



Electronic Court Filing Version 4.1

Committee Specification 01

29 September 2023

This stage:

<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/ecf-v4.1-cs01.docx> (Authoritative)
<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/ecf-v4.1-cs01.html>
<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/ecf-v4.1-cs01.pdf>

Previous stage:

<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd02/ecf-v4.1-csd02.docx> (Authoritative)
<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd02/ecf-v4.1-csd02.html>
<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd02/ecf-v4.1-csd02.pdf>

Latest stage:

<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/ecf-v4.1.docx> (Authoritative)
<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/ecf-v4.1.html>
<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/ecf-v4.1.pdf>

Technical Committee:

OASIS LegalXML Electronic Court Filing TC

Chair:

James Cabral (jim.cabral@infotrack.com), InfoTrack US

Editors:

James Cabral (jim.cabral@infotrack.com), InfoTrack US

Gary Graham (GGraham@courts.az.gov), Arizona Supreme Court

Philip Baughman (Philip.Baughman@tylertech.com), Tyler Technologies, Inc.

Additional artifacts:

This document is one component of a Work Product that also includes:

- XML schemas: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/xsd/>.
- XML sample messages: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/xml/>.
- Model and documentation: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/model/>.
- Generic code lists: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/gcl/>.
- Specification metadata: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/xsd/metadata.xml>.

Related work:

This specification replaces or supersedes:

- *LegalXML Electronic Court Filing 3.0*. Edited by Roger Winters. 15 November 2005.
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v3.0/ecf-v3.0-spec-cd01.zip>.
- *Electronic Court Filing Version 4.0*. Edited by Adam Angione and Roger Winters. Latest stage:
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.0/ecf-v4.0-spec/ecf-v4.0-spec.html>.
- *Electronic Court Filing Version 4.01*. Edited by Adam Angione and James Cabral. Latest stage:
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/ecf-v4.01-spec.html>.
- *Electronic Court Filing Version 4.01 Errata 01*. Edited by James Cabral and Gary Graham. 14 July 2014. OASIS Approved Errata. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata01/os/ecf-v4.01-spec-errata01-os.html>.

- *Electronic Court Filing Version 4.01 Errata 02*. Edited by James Cabral and Gary Graham. 07 July 2015. OASIS Approved Errata. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/os/ecf-v4.01-spec-errata02-os.html>.

This specification is related to:

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

Declared XML namespaces:

- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:AppInfo-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:AppellateCase-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:BankruptcyCase-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseListQueryMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseListResponseMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseQueryMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CaseResponseMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CitationCase-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CivilCase-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CommonTypes-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CoreFilingMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CourtPolicyQueryMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CourtPolicyResponseMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:CriminalCase-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:DocumentQueryMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:DocumentResponseMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:DomesticCase-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FeesCalculationQueryMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FeesCalculationResponseMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FilingListQueryMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FilingListResponseMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FilingStatusQueryMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:FilingStatusResponseMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:JuvenileCase-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:MessageReceiptMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:PaymentMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:PaymentReceiptMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:RecordDocketingCallbackMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:RecordDocketingMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:ReviewFilingCallbackMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:ServiceInformationQueryMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:ServiceInformationResponseMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:ServiceReceiptMessage-4.1
- urn:oasis:names:tc:legalxml-courtfilling:schema:xsd:MessageWrappers-4.1

Abstract:

This document defines the LegalXML Electronic Court Filing 4.1 (ECF 4.1) 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 4.1 is a maintenance release to address several minor schema and definition issues identified by implementers of the ECF 4.0 and 4.01 specifications.

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 stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=legalxml-courtfilling#technical.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "[Send A Comment](#)" button on the TC's web page at <https://www.oasis-open.org/committees/legalxml-courtfilling/>.

This specification is provided under the [RF on Limited Terms](#) Mode of the [OASIS IPR Policy](#), the mode chosen when the Technical Committee was established. 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 TC's web page (<https://www.oasis-open.org/committees/legalxml-courtfilling/ipr.php>).

Note that any machine-readable content ([Computer Language Definitions](#)) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Citation format:

When referencing this document, the following citation format should be used:

[ECF-v4.1]

Electronic Court Filing Version 4.1. Edited by James Cabral, Gary Graham, and Philip Baughman. 29 September 2023. OASIS Committee Specification 01. <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/cs01/ecf-v4.1-cs01.html>. Latest stage: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/ecf-v4.1.html>.

Notices:

Copyright © OASIS Open 2023. All Rights Reserved.

Distributed under the terms of the OASIS IPR Policy, [<https://www.oasis-open.org/policies-guidelines/ipr/>]. For complete copyright information please see the full Notices section in Appendix G below.

Table of Contents

1	Introduction.....	7
1.1	Scope.....	7
1.2	Relationship to Prior Specifications	8
1.3	ECF Version 4.1	8
1.3.1	National Information Exchange Model (NIEM).....	9
1.3.2	OASIS Universal Business Language.....	9
1.3.3	W3C XML-Signature Syntax and Processing	9
1.3.4	OASIS Reference Model for Service Oriented Architecture	10
1.3.5	OASIS Code List Representation (Genericode).....	10
1.4	Terms and Definitions.....	10
1.5	Symbols and Abbreviations	11
1.6	Normative References	12
1.7	Non-Normative References	13
2	ECF 4.1 Architecture	14
2.1	Core vs. Profiles	14
2.2	Major Design Elements.....	14
2.3	Information Model	15
2.3.1	Messages	15
2.3.2	Attachment	16
2.3.3	Sample Message Streams	16
2.4	Court Policy.....	19
2.4.1	Human-Readable Court Policy.....	19
2.4.2	Machine-Readable Court Policy.....	19
2.4.3	Case-Type and Court Extensions	20
2.4.4	Court-Specific Code Lists.....	20
2.4.5	Court-Specific Constraint Schemas	21
3	ECF 4.1 Process Model	22
3.1	The Filing-Preparation-to-Docketing Process Model.....	22
3.2	Business Rules	23
3.2.1	GetPolicy	23
3.2.2	GetServiceInformation.....	23
3.2.3	GetFeesCalculation	24
3.2.4	ReviewFiling	24
3.2.5	ServeFiling.....	24
3.2.6	RecordFiling	24
3.2.7	NotifyDocketingComplete	25
3.2.8	NotifyFilingReviewComplete	25
3.2.9	GetFilingList.....	25
3.2.10	GetFilingStatus.....	25
3.2.11	GetCaseList.....	25
3.2.12	GetCase	26
3.2.13	GetDocument	26
3.3	Message Business Rules	26
3.3.1	Identifiers	26

3.3.1.1	Attachment Identifiers	26
3.3.1.2	Case Identifiers	26
3.3.1.3	Court Identifiers.....	26
3.3.1.4	Document Identifiers	26
3.3.1.5	Filing Identifiers.....	26
3.3.1.6	MDE Identifiers	27
3.3.1.7	Asynchronous responses.....	27
3.3.1.8	Filer and Party Identifiers	27
3.3.2	Code Lists.....	27
3.3.3	Message-Specific Business Rules	29
3.3.3.1	CoreFilingMessage	29
3.3.3.2	PaymentMessage	29
3.3.3.3	RecordDocketingMessage.....	29
3.4	Filing the Record on Appeal	29
4	ECF 4.1 Schemas	32
4.1	ECF 4.1 Case Type Schemas	32
4.2	ECF 4.1 Common Schemas	32
4.3	ECF 4.1 Constraint and Subset Schemas.....	33
4.4	ECF 4.1 Message Schemas.....	33
5	MDE Operations	35
5.1	Filing Assembly MDE.....	35
5.1.1	Provided Operations.....	35
5.1.2	Consumed Operations	35
5.2	Filing Review MDE	36
5.2.1	Provided Operations.....	36
5.2.2	Consumed Operations	36
5.3	Court Record MDE	37
5.3.1	Provided Operations.....	37
5.3.2	Consumed Operations	37
5.4	Legal Service MDE	38
5.4.1	Provided Operations.....	38
5.4.2	Consumed Operations	38
6	Service Interaction Profiles.....	39
6.1	Service Interaction Profile Requirements	39
6.2	Service Interaction Profile Approval and Revision Processes.....	40
6.3	Supported Service Interaction Profiles	41
7	Document Signature Profiles.....	42
7.1	Document Signature Profile Requirements	42
7.2	Document Signature Profile Approval and Revision Processes.....	42
7.3	Supported Document Signature Profiles	43
8	Conformance	44
Appendix A.	(Informative) Release Notes.....	45
A.1	Availability.....	45
A.2	Package Structure	45
A.3	Recursive Structures	45
A.4	Date and Time Formats.....	45
A.5	Known Errata.....	46

Appendix B. (Informative) ECF 4.1 Development Approach and Artifacts	47
B.1 Principles	47
B.2 Approach	47
B.3 ECF 4.1 Exchange Content Models	47
B.4 Spreadsheet Models	49
Appendix C. (Informative) Example Instances.....	50
Appendix D. (Informative) Ongoing Work Items	52
Appendix E. (Informative) Acknowledgments	53
Appendix F. (Informative) Revision History.....	54
Appendix G. Notices	56

1 Introduction

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

5 1.1 Scope

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

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

15
16 This specification supports the following automated information exchanges:

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

42
43 In addition to filing of court case documents, this specification supports “secondary service” – the delivery
44 of copies of filed documents to persons who have already been made parties to a case. This

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

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

52

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

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

63

64 Although ECF 4.1 does not define data structure elements specific to other case types (e.g.,
65 administrative tribunals), the basic structure will support other types of court filings and is extensible
66 through court-specific and case-type-specific extensions.

67 1.2 Relationship to Prior Specifications

68 Electronic Court Filing 4.0 superseded the LegalXML Electronic Court Filing 3.0, 3.01 and 3.1
69 specifications developed by the predecessor organizations to the OASIS Electronic Court Filing Technical
70 Committee. Those specifications were prepared for and approved by the Conference of State Court
71 Administrators COSCA)/National Association for Court Management (NACM) Joint Technology
72 Committee as proposed standards.

73 Relative to the ECF 3.0, 3.01 and 3.1 specifications, the ECF 4.0, 4.01 and 4.1 specifications provide a
74 number of enhancements including:

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

80

81 This specification does not assume that prior specifications will be deprecated. However, ECF 4.1 is not
82 guaranteed to be backward-compatible with previous versions including ECF 4.0 and 4.01, both based on
83 NIEM 2.x. Applications based on ECF versions which themselves are based on NIEM versions other
84 than NIEM 2.x (such as ECF 3.0, 3.01 and 3.1 specifications) will certainly not interoperate successfully
85 with applications using this specification. This fact is indicated by the assignment of a new major and
86 minor version number to the specifications.

87 1.3 ECF Version 4.1

88 ECF 4.1 is a minor enhancement release to address several minor message and schema issues
89 identified by implementers of the ECF 4.0 and 4.01 specifications. All references in this document to ECF
90 4.0 apply to ECF 4.01 and 4.1 as well.

91 The ECF specification incorporates other existing, non-proprietary XML specifications wherever possible.
92 In particular, the specification has dependencies on the **[NIEM]**, the **[UBL]** data library and the World
93 Wide Web Consortium (W3C) XML Digital Signatures specification. The terminology used in this
94 specification to describe the components of the ECF technical architecture conforms to the OASIS
95 Reference Model for Service Oriented Architecture.

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

101 **1.3.1 National Information Exchange Model (NIEM)**

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

113 The **[NIEM]** is most useful for describing common objects such as persons and locations, and criminal
114 justice-specific processes such as arrest, booking, jail and prosecution. The **[NIEM]** is not as well
115 developed for describing non-criminal information exchanges and processes. ECF 4.1 uses the **[NIEM]**
116 version 2.0 where the structures and definitions correspond to the requirements of ECF 4.1. The
117 development process, including the **[NIEM]** modeling process, is described in Appendix B.

118 **1.3.2 OASIS Universal Business Language**

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

124 ECF 4.1 employs the following structures in the **[UBL]** to describe filing payments and payment receipts:

125 <AllowanceCharge>

126 Information about a charge or discount price component.

127 <Address>

128 Information about a structured address.

129 <Payment>

130 Information directly relating to a specific payment.

131 **1.3.3 W3C XML-Signature Syntax and Processing**

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

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

134 sender and be assured of the validity of the electronically transmitted data. [XMLSIG] defines standard
135 means for specifying information content that is to be digitally signed.²
136 ECF 4.1 employs the [XMLSIG] specification to describe digital signatures applied to the entire ECF 4.1
137 message transmission in order to provide authentication, encryption and message integrity. [XMLSIG] is
138 also used in the ECF 4.1 XML Document Signature Profile.

139 1.3.4 OASIS Reference Model for Service Oriented Architecture

140 The [SOA-RM] is a framework for understanding significant entities, and the relationships between those
141 entities, within a service-oriented architecture. ECF 4.1 describes such an architecture and includes
142 terminology that conforms to the [SOA-RM].

143 1.3.5 OASIS Code List Representation (Genericcode)

144 The OASIS Code List Representation format, [Genericcode], is a model and XML schema that can be
145 used to encode a broad range of code list information. The XML format is designed to support
146 interchange or distribution of machine-readable code list information between systems. All ECF 4.1 code
147 lists that are not defined in the NIEM are provided in [Genericcode] 1.0 format.
148

149 1.4 Terms and Definitions

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

153
154 This section defines key terms used in this specification.
155

156 **Attachment**

157 See definition in Section 2.3.2.

158 **Callback message**

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

161 **Document**

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

164 **Document hash**

165 A condensed representation of a document, calculated according to the FIPS 180-4 SHA 256
166 algorithm.

167 **Docketing**

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

170 **Filer**

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

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

- 173 **Filing**
174 An electronic document (with any associated data, attachments and the like) that has been
175 assembled for the purpose of being filed into a specified court case.
- 176 **Hub Service MDE**
177 A centralized Service MDE capable of receiving a single set of service notifications for all parties
178 registered for electronic service in a case and transmitting the service notifications to the Service
179 MDEs registered to each party in the case.
- 180 **Major Design Element (MDE)**
181 A logical grouping of operations representing a significant business process supported by ECF
182 4.1. Each MDE operation receives one or more messages, returning a synchronous response
183 message (a reaction to a message received) and, optionally, returning an asynchronous (later)
184 response message to the originating message sender.
- 185 **Message**
186 See definition in Section 2.3.1.
- 187 **Message Transmission**
188 The sending of one or more messages and associated attachments to an MDE. Each
189 transmission must invoke or respond to an operation on the receiving MDE, as defined in the
190 ECF 4.1 specification.
- 191 **Operation (or MDE Operation)**
192 A function provided by an MDE upon receipt of one or more messages. The function provided by
193 the operation represents a significant step in the court filing business process. A sender invokes
194 an operation on an MDE by transmitting a request with an operation identifier and a set of
195 messages.
- 196 **Operation signature**
197 A definition of the input message and synchronous response message associated with an
198 operation. Each message is given a name and a type by the operation. The type is defined by a
199 single one of the message structures defined in the ECF 4.1 specification.
- 200 **Synchronous response**
201 A message transmission returned immediately (synchronously) as the result of an operation.
202 Every operation has a synchronous response.

203 1.5 Symbols and Abbreviations

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

205

206 ECF 4.1

207 Electronic Court Filing 4.1

208 IEPD

209 Information Exchange Package Documentation

210 MDE

211 Major Design Element

212 NIEM

213 National Information Exchange Model

214 OASIS

215 Organization for the Advancement of Structured Information Standards

216 **XML**
217 eXtensible Markup Language
218 **W3C**
219 World Wide Web Consortium
220 **WS-I**
221 Web Services Interoperability Organization
222

223 1.6 Normative References

224 **[FIPS 180-4]**
225 *Secure Hash Standard*, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>, National Institute for
226 Standards and Technology, March 2012.
227 **[Genericcode]**
228 G. K. Holman, *Code List Representation (Genericcode) 1.0*, [http://docs.oasis-](http://docs.oasis-open.org/codelist/ns/genericcode/1.0/)
229 [open.org/codelist/ns/genericcode/1.0/](http://docs.oasis-open.org/codelist/ns/genericcode/1.0/), OASIS Committee Specification 02, April 06, 2022.
230 **[NIEM]**
231 *National Information Exchange Model 2.0*, <http://niem.gov>, US DOJ and DHS, 2007.
232 **[NIEM Guide]**
233 *NIEM Implementation Guidelines*, <http://www.niem.gov/implementationguide.php>, US DOJ and DHS,
234 2007.
235 **[NIEM Techniques]**
236 *Techniques for Building and Extending NIEM*, [http://reference.niem.gov/niem/guidance/techniques-for-](http://reference.niem.gov/niem/guidance/techniques-for-building-and-extending/2.0.1/techniques-for-building-and-extending-niem-2.0.1.pdf)
237 [building-and-extending/2.0.1/techniques-for-building-and-extending-niem-2.0.1.pdf](http://reference.niem.gov/niem/guidance/techniques-for-building-and-extending/2.0.1/techniques-for-building-and-extending-niem-2.0.1.pdf), Georgia Tech
238 Research Institute, August 2007.
239 **[Namespaces]**
240 T. Bray, *Namespaces in XML*, <http://www.w3.org/TR/1999/REC-xml-names-19990114>, January 14, 1999.
241 **[RFC2046]**
242 N. Freed, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*,
243 <http://www.ietf.org/rfc/rfc2046.txt>, IETF RFC 2046, November 1996.
244 **[RFC2119]**
245 S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>,
246 IETF RFC 2119, March 1997.
247 **[RFC4122]**
248 Leach, et al., *A Universally Unique IDentifier (UUID) URN Namespace*, <http://www.ietf.org/rfc/rfc4122.txt>,
249 IETF RFC 4122, July 2005.
250 **[Schema Part 1]**
251 H. S. Thompson, D. Beech, M. Maloney, N. Mendelsohn, *XML Schema Part 1: Structures Second Edition*,
252 <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>, W3C Recommendation, October 28, 2004.
253 **[Schema Part 2]**
254 P. Biron, A. Malhotra, *XML Schema Part 2: Datatypes Second Edition*, [http://www.w3.org/TR/2004/REC-](http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/)
255 [xmlschema-2-20041028/](http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/), W3C Recommendation, October 28, 2004
256 **[UBL]** *Universal Business Language Version 2.1. 04 November 2013. OASIS Standard.*
257 <http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html>.
258 **[XML 1.0]**
259 T. Bray, *Extensible Markup Language (XML) 1.0 (Third Edition)*, [http://www.w3.org/TR/REC-xml/REC-](http://www.w3.org/TR/REC-xml/REC-XML-20040204)
260 [XML-20040204](http://www.w3.org/TR/REC-xml/REC-XML-20040204), W3C Recommendation, February 4, 2004.
261 **[XMLENC]**

262 D. Eastlake, J. Reagle, *XML Encryption Syntax and Processing*, [http://www.w3.org/TR/2002/REC-](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/)
263 [xmlenc-core-20021210/](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/), W3C Recommendation, December 2002.
264 **[XMLENSIG]**
265 D. Eastlake, J. Reagle, D. Solo, *XML-Signature Syntax and Processing*, [http://www.w3.org/TR/2002/REC-](http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/)
266 [xmldsig-core-20020212/](http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/), W3C Recommendation, February 2002.

267 **1.7 Non-Normative References**

268 **[Court Document]**

269 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)
270 [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).

271 **[NIEM MNRD]**

272 W. Roberts, S. Liebeskind, M. Kindl *National Information Exchange Model Naming and Design Rules Final*
273 *1.3*, <https://reference.niem.gov/niem/specification/naming-and-design-rules/1.3/niem-ndr-1.3.pdf>, October
274 31, 2008.

275 **[Juvenile XML]**

276 S. Rondendell, et. al., *Juvenile Justice XML Report*,
277 [https://web.archive.org/web/20050910223309/http://www.ijis.org/db/share/public/Library/Publications/ju-](https://web.archive.org/web/20050910223309/http://www.ijis.org/db/share/public/Library/Publications/juvenile_justice_xml_final_report_20050630.pdf)
278 [venile_justice_xml_final_report_20050630.pdf](https://web.archive.org/web/20050910223309/http://www.ijis.org/db/share/public/Library/Publications/juvenile_justice_xml_final_report_20050630.pdf), IJIS Institute, July 2005.

279 **[NIEM]**

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

281 **[NCSC Guide]**

282 *State Court Guide to Statistical Reporting*, [https://www.courtstatistics.org/pub-and-def-second-row-](https://www.courtstatistics.org/pub-and-def-second-row-cards/guide-to-statistical-reporting#:~:text=The%20State%20Court%20Guide%20to%20Statistical%20Reporting%20was%20updated%20in,court%20caseload%20statistics%20in%201975.)
283 [cards/guide-to-statistical-](https://www.courtstatistics.org/pub-and-def-second-row-cards/guide-to-statistical-reporting#:~:text=The%20State%20Court%20Guide%20to%20Statistical%20Reporting%20was%20updated%20in,court%20caseload%20statistics%20in%201975.)
284 [reporting#:~:text=The%20State%20Court%20Guide%20to%20Statistical%20Reporting%20was%20updat-](https://www.courtstatistics.org/pub-and-def-second-row-cards/guide-to-statistical-reporting#:~:text=The%20State%20Court%20Guide%20to%20Statistical%20Reporting%20was%20updated%20in,court%20caseload%20statistics%20in%201975.)
285 [ed%20in,court%20caseload%20statistics%20in%201975.](https://www.courtstatistics.org/pub-and-def-second-row-cards/guide-to-statistical-reporting#:~:text=The%20State%20Court%20Guide%20to%20Statistical%20Reporting%20was%20updated%20in,court%20caseload%20statistics%20in%201975.), National Center for State Courts, January
286 2023.

287 **[Rap Sheet]**

288 *Interstate Criminal History Transmission Specification XML Version 3.00*,
289 http://www.search.org/files/pdf/CH_transmission_spec.pdf, Joint Task Force on Rap Sheet
290 Standardization, February 2005.

291 **[SOA-RM]**

292 MacKenzie, et al., *Reference Model for Service Oriented Architecture 1.0*, [https://www.oasis-](https://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf)
293 [open.org/committees/download.php/19679/soa-rm-cs.pdf](https://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf), OASIS Public Review Draft 1.0, February 10,
294 2006.

295 **[Traffic IEPD]**

296 *Traffic Citation IEPD*, <http://www.ncsc.org/~media/Files/ZIPS/Technology/IEPD/TrafficCitation.ashx>,
297 National Center for State Courts, August 8, 2005.

298

2 ECF 4.1 Architecture

299 The ECF 4.1 architecture consists of four Major Design Elements (MDEs), which support operations and
300 messages. An MDE is a logical grouping of operations, such as the operations involved in creating a
301 filing or the operations involved in receiving and recording a filing, that is, incorporating the constituent
302 documents into a court document management system. A message is the data exchanged between
303 MDEs in the form of an XML document that may include one or more additional binary attachments.
304 These messages contain the information to be filed with the court. This section describes the ECF 4.1
305 architecture including the MDEs, the operations and the messages.

306

2.1 Core vs. Profiles

307 The ECF 4.1 architecture can be divided into three principal elements:

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

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

316 The MDEs and messages that make up the core specification are discussed in Sections 2.2 and 2.3
317 below, respectively. Service interaction profiles are discussed in Section 6 below. Document signature
318 profiles are discussed in Section 7 below.

319

2.2 Major Design Elements

320 ECF 4.1 defines four MDEs. They are:

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

334 The MDEs defined in the ECF 4.1 specifications are meant only to define the “interface” to each
335 operation; the specification is not intended to define how operations must be implemented. This strategy
336 allows MDE implementations to interoperate while leaving room for vendors and courts to have differing
337 implementations (e.g., an implementation that supports a particular CMS).

338 An ECF 4.1-compliant implementation may implement one or more of the MDEs defined in the
339 specification but a complete ECF 4.1 system MUST include at least one each of the Filing Assembly,
340 Filing Review and Court Record MDEs. For instance, a court may decide to provide certain MDEs and
341 allow private providers to furnish the remaining MDEs. When multiple MDEs are implemented by a single

342 court, vendor or application, the application MUST maintain the ECF 4.1 specified operations between
343 each MDE so that other applications will be able to interoperate with it.

344 Each of the operations supported by an MDE accepts one or more messages as input and returns an
345 immediate, synchronous response message to the calling MDE. For some operations, the MDE will also
346 return an asynchronous (callback) message at a later time that reports the result of a business process
347 implemented within the MDE. In order to be compliant with ECF 4.1, an MDE MUST support all required
348 operations for that MDE. However, in an ECF 4.1 system that does not support electronic service, the
349 operations associated with the Legal Service MDE are not required.

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

353 2.3 Information Model

354 The ECF information model describes the messages that may be exchanged between MDEs. All ECF
355 4.1 operations use the same core message stream structure, which is implemented in the service
356 interaction profiles. Each ECF core message stream is a stream of bytes that contains at least one
357 message and may also contain attachments.

358 2.3.1 Messages

359 A message is an XML document that is a well-formed XML data structure with a single root element that
360 is transmitted between MDEs and is valid as defined by one of the defined message structure schemas in
361 the ECF 4.1 specification. A message may be related to one or more attachments. A message contains
362 the following information:

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

```
381     <FilingLeadDocument> (or <FilingConnectedDocument>)  
382         <ecf:DocumentRendition>  
383             <DocumentRenditionMetadata>  
384                 <DocumentAttachment>  
385                     <BinaryBase64Object>2345klj345h...<BinaryBase64Objec  
386                     t>  
387                 </DocumentAttachment>  
388             </DocumentRenditionMetadata>  
389         </ecf:DocumentRendition>
```

390 </FilingLeadDocument> (or </FilingConnectedDocument>)

391

392 Elements defined by this specification, whether in core messages, case type-specific extensions or court-
393 specific extensions, are intended to be useful to an automated case management system for the
394 purposes of partially or fully automating case workflow after filing (e.g., filing review, noticing, docketing,
395 judicial assignment, calendaring, standardized forms receipt and generation, fee processing) or
396 ascertaining the adequacy or appropriateness of the filing (e.g., fee or fine calculation, jurisdiction).

397 Elements defined by this specification are not intended to fully populate the automated case management
398 system with all data contained within filed documents. That is, these elements should be useful as “filing
399 metadata” about the case, the filing transaction, parties or documents. These elements may also be “filing
400 data”, or the contents of the filings. For instance, information found on a filing cover sheet can generally
401 be considered filing metadata, even if the information is also repeated in the document(s) being filed.

402

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

- 404 • Elements in the ECF core messages should be applicable to most courts and case types
- 405 • Elements in the ECF case-type-specific extensions should only be applicable to one of the seven
406 case types defined in National Center for State Courts (NCSC) statistical standards
- 407 • Elements in locally-defined court-specific extensions should only be applicable to a particular court or
408 court system but not to courts in general

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

411 **2.3.2 Attachment**

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

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

422 **2.3.3 Sample Message Streams**

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

424

425

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

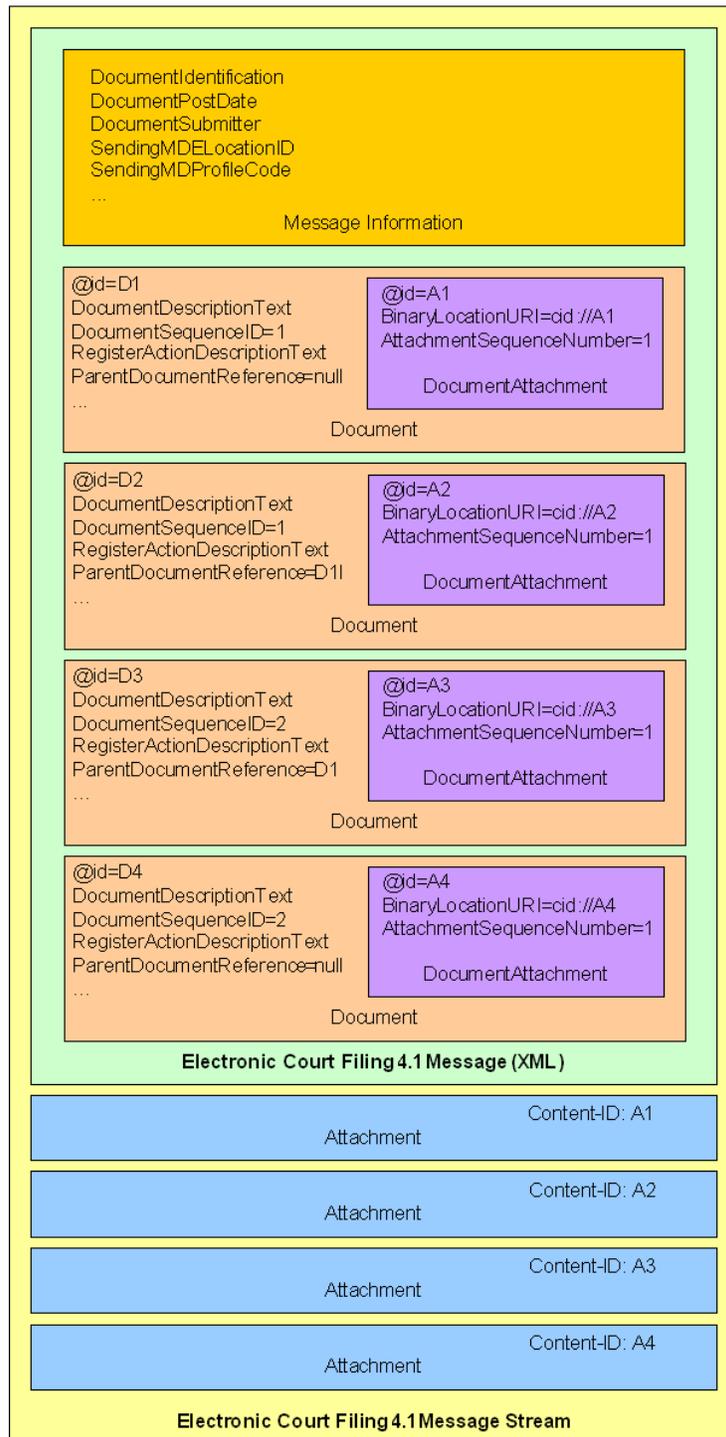
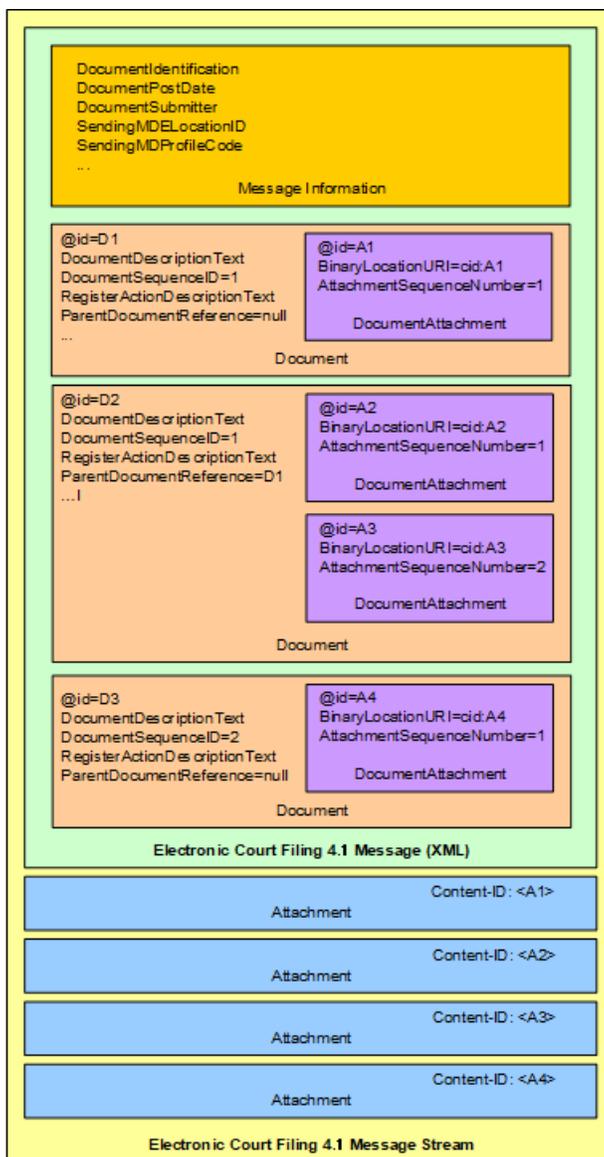


Figure 1. Simple Message Stream

429
 430
 431

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



437
 438
 439

Figure 2. Message Stream with a Document in Multiple Attachments

440 2.4 Court Policy

441 A court's customary practices may influence many aspects of its ECF 4.1 implementation, and those local
442 practices and variations are expressed through the "court policy" component of e-filing, which includes:

- 443 • **Human-readable court policy** – a textual document publishing the court's rules and requirements for
444 electronic filing.
- 445 • **Machine-readable court policy** – an ECF 4.1 message that describes the features of the ECF 4.1
446 implementation supported by this specification, the court's code lists and any other information a
447 Filing Assembly MDE would need to know in order to successfully submit an electronic filing into that
448 court.

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

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

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

467 2.4.1 Human-Readable Court Policy

468 To be compliant with the ECF 4.1 specification, each court **MUST** publish a human-readable court policy
469 that **MUST** include each of the following:

- 470 1. The unique court identifier
- 471 2. The location of the machine-readable court policy
- 472 3. A definition of what constitutes a "lead document" in the court
- 473 4. A description of how filer identifiers are to be maintained during electronic communications regarding
474 the case
- 475 5. A description of how the court processes (dockets) filings
- 476 6. A description of any instances in which the court will mandate an element that the ECF 4.1 schema
477 makes optional
- 478 7. A description of any restrictions to data property values other than code list restrictions. (This
479 restriction may be removed in later versions of the ECF specification)
- 480 8. Any other rules required for electronic filing in the court

481 2.4.2 Machine-Readable Court Policy

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

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

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

- 489 1. The service interaction profile(s) that the court supports
- 490 2. The MDEs, query operations and case types supported by the court's ECF 4.1 system
- 491 3. Whether a court will accept the filing of a URL in lieu of the electronic document itself
- 492 4. Whether the court accepts documents requiring payment of a filing fee
- 493 5. Whether the court accepts electronic filing of sealed documents
- 494 6. Whether the court accepts multiple filings
- 495 7. The court-specific extensions to the ECF 4.1 specification, including the required elements (see
496 below)
- 497 8. The maximum sizes allowed for a single attachment and a complete message stream

498 Some form of machine-readable court policy MUST exist. The machine-readable court policy MUST be
499 provided to the Filing Assembly MDE either by the Filing Review MDE through the GetCourtPolicy query
500 or some other means.

501 2.4.3 Case-Type and Court Extensions

502 Schemas for initiating specific case types (e.g. criminal, civil) are included in the specification. Case-type
503 and court-specific extensions to the ECF core messages are implemented through the methods
504 described in [NIEM Techniques]. The primary extension technique is the use of element substitution, as
505 described in Section 5.3.3 of [NIEM Techniques], in which a more specific element defined in a case-
506 type or court-specific extension is used in place of a generic element in a core message. For instance, a
507 court may add elements required for a particular case type (e.g. civil) by defining an extension schema
508 that includes types (e.g. `court:CivilCaseType`) and elements (e.g., `court:CivilCase`) that
509 substitute for ECF types (e.g. `civil:CivilCaseType`) and elements (e.g., `civil:CivilCase`).
510 Similarly, an implementation may substitute a court-specific code list for a generic code list defined in this
511 specification.

512

513 2.4.4 Court-Specific Code Lists

514 Code lists are used to constrain the allowable values for certain information in an ECF 4.1 message. The
515 court SHOULD publish [GenericCode] 1.0 code lists for each of the following code lists and reference
516 each of these code lists in its court policy:

517

- 518 • ECF Code Lists
 - 519 • [Civil Case Type](#)
 - 520 • `<FiduciaryTypeCode>*`
 - 521 • `<JurisdictionalGroundsCode>`
 - 522 • `<ReliefTypeCode>`
 - 523 • [Domestic Case Type](#)
 - 524 • `<NoContactCode>*`
 - 525 • `<RequestToVacateCode>`
 - 526 • [Common Types](#)
 - 527 • `<AliasAlternateNameTypeCode>*`
 - 528 • `<CaseAssociationTypeCode>*`
 - 529 • `<CaseOfficialRoleText>*`

- 530 • <CaseParticipantRoleCode>*
- 531 • <CauseOfActionCode>
- 532 • <CourtEventTypeCode>
- 533 • <EntityAssociationTypeCode>
- 534 • <ErrorCode>*
- 535 • **Juvenile Case Type**
- 536 • <DelinquentActApplicabilityCode>
- 537 • <DelinquentActDegreeCode>
- 538 • <DelinquentActSeverityCode>
- 539 • <DelinquentActSpecialAllegationCode>
- 540 • <DependencyAllegationCode>
- 541 • <GuardianAssociationTypeCode>*
- 542 • <PlacementTypeCode>
- 543 • **NIEM Code Lists**
- 544 • **JXDM**
- 545 • <ChargeEnhancingFactorText>
- 546 • <CourtLocationCode>
- 547 • <RegisterActionDescriptionText>
- 548 • <StatuteCodeIdentification>
- 549 • <StatuteCodeSectionIdentification>
- 550 • <StatuteOffenseIdentification>
- 551 • <StatusOffenseCodeIdentification>
- 552 • **NIEM Core**
- 553 • <BinaryDescriptionText>*
- 554 • <CaseCategoryText>
- 555 • <DriverLicenseCommercialClassCode>
- 556 • <FamilyKinshipCode>*

557

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

560

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

563

564 2.4.5 Court-Specific Constraint Schemas

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

571 **3 ECF 4.1 Process Model**

572 This section details the interactions of the ECF 4.1 MDEs and the role of each MDE in the electronic filing
573 and electronic service processes. This section also enumerates the operations provided by each MDE
574 and points to the operations, provided by other MDEs, that each MDE consumes.

575 **3.1 The Filing-Preparation-to-Docketing Process Model**

576 This model describes the sequence of operations in a basic filing cycle from Filing Preparation to
577 Docketing. This model involves three parties: a Filer (represented by the Filing Assembly MDE), a Court
578 (represented by the Filing Review and Court Record MDEs) and a Service Recipient (represented by the
579 Legal Service MDE). The operations defined by ECF 4.1 to support the processes in this cycle are listed
580 below. The ReviewFiling and RecordFiling operations are required in a complete ECF 4.1 system as
581 prescribed in Section 2.2. However, when the RecordFiling operation has been implemented within the
582 same system as the ReviewFiling operation, then the RecordFiling operation need not be provided in an
583 ECF 4.1 compliant manner.. The other operations are optional and MAY occur within a given filing:

- 584 • GetPolicy
- 585 • GetServiceInformation
- 586 • GetFeesCalculation
- 587 • **ReviewFiling**
- 588 • ServeFiling
- 589 • **RecordFiling**
- 590 • NotifyDocketingComplete
- 591 • NotifyFilingReviewComplete

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

- 594 • GetFilingList
- 595 • GetFilingStatus

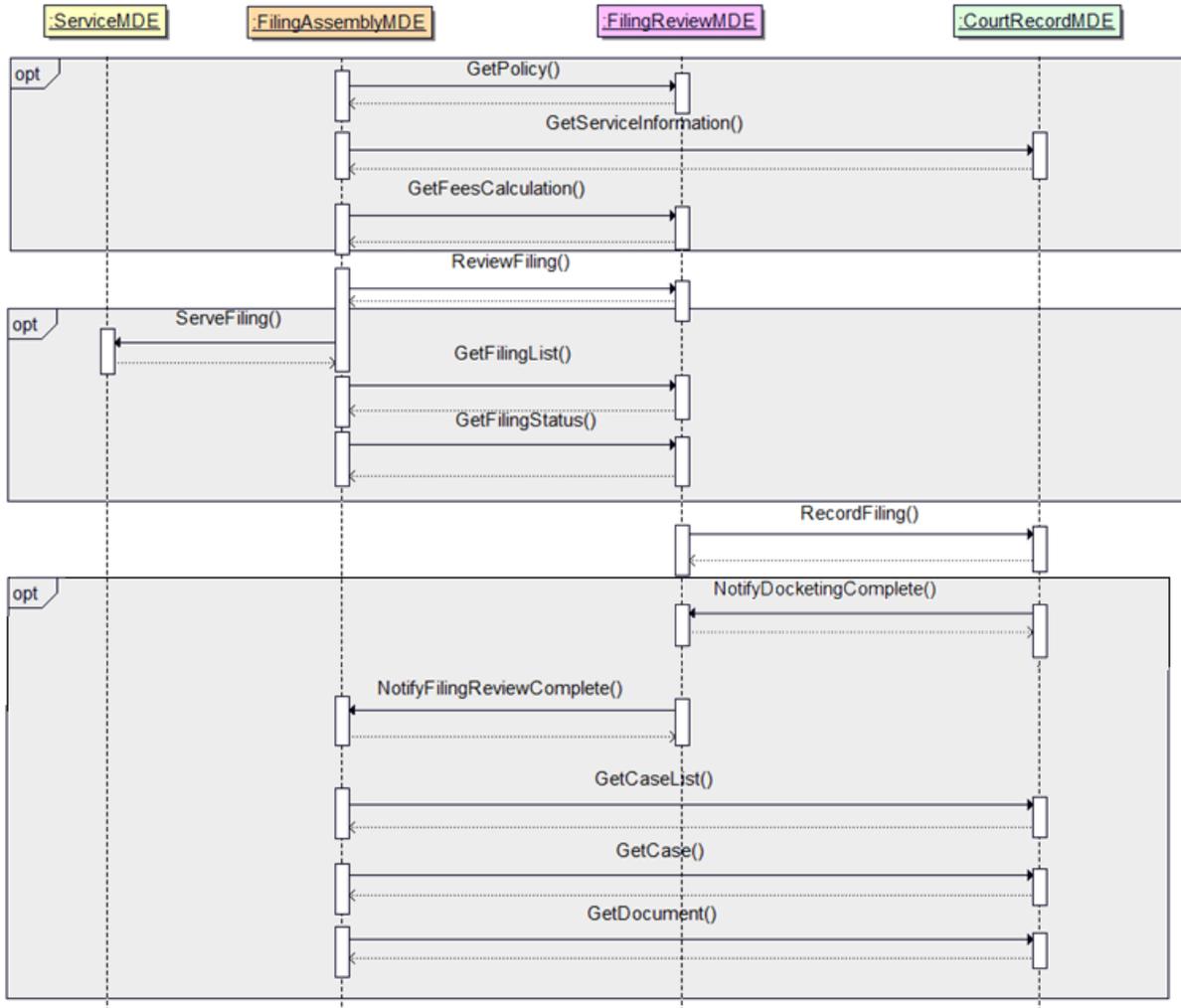
596 At any point, if filing into an existing case, or after the NotifyFilingReviewComplete operation if initiating a
597 case, and if the case is accessible, a party MAY access information through the following operations:

- 598 • GetCaseList
- 599 • GetCase
- 600 • GetDocument

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

603

Figure 4. Filing Preparation to Docketing Process Model



605

606 3.2 Business Rules

607 This section describes the business rules of the generic filing-preparation-to-docketing process that
608 govern the ECF 4.1 operations.

609 ECF 4.1 includes an <ecf:ErrorCode> element for returning errors in response to a query request.
610 Successful queries MUST return an <ecf:ErrorCode> of "0". Failed queries MUST NOT return an
611 <ecf:ErrorCode> of "0" and SHOULD return an appropriate <ecf:ErrorCode> value as defined in
612 court policy.

613 3.2.1 GetPolicy

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

619 3.2.2 GetServiceInformation

620 The Filing Assembly MDE MAY obtain the Court's service information for all parties in an existing case at
621 any time by invoking the GetServiceInformation operation with the appropriate case number on the Court

622 Record MDE. The service list returned by the GetServiceInformation operation assists the filer in
623 maintaining the filer's service list and is not a substitute for the filer's service list. To provide this
624 information, the Court Record MDE MUST have access to the court's registry with all updated information
625 about case participants. There MUST be only one such registry per court, though multiple courts MAY
626 share the same registry. The Court Record MDE responds synchronously to the Filing Assembly MDE
627 with a service list reflecting the most current contact information available to the court, which is necessary
628 to complete secondary service, whether electronically or by other means.

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

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

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

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

637 **3.2.3 GetFeesCalculation**

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

643 **3.2.4 ReviewFiling**

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

648 **3.2.5 ServeFiling**

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

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

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

662 **3.2.6 RecordFiling**

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

667 **3.2.7 NotifyDocketingComplete**

668 RecordDocketingCallbackMessage MAY be provided as a callback message by the Record Filing MDE to
669 the Filing Review MDE to indicate whether the filing was accepted or rejected by the court system. The
670 Filing Review MDE responds synchronously with an acknowledgement of any callback message
671 received.

672

673 When the <RequireAsynchronousResponsesIndicator> in the court policy is “true”, the Court Record
674 MDE MUST invoke the NotifyDocketingComplete operation on the Filing Review MDE, otherwise the
675 callback message is optional.

676

677 If the Court Record MDE rejected the filing, an explanation MUST be provided in the callback message
678 when provided to Filing Review MDE. If the Court Record MDE accepts the filing, the docketing
679 information (e.g., date and time the document was entered into the court record, judge assigned,
680 document identifiers and next court event scheduled) MUST be provided when a callback message is
681 tendered.

682 **3.2.8 NotifyFilingReviewComplete**

683 ReviewFilingCallbackMessage and PaymentReceiptMessage MAY be provided as callback messages by
684 the Review Filing MDE to the Filing Assembly MDE to indicate whether the filings were accepted by the
685 clerk. The Filing Assembly MDE responds synchronously with an acknowledgement of any callback
686 message received.

687

688 When the <RequireAsynchronousResponsesIndicator> in the court policy is “true”, the Filing Review MDE
689 MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE upon receipt of a
690 RecordDocketingCallbackMessage from the Court Record MDE, otherwise the callback message is
691 optional.

692

693 The operation MAY return the filed documents or links to the documents but MUST include the [FIPS
694 180-4] SHA 256 document hash, a condensed representation of the document as currently in the court
695 record.

696

697 If the filing included a payment, and the filing was accepted by the clerk, a receipt for the payment MUST
698 be included in the operation.

699 **3.2.9 GetFilingList**

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

704 **3.2.10 GetFilingStatus**

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

708 **3.2.11 GetCaseList**

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

712 **3.2.12 GetCase**

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

718 **3.2.13 GetDocument**

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

722 **3.3 Message Business Rules**

723 Each operation includes one or more messages as parameters. The following business rules apply to the
724 content of ECF 4.1 messages:

725 **3.3.1 Identifiers**

726 Identifiers are used to uniquely label people, organizations and things in the ECF 4.1 process. The
727 following conventions will be used to produce identifiers.

728 **3.3.1.1 Attachment Identifiers**

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

732 **3.3.1.2 Case Identifiers**

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

735 **3.3.1.3 Court Identifiers**

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

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

740 Examples of conformant court identifiers include:

- 741 • courts.wa.gov:superior.king
- 742 • nmcourts.com:albd.civil
- 743 • uscourts.gov:100
- 744 • courts.gov.bc.ca:appeal

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

746 **3.3.1.4 Document Identifiers**

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

748 **3.3.1.5 Filing Identifiers**

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

751 **3.3.1.6 MDE Identifiers**

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

754 **3.3.1.7 Asynchronous responses**

755 ECF 4.1 messages that support asynchronous responses include <SendingMDELocationID> and
756 <SendingMDEProfileCode> to support the return of the asynchronous response to the sending MDE. If
757 the <RequireAsynchronousResponsesIndicator> in the CourtPolicyResponseMessage is "true", then both
758 <SendingMDELocationID> and <SendingMDEProfileCode> MUST be included in all ECF 4.1 messages
759 that include these elements.

760 **3.3.1.8 Filer and Party Identifiers**

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

764
765 <nc:PersonOtherIdentification>
766 <nc:IdentificationID>100<nc:IdentificationID>
767 <nc:IdentificationCategoryText>ECFFilerID</nc:IdentificationCategoryText>
768 </nc:PersonOtherIdentification>
769

770
771 In addition to <nc:PersonOtherIdentification>, other elements that may contain a filer identifier
772 include <nc:OrganizationOtherIdentification>, <ecf:FilingPartyID> and
773 <ecf:FilingAttorneyID>.

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

778 **3.3.2 Code Lists**

779 Code Lists are used to constrain the allowable values for certain information in a message. The following
780 normative code lists are normative for all ECF 4.1 implementations. Court-specific code lists are listed in
781 Section 2.4.4.

- 782
- 783 • ECF Code Lists
 - 784 • **Bankruptcy Case Type**
 - 785 • <DebtorTypeCode>*
786 • <EstimatedAssetsValueLevelCode>*
787 • <EstimatedDebtsValueLevelCode>*
788 • <NatureOfDebtCode>*789 • <NumberOfCreditorsValueLevelCode>*
- 790 • **Common Types**
 - 791 • <FilingStatusCode>*
- 792 • **Court Policy Response Message**
 - 793 • <MajorDesignElementNameCode>

- 794 • <OperationNameCode>
- 795 • [Service Receipt Message](#)
- 796 • [<ServiceStatusCode>](#)*
- 797 • NIEM Code Lists
- 798 • [ANSI NIST](#)
- 799 • <FingerPositionCode>
- 800 • [JXDM](#)
- 801 • <ChargeNCICCode>
- 802 • <DrivingIncidentHazMatCode>
- 803 • <DrivingJurisdictionAuthorityNCICLSTACode>
- 804 • <IdentificationJurisdictionNCICLISCode>
- 805 • <WarrantExtraditionLimitationCode>
- 806 • [NIEM Core](#)
- 807 • <DocumentLangageCode>
- 808 • <DriverLicenseCommercialClassCode>
- 809 • <DrivingRestrictionCode>
- 810 • <LanguageCode>
- 811 • <LengthUnitCode>
- 812 • <LocationCountryFIPS10-4Code>
- 813 • <LocationCountryISO3166Alpha2Code>
- 814 • <LocationCountyCode>
- 815 • <LocationStateUSPostalServiceCode>
- 816 • <PersonCitizenshipFIPS10-4Code>
- 817 • <PersonCitizenshipISO3166Alpha2Code>
- 818 • <PersonEthnicityCode>
- 819 • <PersonEyeColorCode>
- 820 • <PersonHairColorCode>
- 821 • <PersonRaceCode>
- 822 • <PersonSexCode>
- 823 • <PersonUnionCategoryCode>
- 824 • <PhysicalFeatureCategoryCode>
- 825 • <VehicleColorPrimaryCode>
- 826 • <VehicleMakeCode>
- 827 • <VehicleModelCode>
- 828 • <VehicleStyleCode>
- 829 • <WeightUnitCode>

830

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

833 3.3.3 Message-Specific Business Rules

834 The following business rules apply to specific messages:

835 3.3.3.1 CoreFilingMessage

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

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

847 3.3.3.2 PaymentMessage

848 ECF 4.1 supports multiple particular payment processes. Information about a payment is included in the
849 PaymentMessage including the method of payment of the applicable fees, e.g., electronic funds transfer,
850 credit or debit card, charge to an escrow account held in the court or promise to pay in the future. The
851 payment MAY include a maximum amount for the payment if some latitude is needed to accomplish the
852 filing.

853 3.3.3.3 RecordDocketingMessage

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

856 3.4 Filing the Record on Appeal

857 This section describes the process for filing and subsequently amending the Record on Appeal (ROA)
858 using ECF 4.1.

- 859
- 860 • All ROA transactions, either the original filing or subsequent amendments, MUST contain, as the
861 lead document, an Index of Record document that itemizes the content of the record on appeal.³
862
 - 863 • The documents that comprise the ROA transaction will be identified as supporting documents.
864
 - 865 • The supporting documents that comprise the ROA transaction MAY also have additional attached
866 documents.
867
 - 868 • All ROA documents being submitted, including the Index of Record document and each
869 document within the record, MUST have at least one court-defined document type that indicates
870 the type of transaction to be performed on the document, and whether the document is being
871 added to or stricken from the record.
872

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

- 873 • The Index of Record document and each document within the ROA transaction MAY also have
874 an additional document type or types, which characterize the document for the Court Record
875 MDE.
- 876
- 877 • When a document within the ROA transaction is being stricken from the court record, the
878 document MUST be identified by the unique document identifier, which was provided by the Court
879 Record MDE when the document was initially filed (See section 3.3.1.4).
- 880
- 881 • A hierarchical structure of case lineage elements MUST be used to express the target case's
882 predecessor cases at prior courts. Each predecessor case MAY also have its own predecessor
883 case, as necessary to express the full lineage of an appellate case.⁴
- 884
- 885 • When the ROA transaction is electronically transferred from one court to another, the target case
886 number in the destination court and the case lineage, which includes the predecessor case
887 number in the sending court, MUST be provided.
- 888
- 889 • If the ROA transaction is a case initiating filing in the destination court, then the <FilingCase>
890 object MUST be present and the <CaseTrackingID> MUST be absent.
- 891
- 892 • Each predecessor case identified in the target case's case lineage may include case type-specific
893 and court-specific extensions. The case type and the case type-specific extensions for each
894 predecessor case MUST be consistent throughout the case lineage.
- 895
- 896 • When a ROA amendment transaction is sent, the Index of Record document MUST reflect the
897 status of the record assuming that the transaction will be accepted. If however the transaction is
898 rejected, there will be ramifications for other pending amendment transactions for the same ROA
899 in the same target case.⁵
- 900
- 901 • While an ROA transaction is awaiting acceptance or rejection in the destination court, and when
902 the target case consists of multiple records, courts SHOULD NOT send additional amendment
903 transactions intended for the same record for the same target case.

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.

904

905

906

907

908

- 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.

909 4 ECF 4.1 Schemas

910 The Court Filing XSD schemas are implementations of the ECF 4.1 exchange content models (see
911 Appendix B.3 below). They are the only normative representations of ECF 4.1 messages.

912 All of the ECF 4.1 XSD schemas are contained in the `xsd/` subdirectory of the ECF 4.1 release package
913 (see Appendix A for more information regarding the structure of the release package). The `xsd/`
914 directory is further subdivided into the `xsd/casetype/`, `xsd/common/`, `xsd/constraint/`,
915 `xsd/message/`, and `xsd/Subset/` subdirectories.

916

917 4.1 ECF 4.1 Case Type Schemas

918 The XSD schemas that define extensions specific to certain ECF 4.1 case types are included in the
919 `xsd/casetype/` directory, as listed below:

920

921 **AppellateCase**

922 [xsd/casetype/ECF-4.1-AppellateCase.xsd](#)

923 **BankruptcyCase**

924 [xsd/casetype/ECF-4.1-BankruptcyCase.xsd](#)

925 **CitationCase**

926 [xsd/casetype/ECF-4.1-CitationCase.xsd](#)

927 **CivilCase**

928 [xsd/casetype/ECF-4.1-CivilCase.xsd](#)

929 **CriminalCase**

930 [xsd/casetype/ECF-4.1-CriminalCase.xsd](#)

931 **DomesticCase**

932 [xsd/casetype/ECF-4.1-DomesticCase.xsd](#)

933 **JuvenileCase**

934 [xsd/casetype/ECF-4.1-JuvenileCase.xsd](#)

935

936 4.2 ECF 4.1 Common Schemas

937 The XSD schemas that define the generic elements and types that are common to multiple ECF 4.1
938 messages and/or case types are located in the `xsd/common/` folder, as listed below:

939

940 **AppInfo**

941 [xsd/common/ECF-4.1-AppInfo.xsd](#)

942 **CommonTypes**

943 [xsd/common/ECF-4.1-CommonTypes.xsd](#)

944 **DigitalSignature**

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

946 **Genericcode**

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

948 **4.3 ECF 4.1 Constraint and Subset Schemas**

949 The XSD schemas that define the subset of all NIEM elements and types that are used in ECF 4.1
950 messages and/or case type extensions are located in the `xsd/Subset/niem/` folder. As a general
951 data model, NIEM does not define any constraints regarding the minimum and maximum occurrence of
952 elements contained within types. Therefore, in conformance with NIEM, ECF-specific constraints are not
953 included in the schemas within the `xsd/Subset/niem` folder. The XSD schemas in the
954 `xsd/constraint/niem/` folder represent the NIEM subset schemas with the ECF-specific constraints
955 applied and are the schemas by which the ECF message and case type schemas incorporate NIEM
956 elements and types.

957 **4.4 ECF 4.1 Message Schemas**

958 The XSD schemas defining the messages that support the ECF 4.1 processes are located in the
959 `xsd/messages/` folder, as listed below:

960

961 **CaseListQueryMessage**

962 [xsd/message/ECF-4.1-CaseListQueryMessage.xsd](#)

963 **CaseListResponseMessage**

964 [xsd/message/ECF-4.1-CaseListResponseMessage.xsd](#)

965 **CaseQueryMessage**

966 [xsd/message/ECF-4.1-CaseQueryMessage.xsd](#)

967 **CaseResponseMessage**

968 [xsd/message/ECF-4.1-CaseResponseMessage.xsd](#)

969 **CoreFilingMessage**

970 [xsd/message/ECF-4.1-CoreFilingMessage.xsd](#)

971 **CourtPolicyQueryMessage**

972 [xsd/message/ECF-4.1-CourtPolicyQueryMessage.xsd](#)

973 **CourtPolicyResponseMessage**

974 [xsd/message/ECF-4.1-CourtPolicyResponseMessage.xsd](#)

975 **DocumentQueryMessage**

976 [xsd/message/ECF-4.1-DocumentQueryMessage.xsd](#)

977 **DocumentResponseMessage**

978 [xsd/message/ECF-4.1-DocumentResponseMessage.xsd](#)

979 **FeesCalculationQueryMessage**

980 [xsd/message/ECF-4.1-FeesCalculationQueryMessage.xsd](#)

981 **FeesCalculationResponseMessage**

982 [xsd/message/ECF-4.1-FeesCalculationResponseMessage.xsd](#)

983 **FilingListQueryMessage**

984 [xsd/message/ECF-4.1-FilingListQueryMessage.xsd](#)

985 **FilingListResponseMessage**

986 [xsd/message/ECF-4.1-FilingListResponseMessage.xsd](#)

987 **FilingStatusQueryMessage**

988 [xsd/message/ECF-4.1-FilingStatusQueryMessage.xsd](#)

989 **FilingStatusResponseMessage**

990 [xsd/message/ECF-4.1-FilingStatusResponseMessage.xsd](#)

991 **MessageReceiptMessage**
992 [xsd/message/ECF-4.1-MessageReceiptMessage.xsd](#)
993 **PaymentMessage**
994 [xsd/message/ECF-4.1-PaymentMessage.xsd](#)
995 **PaymentReceiptMessage**
996 [xsd/message/ECF-4.1-PaymentReceiptMessage.xsd](#)
997 **RecordDocketingCallbackMessage**
998 [xsd/message/ECF-4.1-RecordDocketingCallbackMessage.xsd](#)
999 **RecordDocketingMessage**
1000 [xsd/message/ECF-4.1-RecordDocketingMessage.xsd](#)
1001 **ReviewFilingCallbackMessage**
1002 [xsd/message/ECF-4.1-ReviewFilingCallbackMessage.xsd](#)
1003 **ServiceInformationQueryMessage**
1004 [xsd/message/ECF-4.1-ServiceInformationQueryMessage.xsd](#)
1005 **ServiceInformationResponseMessage**
1006 [xsd/message/ECF-4.1-ServiceInformationResponseMessage.xsd](#)
1007 **ServiceReceiptMessage**
1008 [xsd/message/ECF-4.1-ServiceReceiptMessage.xsd](#)
1009

1010 5 MDE Operations

1011 This section details the operations that are provided by each Major Design Element (MDE) and the
 1012 operations, provided by other MDEs that each MDE “consumes.” Each provided operation definition
 1013 includes the input (parameter) and output messages and the required message cardinality in the format:
 1014 (minimum occurrences, maximum occurrences). Implementation of an MDE requires both that the MDE
 1015 provide certain functionality and that the MDE use particular operations provided by other MDEs.

1016 5.1 Filing Assembly MDE

1017 The Filing Assembly MDE supports the preparation and submission of filed documents to a court for
 1018 review, and can receive the results of that process. The Filing Assembly MDE also conveys filings to the
 1019 Legal Service MDE for service on other case participants. The Filing Assembly MDE calls operations in
 1020 other MDEs and provides a single operation for notifying the submitter that the filing has been reviewed
 1021 by a court. A Filing Assembly MDE may be provided by a court or by a third party.

1022 5.1.1 Provided Operations

1023 The Filing Assembly MDE provides the following operations to other MDEs:

Operation	Called By	Output	Parameters
NotifyFilingReviewComplete	Filing Review MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage (1,1)	xsd/message/ECF-4.1-ReviewFilingCallbackMessage.xsd : ReviewFilingCallbackMessage (1,unbounded) xsd/message/ECF-4.1-PaymentReceiptMessage.xsd : PaymentReceiptMessage (1,1)

1024 5.1.2 Consumed Operations

1025 The Filing Assembly MDE calls the following operations in other MDEs:

Operation	Provided By	Return Type
GetPolicy	Filing Review MDE	xsd/message/ECF-4.1-CourtPolicyQueryMessage.xsd : CourtPolicyReponseMessage
ReviewFiling	Filing Review MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage
GetFeesCalculation	Filing Review MDE	xsd/message/ECF-4.1-FeesCalculationResponseMessage.xsd : FeesCalculationResponseMessage
GetFilingStatus	Filing Review MDE	xsd/message/ECF-4.1-FilingStatusResponseMessage.xsd : FilingStatusResponseMessage
GetFilingList	Filing Review MDE	xsd/message/ECF-4.1-FilingListResponseMessage.xsd : FilingListResponseMessage
GetCase	Court Record MDE	xsd/message/ECF-4.1-CaseResponseMessage.xsd : CaseResponseMessage
GetCaseList	Court Record MDE	xsd/message/ECF-4.1-CaseListResponseMessage.xsd : CaseListResponseMessage
GetServiceInformation	Court Record MDE	xsd/message/ECF-4.1-ServiceInformationResponseMessage.xsd :

		ServiceInformationResponseMessage
GetDocument	Court Record MDE	xsd/message/ECF-4.1-DocumentResponseMessage.xsd : DocumentResponseMessage
ServeFiling	Legal Service MDE	xsd/message/ECF-4.1-ServiceReceiptMessage.xsd : ServiceReceiptMessage

1026 5.2 Filing Review MDE

1027 The Filing Review MDE receives, presents and manages the filings. The Filing Review MDE receives
1028 filings in a standard format and presents those filings to a Clerk for review, where they may be accepted
1029 or rejected. The Filing Review MDE transmits data and documents to the Filing Assembly MDE to inform
1030 the filer that the filing has been accepted or rejected. The Filing Review MDE transmits data and
1031 documents for accepted filings to the Court Record MDE for docketing and recording. While there will
1032 generally be one Filing Review MDE per court, there is no physical barrier to having more than one,
1033 particularly if a court wants to support different Filing Review MDEs for particular case types.

1034 5.2.1 Provided Operations

1035 The Filing Review MDE provides the following operations to other MDEs:

Operation	Called By	Output	Parameters
ReviewFiling	Filing Assembly MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage (1,1)	xsd/message/ECF-4.1-CoreFilingMessage.xsd : CoreFilingMessage (1,unbounded)
			xsd/message/ECF-4.1-PaymentMessage.xsd : PaymentMessage (0,1)
NotifyDocketingComplete	Court Docketing MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage (1,1)	xsd/message/ECF-4.1-RecordDocketingCallbackMessage.xsd : RecordDocketingCallbackMessage (1,unbounded)
GetFeesCalculation	Filing Assembly MDE	xsd/message/ECF-4.1-FeesCalculationResponseMessage.xsd : FeesCalculationResponseMessage (1,1)	xsd/message/ECF-4.1-FeesCalculationQueryMessage.xsd : FeesCalculationQueryMessage (1,1)
GetFilingList	Filing Assembly MDE	xsd/message/ECF-4.1-FilingListResponseMessage.xsd : FilingListResponseMessage (1,1)	xsd/message/ECF-4.1-FilingListQueryMessage.xsd : FilingListQueryMessage (1,1)
GetFilingStatus	Filing Assembly MDE	xsd/message/ECF-4.1-FilingStatusResponseMessage.xsd : FilingStatusResponseMessage (1,1)	xsd/message/ECF-4.1-FilingStatusQueryMessage.xsd : FilingStatusQueryMessage (1,1)
GetPolicy	Filing Assembly MDE	xsd/message/ECF-4.1-CourtPolicyQueryMessage.xsd : CourtPolicyResponseMessage (1,1)	xsd/message/ECF-4.1-CourtPolicyQueryMessage.xsd : CourtPolicyQueryMessage (1,1)

1036 5.2.2 Consumed Operations

1037 The Filing Review MDE calls the following operations in other MDEs:

Operation	Provided By	Output
RecordFiling	Court Record MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage
NotifyFilingReviewComplete	Filing Assembly MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage

1038 5.3 Court Record MDE

1039 The Court Record MDE receives the filed documents from the Filing Review MDE and enters them into
 1040 the official case record of the court. The Court Record MDE notifies the Filing Review MDE that the filing
 1041 has been filed.

1042 5.3.1 Provided Operations

1043 The Court Record MDE provides the following operations to other MDEs:

Operation	Called By	Output	Parameters
RecordFiling	Filing Review MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage (1,1)	xsd/message/ECF-4.1-RecordDocketingMessage.xsd : RecordDocketingMessage (1,unbounded)
			xsd/message/ECF-4.14.0-CoreFilingMessage.xsd : CoreFilingMessage (1,unbounded)
GetCase	Filing Assembly MDE	xsd/message/ECF-4.1-CaseResponseMessage.xsd : CaseResponseMessage (1,1)	xsd/message/ECF-4.1-CaseQueryMessage.xsd : CaseQueryMessage (1,1)
GetCaseList	Filing Assembly MDE	xsd/message/ECF-4.1-CaseListResponseMessage.xsd : CaseListResponseMessage (1,1)	xsd/message/ECF-4.1-CaseListQueryMessage.xsd : CaseListQueryMessage (1,1)
GetServiceInformation	Filing Assembly MDE	xsd/message/ECF-4.1-ServiceInformationResponseMessage.xsd : ServiceInformationResponseMessage (1,1)	xsd/message/ECF-4.1-ServiceInformationQueryMessage.xsd : ServiceInformationQueryMessage (1,1)
GetDocument	Filing Assembly MDE	xsd/message/ECF-4.1-DocumentResponseMessage.xsd : DocumentResponseMessage (1,1)	xsd/message/ECF-4.1-DocumentQueryMessage.xsd : DocumentQueryMessage (1,1)

1044 5.3.2 Consumed Operations

1045 The Court Record MDE calls the following operations in other MDEs:

Operation	Provided By	Output
NotifyDocketingComplete	Filing Review MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage

1046 **5.4 Legal Service MDE**

1047 The Legal Service MDE enables a filer or a court to electronically transmit copies of, or links to,
1048 electronically filed documents to other parties who are participating in the case and who are entitled to be
1049 promptly served with the electronically filed documents. The Filing Assembly MDE transmits data and
1050 documents to the Legal Service MDE to inform the case participant that an electronic filing has been
1051 submitted to the court clerk. The Legal Service MDE transmits a callback message to the Filing
1052 Assembly MDE requesting a notification to confirm receipt of the served document.

1053 **5.4.1 Provided Operations**

1054 The Legal Service MDE provides the following operations to other MDEs:

Operation	Called By	Output	Parameters
ServeFiling	Filing Assembly MDE	xsd/message/ECF-4.1-ServiceReceiptMessage.xsd : ServiceReceiptMessage (1,1)	xsd/message/ECF-4.1-CoreFilingMessage.xsd : CoreFilingMessage (1,1)

1055 **5.4.2 Consumed Operations**

1056 The Legal Service MDE does not call operations in other MDEs

1057

6 Service Interaction Profiles

1058

1059 An ECF 4.1 service interaction profile defines a transmission system that supports the functional
1060 requirements of electronic filing, along with the MDE operations and message structures, and implements
1061 certain non-functional requirements. A service interaction profile does not govern the content of
1062 messages – message content is described in Sections 2 and 3 of this specification. A service interaction
1063 profile will define how a message gets from the sending MDE to the receiving MDE in a given messaging
1064 framework.

1065

1066 To support interoperability across Service Interaction Profiles, this specification includes
1067 xsd/wrappers.xsd an optional schema document defining the types and elements for each operation on
1068 all Major Design Elements (MDEs) as defined in Section 5 of this specification. Service Interaction
1069 Profiles MAY require this file.

6.1 Service Interaction Profile Requirements

1070

1071 Each service interaction profile will define standard conventions and configuration details to support
1072 interoperability between and among ECF 4.1 implementations that support the same service interaction
1073 profile. However, compliance with these requirements will not necessarily guarantee interoperability.

1074 To be compliant with the ECF 4.1 specification, a service interaction profile MUST satisfy the following
1075 non-functional requirements:

- 1076 1. **Transport protocol** – A service interaction profile MUST define how messages are physically
1077 transported from a sending MDE to a receiving MDE. In so doing, a profile may identify factors that
1078 restrict the range of environments in which the profile is applicable.
- 1079 2. **MDE addressing** – A service interaction profile MUST include a convention for uniquely addressing
1080 each MDE.
- 1081 3. **Operation addressing** – A service interaction profile MUST describe a convention for uniquely
1082 addressing each MDE operation.
- 1083 4. **Request and operation invocation** – A service interaction profile MUST describe a mechanism for a
1084 sending MDE to invoke an operation on the receiving MDE.
- 1085 5. **Synchronous mode response** – A service interaction profile MUST support synchronous operations
1086 in which the response to an operation is always returned immediately, typically within a matter of
1087 seconds, to the invoking MDE.
- 1088 6. **Asynchronous mode response** – A service interaction profile MUST support asynchronous
1089 operations in which the response to an operation may not necessarily be returned immediately to the
1090 invoking MDE. Instead, the response may be returned at some later time through a callback from the
1091 MDE that received the operations to the invoking MDE. The callback MUST include a reference to
1092 the invoking message transmission.
- 1093 7. **Message/attachment delimiters** – A service interaction profile MUST define how the receiving MDE
1094 distinguishes messages from attachments within a message transmission.
- 1095 8. **Message identifiers** – A service interaction profile MUST provide a means for a sending MDE to
1096 assign a unique identifier to each message (including any attachments) within a message
1097 transmission.

1098 In addition, there are some non-functional features that a service interaction profile SHOULD provide,
1099 including:

- 1100 1. **Message non-repudiation** – A service interaction profile SHOULD provide a mechanism so that the
1101 receiving MDE is provided with evidence that demonstrates:
 - 1102 a. the identity of the sending MDE
 - 1103 b. the content of the message(s) transmitted

- 1104 c. the date and time of the message transmission
- 1105 2. **Message integrity** – A service interaction profile SHOULD provide a mechanism so that the
1106 receiving MDE is able to determine whether the message(s) transmitted (including any attachments)
1107 was (were) modified during the message transmission.
- 1108 3. **Message confidentiality** – A service interaction profile SHOULD provide a mechanism, such as
1109 encryption, that can be used with a sending MDE to ensure that the message(s) in a transmission
1110 (including any attachments) can be processed only by the receiving MDE.
- 1111 4. **Message authentication** – A service interaction profile SHOULD provide a mechanism, such that a
1112 sending MDE is required to include, to display credentials that demonstrate its identity to the receiving
1113 MDE in each message transmission.
- 1114 5. **Message transmission reliability** – A service interaction profile SHOULD provide a mechanism,
1115 such that a sending MDE is required to include, to guarantee that a message transmission will be
1116 delivered to the receiving MDE within a specified period of time, or else the sending MDE will receive
1117 notification at the end of that period of time that the message transmission was not deliverable to the
1118 receiving MDE.
- 1119 6. **Message splitting and assembly** – A service interaction profile SHOULD provide a mechanism by
1120 which a large message and attachments MAY be split into multiple pieces that are transmitted
1121 separately by the sending MDE and reassembled into the complete message by the receiving MDE.
1122 In the HTTP 1.1 protocol, this is called “chunking.”
- 1123 7. **Transmission auditing** – A service interaction profile SHOULD provide a mechanism for the MDE to
1124 receive message transmissions in their entirety (both messaging and “payload” content) for auditing
1125 purposes.

1126 6.2 Service Interaction Profile Approval and Revision Processes

1127 The ECF Technical Committee (TC) will recommend certain service interaction profiles for use in
1128 implementations of the ECF 4.1 specification. The TC will consider a service interaction profile for
1129 recommendation for use in ECF 4.1 implementations provided the profile meets the following
1130 requirements:

- 1131 1. The service interaction profile MUST be described in a document in the format of an OASIS
1132 specification.
- 1133 2. The service interaction profile specification MUST identify a unique URI to identify the service
1134 interaction profile and version.
- 1135 3. The service interaction profile specification MUST describe the binding of MDE operations to the
1136 service interaction profile that satisfies the functional requirements described in Section 3 (“ECF 4.1
1137 Process Model”) and Section 4 (“ECF 4.1 Schema”) of this specification.
- 1138 4. The service interaction profile specification MUST demonstrate that the service interaction profile
1139 satisfies the non-functional service interaction profile requirements described in Section 6.1 (“Service
1140 Interaction Profile Requirements”) of this specification.
- 1141 5. The service interaction profile specification MUST include samples that demonstrate how the
1142 messaging information and “payload” content are combined into message transmissions. These
1143 samples MUST include samples that demonstrate both synchronous and asynchronous mode
1144 operations.
- 1145 6. At least one voting member of the ECF TC MUST agree to sponsor the service interaction profile and
1146 submit the service interaction profile specification to the TC for review as a candidate for approval as
1147 an ECF 4.1 compliant service interaction profile.

1148 Certifying that a candidate service interaction profile meets certain service interaction profile requirements
1149 will necessarily involve some subjectivity since service interaction profile requirements cannot be
1150 expressed algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to assess
1151 whether the proposed profile’s description is adequate in meeting the requirements of ECF 4.1 before
1152 approving the service interaction profile specification as a “Committee Draft” through the OASIS
1153 standards approval process.

1154 From time to time, it may be necessary to revise or update a service interaction profile to bring it into
1155 compliance with changes in network and messaging protocols, or to support additional non-functional
1156 requirements. Any revision(s) to previously approved service interaction profiles will be considered a new
1157 service interaction profile and MUST meet the requirements of a new service interaction profile, including
1158 sponsorship by a voting member of the ECF TC and review and approval by the ECF TC. There will be
1159 no guarantees that future versions of a service interaction profile will be backwardly compatible with the
1160 current version.

1161 6.3 Supported Service Interaction Profiles

1162 The following ECF 4.1 service interaction profile specification is for use in conjunction with
1163 implementations of the ECF 4.1 specification:

- 1164 • **Web Services Service Interaction Profile 4.1 Specification** – This specification defines a
1165 transmission system using the specifications described in the Web Services Interoperability (WS-I)
1166 Basic Profile 1.1, W3C SOAP 1.1 Binding for MTOM 1.0, WS-I Basic Security Profile 1.0 and OASIS
1167 WS-Reliable Messaging 1.1.

1168 Additional service interaction profiles, or revisions to these service interaction profiles, may be approved
1169 by the ECF TC for use in conjunction with implementations of the ECF 4.1 specification according to the
1170 process described in Section 6.2 (“Service Interaction Profile Approval and Revision Processes”) above.

1171

1172 The following service interaction profile was defined for previous versions of ECF. Their use is
1173 deprecated for use in conjunction with the ECF 4.1 specification:

- 1174 • **Portable Media Service Interaction Profile 1.01 Specification** – This specification defines a
1175 transmission system in which the sending MDE stores message transmissions on portable media
1176 (e.g., a compact disc), which is then physically transported to the receiving MDE where it is
1177 connected for retrieval of the message transmissions. This specification may be needed in the
1178 absence of an active network between the sending and receiving MDEs.

1179

1180 7 Document Signature Profiles

1181 An ECF document signature profile defines a mechanism for asserting that a person signed a single
1182 electronic or imaged document, which is an attachment to a message transmission. The signing of an
1183 entire message transmission is described in a service interaction profile and is not supported by a
1184 document signature profile.

1185 7.1 Document Signature Profile Requirements

1186 Each document signature profile will define standard conventions and configuration details to support
1187 interoperability in the creation and verification of document signatures between and among ECF
1188 implementations that support the same document signature profile. However, compliance with these
1189 requirements will not necessarily guarantee interoperability.

1190 Except for the Null Document Signature Profile, to be compliant with the ECF 4.1 specification, a
1191 document signature profile MUST satisfy the following non-functional requirements:

- 1192 1. **Signer name assertion** – A document signature profile MUST make an assertion regarding the
1193 name of the person who signed a document.
- 1194 2. **Signed date assertion** – A document signature profile MUST make an assertion regarding the date
1195 the person signed a document.
- 1196 3. **Multiple signatures** – A document signature profile MUST allow multiple signatures to be associated
1197 with the same document.

1198 A signature profile SHOULD provide the following non-functional features:

- 1199 1. **Signer and date non-repudiation** – A document signature profile SHOULD provide a mechanism so
1200 that the receiving MDE is provided with verifiable evidence that demonstrates:
 - 1201 a. the unique identity of the person who signed the document
 - 1202 b. the date the person signed a document
- 1203 2. **Document integrity** – A document signature profile SHOULD provide a mechanism so that the
1204 receiving MDE is able to determine if the document was modified since the person signed the
1205 document.
- 1206 3. **Document signature auditing** – A document signature profile SHOULD provide a mechanism for
1207 the MDE to receive both the document and signatures for auditing purposes.

1208 7.2 Document Signature Profile Approval and Revision Processes

1209 The ECF Technical Committee will recommend certain document signature profiles for use in
1210 implementations of the ECF 4.1 specification. The TC will consider a document signature profile for
1211 recommendation for use in ECF 4.1 implementations provided the profile meets the following
1212 requirements:

- 1213 1. The document signature profile MUST be described in a document in the format of an OASIS
1214 specification.
- 1215 2. The document signature profile specification MUST identify a unique URI to identify the document
1216 signature profile and version.
- 1217 3. If the document signature is not embedded in the document, the document signature profile
1218 specification MUST include an XML structure for describing precisely how the document signature is
1219 represented.
- 1220 4. The document signature profile specification MUST demonstrate that the document signature profile
1221 satisfies the non-functional requirements described in Section 7.1 (“Document Signature Profile
1222 Requirements”) of this specification.

- 1223 5. The document signature profile specification MUST include samples that demonstrate how the
1224 document signature information and “payload” content are combined into message transmissions.
- 1225 6. At least one voting member of the ECF TC MUST agree to sponsor the document signature profile
1226 and submit the document signature profile specification to the TC for review as a candidate for
1227 approval as an ECF document signature profile.

1228 Certifying that a candidate document signature profile meets certain document signature profile
1229 requirements will necessarily involve some subjectivity, since document signature profile requirements
1230 cannot be expressed algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to
1231 assess whether the proposed profile’s description is adequate to the requirements before approving the
1232 profile specification as a Committee Draft through the OASIS standards approval process.

1233 From time to time, it may be necessary to revise or update a document signature profile to bring it into
1234 compliance with changes in authentication and encryption protocols, or to support additional non-
1235 functional requirements. Any revision(s) to previously approved document signature profiles will be
1236 considered a new document signature profile and MUST meet the requirements of a new document
1237 signature profile, including sponsorship by a voting member of the ECF TC and review and approval by
1238 the ECF TC. There will be no guarantees that future versions of document signature profiles will be
1239 backwardly compatible with the current version.

1240 7.3 Supported Document Signature Profiles

1241 The following ECF document signature profile specifications are candidate Committee Drafts for use in
1242 conjunction with implementations of the ECF 4.1 specification:

- 1243 • **Null Document Signature Profile 1.0 Specification** – This specification defines a default
1244 mechanism to describe documents that do not have any associated signatures.
- 1245 • **XML Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1246 associating a W3C XML Signature with a document.
- 1247 • **Application-Specific Document Signature Profile 1.0 Specification** – This specification defines a
1248 mechanism for embedding an application-specific binary signature with a document. This profile
1249 supports the native capabilities in document formats such as Microsoft Word and the Adobe Portable
1250 Document Format (PDF) for describing and embedding signatures.
- 1251 • **Proxy Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1252 indicating documents that are digitally signed by a court filing infrastructure component on behalf of
1253 an authenticated signer.
- 1254 • **Symmetric Key Document Signature Profile 1.0 Specification** – This specification defines a
1255 mechanism for indicating documents that are digitally signed by a trusted entity on behalf of the
1256 signer using a symmetric key known only to the trusted entity.

1257 Additional document signature profiles, or revisions to these document signatures profiles, may be
1258 approved by the ECF TC for use in conjunction with implementation of the ECF 4.1 specification
1259 according to the process described in Section 7.2 (“Document Signature Profile Approval and Revision
1260 Processes”) above.

1261 **8 Conformance**

1262 *An implementation conforms with the Electronic Court Filing Version 4.1 if the implementation meets the*
1263 *requirements in Sections 1-6 including conformance with the XSD schemas and [Genericcode] code lists*
1264 *referenced in Section 3 and 4.*

1265 Appendix A. (Informative) Release Notes

1266 A.1 Availability

1267 Online and downloadable versions of this release are available from the locations specified at the top of
1268 this document.

1269 A.2 Package Structure

1270 The ECF specification is published as a ZIP archive. Unzipping this archive creates a directory
1271 containing this specification document and a number of subdirectories. The files in these subdirectories,
1272 linked to the specification document, contain the various normative and informational pieces of the
1273 release. A description of each subdirectory is given below.

1274 `gc/`

1275 **[Genericcode]** 1.0 code lists

1276 `model/`

1277 UML exchange content model diagrams and spreadsheet models; see Appendix B.3 and B4

1278 `xml/`

1279 Example instances; see Appendix C

1280 `xsd/`

1281 XSD schemas; see Section 4

1282 A.3 Recursive Structures

1283 Certain components in the **[NIEM]** version 2.0 schemas allow recursive nesting. For example, a
1284 `nc:Case` may be related to another `nc:Case`, etc. These are legitimate business data structures. Most
1285 real-world applications will limit the depth of recursion in such structures, but XSD schemas are incapable
1286 of expressing this constraint. Implementers should be aware of this and may wish to set limits on the
1287 depth of recursive structures in their applications. If so, these limits should be described in human-
1288 readable court policy.

1289 A.4 Date and Time Formats

1290 The date and time elements contained in the messages defined by the ECF 4.1 XSD schemas should be
1291 formatted according to the documentation in the **[NIEM]** version 2.0. The **[NIEM]** documentation
1292 indicates the following:

- 1293 • Calendar date values should be expressed as “CCYY-MM-DD”, with an optional time zone qualifier
1294 designated by appending -hh:00, where hh represent the number of hours the local time zone is
1295 behind Coordinated Universal Time (UTC).
- 1296 • Time values should be expressed as “hh:mm:ss.sss”, with an optional time zone qualifier designated
1297 by appending -hh:00, where hh represent the number of hours the local time zone is behind
1298 Coordinated Universal Time (UTC).
- 1299 • Date and time values should be expressed as “CCYY-MM-DDThh:mm:ss.sss” with an optional time
1300 zone designated by appending -hh:00, where hh represent the number of hours the local time zone is
1301 behind Coordinated Universal Time (UTC).qualifier.

1302 These formats are documented in, but not enforced by, the XSD schema at
1303 `xsd/constraint/niem/proxy/xsd/2.0/xsd.xsd`.

1304 **A.5 Known Errata**

1305 Known errors in the ECF 4.1 specification will be identified in an errata document available at:
1306 <http://www.oasis-open.org/committees/legalxml-courtfilling/>.

1307
1308

Appendix B. (Informative) ECF 4.1 Development Approach and Artifacts

1309 This appendix describes the approach used to develop ECF 4.1 and the modeling artifacts.

1310 B.1 Principles

1311 The key principles that guided the design of the ECF 4.1 message structures were:

- 1312 • **Interoperability** – The ECF 4.1 message structures should provide a means for exchanging court
1313 filings among all types of court information systems.
- 1314 • **Completeness** – The ECF Filing 4.0 message structures format should provide for all the elements
1315 of an electronic filing system.
- 1316 • **Simple implementation** – The design should foster rapid implementation.
- 1317 • **Simple XML and portable structure** – The core messages in an ECF 4.1 exchange will be
1318 formatted as XML documents.
- 1319 • **Familiarity** – The data elements and code values should be meaningful to the legal community and
1320 non-expert recipients alike.
- 1321 • **Interdisciplinary and international utility** – The design should be usable by a broad range of court-
1322 related applications and should be applicable internationally.

1323 B.2 Approach

1324 The ECF 4.1 message schemas were developed as a **[NIEM]** Information Exchange Package Definition
1325 (IEPD). A **[NIEM IEPD]** is a collection of artifacts that describe the structure and content of a set of data
1326 that is transmitted for a specific business purpose. It does not specify other interface layers (such as Web
1327 services).

1328 The NIEM Naming and Design Rules (MNDR) **[NIEM NDR]** describe best practices for the development
1329 of NIEM-conformant Information Exchange Packages and documentation. The Design Rules set forth:

- 1330 • A methodology for the construction of **[NIEM]**-conformant exchange documents
- 1331 • Naming and design rules for the artifacts called for by the methodology
- 1332 • Guidelines for the customization of **[NIEM]** schema structures

1333 B.3 ECF 4.1 Exchange Content Models

1334 The ECF 4.1 exchange content models describe the information components used in all of the messages
1335 defined by ECF 4.1.

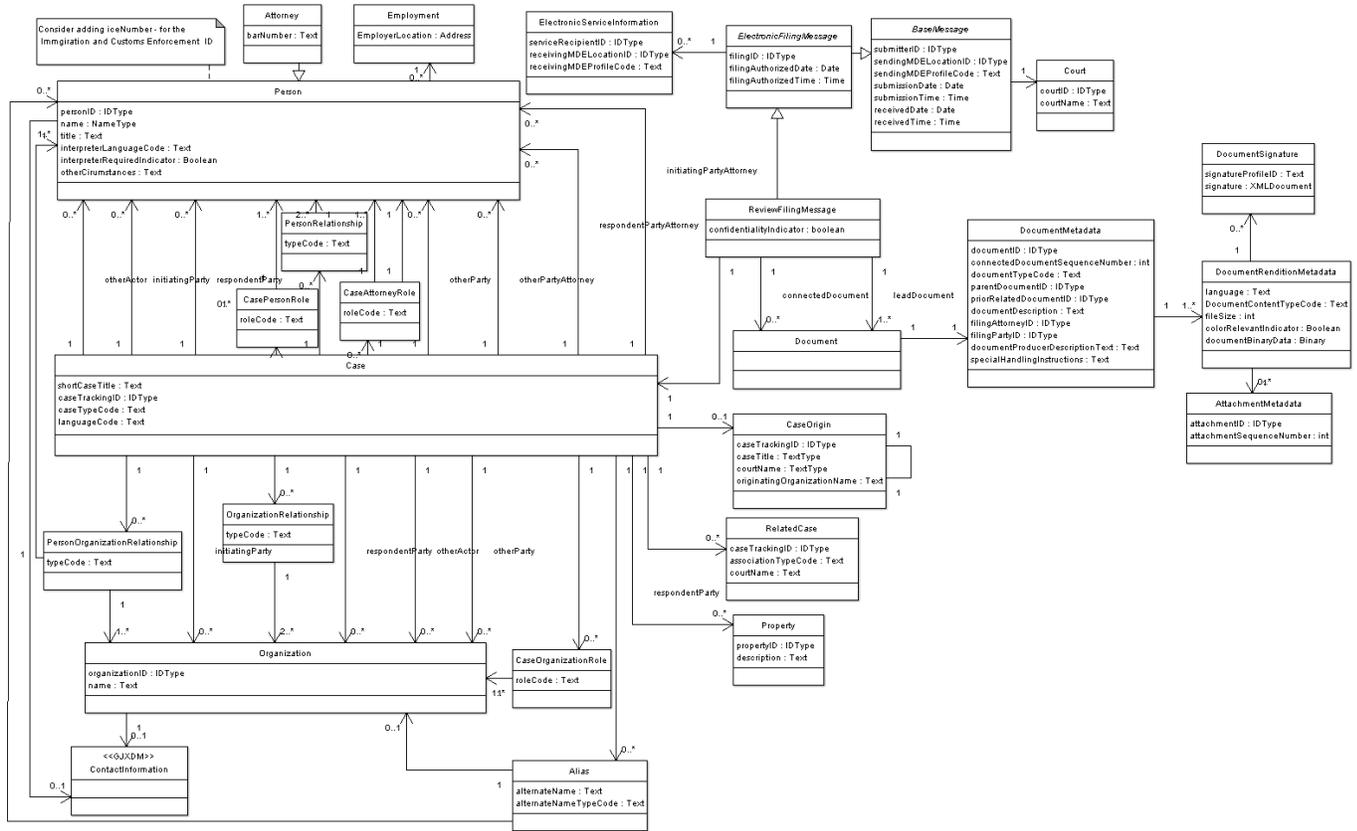
1336 The exchange content models are the result of a detailed analysis of the data requirements to support the
1337 ECF 4.1 Process Model (see Section 3). During the modeling process, common items of data were
1338 identified by a process of normalization to identify aggregates based on functional dependency. Where
1339 appropriate, these were generalized so that they could be re-used to support the various messages.

1340 The exchange content models are used for the following purposes:

- 1341 • They facilitate the identification of the reusable components, i.e., the data structures that are common
1342 across the ECF 4.1 messages.
- 1343 • They aid in understanding the information requirements of the total scenario.
- 1344 • They are the source from which the object classes are derived and documented in the ECF 4.1
1345 schemas (see Section 4).

1346 To facilitate comprehension, the ECF 4.1 is composed of several exchange content model diagrams.
 1347 Each diagram represents a logical grouping of components and displays both the attributes and object
 1348 classes belonging to the components in this grouping. The scope of each diagram is arbitrary and does
 1349 not hold any significance beyond these diagrams.

1350 For example, the ECF 4.1 Review Filing Model diagram is shown below:



1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368

The complete set of exchange content models for all the ECF 4.1 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/CitationFiling.png>

Criminal Filing Model

<model/uml/html/CriminalFiling.png>

Domestic Filing Model

- 1369 [model/uml/html/DomesticFiling.png](#)
- 1370 **Get Calculated Fees Query Model**
- 1371 [model/uml/html/GetFeesCalculationQuery.png](#)
- 1372 **Get Case List Query Model**
- 1373 [model/uml/html/GetCaseListQuery.png](#)
- 1374 **Get Policy Query Model**
- 1375 [model/uml/html/CourtPolicy.png](#)
- 1376 **Get Document Query Model**
- 1377 [model/uml/html/GetDocumentQuery.png](#)
- 1378 **Get Filing List Query Model**
- 1379 [model/uml/html/GetFilingListQuery.png](#)
- 1380 **Get Filing Status Query Model**
- 1381 [model/uml/html/GetFilingStatusQuery.png](#)
- 1382 **Get Service Information Query Model**
- 1383 [model/uml/html/GetServiceInformationQuery.png](#)
- 1384 **Major Design Elements Model**
- 1385 [model/uml/html/MajorDesignElements.png](#)
- 1386 **Juvenile Filing Model**
- 1387 [model/uml/html/JuvenileFiling.png](#)
- 1388 **Record Docketing Model**
- 1389 [model/uml/html/RecordDocketing.png](#)

1390
1391 No specific directions are defined for the associations in these models; they can be navigated in either
1392 direction. The specific navigation path for each association is defined when documents are assembled.

1393 **B.4 Spreadsheet Models**

- 1394 ECF 4.1 uses spreadsheet models to describe the mapping of objects and attributes to **[NIEM]** and ECF
- 1395 4.1 elements. The spreadsheet models use rows to define components. Components are either simple
- 1396 data types or associations. Columns define the metadata associated with each component type.
- 1397 The ECF 4.0 spreadsheet model is located at [model\ECF-4.0-NIEM2-mapping.xls](#).
- 1398

1399

Appendix C. (Informative) Example Instances

1400 Example instances of each ECF 4.1 message are provided in the xml/ subdirectory, as listed below:

1401

1402 **FeesCalculationQueryMessage**

1403 [xml/ECF-4.1-FeesCalculationQueryMessage.xml](#)

1404 **FeesCalculationResponseMessage**

1405 [xml/ECF-4.1-FeesCalculationResponseMessage.xml](#)

1406 **CaseListQueryMessage**

1407 [xml/ECF-4.1-CaseListQueryMessage.xml](#)

1408 **CaseListResponseMessage**

1409 [xml/ECF-4.1-CaseListResponseMessage.xml](#)

1410 **CaseQueryMessage**

1411 [xml/ECF-4.1-CaseQueryMessage.xml](#)

1412 **CaseResponseMessage**

1413 [xml/ECF-4.1-CaseResponseMessage.xml](#)

1414 **CoreFilingMessage (Appellate case type)**

1415 [xml/ECF-4.1-CoreFilingMessage-Appellate.xml](#)

1416 **CoreFilingMessage (Criminal case type)**

1417 [xml/ECF-4.1-CoreFilingMessage-Criminal.xml](#)

1418 **CourtPolicyQueryMessage**

1419 [xml/ECF-4.1-CourtPolicyQueryMessage.xml](#)

1420 **CourtPolicyResponseMessage**

1421 [xml/ECF-4.1-CourtPolicyResponseMessage.xml](#)

1422 **DocumentQueryMessage**

1423 [xml/ECF-4.1-DocumentQueryMessage.xml](#)

1424 **DocumentResponseMessage**

1425 [xml/ECF-4.1-DocumentResponseMessage.xml](#)

1426 **FilingListQueryMessage**

1427 [xml/ECF-4.1-FilingListQueryMessage.xml](#)

1428 **FilingListResponseMessage**

1429 [xml/ECF-4.1-FilingListResponseMessage.xml](#)

1430 **FilingPaymentMessage**

1431 [xml/ECF-4.1-PaymentMessage.xml](#)

1432 **FilingStatusQueryMessage**

1433 [xml/ECF-4.1-FilingStatusQueryMessage.xml](#)

1434 **FilingStatusResponseMessage**

1435 [xml/ECF-4.1-FilingStatusResponseMessage.xml](#)

1436 **MessageReceiptMessage**

1437 [xml/ECF-4.1-MessageReceiptMessage.xml](#)

1438 **PaymentReceiptMessage**

- 1439 [xml/ECF-4.1-PaymentReceiptMessage.xml](#)
- 1440 **RecordDocketingCallbackMessage**
- 1441 [xml/ECF-4.1-RecordDocketingCallbackMessage.xml](#)
- 1442 **RecordDocketingMessage**
- 1443 [xml/ECF-4.1-RecordDocketingMessage.xml](#)
- 1444 **ReviewFilingCallbackMessage**
- 1445 [xml/ECF-4.1-ReviewFilingCallbackMessage.xml](#)
- 1446 **ServiceInformationQueryMessage**
- 1447 [xml/ECF-4.1-ServiceInformationQueryMessage.xml](#)
- 1448 **ServiceInformationResponseMessage**
- 1449 [xml/ECF-4.1-ServiceInformationResponseMessage.xml](#)
- 1450 **ServiceReceiptMessage**
- 1451 [xml/ECF-4.1-ServiceReceiptMessage.xml](#)

1452

Appendix D. (Informative) Ongoing Work Items

1453 The Electronic Court Filing TC plans to continue to revise and expand this specification through future
1454 versions. Future versions of ECF will:

- 1455 • Address filings in administrative tribunals
- 1456 • Address primary service (the delivery of documents such as summonses, subpoenas and warrants
1457 that establish a court's jurisdiction over a party)
- 1458 • Consider how the specifications for filing of documents intended for filing with a court relate to
1459 specifications for filing other documents, e.g., property records, in the offices of elected clerks of
1460 courts
- 1461 • Incorporate feedback from ECF implementations
- 1462 • Support future releases of the **[NIEM]**
- 1463 • Support future **[Court Document]** specifications and integration optimizations
- 1464 • Support non-case related filings into a court clerk's office

1465

Appendix E. (Informative) Acknowledgments

1466 The following court organizations provided lists of data elements required for initiating cases in their case
1467 management information systems:

- 1468 • Administrative Office of United States Courts
 - 1469 ○ Bankruptcy, Civil, Criminal
- 1470 • Arizona Administrative Office of the Courts
 - 1471 ○ Appellate, Civil
- 1472 • Fourth Judicial District Court, Hennepin County, Minneapolis
 - 1473 ○ Criminal
- 1474 • King County Superior Court, Washington
 - 1475 ○ Civil, Criminal
- 1476 • Missouri Office of State Courts Administrator
 - 1477 ○ Citation
- 1478 • Thirteenth Judicial District, Orange County, Florida (through vendor)
 - 1479 ○ Civil, Criminal, Domestic relations, Mental health, Juvenile delinquency/dependency,
 - 1480 Probate, Citation
- 1481 • Utah State Courts
 - 1482 ○ Civil, Criminal

1483

1484 The following individuals have participated in the creation of this specification and are gratefully
1485 acknowledged:

1486 **Participants:**

- 1487 Philip Baughman, Tyler Technologies, Inc.
- 1488 James Cabral, InfoTrack US
- 1489 Eric Eastman, InfoTrack US
- 1490 Ryan Foley, i3-ImageSoft, LLC
- 1491 Gary Graham, Arizona Supreme Court
- 1492 Barbara Holmes, National Center for State Courts
- 1493 George Knecht, InfoTrack US
- 1494 James McMillan, National Center for State Courts
- 1495 Enrique Othon, Tyler Technologies, Inc.
- 1496 Jim Price, Arizona Supreme Court
- 1497 Brock Rogers, File & ServeXpress

Appendix F. (Informative) Revision History

Rev	Date	By Whom	What
Wd01	2022-06-18	James Cabral	Changes to ECF 4.01 OS with errata 02: Relaxed the cardinality of <code><ecf:SendingMDELocationID></code> and <code><ecf:SendingMDEProfileCode></code> in <code><ecf:CaseFilingType></code> to enable MDEs to send messages without requiring an asynchronous message. Added <code><DevelopmentPolicyParametersType>/<RequireAsynchronousResponsesIndicator></code> to <code><CourtPolicyResponseMessage></code> to indicate whether all MDEs MUST support asynchronous responses to messages they send. Relaxed the cardinality of <code><nc:ItemType>/<nc:ItemOtherIdentification></code> , <code><nc:ObligationType>/<nc:ObligationEntity></code> and <code><nc:OrganizationType>/<nc:OrganizationIdentification></code> to allow multiples. Added <code><PersonCitizenshipISO3166Alpha2Code></code> as an alternative to <code><PersonCitizenshipFIPS10-4Code></code> and <code><LocationCountryISO3166Alpha2Code></code> as an alternative to <code><LocationCountryFIPS10-4Code></code> due to the retirement of the FIPS10-4 code list. Added <code>xsd/wrappers.xsd</code> to support document/literal web services.
WD02	2022-06-25	James Cabral Gary Graham	Updated reference to NIEM [MNDR] to version 1.3
WD03	2022-07-04	James Cabral Gary Graham	Updated front matter to conform to current OASIS technical specification template. Updated stage of normative reference to Code List Representation specification to OASIS Committee Specification 02.
WD04	2022-08-23	James Cabral Gary Graham	Replace references to ECF 4.0 with 4.1. In <code>xsd/wrappers.xsd</code> , fixed consistency of message names and changed <code>docketcb:RecordDocketingMessage</code> to <code>docketcb:RecordDocketingCallbackMessage</code> in <code>NotifyDocketingCompleteRequestType</code> .
WD05	2022-09-12	James Cabral Gary Graham	Minor changes to front matter and sections 1.2, 3.1, 3.3.1.8, 3.2.7, 3.2.8 and 5.3.
WD06	2022-11-17	James Cabral Gary Graham	Minor typos corrected in Section 3.
CS01	2022-12-07	James Cabral Gary Graham	Committee Specification Draft 01 approved and posted for public review
WD07	2023-05-10	James Cabral Gary Graham	Moved Appendix C to new Section 5 MDE Operations. Clarified lack of backward compatibility in Section 1.2. Fixed broken links in Section 1.7. Clarified in Section 2.2 that an MDE must support all required operations for the MDE. Clarified in Section 2.4.2 that some form of machine-

Rev	Date	By Whom	What
			readable court policy must exist. Clarified required operations in Section 3.1. Rewrote Sections 3.2.7 and 3.2.8 to improve clarity. Clarified the use of xsd/wrappers.xsd in Section 6.0. Deprecated the use of Portable Media SIP in Section 6.3. Fixed Figure 4 to reflect that NotifyDocketingComplete is optional. Fixed minor formatting issue in Section 7.1, Removed the references to specific versions and filenames in Appendix A.2. Clarified Appendix A.3. Fixed links to images in Appendix B.3. Removed old comments in the files in the /xsd, /xml and /gc folders. Updated the wsu: URI in xsd/wrappers.xsd file.
WD08	2023-05-16	James Cabral Gary Graham	Added cardinality to provided operations in Section 5.
WD09	2023-05-22	James Cabral Gary Graham	Added an example message: ECF-4.1-CoreFilingMessage-Appellate-ROA.xml
WD10	2023-05-31	James Cabral Gary Graham	Under related work, added ECF 4.01 Errata 01. In Section 1,2, updated backward compatibility statement. Fixed minor typos.
WD11	2023-06-23	James Cabral Gary Graham	In 2.4.2 Machine-Readable Court Policy, removed reference to batch filing. In 3.2.8 NotifyFilingReviewComplete, clarified the use of document hash. Relaxed document cardinality by changing cardinality of ReviewFilingRequestType/core:CoreFilingMessage, NotifyFilingReviewCompleteType/reviewcb:ReviewFilingCallbackMessage, RecordFilingRequestType/core:CoreFilingMessage and NotifyDocketingCompleteType/docketcb:RecordDocketingCallbackMessage in xsd/wrappers.xsd and ecf:ElectronicFilingCallbackMessageType/ecf:ReviewedLeadDocument in xsd/common/ECF-4.1-Common.xsd from 1,1 to 1,unbounded.
WD12	2023-06-23	James Cabral Gary Graham	Relaxed document cardinality by changing cardinality of RecordDocketingMessageType/ecf:ReviewedLeadDocument in xsd/message/ECF-4.1-RecordDocketingMessage.xsd from 1,1 to 1,unbounded.
WD13	2023-06-23	James Cabral Gary Graham	Relaxed document cardinality by changing cardinality of feesquery:FeesCalculationQueryMessageType/core:CoreFilingMessage, in xsd/message/ECF-4.1-FeesCalculationQueryMessage.xsd from 1,1 to 1,unbounded. Updated Section 5 with changes to document cardinality.

1499

1500

Appendix G. Notices

1501 Copyright © OASIS Open 2023. All Rights Reserved.

1502 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
1503 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website:
1504 [<https://www.oasis-open.org/policies-guidelines/ipr/>].

1505 This document and translations of it may be copied and furnished to others, and derivative works that
1506 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
1507 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
1508 and this section are included on all such copies and derivative works. However, this document itself may
1509 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
1510 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
1511 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must
1512 be followed) or as required to translate it into languages other than English.

1513 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
1514 or assigns.

1515 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1516 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
1517 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
1518 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1519 PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT,
1520 INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS
1521 DOCUMENT OR ANY PART THEREOF.

1522 As stated in the OASIS IPR Policy, the following three paragraphs in brackets apply to OASIS Standards
1523 Final Deliverable documents (Committee Specifications, OASIS Standards, or Approved Errata).

1524 [OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
1525 necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS
1526 TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims
1527 in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this
1528 deliverable.]

1529 [OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of
1530 any patent claims that would necessarily be infringed by implementations of this OASIS Standards Final
1531 Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner
1532 consistent with the IPR Mode of the OASIS Technical Committee that produced this OASIS Standards
1533 Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so.]

1534 [OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
1535 might be claimed to pertain to the implementation or use of the technology described in this OASIS
1536 Standards Final Deliverable or the extent to which any license under such rights might or might not be
1537 available; neither does it represent that it has made any effort to identify any such rights. Information on
1538 OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS
1539 Technical Committee can be found on the OASIS website. Copies of claims of rights made available for
1540 publication and any assurances of licenses to be made available, or the result of an attempt made to
1541 obtain a general license or permission for the use of such proprietary rights by implementers or users of
1542 this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS
1543 makes no representation that any information or list of intellectual property rights will at any time be
1544 complete, or that any claims in such list are, in fact, Essential Claims.]

1545 The name "OASIS" is a trademark of OASIS, the owner and developer of this document, and should be
1546 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
1547 implementation and use of, documents, while reserving the right to enforce its marks against misleading
1548 uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark/> for above guidance.