



Electronic Court Filing Version 4.01

Committee Specification Draft 01 / Public Review Draft 01

08 February 2011

Specification URIs:

This Version:

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

Previous Version:

<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.0/ecf-v4.0-spec/ecf-v4.0-spec-cd01.doc> (Authoritative)
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.0/ecf-v4.0-spec/ecf-v4.0-spec-cd01.html>
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.0/ecf-v4.0-spec/ecf-v4.0-spec-cd01.pdf>

Latest Version:

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

Technical Committee:

OASIS LegalXML Electronic Court Filing TC

Chair(s):

Ron Bowmaster, Utah Administrative Office of the Courts
John Greacen, Individual Member

Editor(s):

Adam Angione, Courthouse News Service
James Cabral, MTG Management Consultants

Related work:

XML Schemas: [legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/csprd01/xsd/](http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/csprd01/xsd/)

This specification replaces or supersedes:

- OASIS LegalXML Electronic Court Filing Versions 3.0 and 4.0

This specification is related to:

- National Information Exchange Model 2.0

Declared XML Namespace(s):

[urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:AppInfo-4.0](http://uris.oasis-open.org/urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:AppInfo-4.0)
[urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:AppellateCase-4.0](http://uris.oasis-open.org/urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:AppellateCase-4.0)
[urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:BankruptcyCase-4.0](http://uris.oasis-open.org/urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:BankruptcyCase-4.0)
[urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseListQueryMessage-4.0](http://uris.oasis-open.org/urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseListQueryMessage-4.0)

urn:oasis:names:tc:legalxml-court filing:schema:xsd:CaseListResponseMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CaseQueryMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CaseResponseMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CitationCase-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CivilCase-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CommonTypes-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CoreFilingMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CourtPolicyQueryMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CourtPolicyResponseMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:CriminalCase-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:DocumentQueryMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:DocumentResponseMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:DomesticCase-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:FeesCalculationQueryMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:FeesCalculationResponseMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:FilingListQueryMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:FilingListResponseMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:FilingStatusQueryMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:FilingStatusResponseMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:JuvenileCase-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:MessageReceiptMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:PaymentMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:PaymentReceiptMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:RecordDocketingCallbackMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:RecordDocketingMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:ReviewFilingCallbackMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:ServiceInformationQueryMessage-4.0
urn:oasis:names:tc:legalxml-court filing:schema:xsd:ServiceInformationResponseMessage-4.0
urn:oasis:names:tc:legalxml-court filing: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 OASIS LegalXML Electronic Court Filing TC 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 <http://www.oasis-open.org/committees/legalxml-court filing/>.

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 (<http://www.oasis-open.org/committees/legalxml-court filing//ipr.php>).

Citation Format:

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

[ECF v4.01]

Electronic Court Filing Version 4.01. 08 February 2011. OASIS Committee Specification Public Review Draft. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/csprd01/ecf-v4.01-spec-csprd01.doc>

Notices

Copyright © OASIS® 2010. 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 <http://www.oasis-open.org/who/trademark.php> 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
1.4	Terms and Definitions	11
1.5	Symbols and Abbreviations.....	12
1.6	Normative References.....	13
1.7	Non-Normative References.....	13
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	23
3.1	The Filing-Preparation-to-Docketing Process Model	23
3.2	Business Rules.....	25
3.2.1	GetPolicy.....	25
3.2.2	GetServiceInformation	25
3.2.3	GetFeesCalculation	25
3.2.4	ReviewFiling.....	25
3.2.5	ServeFiling.....	25
3.2.6	RecordFiling.....	26
3.2.7	NotifyDocketingComplete	26
3.2.8	NotifyFilingReviewComplete	26
3.2.9	GetFilingList	26
3.2.10	GetFilingStatus	26
3.2.11	GetCaseList	26
3.2.12	GetCase	27
3.2.13	GetDocument.....	27
3.3	Message Business Rules.....	27

3.3.1 Identifiers	27
3.3.2 Code Lists	28
3.3.3 Message-Specific Business Rules.....	29
3.4 Filing the Record on Appeal.....	30
4 ECF 4.0 Schemas.....	32
4.1 ECF 4.0 Case Type Schemas.....	32
4.2 ECF 4.0 Common Schemas	32
4.3 ECF 4.0 Constraint and Subset Schemas	33
4.4 ECF 4.0 Message Schemas	33
5 Service Interaction Profiles	35
5.1 Service Interaction Profile Requirements.....	35
5.2 Service Interaction Profile Approval and Revision Processes	36
5.3 Supported Service Interaction Profiles.....	36
6 Document Signature Profiles	38
6.1 Document Signature Profile Requirements.....	38
6.2 Document Signature Profile Approval and Revision Processes	38
6.3 Supported Document Signature Profiles.....	39
7 Conformance.....	40
Appendix A. (Informative) Release Notes	41
A.1 Availability	41
A.2 Package Structure	41
A.3 Recursive Structures.....	41
A.4 Date and Time Formats	41
A.5 Known Errata	41
Appendix B. (Informative) ECF 4.0 Development Approach and Artifacts.....	43
B.1 Principles	43
B.2 Approach.....	43
B.3 ECF 4.0 Exchange Content Models	43
B.4 Spreadsheet Models	45
Appendix C. (Informative) MDE Operations	46
C.1 Filing Assembly MDE.....	46
C.1.1 Provided Operations.....	46
C.1.2 Consumed Operations	46
C.2 Filing Review MDE	47
C.2.1 Provided Operations.....	47
C.2.2 Consumed Operations	47
C.3 Court Record MDE	48
C.3.1 Provided Operations.....	48
C.3.2 Consumed Operations	48
C.4 Legal Service MDE	48
C.4.1 Provided Operations	49
C.4.2 Consumed Operations.....	49
Appendix D. (Informative) Example Instances	50
Appendix E. (Informative) Ongoing Work Items	52

Appendix F. (Informative) Acknowledgments 53
Appendix G. (Informative) Revision History 54

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 specification does NOT support “primary service,” which entails the service of summonses, subpoenas, warrants and other documents that

44 establish court jurisdiction over persons, making them parties to a case. Therefore, this specification does NOT
45 support the following automated information exchanges:

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

50

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

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

61

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

65 1.2 Relationship to Prior Specifications

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

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

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

75

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

80 1.3 ECF Version 4.01

81 ECF 4.01 is a maintenance release to address several minor schema and definition issues identified by implemen-
82 ters of the ECF 4.0 specification. All references in this document to ECF 4.0 apply to ECF 4.01 as well.

83 Relationship to other XML Specifications

84 The ECF specification incorporates other existing, non-proprietary XML specifications wherever possible. In
85 particular, the specification has dependencies on the **[NIEM]**, the **[UBL]** data library and the World Wide Web
86 Consortium (W3C) XML Digital Signatures specification. The terminology used in this specification to describe the

87 components of the ECF technical architecture conforms to the OASIS Reference Model for Service Oriented
88 Architecture.

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

94 1.3.1 National Information Exchange Model (NIEM)

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

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

110 1.3.2 OASIS Universal Business Language

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

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

117 <AllowanceCharge>

118 Information about a charge or discount price component.

119 <Address>

120 Information about a structured address.

121 <Payment>

122 Information directly relating to a specific payment.

123 1.3.3 W3C XML-Signature Syntax and Processing

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

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

126 assured of the validity of the electronically transmitted data. **[XMLSIG]** defines standard means for specifying
127 information content that is to be digitally signed.²

128 ECF 4.0 employs the **[XMLSIG]** specification to describe digital signatures applied to the entire ECF 4.0 message
129 transmission in order to provide authentication, encryption and message integrity. **[XMLSIG]** are also used in the
130 ECF 4.0 XML Document Signature Profile.

131 **1.3.4 OASIS Reference Model for Service Oriented Architecture**

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

135 **1.4 Terms and Definitions**

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

138
139 This section defines key terms used in this specification.

140

141 **Attachment**

142 See definition in Section 0.

143 **Callback message**

144 A message transmission returned by some operations some time after the operation was invoked (asyn-
145 chronously).

146 **Document**

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

149 **Document hash**

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

152 **Docketing**

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

155 **Filer**

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

158 **Filing**

159 An electronic document (with any associated data, attachments and the like) that has been assembled for
160 the purpose of being filed into a specified court case.

161 **Hub Service MDE**

162 A centralized Service MDE capable of receiving a single set of service notifications for all parties registered
163 for electronic service in a case and transmitting the service notifications to the Service MDEs registered to
164 each party in the case.

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

- 165 **Major Design Element (MDE)**
166 A logical grouping of operations representing a significant business process supported by ECF 4.0. Each
167 MDE operation receives one or more messages, returning a synchronous response message (a reaction to
168 a message received) and, optionally, returning an asynchronous (later) response message to the originat-
169 ing message sender.
- 170 **Message**
171 See definition in Section 2.3.1.
- 172 **Message Transmission**
173 The sending of one or more messages and associated attachments to an MDE. Each transmission must
174 invoke or respond to an operation on the receiving MDE, as defined in the ECF 4.0 specification.
- 175 **Operation (or MDE Operation)**
176 A function provided by an MDE upon receipt of one or more messages. The function provided by the op-
177 eration represents a significant step in the court filing business process. A sender invokes an operation on
178 an MDE by transmitting a request with an operation identifier and a set of messages.
- 179 **Operation signature**
180 A definition of the input message and synchronous response message associated with an operation. Each
181 message is given a name and a type by the operation. The type is defined by a single one of the message
182 structures defined in the ECF 4.0 specification.
- 183 **Synchronous response**
184 A message transmission returned immediately (synchronously) as the result of an operation. Every opera-
185 tion has a synchronous response.

186 **1.5 Symbols and Abbreviations**

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

189 **ECF 4.0**

190 Electronic Court Filing 4.0

191 **IEPD**

192 Information Exchange Package Documentation

193 **MDE**

194 Major Design Element

195 **NIEM**

196 National Information Exchange Model

197 **OASIS**

198 Organization for the Advancement of Structured Information Standards

199 **XML**

200 eXtensible Markup Language

201 **W3C**

202 World Wide Web Consortium

203 **WS-I**

204 Web Services Interoperability Organization

205

206

1.6 Normative References

- 207 **[FIPS 180-2]** Secure Hash Standard, [http://csrc.nist.gov/publications/fips/fips180-2/fips180-](http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf)
208 [2withchangenotice.pdf](http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf), National Institute for Standards and Technology, August 2002.
- 209 **[NIEM]** National Information Exchange Model 2.0, <http://niem.gov>, US DOJ and DHS, 2007.
- 210 **[NIEM Guide]** NIEM Implementation Guidelines, <http://www.niem.gov/implementationguide.php>, US DOJ
211 and DHS, 2007.
- 212 **[NIEM Techniques]** *Techniques for Building and Extending NIEM*,
213 <http://www.niem.gov/topicIndex.php?topic=techPDF>, Georgia Tech Research Institute, Au-
214 gust 2007.
- 215 **[Namespaces]** T. Bray, *Namespaces in XML*, <http://www.w3.org/TR/1999/REC-xml-names-19990114>,
216 January 14, 1999.
- 217 **[RFC2046]** N. Freed, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*,
218 <http://www.ietf.org/rfc/rfc2046.txt>, IETF RFC 2046, November 1996.
- 219 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
220 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 221 **[RFC4122]** Leach, et al., *A Universally Unique IDentifier (UUID) URN Namespace*,
222 <http://www.ietf.org/rfc/rfc4122.txt>, IETF RFC 4112, July 2005.
- 223 **[Schema Part 1]** H. S. Thompson, D. Beech. M. Maloney, N. Mendelsohn, *XML Schema Part 1: Structures*
224 *Second Edition*, <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>, W3C Recom-
225 mendation, October 28, 2004.
- 226 **[Schema Part 2]** P. Biron, A. Malhotra, *XML Schema Part 2: Datatypes Second Edition*,
227 <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>, W3C Recommendation, Octo-
228 ber 28, 2004
- 229 **[UBL]** J. Bozak, T. McGrath, G. K. Holman (editors), *Universal Business Language 2.0*, , OASIS
230 Standard, December 12, 2006.
- 231 **[XML 1.0]** T. Bray, *Extensible Markup Language (XML) 1.0 (Third Edition)*,
232 <http://www.w3.org/TR/REC-xml/REC-XML-20040204>, W3C Recommendation, February 4,
233 2004.
- 234 **[XMLENC]** D. Eastlake, J. Reagle, *XML Encryption Syntax and Processing*,
235 <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>, W3C Recommendation, Decem-
236 ber 2002.
- 237 **[XMLSIG]** D. Eastlake, J. Reagle, D. Solo, *XML-Signature Syntax and Processing*,
238 <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>, W3C Recommendation, Febru-
239 ary 2002.

1.7 Non-Normative References

- 241 **[Court Document]** *OASIS LegalXML Court Document Subcommittee, Charter, July 2006*, [http://www.oasis-](http://www.oasis-open.org/committees/download.php/19120/DocumentSC-Charter-Rev6-1.txt)
242 [open.org/committees/download.php/19120/DocumentSC-Charter-Rev6-1.txt](http://www.oasis-open.org/committees/download.php/19120/DocumentSC-Charter-Rev6-1.txt).
- 243 **[NIEM MNRD]** W. Roberts, S Liebeskind, M. Kindl *National Information Exchange Model Naming and*
244 *Design Rules Draft 1.2*, <http://www.niem.gov/topicIndex.php?topic=file-NDR-lineNum>, Au-
245 gust 7, 2007.
- 246 **[Juvenile XML]** S. Rondendell, et. al., *Juvenile Justice XML Report*,
247 [http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml%5ffinal%5f](http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml%5ffinal%5freport%5f20050630.pdf)
248 [freport%5f20050630.pdf](http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml%5ffinal%5freport%5f20050630.pdf), IJIS Institute, July 2005.
- 249 **[NIEM]** NIEM Concept of Operations, <http://www.niem.gov>, DOJ/DHS, October 7, 2005.
- 250 **[NCSC Guide]** State Court Guide to Statistical Reporting,
251 http://www.ncsconline.org/D_Research/csp/2003_Files/CompleteGuide11_02_04.pdf, Na-
252 tional Center for State Courts, November 2004.

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

256 **[SOA-RM]** MacKenzie, et al., *Reference Model for Service Oriented Architecture 1.0*, [http://www.oasis-](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm)
257 [open.org/committees/tc_home.php?wg_abbrev=soa-rm](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm), Public Review Draft 1.0, February
258 10, 2006.

259 **[Traffic IEPD]** *Traffic Citation IEPD*,
260 http://www.ncsconline.org/d_tech/gjxdm/iepd/citation/Traffic_Citation_08_08_2005.zip, Na-
261 tional Center for State Courts, August 8, 2005.

2 ECF 4.0 Architecture

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

2.1 Core vs. Profiles

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

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

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

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

2.2 Major Design Elements

ECF 4.0 defines four MDEs. They are:

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

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

An ECF 4.0-compliant implementation may implement one or more of the MDEs defined in the specification but a complete ECF 4.0 system **MUST** include at least one each of the Filing Assembly, Filing Review and Court Record MDEs. For instance, a court may decide to provide certain MDEs and allow private providers to furnish the remaining MDEs. When multiple MDEs are implemented by a single court, vendor or application, the application

306 MUST maintain the ECF 4.0 specified operations between each MDE so that other applications will be able to
307 interoperate with it.

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

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

317 2.3 Information Model

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

322 2.3.1 Messages

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

- 327 • Message information about the filing and court case, such as identifiers for the sender and receiver, the sending
328 and receiving MDEs, and the submission date and time, typically a composition of:
 - 329 – A core message which includes basic information common to all courts and case types and Information
330 about each of the documents associated with the message
 - 331 – Case-type-specific extensions that includes information appropriate only for a particular type of filing
 - 332 – Court-specific extensions that includes information appropriate only for cases in a particular court
- 333 • Information about each of the documents associated with the message. A document in this sense is the
334 electronic representation of what would be recognized as a “document” if it were a single, whole, physical paper
335 object. This includes both a lead document, one that will be placed on the court’s register of actions (docketed,
336 indexed) and any supporting document(s), which are present to supplement the lead document in some way.
337 The message includes the document’s metadata, for example, its title, type, identifier, parent document identi-
338 fier and document sequence number. Each document structure may reference one or more attachments,
339 including attachment identifiers and sequence numbers. When included in attachments, a logical document
340 MAY be split into several physical parts if necessary to satisfy a court requirement regarding maximum docu-
341 ment size. The actual binary encoded electronic document MAY be either included in one or more attachments
342 to the message or embedded in the message using the following structure:

```
343     <FilingLeadDocument> (or <FilingConnectedDocument> )  
344         <ecf:DocumentRendition>  
345             <DocumentRenditionMetadata>  
346                 <DocumentAttachment>  
347                     <BinaryBase64Object>2345klj345h...<BinaryBase64Object>  
348                 </DocumentAttachment>  
349             </DocumentRenditionMetadata>  
350         </ecf:DocumentRendition>  
351     </FilingLeadDocument> (or </FilingConnectedDocument> )
```

352

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

363

364 The scope of the ECF core messages and extensions is limited by the following criteria:

- 365 • Elements in the ECF core messages should be applicable to most courts and case types
- 366 • Elements in the ECF case-type-specific extensions should only be applicable to one of the six case types
367 defined in National Center for State Courts (NCSC) statistical standards
- 368 • Elements in locally-defined court-specific extensions should only be applicable to a particular court or court
369 system but not to courts in general

370 All “filing data” elements should be described in the filed documents, whose structure is outside the scope of the
371 ECF specification.

372 **2.3.2 Attachment**

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

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

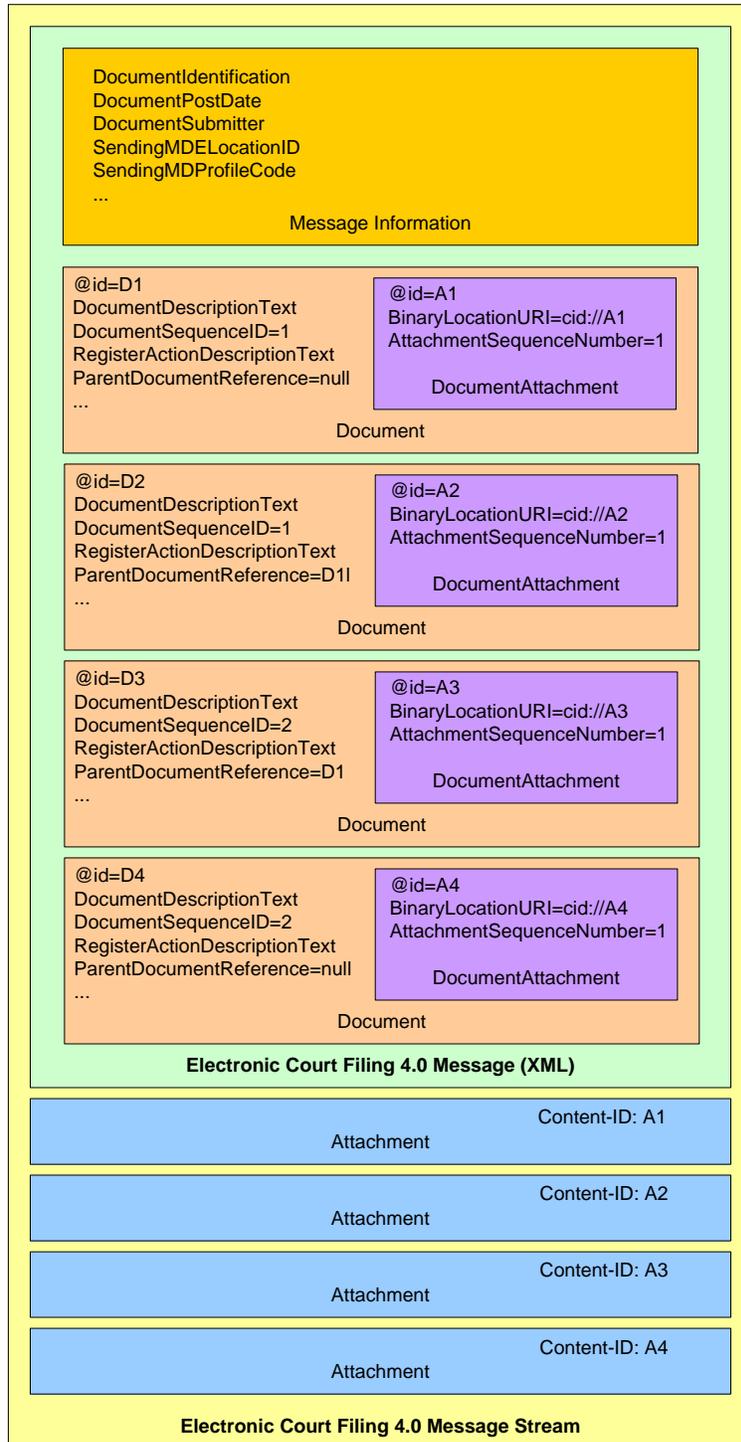
383 **2.3.3 Sample Message Streams**

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

385

386
387
388
389
390

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.



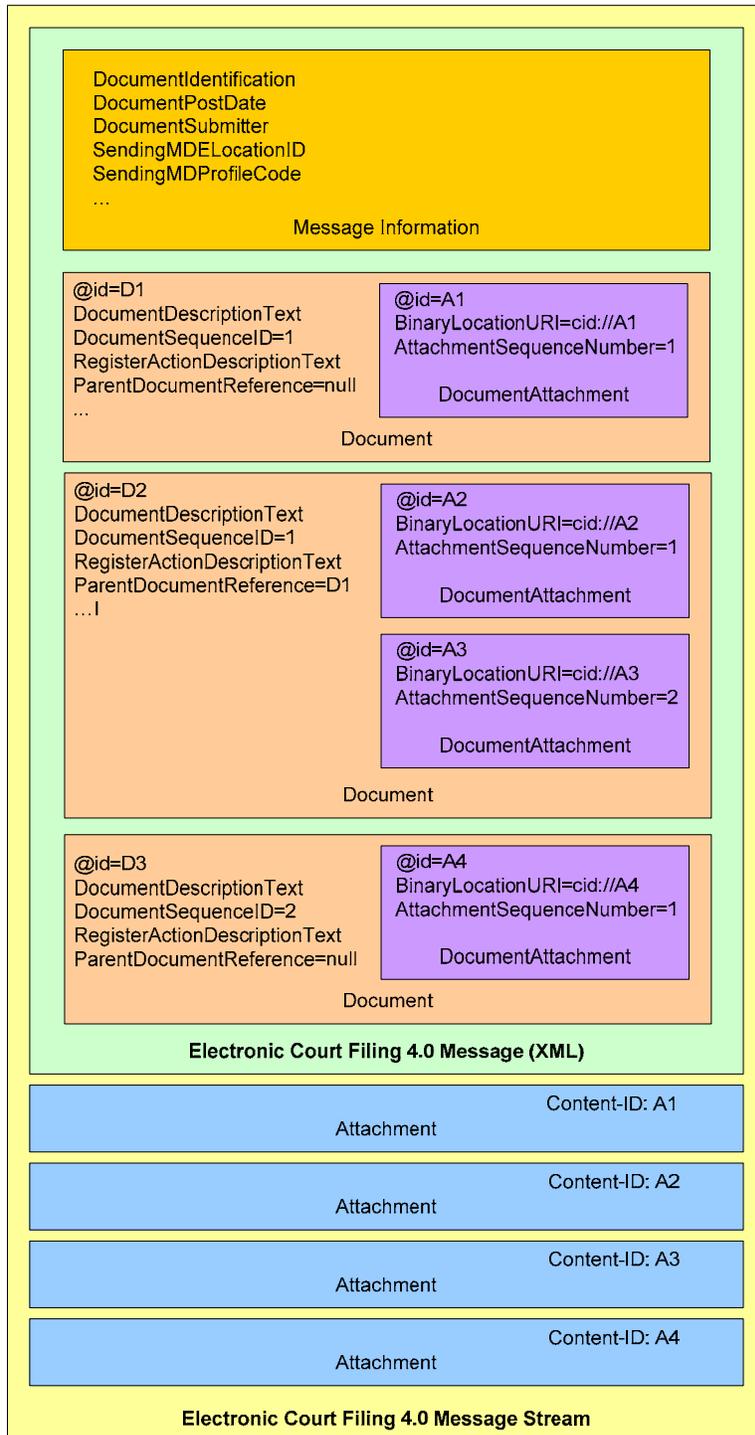
391
392

Figure 1. Simple Message Stream

393

394
395
396
397
398

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.



399
400

Figure 2. Message Stream with a Document in Multiple Attachments

401 2.4 Court Policy

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

- 404 • **Human-readable court policy** – a textual document publishing the court's rules and requirements for
405 electronic filing.
- 406 • **Machine-readable court policy** – an ECF 4.0 message that describes the features of the ECF 4.0 implemen-
407 tation supported by this specification, the court's code lists and any other information a Filing Assembly MDE
408 would need to know in order to successfully submit an electronic filing into that court.

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

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

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

424 2.4.1 Human-Readable Court Policy

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

- 427 1. The unique court identifier
- 428 2. The location of the machine-readable court policy
- 429 3. A definition of what constitutes a "lead document" in the court
- 430 4. A description of how filer identifiers are to be maintained during electronic communications regarding the case
- 431 5. A description of how the court processes (dockets) filings
- 432 6. A description of any instances in which the court will mandate an element that the ECF 4.0 schema makes
433 optional
- 434 7. A description of any restrictions to data property values other than code list restrictions. (This restriction may be
435 removed in later versions of the ECF specification)
- 436 8. Any other rules required for electronic filing in the court

437 2.4.2 Machine-Readable Court Policy

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

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

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

- 444 1. The service interaction profile(s) that the court supports

- 445 2. The MDEs, query operations and case types supported by the court's ECF 4.0 system
- 446 3. Whether a court will accept the filing of a URL in lieu of the electronic document itself
- 447 4. Whether the court accepts documents requiring payment of a filing fee
- 448 5. Whether the court accepts electronic filing of sealed documents
- 449 6. Whether the court accepts multiple (batch) filings
- 450 7. The court-specific extensions to the ECF 4.0 specification, including the required elements (see below)
- 451 8. The maximum sizes allowed for a single attachment and a complete message stream

452 2.4.3 Case-Type and Court Extensions

453 Case-type and court-specific extensions to the ECF core messages are implemented through the methods
454 described in **[NIEM Techniques]**. The primary extension technique is the use of element substitution in which a
455 more specific element defined in a case-type or court-specific extension is used in place of a generic element in a
456 core message. Similarly, an implementation may substitute a court-specific code list for a generic code list defined
457 in this specification.

458

459 2.4.4 Court-Specific Code Lists

460 Code lists are used to constrain the allowable values for certain information in an ECF 4.0 message. The court
461 SHOULD publish Genericcode 1.0 code lists for each of the following code lists and reference each of these code
462 lists in its court policy:

463

- 464 • ECF Code Lists
 - 465 • **Civil Case Type**
 - 466 • `<FiduciaryTypeCode>*`
 - 467 • `<JurisdictionalGroundsCode>`
 - 468 • `<ReliefTypeCode>`
 - 469 • **Domestic Case Type**
 - 470 • `<NoContactCode>*`
 - 471 • `<RequestToVacateCode>`
 - 472 • **Common Types**
 - 473 • `<AliasAlternateNameTypeCode>*`
 - 474 • `<CaseAssociationTypeCode>*`
 - 475 • `<CaseOfficialRoleCode>*`
 - 476 • `<CaseParticipantRoleCode>*`
 - 477 • `<CauseOfActionCode>`
 - 478 • `<CourtEventTypeCode>`
 - 479 • `<EntityAssociationTypeCode>`
 - 480 • `<ErrorCode>*`
 - 481 • **Juvenile Case Type**
 - 482 • `<DelinquentActApplicabilityCode>`
 - 483 • `<DelinquentActDegreeCode>`
 - 484 • `<DelinquentActSeverityCode>`

- 485 • <DelinquentActSpecialAllegationCode>
- 486 • <DependencyAllegationCode>
- 487 • <GuardianAssociationTypeCode>*
- 488 • <PlacementTypeCode>
- 489 • NIEM Code Lists
- 490 • **JXDM**
- 491 • <ChargeEnhancingFactorText>
- 492 • <CourtLocationCode>
- 493 • <RegisterActionDescriptionText>
- 494 • <StatuteCodeIdentification>
- 495 • <StatuteCodeSectionIdentification>
- 496 • <StatuteOffenseIdentification>
- 497 • <StatusOffenseCodeIdentification>
- 498 • **NIEM Core**
- 499 • <BinaryDescriptionText>*
- 500 • <CaseCategoryText>
- 501 • <DriverLicenseCommercialClassCode>
- 502 • <FamilyKinshipCode>*

503

504 A non-normative Genericcode code list with default values is provided for each of the code lists above with asterisks
505 (*).

506

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

509

510 **2.4.5 Court-Specific Constraint Schemas**

511 The cardinality of elements in the NIEM subset imported by the ECF is applied through the use of constraint
512 schemas that define the minimum and maximum occurrence of elements in the NIEM subset. Courts MAY enforce
513 court-specific rules and code lists by creating court-specific constraint schemas. This process creates a duplicate
514 set of the ECF schemas and allows the customization of the cardinality of elements as needed. If court-specific
515 constraint schemas are used, instance documents MUST validate against both the ECF schemas and the court
516 constraint schemas.

517 **3 ECF 4.0 Process Model**

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

521 **3.1 The Filing-Preparation-to-Docketing Process Model**

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

- 528 • GetPolicy
- 529 • GetServiceInformation
- 530 • GetFeesCalculation
- 531 • **ReviewFiling**
- 532 • ServeFiling
- 533 • **RecordFiling**
- 534 • **NotifyDocketingComplete**
- 535 • **NotifyFilingReviewComplete**

536 At any point during or after the ReviewFiling operation, a party MAY access information through the following
537 operations:

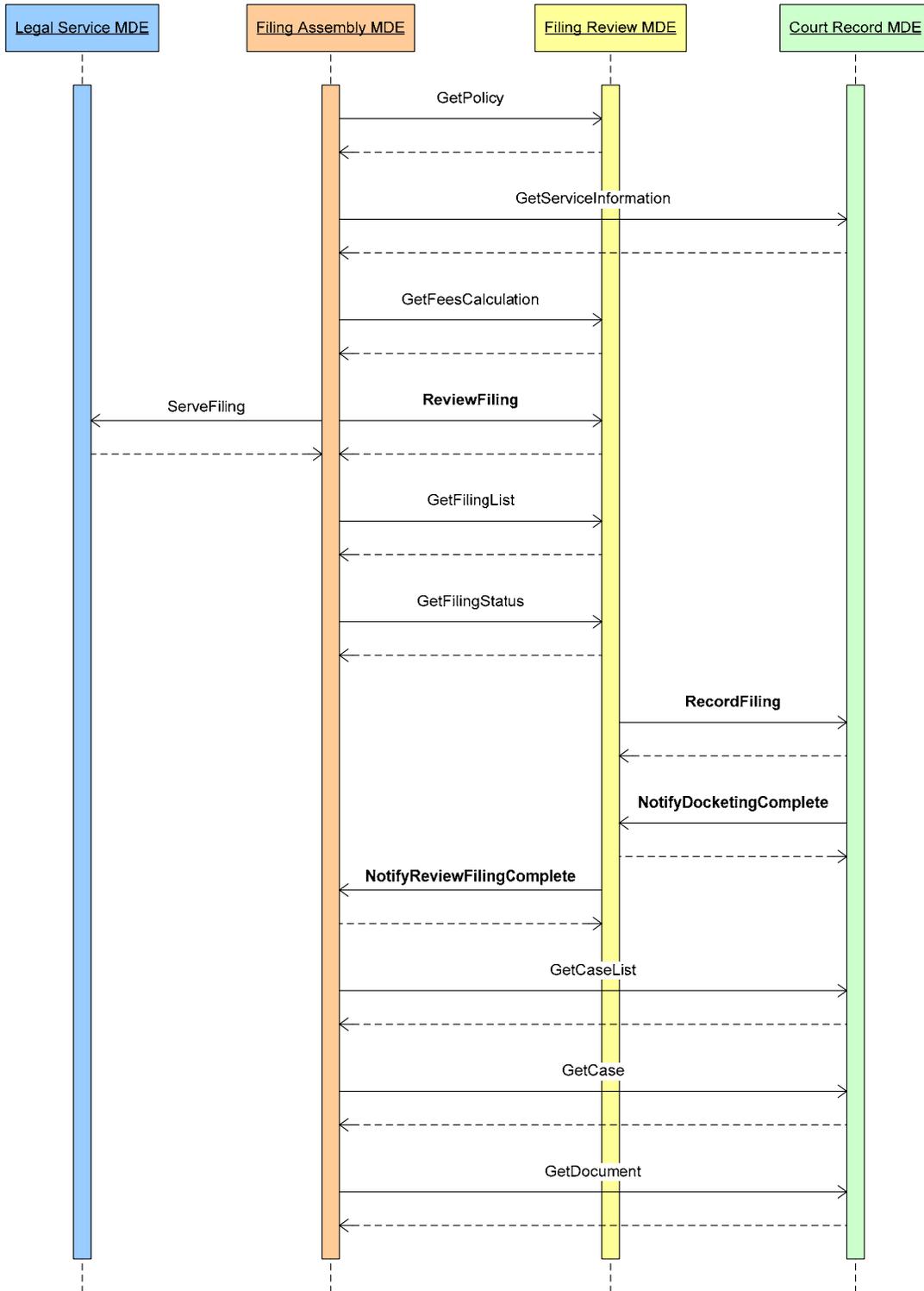
- 538 • GetFilingList
- 539 • GetFilingStatus

540 At any point after the NotifyFilingReviewComplete operation, a party MAY access information through the following
541 operations:

- 542 • GetCaseList
- 543 • GetCase
- 544 • GetDocument

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

Figure 4. Filing Preparation to Docketing Process Model



549 **3.2 Business Rules**

550 This section describes the business rules of the generic filing-preparation-to-docketing process that govern the ECF
551 4.0 operations.

552 ECF 4.0 includes an `<ecf:ErrorCode>` element for returning errors in response to a query request. Successful
553 queries MUST return an `<ecf:ErrorCode>` of "0". Failed queries MUST NOT return an `<ecf:ErrorCode>` of
554 "0" and SHOULD return an appropriate `<ecf:ErrorCode>` value as defined in court policy.

555 **3.2.1 GetPolicy**

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

561 **3.2.2 GetServiceInformation**

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

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

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

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

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

577 **3.2.3 GetFeesCalculation**

578 The Filing Assembly MDE MAY query for the fees associated with a filing by invoking the MDE's GetFeesCalcula-
579 tion operation, with a filing as a parameter, on the Filing Review MDE. The Filing Review MDE responds
580 synchronously with the fee calculation and, optionally, a list of the included charges. This step may be omitted if
581 there are no fees associated with filings in the court or the calculated fees are already known.

582 **3.2.4 ReviewFiling**

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

587 **3.2.5 ServeFiling**

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

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

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

599 **3.2.6 RecordFiling**

600 If the clerk reviews and accepts the filing, the Filing Review MDE MUST invoke the RecordFiling operation on the
601 Court Record MDE. The RecordFiling operation includes information from the ReviewFiling operation with any
602 modifications or comments by the clerk. The Court Record MDE responds synchronously with an acknowledge-
603 ment of the request.

604 **3.2.7 NotifyDocketingComplete**

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

611 **3.2.8 NotifyFilingReviewComplete**

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

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

621 **3.2.9 GetFilingList**

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

625 **3.2.10 GetFilingStatus**

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

629 **3.2.11 GetCaseList**

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

633 **3.2.12 GetCase**

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

638 **3.2.13 GetDocument**

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

642 **3.3 Message Business Rules**

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

645 **3.3.1 Identifiers**

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

648 **3.3.1.1 Attachment Identifiers**

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

651 **3.3.1.2 Case Identifiers**

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

653 **3.3.1.3 Court Identifiers**

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

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

658 Examples of conformant court identifiers include:

- 659 • courts.wa.gov:superior.king
- 660 • nmcourts.com:albd.civil
- 661 • uscourts.gov:100
- 662 • courts.gov.bc.ca:appeal

663 These are strictly examples and do not necessarily indicate actual courts.

664 **3.3.1.4 Document Identifiers**

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

666 **3.3.1.5 Filing Identifiers**

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

669 **3.3.1.6 MDE Identifiers**

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

672 **3.3.1.7 Filer and Party Identifiers**

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

676
677 `<nc:PersonOtherIdentification>`
678 `<nc:IdentificationID>100<nc:IdentificationID>`
679 `<nc:IdentificationCategoryText>ECFFilerID</nc:IdentificationCategoryText>`
680 `</nc:PersonOtherIdentification>`

681

682 In addition to `<nc:PersonOtherIdentification>`, other elements that may contain a filer identifier include
683 `<nc:OrganizationOtherIdentification>`, `<ecf:FilingPartyID>` and `<ecf:FilingAttorneyID>`.

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

688 **3.3.2 Code Lists**

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

691

- 692 • ECF Code Lists
 - 693 • [Bankruptcy Case Type](#)
 - 694 • `<DebtorTypeCode>*`
 - 695 • `<EstimatedAssetsValueLevelCode>*`
 - 696 • `<EstimatedDebtsValueLevelCode>*`
 - 697 • `<NatureOfDebtCode>*`
 - 698 • `<NumberOfCreditorsValueLevelCode>*`
 - 699 • [Common Types](#)
 - 700 • `<FilingStatusCode>*`
 - 701 • [Court Policy Response Message](#)
 - 702 • `<MajorDesignElementNameCode>`
 - 703 • `<OperationNameCode>`
 - 704 • [Service Receipt Message](#)
 - 705 • `<ServiceStatusCode>*`
- 706 • NIEM Code Lists
 - 707 • [ANSI NIST](#)
 - 708 • `<FingerPositionCode>`

- 709 • **JXDM**
- 710 • <ChargeNCICCode>
- 711 • <DrivingIncidentHazMatCode>
- 712 • <DrivingJurisdictionAuthorityNCICLSTACode>
- 713 • <IdentificationJurisdictionNCICLISCode>
- 714 • <WarrantExtraditionLimitationCode>
- 715 • **NIEM Core**
- 716 • <DocumentLangageCode>
- 717 • <DriverLicenseCommercialClassCode>
- 718 • <DrivingRestrictionCode>
- 719 • <LanguageCode>
- 720 • <LengthUnitCode>
- 721 • <LocationCountryFIPS10-4Code>
- 722 • <LocationCountyCode>
- 723 • <LocationStateUSPostalServiceCode>
- 724 • <PersonCitizenshipFIPS10-4Code>
- 725 • <PersonEthnicityCode>
- 726 • <PersonEyeColorCode>
- 727 • <PersonHairColorCode>
- 728 • <PersonRaceCode>
- 729 • <PersonSexCode>
- 730 • <PersonUnionCategoryCode>
- 731 • <PhysicalFeatureCategoryCode>
- 732 • <VehicleColorPrimaryCode>
- 733 • <VehicleMakeCode>
- 734 • <VehicleModelCode>
- 735 • <VehicleStyleCode>
- 736 • <WeightUnitCode>

738 Code lists defined using Genericode 1.0 are indicated with asterisks (*). The remaining code lists are defined in
 739 XSD schema definitions.
 740

741 **3.3.3 Message-Specific Business Rules**

742 The following business rules apply to specific messages:

743 **3.3.3.1 CoreFilingMessage**

744 A CoreFilingMessage MUST express the name or names of the party or parties on whose behalf a document is
 745 filed, and the party whose document is the subject of a responsive document being submitted for filing. If a case

746 refers to a single element using the legal term “In Re,” the filer SHOULD use the NIEM
747 <j:CaseRespondentParty>, not the <j:CaseInitiatingParty> element.

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

754 3.3.3.2 FilingPaymentMessage

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

759 3.3.3.3 RecordDocketingMessage

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

762 3.4 Filing the Record on Appeal

763 This section describes the process for filing and subsequently amending the Record on Appeal (ROA) using ECF
764 4.0.

- 765
- 766 • All ROA transactions, either the original filing or subsequent amendments, MUST contain, as the lead
767 document, an Index of Record document that itemizes the content of the record on appeal.³
- 768
- 769 • The documents that comprise the ROA transaction will be identified as supporting documents.
- 770
- 771 • The supporting documents that comprise the ROA transaction MAY also have additional attached docu-
772 ments.
- 773
- 774 • All ROA documents being submitted, including the Index of Record document and each document within
775 the record, MUST have at least one court-defined document type that indicates the type of transaction to be
776 performed on the document, and whether the document is being added to or stricken from the record.
- 777
- 778 • The Index of Record document and each document within the ROA transaction MAY also have an addi-
779 tional document type or types, which characterize the document for the Court Record MDE.
- 780
- 781 • When a document within the ROA transaction is being stricken from the court record, the document MUST
782 be identified by the unique document identifier, which was provided by the Court Record MDE when the
783 document was initially filed (See section 3.3.1.4).
- 784

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

- 785
- 786
- 787
- 788
- 789
- 790
- 791
- 792
- 793
- 794
- 795
- 796
- 797
- 798
- 799
- 800
- 801
- 802
- 803
- 804
- 805
- 806
- 807
- 808
- 809
- 810
- 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 rejected, there will be ramifications for other pending amendment transactions for the same ROA in the same target case.⁵
 - While an ROA transaction is awaiting acceptance or rejection in the destination court, and when the target case consists of multiple records, courts are cautioned against, but not prohibited from, sending additional amendment transactions intended for the same record for the same target case.
 - Individual documents within the ROA transaction **MUST** not be individually accepted or rejected. All documents within the ROA transaction **MUST** have the same acceptance or rejection disposition.

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).

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.

811

4 ECF 4.0 Schemas

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

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

818

4.1 ECF 4.0 Case Type Schemas

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

822

AppellateCase

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

BankruptcyCase

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

CitationCase

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

CivilCase

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

CriminalCase

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

DomesticCase

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

JuvenileCase

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

837

4.2 ECF 4.0 Common Schemas

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

841

AppInfo

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

CommonTypes

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

DigitalSignature

847 [xsd/common/xmldsig-core-schema.xsd](#)

GenericCode

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

850 4.3 ECF 4.0 Constraint and Subset Schemas

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

858 4.4 ECF 4.0 Message Schemas

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

861 **CaseListQueryMessage**

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

863 **CaseListResponseMessage**

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

865 **CaseQueryMessage**

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

867 **CaseResponseMessage**

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

869 **CoreFilingMessage**

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

871 **CourtPolicyQueryMessage**

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

873 **CourtPolicyResponseMessage**

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

875 **DocumentQueryMessage**

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

877 **DocumentResponseMessage**

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

879 **FeesCalculationQueryMessage**

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

881 **FeesCalculationResponseMessage**

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

883 **FilingListQueryMessage**

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

885 **FilingListResponseMessage**

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

887 **FilingStatusQueryMessage**

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

889 **FilingStatusResponseMessage**

890 [xsd/message/ECF-4.0-FilingStatusResponseMessage.xsd](#)

892 **MessageReceiptMessage**
893 [xsd/message/ECF-4.0-MessageReceiptMessage.xsd](#)
894 **PaymentMessage**
895 [xsd/message/ECF-4.0-PaymentMessage.xsd](#)
896 **PaymentReceiptMessage**
897 [xsd/message/ECF-4.0-PaymentReceiptMessage.xsd](#)
898 **RecordDocketingCallbackMessage**
899 [xsd/message/ECF-4.0-RecordDocketingCallbackMessage.xsd](#)
900 **RecordDocketingMessage**
901 [xsd/message/ECF-4.0-RecordDocketingMessage.xsd](#)
902 **ReviewFilingCallbackMessage**
903 [xsd/message/ECF-4.0-ReviewFilingCallbackMessage.xsd](#)
904 **ServiceInformationQueryMessage**
905 [xsd/message/ECF-4.0-ServiceInformationQueryMessage.xsd](#)
906 **ServiceInformationResponseMessage**
907 [xsd/message/ECF-4.0-ServiceInformationResponseMessage.xsd](#)
908 **ServiceReceiptMessage**
909 [xsd/message/ECF-4.0-ServiceReceiptMessage.xsd](#)
910

5 Service Interaction Profiles

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

5.1 Service Interaction Profile Requirements

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

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

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

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

1. **Message non-repudiation** – A service interaction profile **SHOULD** provide a mechanism so that the receiving MDE is provided with evidence that demonstrates:
 - a. the identity of the sending MDE
 - b. the content of the message(s) transmitted
 - c. the date and time of the message transmission
2. **Message integrity** – A service interaction profile **SHOULD** provide a mechanism so that the receiving MDE is able to determine whether the message(s) transmitted (including any attachments) was (were) modified during the message transmission.
3. **Message confidentiality** – A service interaction profile **SHOULD** provide a mechanism, such as encryption, that can be used with a sending MDE to ensure that the message(s) in a transmission (including any attachments) can be processed only by the receiving MDE.

- 954 4. **Message authentication** – A service interaction profile SHOULD provide a mechanism, such that a sending
955 MDE is required to include, to display credentials that demonstrate its identity to the receiving MDE in each
956 message transmission.
- 957 5. **Message transmission reliability** – A service interaction profile SHOULD provide a mechanism, such that a
958 sending MDE is required to include, to guarantee that a message transmission will be delivered to the receiving
959 MDE within a specified period of time, or else the sending MDE will receive notification at the end of that period
960 of time that the message transmission was not deliverable to the receiving MDE.
- 961 6. **Message splitting and assembly** – A service interaction profile SHOULD provide a mechanism by which a
962 large message and attachments MAY be split into multiple pieces that are transmitted separately by the
963 sending MDE and reassembled into the complete message by the receiving MDE. In the HTTP 1.1 protocol,
964 this is called “chunking.”
- 965 7. **Transmission auditing** – A service interaction profile SHOULD provide a mechanism for the MDE to receive
966 message transmissions in their entirety (both messaging and “payload” content) for auditing purposes.

967 5.2 Service Interaction Profile Approval and Revision Processes

968 The ECF Technical Committee (TC) will recommend certain service interaction profiles for use in implementations
969 of the ECF 4.0 specification. The TC will consider a service interaction profile for recommendation for use in ECF
970 4.0 implementations provided the profile meets the following requirements:

- 971 1. The service interaction profile MUST be described in a document in the format of an OASIS specification.
- 972 2. The service interaction profile specification MUST identify a unique URI to identify the service interaction profile
973 and version.
- 974 3. The service interaction profile specification MUST describe the binding of MDE operations to the service
975 interaction profile that satisfies the functional requirements described in Section 3 (“ECF 4.0 Process Model”)
976 and Section 4 (“ECF 4.0 Schema”) of this specification.
- 977 4. The service interaction profile specification MUST demonstrate that the service interaction profile satisfies the
978 non-functional service interaction profile requirements described in Section 5.1 (“Service Interaction Profile
979 Requirements”) of this specification.
- 980 5. The service interaction profile specification MUST include samples that demonstrate how the messaging
981 information and “payload” content are combined into message transmissions. These samples MUST include
982 samples that demonstrate both synchronous and asynchronous mode operations.
- 983 6. At least one voting member of the ECF TC MUST agree to sponsor the service interaction profile and submit
984 the service interaction profile specification to the TC for review as a candidate for approval as an ECF 4.0
985 compliant service interaction profile.

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

991 From time to time, it may be necessary to revise or update a service interaction profile to bring it into compliance
992 with changes in network and messaging protocols, or to support additional non-functional requirements. Any
993 revision(s) to previously approved service interaction profiles will be considered a new service interaction profile and
994 MUST meet the requirements of a new service interaction profile, including sponsorship by a voting member of the
995 ECF TC and review and approval by the ECF TC. There will be no guarantees that future versions of a service
996 interaction profile will be backwardly compatible with the current version.

997 5.3 Supported Service Interaction Profiles

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

- 1000 • **Web Services Service Interaction Profile 2.0 Specification** – This specification defines a transmission
1001 system using the specifications described in the Web Services Interoperability (WS-I) Basic Profile 1.1, W3C
1002 SOAP 1.1 Binding for MTGOM 1.0, WS-I Basic Security Profile 1.0 and OASIS WS-Reliable Messaging 1.1.
- 1003 • **Web Services Service Interaction Profile 2.1 Specification** – This specification defines a transmission
1004 system using the specifications described in the Web Services Interoperability (WS-I) Basic Profile 1.1, W3C
1005 SOAP 1.1 Binding for MTGOM 1.0 and WS-I Basic Security Profile 1.1 and OASIS WS-Reliable Messaging 1.1.
- 1006 • **Portable Media Service Interaction Profile 1.01 Specification** – This specification defines a transmission
1007 system in which the sending MDE stores message transmissions on portable media (e.g., a compact disc),
1008 which is then physically transported to the receiving MDE where it is connected for retrieval of the message
1009 transmissions. This specification may be needed in the absence of an active network between the sending and
1010 receiving MDEs.

1011 Additional service interaction profiles, or revisions to these service interaction profiles, may be approved by the ECF
1012 TC for use in conjunction with implementations of the ECF 4.0 specification according to the process described in
1013 Section 5.2 (“Service Interaction Profile Approval and Revision Processes”) above.

6 Document Signature Profiles

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

6.1 Document Signature Profile Requirements

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

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

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

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

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

6.2 Document Signature Profile Approval and Revision Processes

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

1. The document signature profile MUST be described in a document in the format of an OASIS specification.
2. The document signature profile specification MUST identify a unique URI to identify the document signature profile and version.
3. If the document signature is not embedded in the document, the document signature profile specification MUST include an XML structure for describing precisely how the document signature is represented.
4. The document signature profile specification MUST demonstrate that the document signature profile satisfies the non-functional requirements described in Section 6.1 (“Document Signature Profile Requirements”) of this specification.
5. The document signature profile specification MUST include samples that demonstrate how the document signature information and “payload” content are combined into message transmissions.
6. At least one voting member of the ECF TC MUST agree to sponsor the document signature profile and submit the document signature profile specification to the TC for review as a candidate for approval as an ECF 4.0 document signature profile.

1057 Certifying that a candidate document signature profile meets certain document signature profile requirements will
1058 necessarily involve some subjectivity, since document signature profile requirements cannot be expressed
1059 algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to assess whether the proposed
1060 profile's description is adequate to the requirements before approving the profile specification as a Committee Draft
1061 through the OASIS standards approval process.

1062 From time to time, it may be necessary to revise or update a document signature profile to bring it into compliance
1063 with changes in authentication and encryption protocols, or to support additional non-functional requirements. Any
1064 revision(s) to previously approved document signature profiles will be considered a new document signature profile
1065 and MUST meet the requirements of a new document signature profile, including sponsorship by a voting member
1066 of the ECF TC and review and approval by the ECF TC. There will be no guarantees that future versions of
1067 document signature profiles will be backwardly compatible with the current version.

1068 6.3 Supported Document Signature Profiles

1069 The following ECF 4.0 document signature profile specifications are candidate Committee Drafts for use in
1070 conjunction with implementations of the ECF 4.0 specification:

- 1071 • **Null Document Signature Profile 1.0 Specification** – This specification defines a default mechanism to
1072 describe documents that do not have any associated signatures.
- 1073 • **XML Document Signature Profile 1.0 Specification** – This specification defines a mechanism for associating
1074 a W3C XML Signature with a document.
- 1075 • **Application-Specific Document Signature Profile 1.0 Specification** – This specification defines a
1076 mechanism for embedding an application-specific binary signature with a document. This profile supports the
1077 native capabilities in document formats such as Microsoft Word and the Adobe Portable Document Format
1078 (PDF) for describing and embedding signatures.
- 1079 • **Proxy Document Signature Profile 1.0 Specification** – This specification defines a mechanism for indicating
1080 documents that are digitally signed by a court filing infrastructure component on behalf of an authenticated
1081 signer.
- 1082 • **Symmetric Key Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1083 indicating documents that are digitally signed by a trusted entity on behalf of the signer using a symmetric key
1084 known only to the trusted entity.

1085 Additional document signature profiles, or revisions to these document signatures profiles, may be approved by the
1086 ECF TC for use in conjunction with implementation of the ECF 4.0 specification according to the process described
1087 in Section 6.2 ("Document Signature Profile Approval and Revision Processes") above.

7 Conformance

1088

1089

1090

An implementation conforms with the Electronic Court Filing Version 4.01 if the implementation meets the requirements in Sections 1-6 including conformance with the referenced XSD schemas and Genericcode code lists.

1091 Appendix A. (Informative) Release Notes

1092 A.1 Availability

1093 Online and downloadable versions of this release are available from the locations specified at the top of this
1094 document.

1095 A.2 Package Structure

1096 The ECF 4.0 specification is published as a ZIP archive named `ecf-v4.0.zip`. Unzipping this archive creates a
1097 directory named `ecf-4.0/` containing this specification document and a number of subdirectories. The files in
1098 these subdirectories, linked to the specification document, contain the various normative and informational pieces of
1099 the 1.0 release. A description of each subdirectory is given below.

1100 `gc/`

1101 Genericcode 1.0 code lists

1102 `model/`

1103 ECF 4.0 UML exchange content model diagrams and spreadsheet models; see Appendix B.3 and B4

1104 `xml/`

1105 Example instances; see Appendix D

1106 `xsd/`

1107 XSD schemas; see Section 4

1108 A.3 Recursive Structures

1109 Certain components in the **[NIEM]** version 2.0 schemas allow recursive nesting. For example, a `nc:Case` may be
1110 related to another `nc:Case`, etc. These are legitimate business data structures. Most real-world applications will
1111 limit the depth of recursion in such structures, but XSD schemas are incapable of expressing this constraint.
1112 Implementers should be aware of this and may wish to set limits on the depth of recursive structures in their
1113 applications.

1114 A.4 Date and Time Formats

1115 The date and time elements contained in the messages defined by the ECF 4.0 XSD schemas should be formatted
1116 according to the documentation in the **[NIEM]** version 2.0. The **[NIEM]** documentation indicates the following:

- 1117 • Calendar date values should be expressed as “CCYY-MM-DD”, with an optional time zone qualifier designated
1118 by appending -hh:00, where hh represent the number of hours the local time zone is behind Coordinated
1119 Universal Time (UTC).
- 1120 • Time values should be expressed as “hh:mm:ss.sss”, with an optional time zone qualifier designated by
1121 appending -hh:00, where hh represent the number of hours the local time zone is behind Coordinated Universal
1122 Time (UTC).
- 1123 • Date and time values should be expressed as “CCYY-MM-DDThh:mm:ss.sss” with an optional time zone
1124 designated by appending -hh:00, where hh represent the number of hours the local time zone is behind Coordi-
1125 nated Universal Time (UTC).qualifier.

1126 These formats are documented in, but not enforced by, the XSD schema at
1127 `xsd/constraint/niem/proxy/xsd/2.0/xsd.xsd`.

1128 A.5 Known Errata

1129 Known errors in the ECF 4.0 specification will be identified in an errata document available at:

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

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

B.1 Principles

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

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

B.2 Approach

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

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

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

B.3 ECF 4.0 Exchange Content Models

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

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

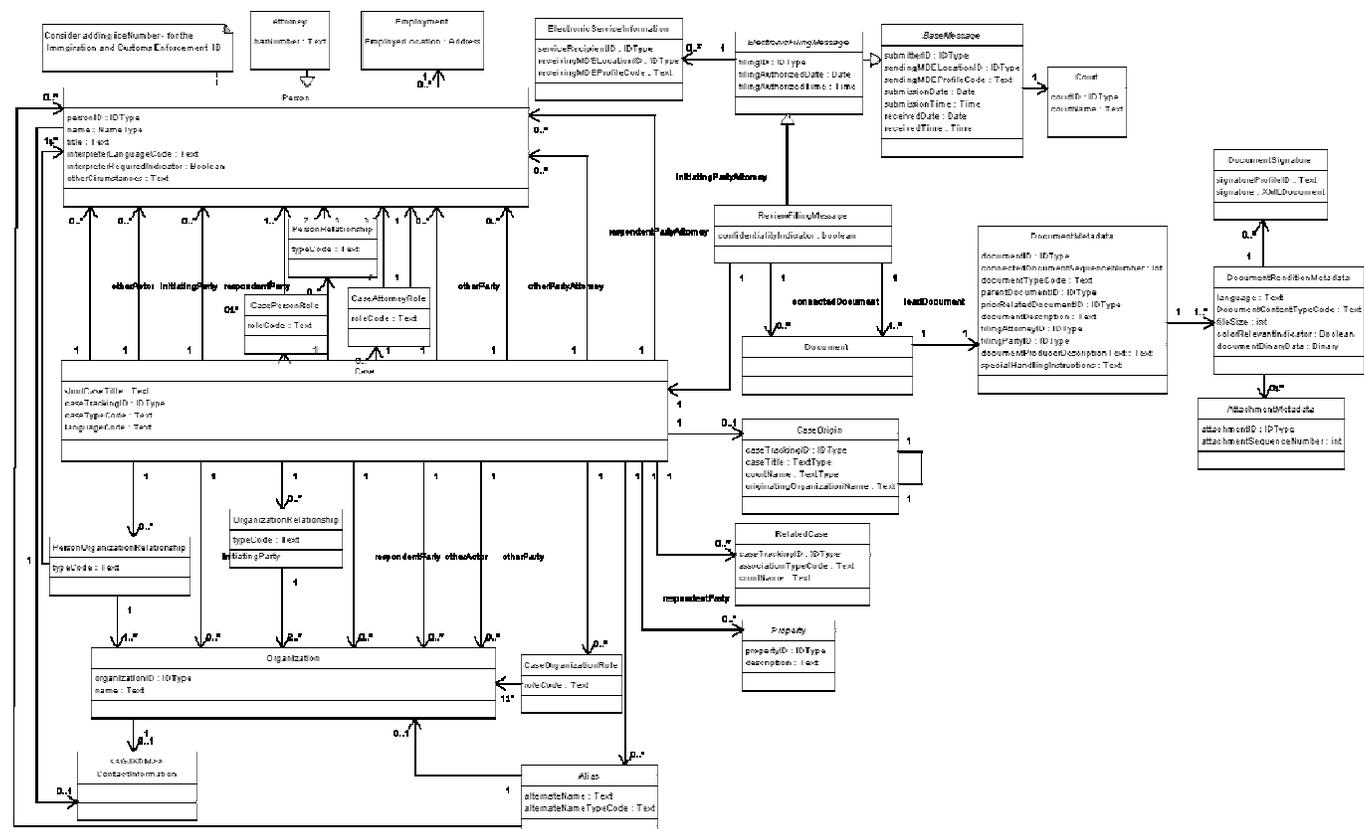
The exchange content models are used for the following purposes:

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

1169
1170
1171
1172
1173

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

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



1174
1175
1176

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

- 1192 [model/uml/html/DomesticFiling.png](#)
- 1193 **Extended Person Information Model**
- 1194 [model/uml/html/ExtendedPersonInformation.png](#)
- 1195 **Get Calculated Fees Query Model**
- 1196 [model/uml/html/GetFeesCalculationQuery.png](#)
- 1197 **Get Case List Query Model**
- 1198 [model/uml/html/GetCaseListQuery.png](#)
- 1199 **Get Document Query Model**
- 1200 [model/uml/html/GetDocumentQuery.png](#)
- 1201 **Get Filing List Query Model**
- 1202 [model/uml/html/GetFilingListQuery.png](#)
- 1203 **Get Filing Status Query Model**
- 1204 [model/uml/html/GetFilingStatusQuery.png](#)
- 1205 **Get Service Information Query Model**
- 1206 [model/uml/html/GetServiceInformationQuery.png](#)
- 1207 **Major Design Elements Model**
- 1208 [model/uml/html/MajorDesignElements.png](#)
- 1209 **Juvenile Filing Model**
- 1210 [model/uml/html/JuvenileFiling.png](#)
- 1211 **Record Docketing Model**
- 1212 [model/uml/html/RecordDocketing.png](#)
- 1213 **Review Filing Model**
- 1214 [model/uml/html/ReviewFiling.png](#)

1215

1216 No specific directions are defined for the associations in these models; they can be navigated in either direction.

1217 The specific navigation path for each association is defined when documents are assembled.

1218 **B.4 Spreadsheet Models**

1219 ECF 4.0 uses spreadsheet models to describe the mapping of objects and attributes to **[NIEM]** and ECF 4.0

1220 elements. The spreadsheet models use rows to define components. Components are either simple data types or

1221 associations. Columns define the metadata associated with each component type.

1222 The ECF 4.0 spreadsheet model is located at [mod/ECF-4.0-NIEM-mapping.xls](#).

1223

Appendix C. (Informative) MDE Operations

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.

C.1 Filing Assembly MDE

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.

C.1.1 Provided Operations

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

Operation	Called By	Output	Parameters
NotifyFilingReview-Complete	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

C.1.2 Consumed Operations

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
-------------	-------------------	--

C.2 Filing Review MDE

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

C.2.1 Provided Operations

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
NotifyDocketing-Complete	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

C.2.2 Consumed Operations

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

1250

C.3 Court Record MDE

1251

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

1252

1253

C.3.1 Provided Operations

1254

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
			xsd/message/ECF-4.0-CaseTypeSpecificMessage.xsd : CaseTypeSpecificMessage
			xsd/message/ECF-4.0-CaseTypeSpecificMessage.xsd : CourtSpecificMessage
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

1255

C.3.2 Consumed Operations

1256

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

1257

C.4 Legal Service MDE

1258

The Legal Service MDE enables a filer or a court to electronically transmit copies of, or links to, electronically filed documents to other parties who are participating in the case and who are entitled to be promptly served with the electronically filed documents. The Filing Assembly MDE transmits data and documents to the Legal Service MDE to inform the case participant that an electronic filing has been submitted to the court clerk. The Legal Service MDE transmits a callback message to the Filing Assembly MDE requesting a notification to confirm receipt of the served document.

1259

1260

1261

1262

1263

1264

C.4.1 Provided Operations

1265

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

1266

C.4.2 Consumed Operations

1267

The Legal Service MDE does not call operations in other MDEs

Appendix D. (Informative) Example Instances

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

FeesCalculationQueryMessage

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

FeesCalculationResponseMessage

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

CaseListQueryMessage

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

CaseListResponseMessage

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

CaseQueryMessage

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

CaseResponseMessage

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

CoreFilingMessage (Appellate case type)

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

CoreFilingMessage (Criminal case type)

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

CourtPolicyQueryMessage

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

CourtPolicyResponseMessage

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

DocumentQueryMessage

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

DocumentResponseMessage

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

FilingListQueryMessage

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

FilingListResponseMessage

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

FilingPaymentMessage

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

FilingStatusQueryMessage

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

FilingStatusResponseMessage

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

MessageReceiptMessage

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

- 1307 **PaymentReceiptMessage**
- 1308 [xml/ECF-4.0-PaymentReceiptMessage.xml](#)
- 1309 **RecordDocketingCallbackMessage**
- 1310 [xml/ECF-4.0-RecordDocketingCallbackMessage.xml](#)
- 1311 **RecordDocketingMessage**
- 1312 [xml/ECF-4.0-RecordDocketingMessage.xml](#)
- 1313 **ReviewFilingCallbackMessage**
- 1314 [xml/ECF-4.0-ReviewFilingCallbackMessage.xml](#)
- 1315 **ServiceInformationQueryMessage**
- 1316 [xml/ECF-4.0-ServiceInformationQueryMessage.xml](#)
- 1317 **ServiceInformationResponseMessage**
- 1318 [xml/ECF-4.0-ServiceInformationResponseMessage.xml](#)
- 1319 **ServiceReceiptMessage**
- 1320 [xml/ECF-4.0-ServiceReceiptMessage.xml](#)

1321 Appendix E. (Informative) Ongoing Work Items

1322 The Electronic Court Filing TC plans to continue to revise and expand this specification through future versions.
1323 Future versions of ECF will:

- 1324 • Address filings in administrative tribunals
- 1325 • Address primary service (the delivery of documents such as summonses, subpoenas and warrants that
1326 establish a court's jurisdiction over a party)
- 1327 • Consider how the specifications for filing of documents intended for filing with a court relate to specifications for
1328 filing other documents, e.g., property records, in the offices of elected clerks of courts
- 1329 • Incorporate feedback from ECF implementations
- 1330 • Support future releases of the **[NIEM]**
- 1331 • Support future **[Court Document]** specifications and integration optimizations
- 1332 • Support non-case related filings into a court clerk's office

Appendix F. (Informative) Acknowledgments

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

- Administrative Office of United States Courts
 - Bankruptcy, Civil, Criminal
- Arizona Administrative Office of the Courts
 - Appellate
- Fourth Judicial District Court, Hennepin County, Minneapolis
 - Criminal
- King County Superior Court, Washington
 - Civil, Criminal
- Missouri Office of State Courts Administrator
 - Citation
- Thirteenth Judicial District, Orange County, Florida (through vendor)
 - Civil, Criminal, Domestic relations, Mental health, Juvenile delinquency/dependency, Probate, Citation
- Utah State Courts
 - Civil, Criminal

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

- Rolly Chambers, American Bar Association
- John Messing, American Bar Association
- Adam Angione, Courthouse News Service
- Eric Eastman Doxpop, LLC
- Robert DeFilippis, Associate
- Chester Ensign, Associate
- Gary Poindexter, Associate
- Michael Alexandrou, Judicial Council of Georgia
- Shawn Artrip, Judicial Council of Georgia,
- Robbie Diaz, Judicial Council of Georgia
- Hui Ji, Judicial Council of Georgia
- Morgan Medders, Judicial Council of Georgia
- Scott Edson, LA County Information Systems Advisory Body
- Ali Farahani, LA County Information Systems Advisory Body
- John Ruegg, LA County Information Systems Advisory Body
- CJ Allen, Maricopa County
- Robin Gibson, Missouri Office of State Courts Admin
- James Cabral, MTG Management Consultants, LLC
- Thomas Clarke, National Center for State Courts
- Diana Graski, National Center for State Courts
- Jim Harris, National Center for State Courts
- Jason Hill, New York State Office of Court Administration, DoT
- Robert O'Brien, Ottawa Courts Administration Service
- George Knecht, PC Intellect, LLC
- Mark Ladd, Property Records Industry Assn.
- Ron Bowmaster, Utah Administrative Office of the Courts

1380

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.

1381