court
- 488 4. A description of how filer identifiers are to be maintained during electronic communications regarding
489 the case
- 490 5. A description of how the court processes (dockets) filings
- 491 6. A description of any instances in which the court will mandate an element that the ECF 4.0 schema
492 makes optional
- 493 7. A description of any restrictions to data property values other than code list restrictions. (This
494 restriction may be removed in later versions of the ECF specification)
- 495 8. Any other rules required for electronic filing in the court

496 2.4.2 Machine-Readable Court Policy

497 Machine-readable Court Policy includes structures for identifying run-time and development-time policy
498 information.

499 Run-time information includes information that will be updated from time to time, such as code lists (e.g.,
500 acceptable document types, codes for various criminal charges and civil causes of action) and the court's
501 public key for digital signatures and encryption.

502 Development-time information includes court rules governing electronic filing that are needed at the time
503 an application is developed but which are not likely to change. These include:

- 504 1. The service interaction profile(s) that the court supports
- 505 2. The MDEs, query operations and case types supported by the court's ECF 4.0 system
- 506 3. Whether a court will accept the filing of a URL in lieu of the electronic document itself
- 507 4. Whether the court accepts documents requiring payment of a filing fee
- 508 5. Whether the court accepts electronic filing of sealed documents
- 509 6. Whether the court accepts multiple (batch) filings
- 510 7. The court-specific extensions to the ECF 4.0 specification, including the required elements (see
511 below)
- 512 8. The maximum sizes allowed for a single attachment and a complete message stream

513 The machine readable court policy MUST be provided to the Filing Assembly MDE either by the Filing
514 Review MDE through the GetCourtPolicy query or some other means.

515 **2.4.3 Case-Type and Court Extensions**

516 Schemas for initiating specific case types (e.g. criminal, civil) are included in the specification. Case-type
517 and court-specific extensions to the ECF core messages are implemented through the methods
518 described in **[NIEM Techniques]**. The primary extension technique is the use of element substitution, as
519 described in Section 5.3.3 of **[NIEM Techniques]**, in which a more specific element defined in a case-
520 type or court-specific extension is used in place of a generic element in a core message. For instance, a
521 court may add elements required for a particular case type (e.g. civil) by defining an extension schema
522 that includes types (e.g. `court:CivilCaseType`) and elements (e.g., `court:CivilCase`) that
523 substitute for ECF types (e.g. `civil:CivilCaseType`) and elements (e.g., `civil:CivilCase`).
524 Similarly, an implementation may substitute a court-specific code list for a generic code list defined in this
525 specification.

526

527 **2.4.4 Court-Specific Code Lists**

528 Code lists are used to constrain the allowable values for certain information in an ECF 4.0 message. The
529 court SHOULD publish **[Genericcode]** 1.0 code lists for each of the following code lists and reference
530 each of these code lists in its court policy:

531

- 532 • ECF Code Lists
 - 533 • **Civil Case Type**
 - 534 • `<FiduciaryTypeCode>*`
 - 535 • `<JurisdictionalGroundsCode>`
 - 536 • `<ReliefTypeCode>`
 - 537 • **Domestic Case Type**
 - 538 • `<NoContactCode>*`
 - 539 • `<RequestToVacateCode>`
 - 540 • **Common Types**
 - 541 • `<AliasAlternateNameTypeCode>*`

- 542 • [<CaseAssociationTypeCode>*](#)
- 543 • [<CaseOfficialRoleText>*](#)
- 544 • [<CaseParticipantRoleCode>*](#)
- 545 • [<CauseOfActionCode>](#)
- 546 • [<CourtEventTypeCode>](#)
- 547 • [<EntityAssociationTypeCode>](#)
- 548 • [<ErrorCode>*](#)
- 549 • **Juvenile Case Type**
- 550 • [<DelinquentActApplicabilityCode>](#)
- 551 • [<DelinquentActDegreeCode>](#)
- 552 • [<DelinquentActSeverityCode>](#)
- 553 • [<DelinquentActSpecialAllegationCode>](#)
- 554 • [<DependencyAllegationCode>](#)
- 555 • [<GuardianAssociationTypeCode>*](#)
- 556 • [<PlacementTypeCode>](#)
- 557 • **NIEM Code Lists**
- 558 • **JXDM**
- 559 • [<ChargeEnhancingFactorText>](#)
- 560 • [<CourtLocationCode>](#)
- 561 • [<RegisterActionDescriptionText>](#)
- 562 • [<StatuteCodeIdentification>](#)
- 563 • [<StatuteCodeSectionIdentification>](#)
- 564 • [<StatuteOffenseIdentification>](#)
- 565 • [<StatusOffenseCodeIdentification>](#)
- 566 • **NIEM Core**
- 567 • [<BinaryDescriptionText>*](#)
- 568 • [<CaseCategoryText>](#)
- 569 • [<DriverLicenseCommercialClassCode>](#)
- 570 • [<FamilyKinshipCode>*](#)

571

572 A non-normative **[Genericcode]** code list with default values is provided for each of the code lists above
 573 with asterisks (*).

574

575 If a court does not define allowable values for any of the above code lists in court policy, then any value
 576 MUST be considered acceptable for that code.

577

578 2.4.5 Court-Specific Constraint Schemas

579 The cardinality of elements in the NIEM subset imported by the ECF is applied through the use of
 580 constraint schemas that define the minimum and maximum occurrence of elements in the NIEM subset.

581 Courts MAY enforce court-specific rules and code lists by creating court-specific constraint schemas.
582 This process creates a duplicate set of the ECF schemas and allows the customization of the cardinality
583 of elements as needed. If court-specific constraint schemas are used, instance documents MUST
584 validate against both the ECF schemas and the court constraint schemas.

585 3 ECF 4.0 Process Model

586 This section details the interactions of the ECF 4.0 MDEs and the role of each MDE in the electronic filing
587 and electronic service processes. This section also enumerates the operations provided by each MDE
588 and points to the operations, provided by other MDEs, that each MDE consumes.

589 3.1 The Filing-Preparation-to-Docketing Process Model

590 This model describes the sequence of operations in a basic filing cycle from Filing Preparation to
591 Docketing. This model involves three parties: a Filer (represented by the Filing Assembly MDE), a Court
592 (represented by the Filing Review and Court Record MDEs) and a Service Recipient (represented by the
593 Legal Service MDE). The operations defined by ECF 4.0 to support the processes in this cycle are listed
594 below. The operations in bold are required and MUST occur in every successful filing as long as sending
595 and receiving MDEs are implemented. The other operations are optional and MAY occur within a given
596 filing:

- 597 • GetPolicy
- 598 • GetServiceInformation
- 599 • GetFeesCalculation
- 600 • **ReviewFiling**
- 601 • ServeFiling
- 602 • **RecordFiling**
- 603 • **NotifyDocketingComplete**
- 604 • **NotifyFilingReviewComplete**

605 At any point during or after the ReviewFiling operation, if the filing is accessible, a party MAY access
606 information through the following operations:

- 607 • GetFilingList
- 608 • GetFilingStatus

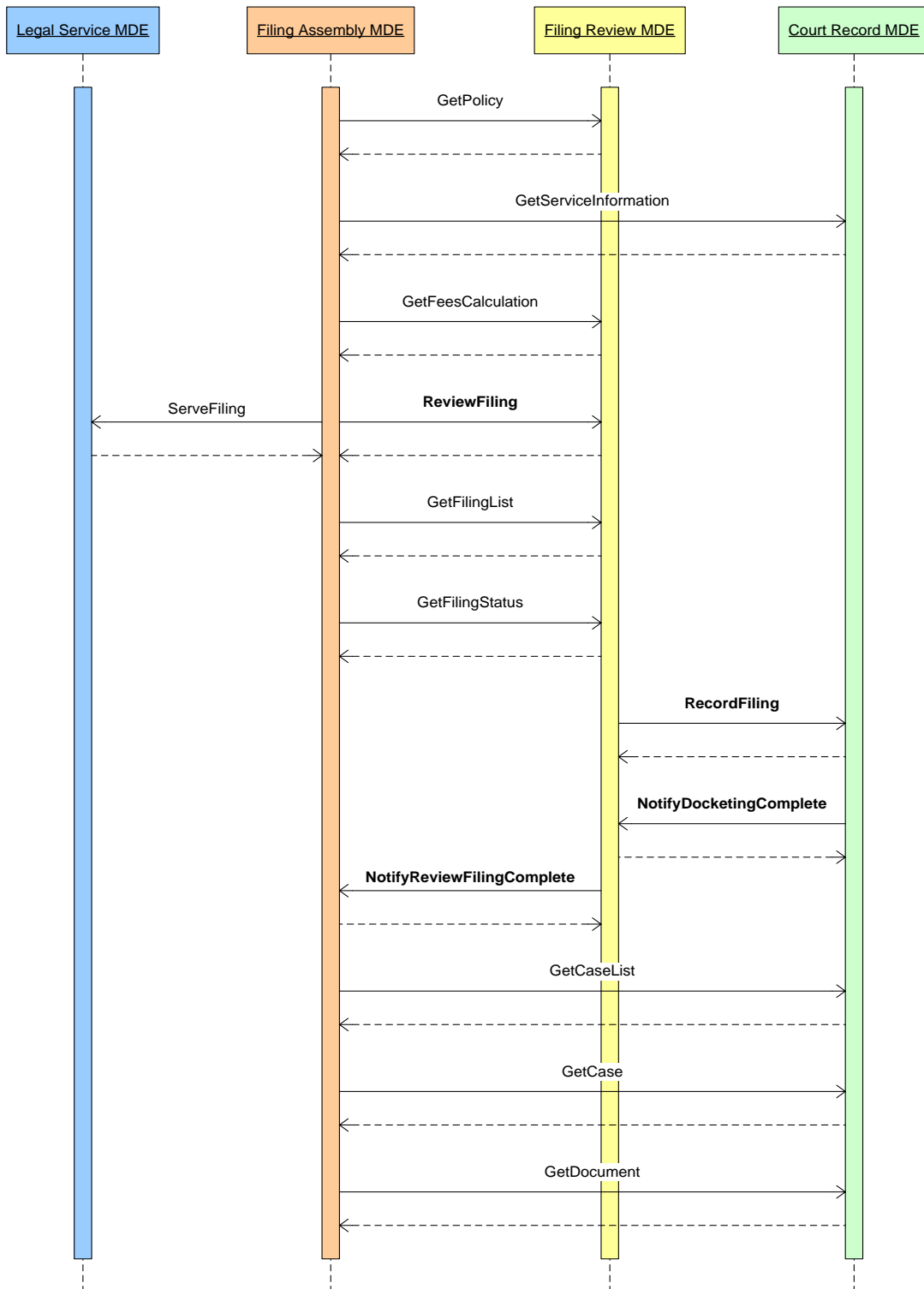
609 At any point after the NotifyFilingReviewComplete operation, if the case is accessible, a party MAY
610 access information through the following operations:

- 611 • GetCaseList
- 612 • GetCase
- 613 • GetDocument

614 These operations are depicted in the sequence diagram below. The solid lines indicate invoked
615 operations and the dashed lines indicate the synchronous responses to those operations.

616

Figure 4. Filing Preparation to Docketing Process Model



620 **3.2 Business Rules**

621 This section describes the business rules of the generic filing-preparation-to-docketing process that
622 govern the ECF 4.0 operations.

623 ECF 4.0 includes an `<ecf:ErrorCode>` element for returning errors in response to a query request.
624 Successful queries MUST return an `<ecf:ErrorCode>` of "0". Failed queries MUST NOT return an
625 `<ecf:ErrorCode>` of "0" and SHOULD return an appropriate `<ecf:ErrorCode>` value as defined in
626 court policy.

627 **3.2.1 GetPolicy**

628 The Filing Assembly MDE MAY obtain a court's machine-readable court policy at any time by invoking the
629 GetPolicy operation on the Filing Review MDE with the identifier for the court. The Filing Review MDE
630 returns the machine-readable court policy in a synchronous response. The content of the machine-
631 readable court policy is described in Section 2.4.2. This step may be omitted if the Filing Assembly MDE
632 already has the current court policy.

633 **3.2.2 GetServiceInformation**

634 The Filing Assembly MDE MAY obtain the Court's service information for all parties in an existing case at
635 any time by invoking the GetServiceInformation operation with the appropriate case number on the Court
636 Record MDE. The service list returned by the GetServiceInformation operation assists the filer in
637 maintaining the filer's service list and is not a substitute for the filer's service list. To provide this
638 information, the Court Record MDE MUST have access to the court's registry with all updated information
639 about case participants. There MUST be only one such registry per court, though multiple courts MAY
640 share the same registry. The Court Record MDE responds synchronously to the Filing Assembly MDE
641 with a service list reflecting the most current contact information available to the court, which is necessary
642 to complete secondary service, whether electronically or by other means.

643 If the court provides a Hub Service MDE, the electronic service information returned from this query
644 MUST include the court's Service MDE ID for all case participants who have one.

645 A party to a case is always the official target of service. In practice, the system will actually deliver to pro
646 se litigants and to attorneys as intermediaries.

647 The duty to complete secondary service is upon the filer, and not the court, except when the court is the
648 filer.

649 The GetServiceInformation operation returns a service list current as of the transaction. No assumption
650 can be made that the data returned by the operation will remain current for use at any future point in time.

651 **3.2.3 GetFeesCalculation**

652 The Filing Assembly MDE MAY query for the fees associated with a filing by invoking the MDE's
653 GetFeesCalculation operation, with a filing as a parameter, on the Filing Review MDE. The Filing Review
654 MDE responds synchronously with the fee calculation and, optionally, a list of the included charges. This
655 step may be omitted if there are no fees associated with filings in the court or the calculated fees are
656 already known.

657 **3.2.4 ReviewFiling**

658 The Filing Assembly MDE MUST submit the filing to the court by invoking the ReviewFiling operation on
659 the Filing Review MDE. The ReviewFiling operation includes messages for the core filing, including the
660 case type-specific and court-specific extensions and the filing payment. The Filing Review MDE
661 responds synchronously with a receipt message that includes the filing identifier issued by the court.

662 **3.2.5 ServeFiling**

663 At approximately the same time the Filing Assembly MDE submits the filing to the court, the Filing
664 Assembly MDE MAY serve the entire filing, to other parties in the case by invoking the ServeFiling
665 operation on the ServiceMDE associated with the service recipient. This operation MUST NOT be used
666 to serve parties in a new case or to persons or organizations that have not yet been made party to the
667 case. The Legal Service MDE responds synchronously with an acknowledgement that the message will
668 be delivered to the service recipient or with an error.

669 If the court hosts a hub Service MDE, the Filing Assembly MDE MAY send a message to the hub Service
670 MDE's ServeFiling operation. The hub Service MDE MUST then broadcast the message to each of the
671 individual Legal Service MDE's ServeFiling operations and respond synchronously with a single
672 ServiceResponseMessage to the Filing Assembly MDE, conveying the results of each individual service
673 transaction.

674 If a court chooses to support electronic service, then each Filing Assembly MDE MUST support service
675 operations for the clients for which it provides Filing Assembly functionality.

676 **3.2.6 RecordFiling**

677 If the clerk reviews and accepts the filing, the Filing Review MDE MUST invoke the RecordFiling
678 operation on the Court Record MDE. The RecordFiling operation includes information from the
679 ReviewFiling operation with any modifications or comments by the clerk. The Court Record MDE
680 responds synchronously with an acknowledgement of the request.

681 **3.2.7 NotifyDocketingComplete**

682 The Court Record MDE MUST invoke the NotifyDocketingComplete operation on the Filing Review MDE
683 as a callback message to the RecordFiling operation to indicate whether the filing was accepted or
684 rejected by the court record system. If the Court Record MDE rejected the filing, an explanation MUST be
685 provided. If the Court Record MDE accepts the filing, the docketing information (e.g. date and time the
686 document was entered into the court record, judge assigned, document identifiers and next court event
687 scheduled) MUST be provided. The Filing Review MDE responds synchronously with an
688 acknowledgement of the callback message.

689 **3.2.8 NotifyFilingReviewComplete**

690 If the clerk rejects the filings or the Filing Review MDE receives the Notify Docketing Complete message,
691 the Filing Review MDE MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly
692 MDE as a callback message to the ReviewFiling operation to indicate whether the filing was accepted
693 and docketed by the clerk and court record system. The operation MAY return the filed documents or
694 links to the documents, but MUST include the **[FIPS 180-2]** SHA 256 document hash, a condensed
695 representation of a document intended to protect document integrity.

696 If the filing included a payment, and the filing was accepted by the clerk and court record system, a
697 receipt for the payment MUST be included in the operation. The Filing Assembly MDE responds
698 synchronously with an acknowledgement of the callback message.

699 **3.2.9 GetFilingList**

700 The Filing Assembly MDE MAY invoke the GetFilingList query operation on the Filing Review MDE to
701 return a list of filings matching several criteria including the filer identifier, the case number and the filed
702 date within a certain time range. The Filing Review MDE responds synchronously with a list of matching
703 filings and the status of each filing.

704 **3.2.10 GetFilingStatus**

705 The Filing Assembly MDE MAY invoke the GetFilingStatus query operation with the filing Identifier on the
706 Filing Review MDE to return the status of the selected filing. The Filing Review MDE responds
707 synchronously with the matching filing and the status of the filing.

708 **3.2.11 GetCaseList**

709 The Filing Assembly MDE MAY invoke the GetCaseList query operation on the Court Record MDE to
710 return a list of cases matching several criteria including case number, case participant, or the filed date
711 over a specific time range. The Court Record MDE responds synchronously with a list of matching cases.

712 **3.2.12 GetCase**

713 The Filing Assembly MDE MAY invoke the GetCase query operation with a case number on the Court
714 Record MDE to return information about the case including the case participants, court docket and
715 calendar events. The Filing Assembly MDE may also limit the amount of case detail returned from the
716 Court Record MDE by using a set of filters. The Court Record MDE responds synchronously with the
717 selected case information.

718 **3.2.13 GetDocument**

719 The Filing Assembly MDE MAY invoke the GetDocument query operation, including the case number and
720 document number, on the Court Record MDE to retrieve a particular document from a case. The Court
721 Record MDE will respond synchronously with the requested document or instructions on how to access it.

722 **3.3 Message Business Rules**

723 Each operation includes one or more messages as parameters. The following business rules apply to the
724 content of ECF 4.0 messages:

725 **3.3.1 Identifiers**

726 Identifiers are used to uniquely label people, organizations and things in the ECF 4.0 process. The
727 following conventions will be used to produce identifiers.

728 **3.3.1.1 Attachment Identifiers**

729 Attachment identifiers MUST be unique within a message transmission. A convention for assigning
730 identifiers to each message and attachment in a message transmission has to be defined in each service
731 interaction profile.

732 **3.3.1.2 Case Identifiers**

733 Case identifiers (case numbers) are assigned by the court record system and MUST be unique within a
734 court.

735 **3.3.1.3 Court Identifiers**

736 Court identifiers are locally assigned by the court administrator for a region (typically a state, provincial or
737 federal court administrator) and MUST be universally unique to a court but not necessarily to a particular
738 court house, branch or subunit of a court. Court identifiers MUST conform to following convention:

739 <Internet domain of the court administrator>:<unique identifier within the court system>.

740 Examples of conformant court identifiers include:

- 741 • courts.wa.gov:superior.king
- 742 • nmcourts.com:albd.civil

- 743 • uscourts.gov:100
744 • courts.gov.bc.ca:appeal
745 These are strictly examples and do not necessarily indicate actual courts.

746 3.3.1.4 Document Identifiers

747 Document identifiers are assigned by the court record system and MUST be unique within a court.

748 3.3.1.5 Filing Identifiers

749 Filing identifiers MUST be unique within a court and will be generated by the court in response to a
750 ReviewFiling operation.

751 3.3.1.6 MDE Identifiers

752 The address of an MDE MUST be unique within a given communications infrastructure. The convention
753 for defining MDE identifiers will be defined in each service interaction profile.

754 3.3.1.7 Filer and Party Identifiers

755 Identifiers for filers and parties to a case, both persons and organizations, MUST be unique within a case
756 and will be generated by the court in response to a ReviewFiling operation. The following is a non-
757 normative example of an identifier for filer number 100:

```
758  
759 <nc:PersonOtherIdentification>  
760     <nc:IdentificationID>100<nc:IdentificationID>  
761     <nc:IdentificationCategoryText>ECFFilerID</nc:IdentificationCategoryTex  
762 t>  
763 </nc:PersonOtherIdentification>
```

764
765 In addition to <nc:PersonOtherIdentification>, other elements that may contain a filer identifier
766 include <nc:OrganizationOtherIdentification>, <ecf:FilingPartyID> and
767 <ecf:FilingAttorneyID>.

768 Attorneys MAY reference the parties they represent with party identifiers. Self-represented litigants MAY
769 be represented using both attorney and party elements for the same individual, with a reference from the
770 attorney element to the party element. The attorney elements for a self-represented litigant SHOULD
771 NOT include a bar number.

772 3.3.2 Code Lists

773 Code Lists are used to constrain the allowable values for certain information in a message. The following
774 normative code lists are normative for all ECF 4.0 implementations. Court-specific code lists are listed in
775 Section 2.4.4.

- 776
777 • ECF Code Lists
- 778 • [Bankruptcy Case Type](#)
 - 779 • <DebtorTypeCode>*
 - 780 • <EstimatedAssetsValueLevelCode>*
 - 781 • <EstimatedDebtsValueLevelCode>*
 - 782 • <NatureOfDebtCode>*

- 783 • [<NumberOfCreditorsValueLevelCode>*](#)
- 784 • [Common Types](#)
- 785 • [<FilingStatusCode>*](#)
- 786 • [Court Policy Response Message](#)
- 787 • [<MajorDesignElementNameCode>](#)
- 788 • [<OperationNameCode>](#)
- 789 • [Service Receipt Message](#)
- 790 • [<ServiceStatusCode>*](#)
- 791 • [NIEM Code Lists](#)
- 792 • [ANSI NIST](#)
- 793 • [<FingerPositionCode>](#)
- 794 • [JXDM](#)
- 795 • [<ChargeNCICCode>](#)
- 796 • [<DrivingIncidentHazMatCode>](#)
- 797 • [<DrivingJurisdictionAuthorityNCICLSTACode>](#)
- 798 • [<IdentificationJurisdictionNCICLISCode>](#)
- 799 • [<WarrantExtraditionLimitationCode>](#)
- 800 • [NIEM Core](#)
- 801 • [<DocumentLangageCode>](#)
- 802 • [<DriverLicenseCommercialClassCode>](#)
- 803 • [<DrivingRestrictionCode>](#)
- 804 • [<LanguageCode>](#)
- 805 • [<LengthUnitCode>](#)
- 806 • [<LocationCountryFIPS10-4Code>](#)
- 807 • [<LocationCountyCode>](#)
- 808 • [<LocationStateUSPostalServiceCode>](#)
- 809 • [<PersonCitizenshipFIPS10-4Code>](#)
- 810 • [<PersonEthnicityCode>](#)
- 811 • [<PersonEyeColorCode>](#)
- 812 • [<PersonHairColorCode>](#)
- 813 • [<PersonRaceCode>](#)
- 814 • [<PersonSexCode>](#)
- 815 • [<PersonUnionCategoryCode>](#)
- 816 • [<PhysicalFeatureCategoryCode>](#)
- 817 • [<VehicleColorPrimaryCode>](#)
- 818 • [<VehicleMakeCode>](#)
- 819 • [<VehicleModelCode>](#)
- 820 • [<VehicleStyleCode>](#)

821 • <WeightUnitCode>

822

823 Code lists defined using **[Genericcode]** 1.0 are indicated with asterisks (*). The remaining code lists are
824 defined in XSD schema definitions.

825

826 **3.3.3 Message-Specific Business Rules**

827 The following business rules apply to specific messages:

828 **3.3.3.1 CoreFilingMessage**

829 A CoreFilingMessage **MUST** express the name or names of the party or parties on whose behalf a
830 document is filed, and the party whose document is the subject of a responsive document being
831 submitted for filing. If a case refers to a single element using the legal term “In Re,” the filer **SHOULD** use
832 the NIEM <j:CaseRespondentParty>, **not** the <j:CaseInitiatingParty> element.

833 A CoreFilingMessage **MAY NOT** include documents for transactions such as the payment of a criminal
834 fine. If a CoreFilingMessage includes documents, the message **MUST** include only one level of
835 connected and supporting documents. If a CoreFilingMessage includes multiple renditions of the same
836 document, the <nc:BinaryDescriptionText> element **SHOULD** be used to determine how to
837 process multiple renditions of the same document. The <ecf:DocumentMetadata> and
838 <ecf:DocumentRenditionMetadata> structures **MAY** be extended to support more sophisticated
839 workflow processes.

840 **3.3.3.2 FilingPaymentMessage**

841 ECF 4.0 supports multiple particular payment processes. Information about a payment is included in the
842 FilingPaymentMessage including the method of payment of the applicable fees, e.g., electronic funds
843 transfer, credit or debit card, charge to an escrow account held in the court or promise to pay in the
844 future. The payment **MAY** include a maximum amount for the payment if some latitude is needed to
845 accomplish the filing.

846 **3.3.3.3 RecordDocketingMessage**

847 The court record system **SHOULD** retain all complete message transmissions, including any message
848 envelopes and headers defined by the service interaction profile, for evidentiary purposes.

849 **3.4 Filing the Record on Appeal**

850 This section describes the process for filing and subsequently amending the Record on Appeal (ROA)
851 using ECF 4.0.

852

853 • All ROA transactions, either the original filing or subsequent amendments, **MUST** contain, as the
854 lead document, an Index of Record document that itemizes the content of the record on appeal.³

855

856 • The documents that comprise the ROA transaction will be identified as supporting documents.

857

858 • The supporting documents that comprise the ROA transaction **MAY** also have additional attached
859 documents.

³ There are no set requirements for the structure or content of the Index of Record document

- 860
- 861
- 862
- 863
- 864
- 865
- 866
- 867
- 868
- 869
- 870
- 871
- 872
- 873
- 874
- 875
- 876
- 877
- 878
- 879
- 880
- 881
- 882
- 883
- 884
- 885
- 886
- 887
- 888
- 889
- 890
- All ROA documents being submitted, including the Index of Record document and each document within the record, MUST have at least one court-defined document type that indicates the type of transaction to be performed on the document, and whether the document is being added to or stricken from the record.
 - The Index of Record document and each document within the ROA transaction MAY also have an additional document type or types, which characterize the document for the Court Record MDE.
 - When a document within the ROA transaction is being stricken from the court record, the document MUST be identified by the unique document identifier, which was provided by the Court Record MDE when the document was initially filed (See section 3.3.1.4).
 - A hierarchical structure of case lineage elements MUST be used to express the target case's predecessor cases at prior courts. Each predecessor case MAY also have its own predecessor case, as necessary to express the full lineage of an appellate case.⁴
 - When the ROA transaction is electronically transferred from one court to another, the target case number in the destination court and the case lineage, which includes the predecessor case number in the sending court, MUST be provided.
 - If the ROA transaction is a case initiating filing in the destination court, then the FilingCase object MUST be present and the CaseTrackingID MUST be absent.
 - Each predecessor case identified in the target case's case lineage may include case type-specific and court-specific extensions. The case type and the case type-specific extensions for each predecessor case MUST be consistent throughout the case lineage.
 - When a ROA amendment transaction is sent, the Index of Record document MUST reflect the status of the record assuming that the transaction will be accepted. If however the transaction is

⁴ Explanation (non-normative): There is not always a one to one correspondence between a lower court case (i.e. a trial court case) and the target appellate case. A single trial court case could have multiple descendent cases, and a single appellate case can have multiple predecessors. In the situation where an appellate case has multiple predecessor cases, each predecessor case will send a record on appeal to the target court for the appellate case. Each individual record will have an independent index of record. The warning above against sending multiple ROA transactions while a prior transaction is still pending must be regarded in light of the record to which the transaction is intended (or if you prefer, the predecessor case from which it originates). For example, let's say an appellate case has two predecessor cases, case A and case B. If an ROA transaction for the record from case A is pending (awaiting acceptance or rejection), this will not have any potential adverse impact on an ROA transaction from case B. Similarly, if a single lower court case were on appeal in two different appellate cases (say case Y and case Z), then while an ROA transaction targeted to case Y is pending, there is no potential adverse impact to case Z receiving an ROA transaction (assuming of course that case Z does not also have a pending ROA transaction from the same predecessor case).

891 rejected, there will be ramifications for other pending amendment transactions for the same ROA
892 in the same target case.⁵

893

894 • While an ROA transaction is awaiting acceptance or rejection in the destination court, and when
895 the target case consists of multiple records, courts SHOULD NOT send additional amendment
896 transactions intended for the same record for the same target case.

897

898 • Individual documents within the ROA transaction MUST not be individually accepted or rejected.
899 All documents within the ROA transaction MUST have the same acceptance or rejection
900 disposition.

901

⁵ While an ROA transaction is awaiting acceptance or rejection in the destination court, courts are cautioned against, but not prohibited from, sending additional amendment transactions for the same record in the same target case, regardless of whether the case contains one or many records.

902 4 ECF 4.0 Schemas

903 The Court Filing XSD schemas are implementations of the ECF 4.0 exchange content models (see
904 Appendix B.3 below). They are the only normative representations of ECF 4.0 messages.

905 All of the ECF 4.0 XSD schemas are contained in the `xsd/` subdirectory of the ECF 4.0 release package
906 (see Appendix A for more information regarding the structure of the release package). The `xsd/`
907 directory is further subdivided into the `xsd/casetype/`, `xsd/common/`, `xsd/constraint/`,
908 `xsd/message/`, and `xsd/Subset/` subdirectories.

909

910 4.1 ECF 4.0 Case Type Schemas

911 The XSD schemas that define extensions specific to certain ECF 4.0 case types are included in the
912 `xsd/casetype/` directory, as listed below:

913

914 **AppellateCase**

915 [xsd/casetype/ECF-4.0-AppellateCase.xsd](#)

916 **BankruptcyCase**

917 [xsd/casetype/ECF-4.0-BankruptcyCase.xsd](#)

918 **CitationCase**

919 [xsd/casetype/ECF-4.0-CitationCase.xsd](#)

920 **CivilCase**

921 [xsd/casetype/ECF-4.0-CivilCase.xsd](#)

922 **CriminalCase**

923 [xsd/casetype/ECF-4.0-CriminalCase.xsd](#)

924 **DomesticCase**

925 [xsd/casetype/ECF-4.0-DomesticCase.xsd](#)

926 **JuvenileCase**

927 [xsd/casetype/ECF-4.0-JuvenileCase.xsd](#)

928

929 4.2 ECF 4.0 Common Schemas

930 The XSD schemas that define the generic elements and types that are common to multiple ECF 4.0
931 messages and/or case types are located in the `xsd/common/` folder, as listed below:

932

933 **AppInfo**

934 [xsd/common/ECF-4.0-AppInfo.xsd](#)

935 **CommonTypes**

936 [xsd/common/ECF-4.0-CommonTypes.xsd](#)

937 **DigitalSignature**

938 [xsd/common/xmlsig-core-schema.xsd](#)

939 **Genericcode**

940 [xsd/common/genericcode.xsd](#)

941 **4.3 ECF 4.0 Constraint and Subset Schemas**

942 The XSD schemas that define the subset of all NIEM elements and types that are used in ECF 4.0
943 messages and/or case type extensions are located in the `xsd/Subset/niem/` folder. As a general
944 data model, NIEM does not define any constraints regarding the minimum and maximum occurrence of
945 elements contained within types. Therefore, in conformance with NIEM, ECF-specific constraints are not
946 included in the schemas within the `xsd/Subset/niem` folder. The XSD schemas in the
947 `xsd/constraint/niem/` folder represent the NIEM subset schemas with the ECF-specific constraints
948 applied and are the schemas by which the ECF message and case type schemas incorporate NIEM
949 elements and types.

950 **4.4 ECF 4.0 Message Schemas**

951 The XSD schemas defining the messages that support the ECF 4.0 processes are located in the
952 `xsd/messages/` folder, as listed below:

953

954 **CaseListQueryMessage**

955 [xsd/message/ECF-4.0-CaseListQueryMessage.xsd](#)

956 **CaseListResponseMessage**

957 [xsd/message/ECF-4.0-CaseListResponseMessage.xsd](#)

958 **CaseQueryMessage**

959 [xsd/message/ECF-4.0-CaseQueryMessage.xsd](#)

960 **CaseResponseMessage**

961 [xsd/message/ECF-4.0-CaseResponseMessage.xsd](#)

962 **CoreFilingMessage**

963 [xsd/message/ECF-4.0-CoreFilingMessage.xsd](#)

964 **CourtPolicyQueryMessage**

965 [xsd/message/ECF-4.0-CourtPolicyQueryMessage.xsd](#)

966 **CourtPolicyResponseMessage**

967 [xsd/message/ECF-4.0-CourtPolicyResponseMessage.xsd](#)

968 **DocumentQueryMessage**

969 [xsd/message/ECF-4.0-DocumentQueryMessage.xsd](#)

970 **DocumentResponseMessage**

971 [xsd/message/ECF-4.0-DocumentResponseMessage.xsd](#)

972 **FeesCalculationQueryMessage**

973 [xsd/message/ECF-4.0-FeesCalculationQueryMessage.xsd](#)

974 **FeesCalculationResponseMessage**

975 [xsd/message/ECF-4.0-FeesCalculationResponseMessage.xsd](#)

976 **FilingListQueryMessage**

977 [xsd/message/ECF-4.0-FilingListQueryMessage.xsd](#)

978 **FilingListResponseMessage**

979 [xsd/message/ECF-4.0-FilingListResponseMessage.xsd](#)

980 **FilingStatusQueryMessage**

981 [xsd/message/ECF-4.0-FilingStatusQueryMessage.xsd](#)

982 **FilingStatusResponseMessage**

983 [xsd/message/ECF-4.0-FilingStatusResponseMessage.xsd](#)
984 **MessageReceiptMessage**
985 [xsd/message/ECF-4.0-MessageReceiptMessage.xsd](#)
986 **PaymentMessage**
987 [xsd/message/ECF-4.0-PaymentMessage.xsd](#)
988 **PaymentReceiptMessage**
989 [xsd/message/ECF-4.0-PaymentReceiptMessage.xsd](#)
990 **RecordDocketingCallbackMessage**
991 [xsd/message/ECF-4.0-RecordDocketingCallbackMessage.xsd](#)
992 **RecordDocketingMessage**
993 [xsd/message/ECF-4.0-RecordDocketingMessage.xsd](#)
994 **ReviewFilingCallbackMessage**
995 [xsd/message/ECF-4.0-ReviewFilingCallbackMessage.xsd](#)
996 **ServiceInformationQueryMessage**
997 [xsd/message/ECF-4.0-ServiceInformationQueryMessage.xsd](#)
998 **ServiceInformationResponseMessage**
999 [xsd/message/ECF-4.0-ServiceInformationResponseMessage.xsd](#)
1000 **ServiceReceiptMessage**
1001 [xsd/message/ECF-4.0-ServiceReceiptMessage.xsd](#)
1002

5 Service Interaction Profiles

1003

1004 An ECF 4.0 service interaction profile defines a transmission system that supports the functional
1005 requirements of electronic filing, along with the MDE operations and message structures, and implements
1006 certain non-functional requirements. A service interaction profile does not govern the content of
1007 messages – message content is described in Sections 2 and 3 of this specification. A service interaction
1008 profile will define how a message gets from the sending MDE to the receiving MDE in a given messaging
1009 framework.

5.1 Service Interaction Profile Requirements

1010
1011 Each service interaction profile will define standard conventions and configuration details to support
1012 interoperability between and among ECF 4.0 implementations that support the same service interaction
1013 profile. However, compliance with these requirements will not necessarily guarantee interoperability.

1014 To be compliant with the ECF 4.0 specification, a service interaction profile **MUST** satisfy the following
1015 non-functional requirements:

- 1016 1. **Transport protocol** – A service interaction profile **MUST** define how messages are physically
1017 transported from a sending MDE to a receiving MDE. In so doing, a profile may identify factors that
1018 restrict the range of environments in which the profile is applicable.
- 1019 2. **MDE addressing** – A service interaction profile **MUST** include a convention for uniquely addressing
1020 each MDE.
- 1021 3. **Operation addressing** – A service interaction profile **MUST** describe a convention for uniquely
1022 addressing each MDE operation.
- 1023 4. **Request and operation invocation** – A service interaction profile **MUST** describe a mechanism for a
1024 sending MDE to invoke an operation on the receiving MDE.
- 1025 5. **Synchronous mode response** – A service interaction profile **MUST** support synchronous operations
1026 in which the response to an operation is always returned immediately, typically within a matter of
1027 seconds, to the invoking MDE.
- 1028 6. **Asynchronous mode response** – A service interaction profile **MUST** support asynchronous
1029 operations in which the response to an operation may not necessarily be returned immediately to the
1030 invoking MDE. Instead, the response may be returned at some later time through a callback from the
1031 MDE that received the operations to the invoking MDE. The callback **MUST** include a reference to
1032 the invoking message transmission.
- 1033 7. **Message/attachment delimiters** – A service interaction profile **MUST** define how the receiving MDE
1034 distinguishes messages from attachments within a message transmission.
- 1035 8. **Message identifiers** – A service interaction profile **MUST** provide a means for a sending MDE to
1036 assign a unique identifier to each message (including any attachments) within a message
1037 transmission.

1038 In addition, there are some non-functional features that a service interaction profile **SHOULD** provide,
1039 including:

- 1040 1. **Message non-repudiation** – A service interaction profile **SHOULD** provide a mechanism so that the
1041 receiving MDE is provided with evidence that demonstrates:
 - 1042 a. the identity of the sending MDE
 - 1043 b. the content of the message(s) transmitted
 - 1044 c. the date and time of the message transmission
- 1045 2. **Message integrity** – A service interaction profile **SHOULD** provide a mechanism so that the
1046 receiving MDE is able to determine whether the message(s) transmitted (including any attachments)
1047 was (were) modified during the message transmission.

- 1048 3. **Message confidentiality** – A service interaction profile SHOULD provide a mechanism, such as
1049 encryption, that can be used with a sending MDE to ensure that the message(s) in a transmission
1050 (including any attachments) can be processed only by the receiving MDE.
- 1051 4. **Message authentication** – A service interaction profile SHOULD provide a mechanism, such that a
1052 sending MDE is required to include, to display credentials that demonstrate its identity to the receiving
1053 MDE in each message transmission.
- 1054 5. **Message transmission reliability** – A service interaction profile SHOULD provide a mechanism,
1055 such that a sending MDE is required to include, to guarantee that a message transmission will be
1056 delivered to the receiving MDE within a specified period of time, or else the sending MDE will receive
1057 notification at the end of that period of time that the message transmission was not deliverable to the
1058 receiving MDE.
- 1059 6. **Message splitting and assembly** – A service interaction profile SHOULD provide a mechanism by
1060 which a large message and attachments MAY be split into multiple pieces that are transmitted
1061 separately by the sending MDE and reassembled into the complete message by the receiving MDE.
1062 In the HTTP 1.1 protocol, this is called “chunking.”
- 1063 7. **Transmission auditing** – A service interaction profile SHOULD provide a mechanism for the MDE to
1064 receive message transmissions in their entirety (both messaging and “payload” content) for auditing
1065 purposes.

1066 5.2 Service Interaction Profile Approval and Revision Processes

1067 The ECF Technical Committee (TC) will recommend certain service interaction profiles for use in
1068 implementations of the ECF 4.0 specification. The TC will consider a service interaction profile for
1069 recommendation for use in ECF 4.0 implementations provided the profile meets the following
1070 requirements:

- 1071 1. The service interaction profile MUST be described in a document in the format of an OASIS
1072 specification.
- 1073 2. The service interaction profile specification MUST identify a unique URI to identify the service
1074 interaction profile and version.
- 1075 3. The service interaction profile specification MUST describe the binding of MDE operations to the
1076 service interaction profile that satisfies the functional requirements described in Section 3 (“ECF 4.0
1077 Process Model”) and Section 4 (“ECF 4.0 Schema”) of this specification.
- 1078 4. The service interaction profile specification MUST demonstrate that the service interaction profile
1079 satisfies the non-functional service interaction profile requirements described in Section 5.1 (“Service
1080 Interaction Profile Requirements”) of this specification.
- 1081 5. The service interaction profile specification MUST include samples that demonstrate how the
1082 messaging information and “payload” content are combined into message transmissions. These
1083 samples MUST include samples that demonstrate both synchronous and asynchronous mode
1084 operations.
- 1085 6. At least one voting member of the ECF TC MUST agree to sponsor the service interaction profile and
1086 submit the service interaction profile specification to the TC for review as a candidate for approval as
1087 an ECF 4.0 compliant service interaction profile.

1088 Certifying that a candidate service interaction profile meets certain service interaction profile requirements
1089 will necessarily involve some subjectivity since service interaction profile requirements cannot be
1090 expressed algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to assess
1091 whether the proposed profile’s description is adequate in meeting the requirements of ECF 4.0 before
1092 approving the service interaction profile specification as a “Committee Draft” through the OASIS
1093 standards approval process.

1094 From time to time, it may be necessary to revise or update a service interaction profile to bring it into
1095 compliance with changes in network and messaging protocols, or to support additional non-functional
1096 requirements. Any revision(s) to previously approved service interaction profiles will be considered a new
1097 service interaction profile and MUST meet the requirements of a new service interaction profile, including

1098 sponsorship by a voting member of the ECF TC and review and approval by the ECF TC. There will be
1099 no guarantees that future versions of a service interaction profile will be backwardly compatible with the
1100 current version.

1101 **5.3 Supported Service Interaction Profiles**

1102 The following ECF 4.0 service interaction profile specifications are for use in conjunction with
1103 implementations of the ECF 4.0 specification:

- 1104 • **Web Services Service Interaction Profile 2.0 Specification** – This specification defines a
1105 transmission system using the specifications described in the Web Services Interoperability (WS-I)
1106 Basic Profile 1.1, W3C SOAP 1.1 Binding for MTOM 1.0, WS-I Basic Security Profile 1.0 and OASIS
1107 WS-Reliable Messaging 1.1.
- 1108 • **Web Services Service Interaction Profile 2.1 Specification** – This specification defines a
1109 transmission system using the specifications described in the Web Services Interoperability (WS-I)
1110 Basic Profile 1.1, W3C SOAP 1.1 Binding for MTOM 1.0 and WS-I Basic Security Profile 1.1 and
1111 OASIS WS-Reliable Messaging 1.1.
- 1112 • **Portable Media Service Interaction Profile 1.01 Specification** – This specification defines a
1113 transmission system in which the sending MDE stores message transmissions on portable media
1114 (e.g., a compact disc), which is then physically transported to the receiving MDE where it is
1115 connected for retrieval of the message transmissions. This specification may be needed in the
1116 absence of an active network between the sending and receiving MDEs.

1117 Additional service interaction profiles, or revisions to these service interaction profiles, may be approved
1118 by the ECF TC for use in conjunction with implementations of the ECF 4.0 specification according to the
1119 process described in Section 5.2 (“Service Interaction Profile Approval and Revision Processes”) above.

1120 6 Document Signature Profiles

1121 An ~~ECF 4.0~~ ECF document signature profile defines a mechanism for asserting that a person signed a
1122 single electronic or imaged document, which is an attachment to a message transmission. The signing of
1123 an entire message transmission is described in a service interaction profile and is not supported by a
1124 document signature profile.

1125 6.1 Document Signature Profile Requirements

1126 Each document signature profile will define standard conventions and configuration details to support
1127 interoperability in the creation and verification of document signatures between and among ~~ECF 4.0~~ ECF
1128 implementations that support the same document signature profile. However, compliance with these
1129 requirements will not necessarily guarantee interoperability.

1130 Except for the Null Document Signature Profile, to be compliant with the ECF 4.0 specification, a
1131 document signature profile MUST satisfy the following non-functional requirements:

- 1132 1. **Signer name assertion** – A document signature profile MUST make an assertion regarding the
1133 name of the person who signed a document.
- 1134 2. **Signed date assertion** – A document signature profile MUST make an assertion regarding the date
1135 the person signed a document.
- 1136 3. **Multiple signatures** – A document signature profile MUST allow multiple signatures to be associated
1137 with the same document.

1138 A signature profile SHOULD provide the following non-functional features:

- 1139 1. **Signer and date non-repudiation** – A document signature profile SHOULD provide a mechanism so
1140 that the receiving MDE is provided with verifiable evidence that demonstrates:
 - 1141 a. the unique identity of the person who signed the document
 - 1142 b. the date the person signed a document
- 1143 2. **Document integrity** – A document signature profile SHOULD provide a mechanism so that the
1144 receiving MDE is able to determine if the document was modified since the person signed the
1145 document.
- 1146 3. **Document signature auditing** – A document signature profile SHOULD provide a mechanism for
1147 the MDE to receive both the document and signatures for auditing purposes.

1148 6.2 Document Signature Profile Approval and Revision Processes

1149 The ECF Technical Committee will recommend certain document signature profiles for use in
1150 implementations of the ECF 4.0 specification. The TC will consider a document signature profile for
1151 recommendation for use in ECF 4.0 implementations provided the profile meets the following
1152 requirements:

- 1153 1. The document signature profile MUST be described in a document in the format of an OASIS
1154 specification.
- 1155 2. The document signature profile specification MUST identify a unique URI to identify the document
1156 signature profile and version.
- 1157 3. If the document signature is not embedded in the document, the document signature profile
1158 specification MUST include an XML structure for describing precisely how the document signature is
1159 represented.
- 1160 4. The document signature profile specification MUST demonstrate that the document signature profile
1161 satisfies the non-functional requirements described in Section 6.1 (“Document Signature Profile
1162 Requirements”) of this specification.

- 1163 5. The document signature profile specification MUST include samples that demonstrate how the
1164 document signature information and “payload” content are combined into message transmissions.
- 1165 6. At least one voting member of the ECF TC MUST agree to sponsor the document signature profile
1166 and submit the document signature profile specification to the TC for review as a candidate for
1167 approval as an ~~ECF 4.0~~ ECF document signature profile.

1168 Certifying that a candidate document signature profile meets certain document signature profile
1169 requirements will necessarily involve some subjectivity, since document signature profile requirements
1170 cannot be expressed algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to
1171 assess whether the proposed profile’s description is adequate to the requirements before approving the
1172 profile specification as a Committee Draft through the OASIS standards approval process.

1173 From time to time, it may be necessary to revise or update a document signature profile to bring it into
1174 compliance with changes in authentication and encryption protocols, or to support additional non-
1175 functional requirements. Any revision(s) to previously approved document signature profiles will be
1176 considered a new document signature profile and MUST meet the requirements of a new document
1177 signature profile, including sponsorship by a voting member of the ECF TC and review and approval by
1178 the ECF TC. There will be no guarantees that future versions of document signature profiles will be
1179 backwardly compatible with the current version.

1180 6.3 Supported Document Signature Profiles

1181 The following ~~ECF 4.0~~ ECF document signature profile specifications are candidate Committee Drafts for
1182 use in conjunction with implementations of the ECF 4.0 specification:

- 1183 • **Null Document Signature Profile 1.0 Specification** – This specification defines a default
1184 mechanism to describe documents that do not have any associated signatures.
- 1185 • **XML Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1186 associating a W3C XML Signature with a document.
- 1187 • **Application-Specific Document Signature Profile 1.0 Specification** – This specification defines a
1188 mechanism for embedding an application-specific binary signature with a document. This profile
1189 supports the native capabilities in document formats such as Microsoft Word and the Adobe Portable
1190 Document Format (PDF) for describing and embedding signatures.
- 1191 • **Proxy Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1192 indicating documents that are digitally signed by a court filing infrastructure component on behalf of
1193 an authenticated signer.
- 1194 • **Symmetric Key Document Signature Profile 1.0 Specification** – This specification defines a
1195 mechanism for indicating documents that are digitally signed by a trusted entity on behalf of the
1196 signer using a symmetric key known only to the trusted entity.

1197 Additional document signature profiles, or revisions to these document signatures profiles, may be
1198 approved by the ECF TC for use in conjunction with implementation of the ECF 4.0 specification
1199 according to the process described in Section 6.2 (“Document Signature Profile Approval and Revision
1200 Processes”) above.

1201 **7 Conformance**

1202 *An implementation conforms with the Electronic Court Filing Version 4.01 if the implementation meets the*
1203 *requirements in Sections 1-6 including conformance with the XSD schemas and [Genericcode] code lists*
1204 *referenced in Section 3 and 4.*

1205 Appendix A. (Informative) Release Notes

1206 A.1 Availability

1207 Online and downloadable versions of this release are available from the locations specified at the top of
1208 this document.

1209 A.2 Package Structure

1210 The ECF 4.0 specification is published as a ZIP archive named `ecf-v4.0.zip`. Unzipping this archive
1211 creates a directory named `ecf-4.0/` containing this specification document and a number of
1212 subdirectories. The files in these subdirectories, linked to the specification document, contain the various
1213 normative and informational pieces of the 1.0 release. A description of each subdirectory is given below.

1214 `gc/`

1215 **[Genericcode]** 1.0 code lists

1216 `model/`

1217 ECF 4.0 UML exchange content model diagrams and spreadsheet models; see Appendix B.3 and

1218 B4

1219 `xml/`

1220 Example instances; see Appendix D

1221 `xsd/`

1222 XSD schemas; see Section 4

1223 A.3 Recursive Structures

1224 Certain components in the **[NIEM]** version 2.0 schemas allow recursive nesting. For example, a
1225 `nc:Case` may be related to another `nc:Case`, etc. These are legitimate business data structures. Most
1226 real-world applications will limit the depth of recursion in such structures, but XSD schemas are incapable
1227 of expressing this constraint. Implementers should be aware of this and may wish to set limits on the
1228 depth of recursive structures in their applications.

1229 A.4 Date and Time Formats

1230 The date and time elements contained in the messages defined by the ECF 4.0 XSD schemas should be
1231 formatted according to the documentation in the **[NIEM]** version 2.0. The **[NIEM]** documentation
1232 indicates the following:

- 1233 • Calendar date values should be expressed as “CCYY-MM-DD”, with an optional time zone qualifier
1234 designated by appending `-hh:00`, where `hh` represent the number of hours the local time zone is
1235 behind Coordinated Universal Time (UTC).
- 1236 • Time values should be expressed as “hh:mm:ss.sss”, with an optional time zone qualifier designated
1237 by appending `-hh:00`, where `hh` represent the number of hours the local time zone is behind
1238 Coordinated Universal Time (UTC).
- 1239 • Date and time values should be expressed as “CCYY-MM-DDThh:mm:ss.sss” with an optional time
1240 zone designated by appending `-hh:00`, where `hh` represent the number of hours the local time zone is
1241 behind Coordinated Universal Time (UTC).qualifier.

1242 These formats are documented in, but not enforced by, the XSD schema at
1243 `xsd/constraint/niem/proxy/xsd/2.0/xsd.xsd`.

1244 **A.5 Known Errata**

1245 Known errors in the ECF 4.0 specification will be identified in an errata document available at:

1246 <http://www.oasis-open.org/committees/legalxml-courtfilling/>.

1247
1248

Appendix B. (Informative) ECF 4.0 Development Approach and Artifacts

1249 This appendix describes the approach used to develop ECF 4.0 and the modeling artifacts.

1250 B.1 Principles

1251 The key principles that guided the design of the ECF 4.0 message structures were:

- 1252 • **Interoperability** – The ECF 4.0 message structures should provide a means for exchanging court
1253 filings among all types of court information systems.
- 1254 • **Completeness** – The ECF Filing 4.0 message structures format should provide for all the elements
1255 of an electronic filing system.
- 1256 • **Simple implementation** – The design should foster rapid implementation.
- 1257 • **Simple XML and portable structure** – The core messages in an ECF 4.0 exchange will be
1258 formatted as XML documents.
- 1259 • **Familiarity** – The data elements and code values should be meaningful to the legal community and
1260 non-expert recipients alike.
- 1261 • **Interdisciplinary and international utility** – The design should be usable by a broad range of court-
1262 related applications and should be applicable internationally.

1263 B.2 Approach

1264 The ECF 4.0 message schemas were developed as a **[NIEM]** Information Exchange Package Definition
1265 (IEPD). A **[NIEM IEPD]** is a collection of artifacts that describe the structure and content of a set of data
1266 that is transmitted for a specific business purpose. It does not specify other interface layers (such as Web
1267 services).

1268 The NIEM Naming and Design Rules (MNDR) **[NIEM NDR]** describe best practices for the development
1269 of NIEM-conformant Information Exchange Packages and documentation. The Design Rules set forth:

- 1270 • A methodology for the construction of **[NIEM]**-conformant exchange documents
- 1271 • Naming and design rules for the artifacts called for by the methodology
- 1272 • Guidelines for the customization of **[NIEM]** schema structures

1273 B.3 ECF 4.0 Exchange Content Models

1274 The ECF 4.0 exchange content models describe the information components used in all of the messages
1275 defined by ECF 4.0.

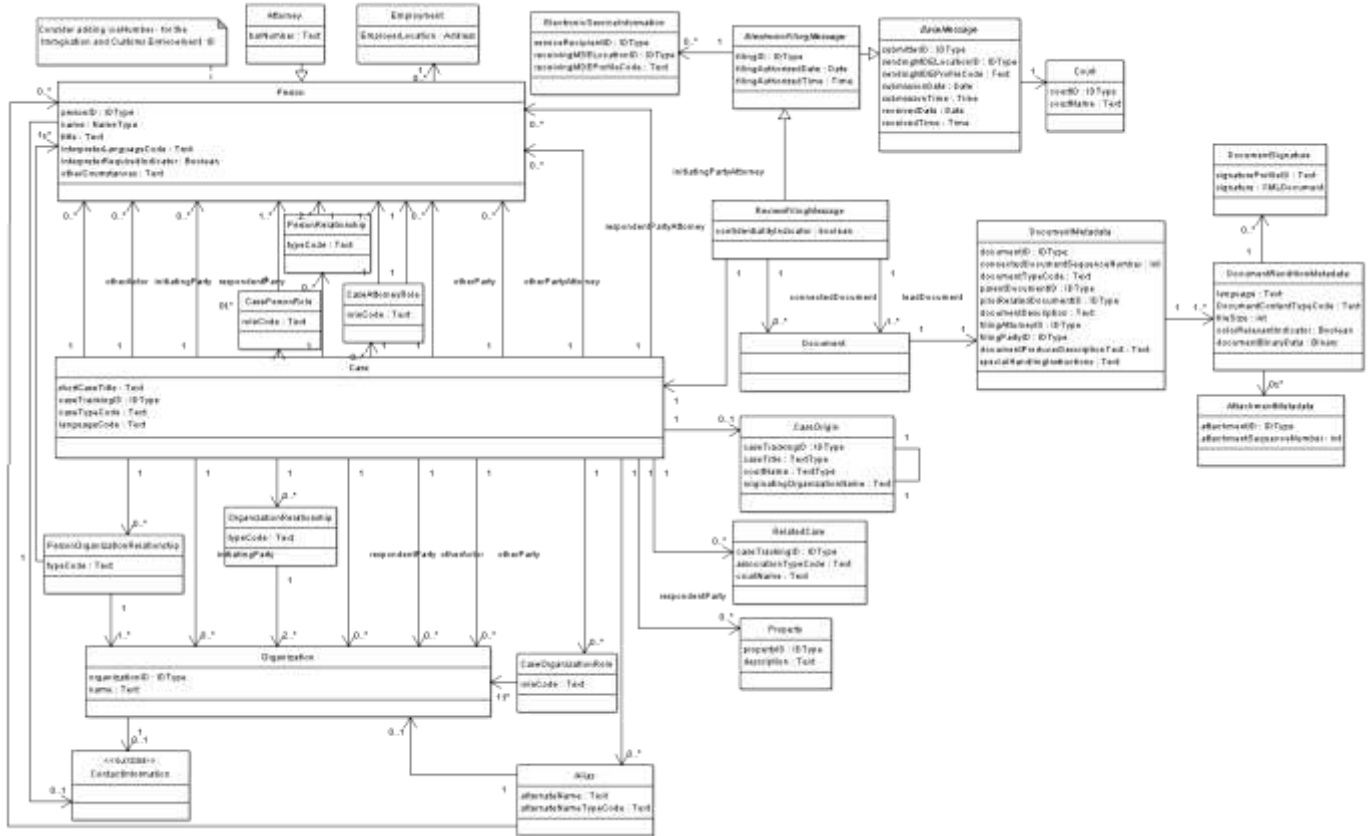
1276 The exchange content models are the result of a detailed analysis of the data requirements to support the
1277 ECF 4.0 Process Model (see Section 3). During the modeling process, common items of data were
1278 identified by a process of normalization to identify aggregates based on functional dependency. Where
1279 appropriate, these were generalized so that they could be re-used to support the various messages.

1280 The exchange content models are used for the following purposes:

- 1281 • They facilitate the identification of the reusable components, i.e., the data structures that are common
1282 across the ECF 4.0 messages.
- 1283 • They aid in understanding the information requirements of the total scenario.
- 1284 • They are the source from which the object classes are derived and documented in the ECF 4.0
1285 schemas (see Section 4).

1286 To facilitate comprehension, the ECF 4.0 is composed of several exchange content model diagrams.
 1287 Each diagram represents a logical grouping of components and displays both the attributes and object
 1288 classes belonging to the components in this grouping. The scope of each diagram is arbitrary and does
 1289 not hold any significance beyond these diagrams.

1290 For example, the ECF 4.0 Review Filing Model diagram is shown below:



1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308

The complete set of exchange content models for all the ECF 4.0 components is listed below:

Appellate Filing Model

<model/uml/html/AppellateFiling.png>

Bankruptcy Filing Model

<model/uml/html/BankruptcyFiling.png>

Base Message Model

<model/uml/html/BaseMessage.png>

Civil Filing Model

<model/uml/html/CivilFiling.png>

Citation Filing Model

<model/uml/html/Violation Filing.png>

Criminal Filing Model

<model/uml/html/CriminalFiling.png>

Domestic Filing Model

- 1309 <model/uml/html/DomesticFiling.png>
- 1310 **Extended Person Information Model**
- 1311 <model/uml/html/ExtendedPersonInformation.png>
- 1312 **Get Calculated Fees Query Model**
- 1313 <model/uml/html/GetFeesCalculationQuery.png>
- 1314 **Get Case List Query Model**
- 1315 <model/uml/html/GetCaseListQuery.png>
- 1316 **Get Document Query Model**
- 1317 <model/uml/html/GetDocumentQuery.png>
- 1318 **Get Filing List Query Model**
- 1319 <model/uml/html/GetFilingListQuery.png>
- 1320 **Get Filing Status Query Model**
- 1321 <model/uml/html/GetFilingStatusQuery.png>
- 1322 **Get Service Information Query Model**
- 1323 <model/uml/html/GetServiceInformationQuery.png>
- 1324 **Major Design Elements Model**
- 1325 <model/uml/html/MajorDesignElements.png>
- 1326 **Juvenile Filing Model**
- 1327 <model/uml/html/JuvenileFiling.png>
- 1328 **Record Docketing Model**
- 1329 <model/uml/html/RecordDocketing.png>
- 1330 **Review Filing Model**
- 1331 <model/uml/html/ReviewFiling.png>

1332

No specific directions are defined for the associations in these models; they can be navigated in either direction. The specific navigation path for each association is defined when documents are assembled.

1335 **B.4 Spreadsheet Models**

1336 ECF 4.0 uses spreadsheet models to describe the mapping of objects and attributes to **[NIEM]** and ECF
1337 4.0 elements. The spreadsheet models use rows to define components. Components are either simple
1338 data types or associations. Columns define the metadata associated with each component type.

1339 The ECF 4.0 spreadsheet model is located at <model/ECF-4.0-NIEM2-mapping.xls>.

1340

1341

Appendix C. (Informative) MDE Operations

1342
1343
1344
1345

This appendix details the operations that are provided by each Major Design Element (MDE) and the operations, provided by other MDEs that each MDE “consumes.” Implementation of an MDE requires both that the MDE provide certain functionality and that the MDE use particular operations provided by other MDEs.

1346

C.1 Filing Assembly MDE

1347
1348
1349
1350
1351

The Filing Assembly MDE supports the preparation and submission of filed documents to a court for review, and can receive the results of that process. The Filing Assembly MDE also conveys filings to the Legal Service MDE for service on other case participants. The Filing Assembly MDE calls operations in other MDEs and provides a single operation for notifying the submitter that the filing has been reviewed by a court. A Filing Assembly MDE may be provided by a court or by a third party.

1352

C.1.1 Provided Operations

1353

The Filing Assembly MDE provides the following operations to other MDEs:

| Operation | Called By | Output | Parameters |
|----------------------------|-------------------|--|--|
| NotifyFilingReviewComplete | Filing Review MDE | xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage | xsd/message/ECF-4.0-ReviewFilingCallbackMessage.xsd : ReviewFilingCallbackMessage xsd/message/ECF-4.0-PaymentMessage.xsd : PaymentMessage |

1354

C.1.2 Consumed Operations

1355

The Filing Assembly MDE calls the following operations in other MDEs:

| Operation | Provided By | Return Type |
|-----------------------|-------------------|--|
| GetPolicy | Filing Review MDE | xsd/message/ECF-4.0-CourtPolicyQueryMessage.xsd : CourtPolicyReponseMessage |
| ReviewFiling | Filing Review MDE | xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage |
| GetFeesCalculation | Filing Review MDE | xsd/message/ECF-4.0-FeesCalculationResponseMessage.xsd : FeesCalculationResponseMessage |
| GetFilingStatus | Filing Review MDE | xsd/message/ECF-4.0-FilingStatusResponseMessage.xsd : FilingStatusResponseMessage |
| GetFilingList | Filing Review MDE | xsd/message/ECF-4.0-FilingListResponseMessage.xsd : FilingListResponseMessage |
| GetCase | Court Record MDE | xsd/message/ECF-4.0-CaseResponseMessage.xsd : CaseResponseMessage |
| GetCaseList | Court Record MDE | xsd/message/ECF-4.0-CaseListResponseMessage.xsd : CaseListResponseMessage |
| GetServiceInformation | Court Record MDE | xsd/message/ECF-4.0-ServiceInformationResponseMessage.xsd : ServiceInformationResponseMessage |

| | | |
|-------------|-------------------|---|
| GetDocument | Court Record MDE | xsd/message/ECF-4.0-DocumentResponseMessage.xsd : DocumentResponseMessage |
| ServeFiling | Legal Service MDE | xsd/message/ECF-4.0-ServiceReceiptMessage.xsd : ServiceReceiptMessage |

1356 C.2 Filing Review MDE

1357 The Filing Review MDE receives, presents and manages the filings. The Filing Review MDE receives
1358 filings in a standard format and presents those filings to a Clerk for review, where they may be accepted
1359 or rejected. The Filing Review MDE transmits data and documents to the Filing Assembly MDE to inform
1360 the filer that the filing has been accepted or rejected. The Filing Review MDE transmits data and
1361 documents for accepted filings to the Court Record MDE for docketing and recording. While there will
1362 generally be one Filing Review MDE per court, there is no physical barrier to having more than one,
1363 particularly if a court wants to support different Filing Review MDEs for particular case types.

1364 C.2.1 Provided Operations

1365 The Filing Review MDE provides the following operations to other MDEs:

| Operation | Called By | Output | Parameters |
|-------------------------|---------------------|---|---|
| ReviewFiling | Filing Assembly MDE | xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage | xsd/message/ECF-4.0-CoreFilingMessage.xsd : CoreFilingMessage |
| | | | xsd/message/ECF-4.0-PaymentMessage.xsd : PaymentMessage |
| NotifyDocketingComplete | Court Docketing MDE | xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage | xsd/message/ECF-4.0-RecordDocketingCallbackMessage.xsd : RecordDocketingCallbackMessage |
| GetFeesCalculation | Filing Assembly MDE | xsd/message/ECF-4.0-FeesCalculationResponseMessage.xsd : FeesCalculationResponseMessage | xsd/message/ECF-4.0-FeesCalculationQueryMessage.xsd : FeesCalculationQueryMessage |
| GetFilingList | Filing Assembly MDE | xsd/message/ECF-4.0-FilingListResponseMessage.xsd : FilingListResponseMessage | xsd/message/ECF-4.0-FilingListQueryMessage.xsd : FilingListQueryMessage |
| GetFilingStatus | Filing Assembly MDE | xsd/message/ECF-4.0-FilingStatusResponseMessage.xsd : FilingStatusResponseMessage | xsd/message/ECF-4.0-FilingStatusQueryMessage.xsd : FilingStatusQueryMessage |
| GetPolicy | Filing Assembly MDE | xsd/message/ECF-4.0-CourtPolicyQueryMessage.xsd : CourtPolicyResponseMessage | xsd/message/ECF-4.0-CourtPolicyQueryMessage.xsd : CourtPolicyQueryMessage |

1366 C.2.2 Consumed Operations

1367 The Filing Review MDE calls the following operations in other MDEs:

| Operation | Provided By | Output |
|----------------------------|---------------------|--|
| RecordFiling | Court Record MDE | xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage |
| NotifyFilingReviewComplete | Filing Assembly MDE | xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage |

1368 C.3 Court Record MDE

1369 The Court Record MDE receives the filed documents from the Filing Review MDE and enters them into
1370 the official case record of the court. The Court Record MDE notifies the Filing Review MDE that the filing
1371 has been filed.

1372 C.3.1 Provided Operations

1373 The Court Record MDE provides the following operations to other MDEs:

| Operation | Called By | Output | Parameters |
|-----------------------|---------------------|--|---|
| RecordFiling | Filing Review MDE | xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage | xsd/message/ECF-4.0-RecordDocketingMessage.xsd : RecordDocketingMessage xsd/message/ECF-4.04.0-CoreFilingMessage.xsd : CoreFilingMessage |
| GetCase | Filing Assembly MDE | xsd/message/ECF-4.0-CaseResponseMessage.xsd : CaseResponseMessage | xsd/message/ECF-4.0-CaseQueryMessage.xsd : CaseQueryMessage |
| GetCaseList | Filing Assembly MDE | xsd/message/ECF-4.0-CaseListResponseMessage.xsd : CaseListResponseMessage | xsd/message/ECF-4.0-CaseListQueryMessage.xsd : CaseListQueryMessage |
| GetServiceInformation | Filing Assembly MDE | xsd/message/ECF-4.0-ServiceInformationResponseMessage.xsd : ServiceInformationResponseMessage | xsd/message/ECF-4.0-ServiceInformationQueryMessage.xsd : ServiceInformationQueryMessage |
| GetDocument | Filing Assembly MDE | xsd/message/ECF-4.0-DocumentResponseMessage.xsd : DocumentResponseMessage | xsd/message/ECF-4.0-DocumentQueryMessage.xsd : DocumentQueryMessage |

1374 C.3.2 Consumed Operations

1375 The Court Record MDE calls the following operations in other MDEs:

| Operation | Provided By | Output |
|-------------------------|-------------------|--|
| NotifyDocketingComplete | Filing Review MDE | xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage |

1376 C.4 Legal Service MDE

1377 The Legal Service MDE enables a filer or a court to electronically transmit copies of, or links to,
1378 electronically filed documents to other parties who are participating in the case and who are entitled to be

1379 promptly served with the electronically filed documents. The Filing Assembly MDE transmits data and
1380 documents to the Legal Service MDE to inform the case participant that an electronic filing has been
1381 submitted to the court clerk. The Legal Service MDE transmits a callback message to the Filing
1382 Assembly MDE requesting a notification to confirm receipt of the served document.

1383 **C.4.1 Provided Operations**

1384 The Legal Service MDE provides the following operations to other MDEs:

| Operation | Called By | Output | Parameters |
|-------------|---------------------|---|---|
| ServeFiling | Filing Assembly MDE | xsd/message/ECF-4.0-ServiceReceiptMessage.xsd : ServiceReceiptMessage | xsd/message/ECF-4.0-CoreFilingMessage.xsd : CoreFilingMessage |

1385 **C.4.2 Consumed Operations**

1386 The Legal Service MDE does not call operations in other MDEs

1387

Appendix D. (Informative) Example Instances

1388 Example instances of each ECF 4.0 message are provided in the xml/ subdirectory, as listed below:

1389

1390 **FeesCalculationQueryMessage**

1391 [xml/ECF-4.0-FeesCalculationQueryMessage.xml](#)

1392 **FeesCalculationResponseMessage**

1393 [xml/ECF-4.0-FeesCalculationResponseMessage.xml](#)

1394 **CaseListQueryMessage**

1395 [xml/ECF-4.0-CaseListQueryMessage.xml](#)

1396 **CaseListResponseMessage**

1397 [xml/ECF-4.0-CaseListResponseMessage.xml](#)

1398 **CaseQueryMessage**

1399 [xml/ECF-4.0-CaseQueryMessage.xml](#)

1400 **CaseResponseMessage**

1401 [xml/ECF-4.0-CaseResponseMessage.xml](#)

1402 **CoreFilingMessage (Appellate case type)**

1403 [xml/ECF-4.0-CoreFilingMessage-Appellate.xml](#)

1404 **CoreFilingMessage (Criminal case type)**

1405 [xml/ECF-4.0-CoreFilingMessage-Criminal.xml](#)

1406 **CourtPolicyQueryMessage**

1407 [xml/ECF-4.0-CourtPolicyQueryMessage.xml](#)

1408 **CourtPolicyResponseMessage**

1409 [xml/ECF-4.0-CourtPolicyResponseMessage.xml](#)

1410 **DocumentQueryMessage**

1411 [xml/ECF-4.0-DocumentQueryMessage.xml](#)

1412 **DocumentResponseMessage**

1413 [xml/ECF-4.0-DocumentResponseMessage.xml](#)

1414 **FilingListQueryMessage**

1415 [xml/ECF-4.0-FilingListQueryMessage.xml](#)

1416 **FilingListResponseMessage**

1417 [xml/ECF-4.0-FilingListResponseMessage.xml](#)

1418 **FilingPaymentMessage**

1419 [xml/ECF-4.0-PaymentMessage.xml](#)

1420 **FilingStatusQueryMessage**

1421 [xml/ECF-4.0-FilingStatusQueryMessage.xml](#)

1422 **FilingStatusResponseMessage**

1423 [xml/ECF-4.0-FilingStatusResponseMessage.xml](#)

1424 **MessageReceiptMessage**

1425 [xml/ECF-4.0-MessageReceiptMessage.xml](#)

1426 **PaymentReceiptMessage**

- 1427 [xml/ECF-4.0-PaymentReceiptMessage.xml](#)
- 1428 **RecordDocketingCallbackMessage**
- 1429 [xml/ECF-4.0-RecordDocketingCallbackMessage.xml](#)
- 1430 **RecordDocketingMessage**
- 1431 [xml/ECF-4.0-RecordDocketingMessage.xml](#)
- 1432 **ReviewFilingCallbackMessage**
- 1433 [xml/ECF-4.0-ReviewFilingCallbackMessage.xml](#)
- 1434 **ServiceInformationQueryMessage**
- 1435 [xml/ECF-4.0-ServiceInformationQueryMessage.xml](#)
- 1436 **ServiceInformationResponseMessage**
- 1437 [xml/ECF-4.0-ServiceInformationResponseMessage.xml](#)
- 1438 **ServiceReceiptMessage**
- 1439 [xml/ECF-4.0-ServiceReceiptMessage.xml](#)

1440 **Appendix E. (Informative) Ongoing Work Items**

1441 The Electronic Court Filing TC plans to continue to revise and expand this specification through future
1442 versions. Future versions of ECF will:

- 1443 • Address filings in administrative tribunals
- 1444 • Address primary service (the delivery of documents such as summonses, subpoenas and warrants
1445 that establish a court's jurisdiction over a party)
- 1446 • Consider how the specifications for filing of documents intended for filing with a court relate to
1447 specifications for filing other documents, e.g., property records, in the offices of elected clerks of
1448 courts
- 1449 • Incorporate feedback from ECF implementations
- 1450 • Support future releases of the **[NIEM]**
- 1451 • Support future **[Court Document]** specifications and integration optimizations
- 1452 • Support non-case related filings into a court clerk's office

1453

Appendix F. (Informative) Acknowledgments

1454 The following court organizations provided lists of data elements required for initiating cases in their case
1455 management information systems:

- 1456 • Administrative Office of United States Courts
 - 1457 ○ Bankruptcy, Civil, Criminal
- 1458 • Arizona Administrative Office of the Courts
 - 1459 ○ Appellate
- 1460 • Fourth Judicial District Court, Hennepin County, Minneapolis
 - 1461 ○ Criminal
- 1462 • King County Superior Court, Washington
 - 1463 ○ Civil, Criminal
- 1464 • Missouri Office of State Courts Administrator
 - 1465 ○ Citation
- 1466 • Thirteenth Judicial District, Orange County, Florida (through vendor)
 - 1467 ○ Civil, Criminal, Domestic relations, Mental health, Juvenile delinquency/dependency,
 - 1468 Probate, Citation
- 1469 • Utah State Courts
 - 1470 ○ Civil, Criminal

1471

1472 The following individuals were members or voting members of the committee during the development of
1473 this specification:

- 1474 • Rolly Chambers, American Bar Association
- 1475 • John Messing, American Bar Association
- 1476 • Adam Angione, Courthouse News Service
- 1477 • Eric Eastman Doxpop, LLC
- 1478 • Robert DeFilippis, Associate
- 1479 • Chester Ensign, Associate
- 1480 • Gary Poindexter, Associate
- 1481 • Michael Alexandrou, Judicial Council of Georgia
- 1482 • Shawn Artrip, Judicial Council of Georgia,
- 1483 • Robbie Diaz, Judicial Council of Georgia
- 1484 • Hui Ji, Judicial Council of Georgia
- 1485 • Morgan Medders, Judicial Council of Georgia
- 1486 • Scott Edson, LA County Information Systems Advisory Body
- 1487 • Ali Farahani, LA County Information Systems Advisory Body
- 1488 • John Ruegg, LA County Information Systems Advisory Body
- 1489 • CJ Allen, Maricopa County
- 1490 • Robin Gibson, Missouri Office of State Courts Admin
- 1491 • James Cabral, MTG Management Consultants, LLC
- 1492 • Thomas Clarke, National Center for State Courts
- 1493 • Diana Graski, National Center for State Courts
- 1494 • Jim Harris, National Center for State Courts
- 1495 • Jason Hill, New York State Office of Court Administration, DoT
- 1496 • Robert O'Brien, Ottawa Courts Administration Service
- 1497 • George Knecht, PC Intellect, LLC
- 1498 • Mark Ladd, Property Records Industry Assn.
- 1499 • Ron Bowmaster. Utah Administrative Office of the Courts

1500

Appendix G. (Informative) Revision History

| Rev | Date | By Whom | What |
|--------|------------|--------------|--|
| Wd01 | 2008-03-17 | James Cabral | Initial version |
| Wd02 | 2008-08-15 | James Cabral | Revision including complete IEPD. |
| Wd03 | 2008-08-25 | James Cabral | Revisions based on August face to face meeting and initial testing. |
| Wd04 | 2008-09-03 | James Cabral | Revised guidance on filing record on appeal (Section 3.4) |
| Wd01 | 2008-09-20 | James Cabral | Committee draft |
| 4.01 | 2010-03-15 | James Cabral | Minor schema and definition changes based on feedback from implementers of the ECF 4.0 specification. |
| Csd-01 | 2011-08-08 | James Cabral | Revised reference format. Minor schema changes including correction of several constraints. Addition of Section 1.3.5 and revision of Section 2.4.3 based on feedback from implementers. |
| Csd-02 | 2011-10-18 | James Cabral | Updated UBL reference version 2.1. |
| Csd-03 | 2012-02-07 | James Cabral | Minor changes to Appendix C (non-normative). Minor additions to the schema. Revision of sections 2.4.1, 2.4.2, 3.1 and 3.3.3.2, based on feedback from implementers. |

1501

1502