



Electronic Court Filing Version 4.1

Committee Specification Draft ~~0102~~

~~07 December 2022~~

17 July 2023

This 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>

Previous stage:

<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd01/ecf-v4.1-csd01.docx> (Authoritative)
<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd01/ecf-v4.1-csd01.html>
<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd01/ecf-v4.1-csd01.pdf>

~~Previous stage~~:

~~N/A~~

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/csd02/xsd/>.
- XML sample messages: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd02/xml/>.
- Model and documentation: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd02/model/>.
- Generic code lists: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd02/gcl/>.
- Specification metadata: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v4.1/csd02/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-courtfileing#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-courtfileing/>.

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-courtfileing/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. ~~07 December 2022~~ 17 July 2023. OASIS Committee Specification Draft ~~0102~~. <https://docs.oasis-open.org/legalxml-courtfileing/ecf/v4.1/csd02/ecf-v4.1-csd02.html>. Latest stage: <https://docs.oasis-open.org/legalxml-courtfileing/ecf/v4.1/ecf-v4.1.html>.

Notices:

Copyright © OASIS Open 202~~2~~3. 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 ~~H~~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.....	9
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.....	10
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.....	24
3.2.1	GetPolicy.....	24
3.2.2	GetServiceInformation.....	24
3.2.3	GetFeesCalculation.....	25
3.2.4	ReviewFiling.....	25
3.2.5	ServeFiling.....	25
3.2.6	RecordFiling.....	25
3.2.7	NotifyDocketingComplete.....	26
3.2.8	NotifyFilingReviewComplete.....	26
3.2.9	GetFilingList.....	26
3.2.10	GetFilingStatus.....	26
3.2.11	GetCaseList.....	27
3.2.12	GetCase.....	27
3.2.13	GetDocument.....	27
3.3	Message Business Rules.....	27
3.3.1	Identifiers.....	27

3.3.1.1	Attachment Identifiers	27
3.3.1.2	Case Identifiers	27
3.3.1.3	Court Identifiers.....	27
3.3.1.4	Document Identifiers	28
3.3.1.5	Filing Identifiers.....	28
3.3.1.6	MDE Identifiers	28
3.3.1.7	Asynchronous responses.....	28
3.3.1.8	Filer and Party Identifiers	28
3.3.2	Code Lists.....	28
3.3.3	Message-Specific Business Rules	30
3.3.3.1	CoreFilingMessage	30
3.3.3.2	PaymentMessage	30
3.3.3.3	RecordDocketingMessage.....	30
3.4	Filing the Record on Appeal	30
4	ECF 4.1 Schemas	33
4.1	ECF 4.1 Case Type Schemas	33
4.2	ECF 4.1 Common Schemas	33
4.3	ECF 4.1 Constraint and Subset Schemas.....	34
4.4	ECF 4.1 Message Schemas.....	34
5	MDE Operations	36
5.1	Filing Assembly MDE.....	36
5.1.1	Provided Operations.....	36
5.1.2	Consumed Operations	36
5.2	Filing Review MDE	37
5.2.1	Provided Operations.....	37
5.2.2	Consumed Operations	37
5.3	Court Record MDE	38
5.3.1	Provided Operations.....	38
5.3.2	Consumed Operations	38
5.4	Legal Service MDE	39
5.4.1	Provided Operations.....	39
5.4.2	Consumed Operations	39
6	Service Interaction Profiles.....	40
6.1	Service Interaction Profile Requirements	40
6.2	Service Interaction Profile Approval and Revision Processes.....	41
6.3	Supported Service Interaction Profiles	42
7	Document Signature Profiles.....	43
7.1	Document Signature Profile Requirements	43
7.2	Document Signature Profile Approval and Revision Processes.....	43
7.3	Supported Document Signature Profiles	44
8	Conformance	45
Appendix A.	(Informative) Release Notes.....	46
A.1	Availability.....	46
A.2	Package Structure	46
A.3	Recursive Structures	46
A.4	Date and Time Formats.....	46
A.5	Known Errata.....	47

Appendix B. (Informative) ECF 4.1 Development Approach and Artifacts	48
B.1 Principles	48
B.2 Approach	48
B.3 ECF 4.1 Exchange Content Models	48
B.4 Spreadsheet Models	50
Appendix C. (Informative) Example Instances.....	55
Appendix D. (Informative) Ongoing Work Items	57
Appendix E. (Informative) Acknowledgments	58
Appendix F. (Informative) Revision History.....	59
Appendix G. Notices	61

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 applications using
83 the 4.01, both based on NIEM 2.x. Applications based on ECF versions which themselves are based on
84 NIEM versions other than NIEM 2.x (such as ECF 3.0, 3.01 and 3.1 specifications) will **certainly not**
85 interoperate successfully with applications using ~~these specifications.~~ **this specification.** This fact is
86 indicated by the assignment of a new major and minor version number to the ~~ECF 4.0, 4.01 and 4.1~~
87 specifications.

88 1.3 ECF Version 4.1

89 ECF 4.1 is a minor enhancement release to address several minor message and schema issues
90 identified by implementers of the ECF 4.0 and 4.01 specifications. All references in this document to ECF
91 4.0 apply to ECF 4.01 and 4.1 as well.

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

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

102 1.3.1 National Information Exchange Model (NIEM)

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

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

119 1.3.2 OASIS Universal Business Language

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

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

126 <AllowanceCharge>
127 Information about a charge or discount price component.
128 <Address>
129 Information about a structured address.
130 <Payment>

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

131 Information directly relating to a specific payment.

132 **1.3.3 W3C XML-Signature Syntax and Processing**

133 The W3C XML Signature Syntax and Processing (**[XMLSIG]**) specification describes a mechanism for
134 signing electronic documents. This mechanism allows recipients of electronic documents to identify the
135 sender and be assured of the validity of the electronically transmitted data. **[XMLSIG]** defines standard
136 means for specifying information content that is to be digitally signed.²

137 ECF 4.1 employs the **[XMLSIG]** specification to describe digital signatures applied to the entire ECF 4.1
138 message transmission in order to provide authentication, encryption and message integrity. **[XMLSIG]** is
139 also used in the ECF 4.1 XML Document Signature Profile.

140 **1.3.4 OASIS Reference Model for Service Oriented Architecture**

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

144 **1.3.5 OASIS Code List Representation (Genericode)**

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

149

150 **1.4 Terms and Definitions**

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

154

155 This section defines key terms used in this specification.

156

157 **Attachment**

158 See definition in Section 2.3.2.

159 **Callback message**

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

162 **Document**

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

165 **Document hash**

166 A condensed representation of a document ~~intended to protect document integrity~~, calculated
167 according to the FIPS 180-4 SHA 256 algorithm.

168 **Docketing**

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

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

171 **Filer**

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

174 **Filing**

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

177 **Hub Service MDE**

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

181 **Major Design Element (MDE)**

182 A logical grouping of operations representing a significant business process supported by ECF
183 4.1. Each MDE operation receives one or more messages, returning a synchronous response
184 message (a reaction to a message received) and, optionally, returning an asynchronous (later)
185 response message to the originating message sender.

186 **Message**

187 See definition in Section 2.3.1.

188 **Message Transmission**

189 The sending of one or more messages and associated attachments to an MDE. Each
190 transmission must invoke or respond to an operation on the receiving MDE, as defined in the
191 ECF 4.1 specification.

192 **Operation (or MDE Operation)**

193 A function provided by an MDE upon receipt of one or more messages. The function provided by
194 the operation represents a significant step in the court filing business process. A sender invokes
195 an operation on an MDE by transmitting a request with an operation identifier and a set of
196 messages.

197 **Operation signature**

198 A definition of the input message and synchronous response message associated with an
199 operation. Each message is given a name and a type by the operation. The type is defined by a
200 single one of the message structures defined in the ECF 4.1 specification.

201 **Synchronous response**

202 A message transmission returned immediately (synchronously) as the result of an operation.
203 Every operation has a synchronous response.

204 **1.5 Symbols and Abbreviations**

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

206

207 **ECF 4.1**

208 Electronic Court Filing 4.1

209 **IEPD**

210 Information Exchange Package Documentation

211 **MDE**

212 Major Design Element

- 213 **NIEM**
214 National Information Exchange Model
- 215 **OASIS**
216 Organization for the Advancement of Structured Information Standards
- 217 **XML**
218 eXtensible Markup Language
- 219 **W3C**
220 World Wide Web Consortium
- 221 **WS-I**
222 Web Services Interoperability Organization
- 223

224 1.6 Normative References

- 225 **[FIPS 180-4]**
226 *Secure Hash Standard*, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>, National Institute for
227 Standards and Technology, March 2012.
- 228 **[Genericode]**
229 G. K. Holman, *Code List Representation (Genericode) 1.0*, [http://docs.oasis-](http://docs.oasis-open.org/codelist/ns/genericode/1.0/)
230 [open.org/codelist/ns/genericode/1.0/](http://docs.oasis-open.org/codelist/ns/genericode/1.0/), OASIS Committee Specification 02, April 06, 2022.
- 231 **[NIEM]**
232 *National Information Exchange Model 2.0*, <http://niem.gov>, US DOJ and DHS, 2007.
- 233 **[NIEM Guide]**
234 *NIEM Implementation Guidelines*, <http://www.niem.gov/implementationguide.php>, US DOJ and DHS,
235 2007.
- 236 **[NIEM Techniques]**
237 *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)
238 [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
239 Research Institute, August 2007.
- 240 **[Namespaces]**
241 T. Bray, *Namespaces in XML*, <http://www.w3.org/TR/1999/REC-xml-names-19990114>, January 14, 1999.
- 242 **[RFC2046]**
243 N. Freed, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*,
244 <http://www.ietf.org/rfc/rfc2046.txt>, IETF RFC 2046, November 1996.
- 245 **[RFC2119]**
246 S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>,
247 IETF RFC 2119, March 1997.
- 248 **[RFC4122]**
249 Leach, et al., *A Universally Unique IDentifier (UUID) URN Namespace*, <http://www.ietf.org/rfc/rfc4122.txt>,
250 IETF RFC 4112, July 2005.
- 251 **[Schema Part 1]**
252 H. S. Thompson, D. Beech, M. Maloney, N. Mendelsohn, *XML Schema Part 1: Structures Second Edition*,
253 <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>, W3C Recommendation, October 28, 2004.
- 254 **[Schema Part 2]**
255 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/)
256 [xmlschema-2-20041028/](http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/), W3C Recommendation, October 28, 2004
- 257 **[UBL]** *Universal Business Language Version 2.1*. 04 November 2013. OASIS Standard.
258 <http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html>.

259 **[XML 1.0]**
 260 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)
 261 [XML-20040204](http://www.w3.org/TR/REC-xml/REC-XML-20040204), W3C Recommendation, February 4, 2004.

262 **[XMLENC]**
 263 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/)
 264 [xmlenc-core-20021210/](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/), W3C Recommendation, December 2002.

265 **[XMLSIG]**
 266 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/)
 267 [xmldsig-core-20020212/](http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/), W3C Recommendation, February 2002.

268 1.7 Non-Normative References

269 **[Court Document]**
 270 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)
 271 [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).

272 **[NIEM MNRD]**
 273 W. Roberts, S. Liebeskind, M. Kindl *National Information Exchange Model Naming and Design Rules Final*
 274 *1.3*, [https://reference.niem.gov/niem/specification/naming-and-design-rules/1.3/niem-ndr-](https://reference.niem.gov/niem/specification/naming-and-design-rules/1.3/niem-ndr-1.3.pdf)
 275 [1.3.pdf](https://reference.niem.gov/niem/specification/naming-and-design-rules/1.3/niem-ndr-1.3.pdf)<https://reference.niem.gov/niem/specification/naming-and-design-rules/1.3/niem-ndr-1.3.pdf>,
 276 October 31, 2008.

277 **[Juvenile XML]**
 278 S. Rondendell, et. al., *Juvenile Justice XML Report*,
 279 [https://web.archive.org/web/20050910223309/http://www.ijis.org/db/share/public/Library/Publications/juve-](https://web.archive.org/web/20050910223309/http://www.ijis.org/db/share/public/Library/Publications/juvenile_justice_xml_final_report_20050630.pdf)
 280 [nile_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.

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

283 **[NCSC Guide]**
 284 *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%201975-77)
 285 [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%201975-77)
 286 [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%201975-77)
 287 [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%201975-77)
 288 [77](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%201975-77), National Center for State Courts, ~~November~~
~~2004~~ [January 2023](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%201975-77).

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

293 **[SOA-RM]**
 294 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)
 295 [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,
 296 2006.

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

2 ECF 4.1 Architecture

300

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

2.1 Core vs. Profiles

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

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

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

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

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.

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

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

355 2.3 Information Model

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

360 2.3.1 Messages

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

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

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

393

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

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

404

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

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

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

413 **2.3.2 Attachment**

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

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

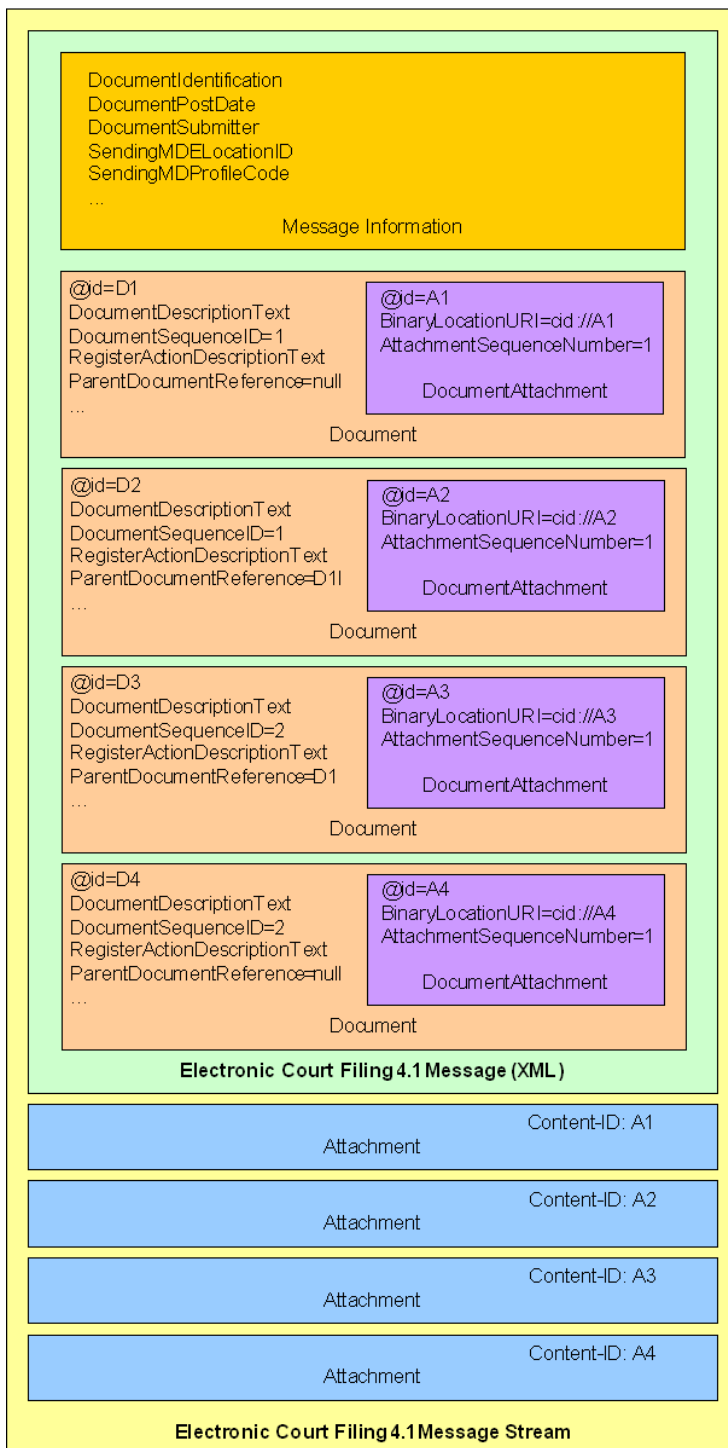
424 **2.3.3 Sample Message Streams**

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

426

427

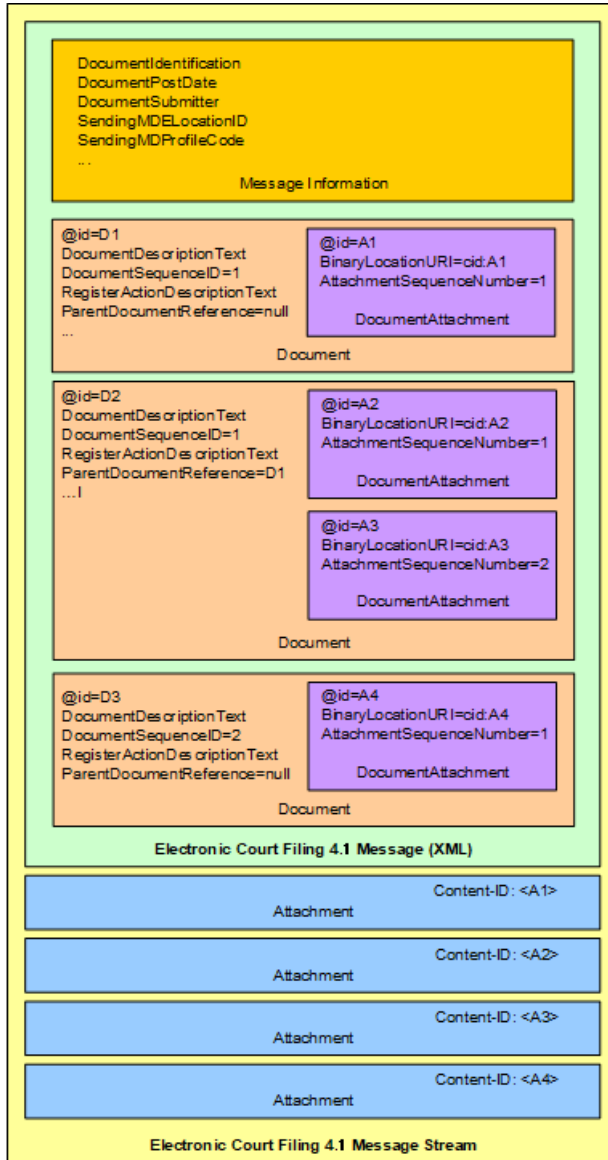
428 Figure 1 illustrates a message stream involving two lead documents, the first of which has two supporting
 429 documents. The second lead document has no supporting documents. Each document is associated
 430 with a single attachment.



431
 432
 433

Figure 1. Simple Message Stream

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



439 **Figure 2. Message Stream with a Document in Multiple Attachments**
 440
 441

442 2.4 Court Policy

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

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

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

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

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

469 2.4.1 Human-Readable Court Policy

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

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

483 2.4.2 Machine-Readable Court Policy

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

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

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

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

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

503 2.4.3 Case-Type and Court Extensions

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

514

515 2.4.4 Court-Specific Code Lists

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

519

- 520 • ECF Code Lists
 - 521 • **Civil Case Type**
 - 522 • `<FiduciaryTypeCode>*`
 - 523 • `<JurisdictionalGroundsCode>`
 - 524 • `<ReliefTypeCode>`
 - 525 • **Domestic Case Type**
 - 526 • `<NoContactCode>*`
 - 527 • `<RequestToVacateCode>`
 - 528 • **Common Types**
 - 529 • `<AliasAlternateNameTypeCode>*`
 - 530 • `<CaseAssociationTypeCode>*`
 - 531 • `<CaseOfficialRoleText>*`

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

559

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

562

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

565

566 2.4.5 Court-Specific Constraint Schemas

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

573

3 ECF 4.1 Process Model

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

577

3.1 The Filing-Preparation-to-Docketing Process Model

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

- 588 • GetPolicy
- 589 • GetServiceInformation
- 590 • GetFeesCalculation
- 591 • **ReviewFiling**
- 592 • ServeFiling
- 593 • **RecordFiling**
- 594 • NotifyDocketingComplete
- 595 • NotifyFilingReviewComplete

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

- 598 • GetFilingList
- 599 • GetFilingStatus

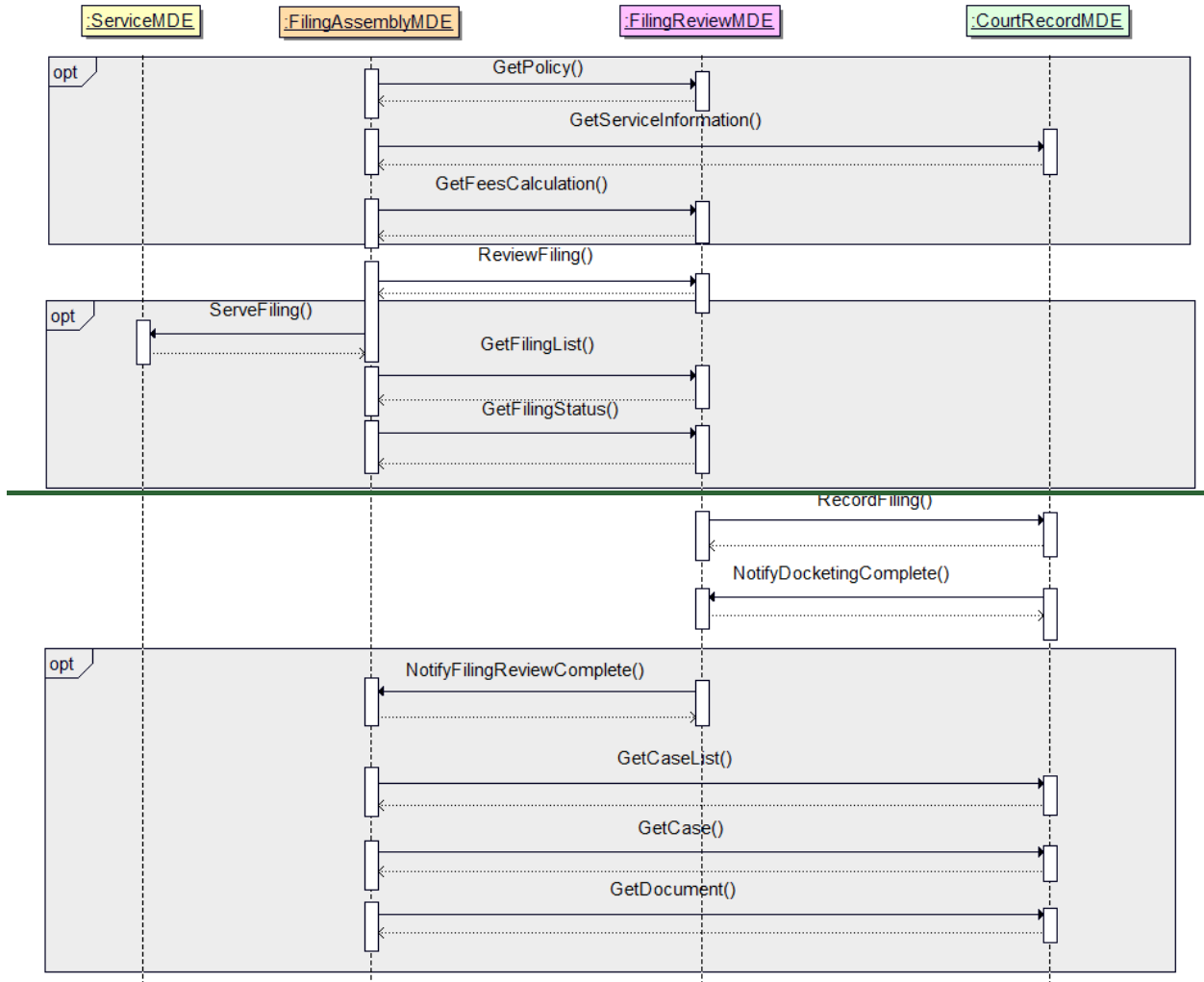
600 At any point, if filing into an existing case, or after the NotifyFilingReviewComplete operation if initiating a
601 case, and if the case is accessible, a party MAY access information through the following operations:

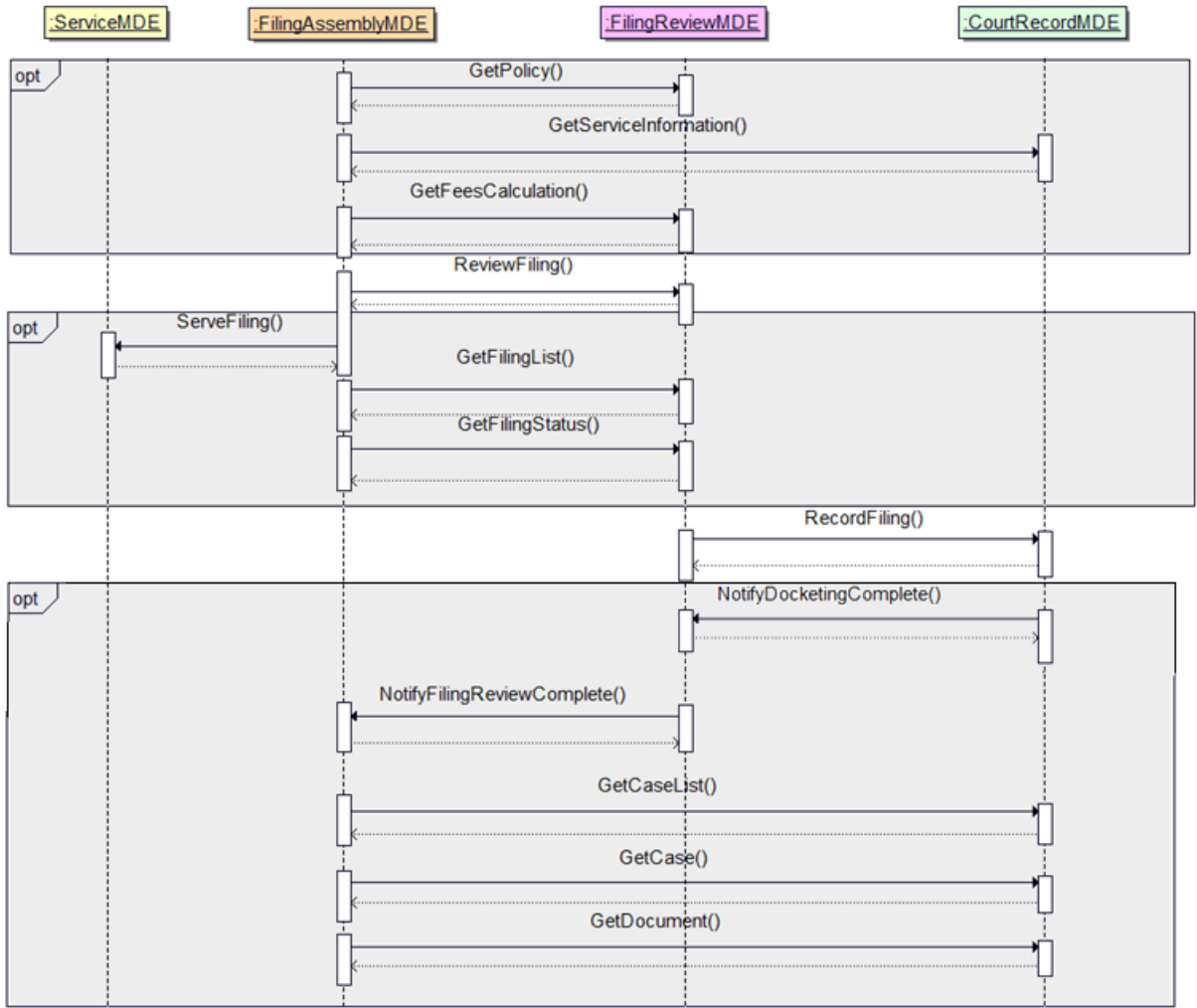
- 602 • GetCaseList
- 603 • GetCase
- 604 • GetDocument

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

607

Figure 4. Filing Preparation to Docketing Process Model





610

611 3.2 Business Rules

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

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

618 3.2.1 GetPolicy

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

624 3.2.2 GetServiceInformation

625 The Filing Assembly MDE MAY obtain the Court's service information for all parties in an existing case at
 626 any time by invoking the GetServiceInformation operation with the appropriate case number on the Court
 627 Record MDE. The service list returned by the GetServiceInformation operation assists the filer in

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

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

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

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

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

642 **3.2.3 GetFeesCalculation**

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

648 **3.2.4 ReviewFiling**

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

653 **3.2.5 ServeFiling**

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

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

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

667 **3.2.6 RecordFiling**

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

672 3.2.7 NotifyDocketingComplete

673 ~~#RecordDocketingCallbackMessage MAY be provided as a callback message by the Record Filing MDE~~
674 ~~to the Filing Review MDE to indicate whether the filing was accepted or rejected by the court system. The~~
675 ~~Filing Review MDE responds synchronously with an acknowledgement of any callback message~~
676 ~~received.~~

677
678 ~~When~~ the <RequireAsynchronousResponsesIndicator> in the court policy is “true”, the Court Record
679 MDE MUST invoke the NotifyDocketingComplete operation on the Filing Review MDE ~~as a~~, otherwise
680 ~~the~~ callback message ~~to the RecordFiling operation to indicate whether the filing was accepted or~~
681 ~~rejected by the court record system.~~ is optional.

682
683 If the Court Record MDE rejected the filing, an explanation MUST be provided, in the callback message
684 when provided to Filing Review MDE. If the Court Record MDE accepts the filing, the docketing
685 information (e.g., date and time the document was entered into the court record, judge assigned,
686 document identifiers and next court event scheduled) MUST be provided. ~~The Filing Review MDE~~
687 ~~responds synchronously with an acknowledgement of the~~ when a callback message. is tendered.

688 3.2.8 NotifyFilingReviewComplete

689 ~~If the clerk rejects the filings or the Filing Review MDE receives the NotifyDocketingComplete message~~
690 ~~and~~ ReviewFilingCallbackMessage and PaymentReceiptMessage MAY be provided as callback
691 messages by the Review Filing MDE to the Filing Assembly MDE to indicate whether the filings were
692 accepted by the clerk. The Filing Assembly MDE responds synchronously with an acknowledgement of
693 any callback message received.

694
695 ~~When~~ the <RequireAsynchronousResponsesIndicator> in the court policy is “true”, the Filing Review MDE
696 MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE ~~as a callback~~
697 ~~message to the ReviewFiling operation to indicate whether the filing was accepted and docketed by the~~
698 ~~clerk and court record system. The operation MAY return the filed documents or links to the documents,~~
699 ~~but MUST include the [FIPS 180-4] SHA 256 document hash, a condensed representation of a document~~
700 ~~intended to protect document integrity.~~ upon receipt of a RecordDocketingCallbackMessage from the
701 Court Record MDE, otherwise the callback message is optional.

702
703 The operation MAY return the filed documents or links to the documents but MUST include the [FIPS
704 180-4] SHA 256 document hash, a condensed representation of the document as currently in the court
705 record.

706
707 If the filing included a payment, and the filing was accepted by the clerk ~~and court record system~~, a
708 receipt for the payment MUST be included in the operation. ~~The Filing Assembly MDE responds~~
709 ~~synchronously with an acknowledgement of the callback message.~~

710 3.2.9 GetFilingList

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

715 3.2.10 GetFilingStatus

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

719 **3.2.11 GetCaseList**

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

723 **3.2.12 GetCase**

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

729 **3.2.13 GetDocument**

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

733 **3.3 Message Business Rules**

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

736 **3.3.1 Identifiers**

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

739 **3.3.1.1 Attachment Identifiers**

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

743 **3.3.1.2 Case Identifiers**

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

746 **3.3.1.3 Court Identifiers**

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

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

751 Examples of conformant court identifiers include:

- 752 • courts.wa.gov:superior.king
- 753 • nmcourts.com:albd.civil
- 754 • uscourts.gov:100
- 755 • courts.gov.bc.ca:appeal

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

757 **3.3.1.4 Document Identifiers**

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

759 **3.3.1.5 Filing Identifiers**

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

762 **3.3.1.6 MDE Identifiers**

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

765 **3.3.1.7 Asynchronous responses**

766 ECF 4.1 messages that support asynchronous responses include <SendingMDELocationID> and
767 <SendingMDEProfileCode> to support the return of the asynchronous response to the sending MDE. If
768 the <RequireAsynchronousResponsesIndicator> in the CourtPolicyResponseMessage is “true”, then both
769 <SendingMDELocationID> and <SendingMDEProfileCode> MUST be included in all ECF 4.1 messages
770 that include these elements.

771 **3.3.1.8 Filer and Party Identifiers**

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

```
775  
776 <nc:PersonOtherIdentification>  
777     <nc:IdentificationID>100<nc:IdentificationID>  
778     <nc:IdentificationCategoryText>ECFFilerID</nc:IdentificationCategoryTex  
779 t>  
780 </nc:PersonOtherIdentification>
```

781
782 In addition to <nc:PersonOtherIdentification>, other elements that may contain a filer identifier
783 include <nc:OrganizationOtherIdentification>, <ecf:FilingPartyID> and
784 <ecf:FilingAttorneyID>.

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

789 **3.3.2 Code Lists**

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

793

- 794 • ECF Code Lists
 - 795 • **Bankruptcy Case Type**
 - 796 • <DebtorTypeCode>*
 - 797 • <EstimatedAssetsValueLevelCode>*
 - 798 • <EstimatedDebtsValueLevelCode>*

- 799 • <NatureOfDebtCode>*
- 800 • <NumberOfCreditorsValueLevelCode>*
- 801 • **Common Types**
- 802 • <FilingStatusCode>*
- 803 • **Court Policy Response Message**
- 804 • <MajorDesignElementNameCode>
- 805 • <OperationNameCode>
- 806 • **Service Receipt Message**
- 807 • <ServiceStatusCode>*
- 808 • **NIEM Code Lists**
- 809 • **ANSI NIST**
- 810 • <FingerPositionCode>
- 811 • **JXDM**
- 812 • <ChargeNCICCode>
- 813 • <DrivingIncidentHazMatCode>
- 814 • <DrivingJurisdictionAuthorityNCICLSTACode>
- 815 • <IdentificationJurisdictionNCICLISCode>
- 816 • <WarrantExtraditionLimitationCode>
- 817 • **NIEM Core**
- 818 • <DocumentLangageCode>
- 819 • <DriverLicenseCommercialClassCode>
- 820 • <DrivingRestrictionCode>
- 821 • <LanguageCode>
- 822 • <LengthUnitCode>
- 823 • <LocationCountryFIPS10-4Code>
- 824 • <LocationCountryISO3166Alpha2Code>
- 825 • <LocationCountyCode>
- 826 • <LocationStateUSPostalServiceCode>
- 827 • <PersonCitizenshipFIPS10-4Code>
- 828 • <PersonCitizenshipISO3166Alpha2Code>
- 829 • <PersonEthnicityCode>
- 830 • <PersonEyeColorCode>
- 831 • <PersonHairColorCode>
- 832 • <PersonRaceCode>
- 833 • <PersonSexCode>
- 834 • <PersonUnionCategoryCode>
- 835 • <PhysicalFeatureCategoryCode>
- 836 • <VehicleColorPrimaryCode>
- 837 • <VehicleMakeCode>

- 838 • <VehicleModelCode>
- 839 • <VehicleStyleCode>
- 840 • <WeightUnitCode>

841

842 Code lists defined using **[GenericCode]** 1.0 are indicated with asterisks (*). The remaining code lists are
843 defined in XSD schema definitions.

844 **3.3.3 Message-Specific Business Rules**

845 The following business rules apply to specific messages:

846 **3.3.3.1 CoreFilingMessage**

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

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

858 **3.3.3.2 PaymentMessage**

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

864 **3.3.3.3 RecordDocketingMessage**

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

867 **3.4 Filing the Record on Appeal**

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

870

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

873

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

875

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

- 876 • The supporting documents that comprise the ROA transaction MAY also have additional attached
877 documents.
- 878
- 879 • All ROA documents being submitted, including the Index of Record document and each
880 document within the record, MUST have at least one court-defined document type that indicates
881 the type of transaction to be performed on the document, and whether the document is being
882 added to or stricken from the record.
- 883
- 884 • The Index of Record document and each document within the ROA transaction MAY also have
885 an additional document type or types, which characterize the document for the Court Record
886 MDE.
- 887
- 888 • When a document within the ROA transaction is being stricken from the court record, the
889 document MUST be identified by the unique document identifier, which was provided by the Court
890 Record MDE when the document was initially filed (See section 3.3.1.4).
- 891
- 892 • A hierarchical structure of case lineage elements MUST be used to express the target case's
893 predecessor cases at prior courts. Each predecessor case MAY also have its own predecessor
894 case, as necessary to express the full lineage of an appellate case.⁴
- 895
- 896 • When the ROA transaction is electronically transferred from one court to another, the target case
897 number in the destination court and the case lineage, which includes the predecessor case
898 number in the sending court, MUST be provided.
- 899
- 900 • If the ROA transaction is a case initiating filing in the destination court, then the <FilingCase>
901 object MUST be present and the <CaseTrackingID> MUST be absent.
- 902
- 903 • Each predecessor case identified in the target case's case lineage may include case type-specific
904 and court-specific extensions. The case type and the case type-specific extensions for each
905 predecessor case MUST be consistent throughout the case lineage.
- 906
- 907 • When a ROA amendment transaction is sent, the Index of Record document MUST reflect the
908 status of the record assuming that the transaction will be accepted. If however the transaction is

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

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

911

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

915

916 • Individual documents within the ROA transaction MUST not be individually accepted or rejected.
917 All documents within the ROA transaction MUST have the same acceptance or rejection
918 disposition.

919

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.

920 4 ECF 4.1 Schemas

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

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

927

928 4.1 ECF 4.1 Case Type Schemas

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

931

932 **AppellateCase**

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

934 **BankruptcyCase**

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

936 **CitationCase**

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

938 **CivilCase**

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

940 **CriminalCase**

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

942 **DomesticCase**

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

944 **JuvenileCase**

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

946

947 4.2 ECF 4.1 Common Schemas

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

950

951 **AppInfo**

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

953 **CommonTypes**

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

955 **DigitalSignature**

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

957 **Genericcode**

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

959 **4.3 ECF 4.1 Constraint and Subset Schemas**

960 The XSD schemas that define the subset of all NIEM elements and types that are used in ECF 4.1
961 messages and/or case type extensions are located in the `xsd/Subset/niem/` folder. As a general
962 data model, NIEM does not define any constraints regarding the minimum and maximum occurrence of
963 elements contained within types. Therefore, in conformance with NIEM, ECF-specific constraints are not
964 included in the schemas within the `xsd/Subset/niem` folder. The XSD schemas in the
965 `xsd/constraint/niem/` folder represent the NIEM subset schemas with the ECF-specific constraints
966 applied and are the schemas by which the ECF message and case type schemas incorporate NIEM
967 elements and types.

968 **4.4 ECF 4.1 Message Schemas**

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

971

972 **CaseListQueryMessage**

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

974 **CaseListResponseMessage**

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

976 **CaseQueryMessage**

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

978 **CaseResponseMessage**

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

980 **CoreFilingMessage**

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

982 **CourtPolicyQueryMessage**

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

984 **CourtPolicyResponseMessage**

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

986 **DocumentQueryMessage**

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

988 **DocumentResponseMessage**

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

990 **FeesCalculationQueryMessage**

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

992 **FeesCalculationResponseMessage**

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

994 **FilingListQueryMessage**

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

996 **FilingListResponseMessage**

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

998 **FilingStatusQueryMessage**

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

1000 **FilingStatusResponseMessage**

1001 [xsd/message/ECF-4.1-FilingStatusResponseMessage.xsd](#)

1002 **MessageReceiptMessage**
1003 [xsd/message/ECF-4.1-MessageReceiptMessage.xsd](#)
1004 **PaymentMessage**
1005 [xsd/message/ECF-4.1-PaymentMessage.xsd](#)
1006 **PaymentReceiptMessage**
1007 [xsd/message/ECF-4.1-PaymentReceiptMessage.xsd](#)
1008 **RecordDocketingCallbackMessage**
1009 [xsd/message/ECF-4.1-RecordDocketingCallbackMessage.xsd](#)
1010 **RecordDocketingMessage**
1011 [xsd/message/ECF-4.1-RecordDocketingMessage.xsd](#)
1012 **ReviewFilingCallbackMessage**
1013 [xsd/message/ECF-4.1-ReviewFilingCallbackMessage.xsd](#)
1014 **ServiceInformationQueryMessage**
1015 [xsd/message/ECF-4.1-ServiceInformationQueryMessage.xsd](#)
1016 **ServiceInformationResponseMessage**
1017 [xsd/message/ECF-4.1-ServiceInformationResponseMessage.xsd](#)
1018 **ServiceReceiptMessage**
1019 [xsd/message/ECF-4.1-ServiceReceiptMessage.xsd](#)
1020

5 MDE Operations

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

5.1 Filing Assembly MDE

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

5.1.1 Provided Operations

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

<u>Operation</u>	<u>Called By</u>	<u>Output</u>	<u>Parameters</u>
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)

5.1.2 Consumed Operations

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

<u>Operation</u>	<u>Provided By</u>	<u>Return Type</u>
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

5.2 Filing Review MDE

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

5.2.1 Provided Operations

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

Operation	Called By	Output	Parameters
ReviewFiling	Filing Assembly MDE	xsd/message/ECF-4.1-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)

5.2.2 Consumed Operations

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

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

1049 **5.3 Court Record MDE**

1050 [The Court Record MDE receives the filed documents from the Filing Review MDE and enters them into](#)
 1051 [the official case record of the court. The Court Record MDE notifies the Filing Review MDE that the filing](#)
 1052 [has been filed.](#)

1053 **5.3.1 Provided Operations**

1054 [The Court Record MDE provides the following operations to other MDEs:](#)

<u>Operation</u>	<u>Called By</u>	<u>Output</u>	<u>Parameters</u>
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)

1055 **5.3.2 Consumed Operations**

1056 [The Court Record MDE calls the following operations in other MDEs:](#)

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

1057 **5.4 Legal Service MDE**

1058 The Legal Service MDE enables a filer or a court to electronically transmit copies of, or links to,
1059 electronically filed documents to other parties who are participating in the case and who are entitled to be
1060 promptly served with the electronically filed documents. The Filing Assembly MDE transmits data and
1061 documents to the Legal Service MDE to inform the case participant that an electronic filing has been
1062 submitted to the court clerk. The Legal Service MDE transmits a callback message to the Filing
1063 Assembly MDE requesting a notification to confirm receipt of the served document.

1064 **5.4.1 Provided Operations**

1065 The Legal Service MDE provides the following operations to other MDEs:

<u>Operation</u>	<u>Called By</u>	<u>Output</u>	<u>Parameters</u>
<u>ServeFiling</u>	<u>Filing Assembly MDE</u>	<u>xsd/message/ECF-4.1- ServiceReceiptMessag e.xsd : ServiceReceiptMessag e (1,1)</u>	<u>xsd/message/ECF-4.1-CoreFilingMessage.xsd : CoreFilingMessage (1,1)</u>

1066 **5.4.2 Consumed Operations**

1067 The Legal Service MDE does not call operations in other MDEs
1068

1069

56 Service Interaction Profiles

1070 An ECF 4.1 service interaction profile defines a transmission system that supports the functional
1071 requirements of electronic filing, along with the MDE operations and message structures, and implements
1072 certain non-functional requirements. A service interaction profile does not govern the content of
1073 messages – message content is described in Sections 2 and 3 of this specification. A service interaction
1074 profile will define how a message gets from the sending MDE to the receiving MDE in a given messaging
1075 framework.

1076

1077 To support interoperability across Service Interaction Profiles, this specification includes
1078 xsd/wrappers.xsd an optional schema document defining the types and elements for each operation on
1079 all Major Design Elements (MDEs) as defined in Section 5 of this specification. Service Interaction
1080 Profiles MAY require this file.

5.16.1 Service Interaction Profile Requirements

1082 Each service interaction profile will define standard conventions and configuration details to support
1083 interoperability between and among ECF 4.1 implementations that support the same service interaction
1084 profile. However, compliance with these requirements will not necessarily guarantee interoperability.

1085 To be compliant with the ECF 4.1 specification, a service interaction profile MUST satisfy the following
1086 non-functional requirements:

- 1087 1. **Transport protocol** – A service interaction profile MUST define how messages are physically
1088 transported from a sending MDE to a receiving MDE. In so doing, a profile may identify factors that
1089 restrict the range of environments in which the profile is applicable.
- 1090 2. **MDE addressing** – A service interaction profile MUST include a convention for uniquely addressing
1091 each MDE.
- 1092 3. **Operation addressing** – A service interaction profile MUST describe a convention for uniquely
1093 addressing each MDE operation.
- 1094 4. **Request and operation invocation** – A service interaction profile MUST describe a mechanism for a
1095 sending MDE to invoke an operation on the receiving MDE.
- 1096 5. **Synchronous mode response** – A service interaction profile MUST support synchronous operations
1097 in which the response to an operation is always returned immediately, typically within a matter of
1098 seconds, to the invoking MDE.
- 1099 6. **Asynchronous mode response** – A service interaction profile MUST support asynchronous
1100 operations in which the response to an operation may not necessarily be returned immediately to the
1101 invoking MDE. Instead, the response may be returned at some later time through a callback from the
1102 MDE that received the operations to the invoking MDE. The callback MUST include a reference to
1103 the invoking message transmission.
- 1104 7. **Message/attachment delimiters** – A service interaction profile MUST define how the receiving MDE
1105 distinguishes messages from attachments within a message transmission.
- 1106 8. **Message identifiers** – A service interaction profile MUST provide a means for a sending MDE to
1107 assign a unique identifier to each message (including any attachments) within a message
1108 transmission.

1109 In addition, there are some non-functional features that a service interaction profile SHOULD provide,
1110 including:

- 1111 1. **Message non-repudiation** – A service interaction profile SHOULD provide a mechanism so that the
1112 receiving MDE is provided with evidence that demonstrates:
 - 1113 a. the identity of the sending MDE
 - 1114 b. the content of the message(s) transmitted
 - 1115 c. the date and time of the message transmission

- 1116 2. **Message integrity** – A service interaction profile SHOULD provide a mechanism so that the
1117 receiving MDE is able to determine whether the message(s) transmitted (including any attachments)
1118 was (were) modified during the message transmission.
- 1119 3. **Message confidentiality** – A service interaction profile SHOULD provide a mechanism, such as
1120 encryption, that can be used with a sending MDE to ensure that the message(s) in a transmission
1121 (including any attachments) can be processed only by the receiving MDE.
- 1122 4. **Message authentication** – A service interaction profile SHOULD provide a mechanism, such that a
1123 sending MDE is required to include, to display credentials that demonstrate its identity to the receiving
1124 MDE in each message transmission.
- 1125 5. **Message transmission reliability** – A service interaction profile SHOULD provide a mechanism,
1126 such that a sending MDE is required to include, to guarantee that a message transmission will be
1127 delivered to the receiving MDE within a specified period of time, or else the sending MDE will receive
1128 notification at the end of that period of time that the message transmission was not deliverable to the
1129 receiving MDE.
- 1130 6. **Message splitting and assembly** – A service interaction profile SHOULD provide a mechanism by
1131 which a large message and attachments MAY be split into multiple pieces that are transmitted
1132 separately by the sending MDE and reassembled into the complete message by the receiving MDE.
1133 In the HTTP 1.1 protocol, this is called “chunking.”
- 1134 7. **Transmission auditing** – A service interaction profile SHOULD provide a mechanism for the MDE to
1135 receive message transmissions in their entirety (both messaging and “payload” content) for auditing
1136 purposes.

1137 **5.26.2 Service Interaction Profile Approval and Revision Processes**

1138 The ECF Technical Committee (TC) will recommend certain service interaction profiles for use in
1139 implementations of the ECF 4.1 specification. The TC will consider a service interaction profile for
1140 recommendation for use in ECF 4.1 implementations provided the profile meets the following
1141 requirements:

- 1142 1. The service interaction profile MUST be described in a document in the format of an OASIS
1143 specification.
- 1144 2. The service interaction profile specification MUST identify a unique URI to identify the service
1145 interaction profile and version.
- 1146 3. The service interaction profile specification MUST describe the binding of MDE operations to the
1147 service interaction profile that satisfies the functional requirements described in Section 3 (“ECF 4.1
1148 Process Model”) and Section 4 (“ECF 4.1 Schema”) of this specification.
- 1149 4. The service interaction profile specification MUST demonstrate that the service interaction profile
1150 satisfies the non-functional service interaction profile requirements described in Section 6.1 (“Service
1151 Interaction Profile Requirements”) of this specification.
- 1152 5. The service interaction profile specification MUST include samples that demonstrate how the
1153 messaging information and “payload” content are combined into message transmissions. These
1154 samples MUST include samples that demonstrate both synchronous and asynchronous mode
1155 operations.
- 1156 6. At least one voting member of the ECF TC MUST agree to sponsor the service interaction profile and
1157 submit the service interaction profile specification to the TC for review as a candidate for approval as
1158 an ECF 4.1 compliant service interaction profile.

1159 Certifying that a candidate service interaction profile meets certain service interaction profile requirements
1160 will necessarily involve some subjectivity since service interaction profile requirements cannot be
1161 expressed algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to assess
1162 whether the proposed profile’s description is adequate in meeting the requirements of ECF 4.1 before
1163 approving the service interaction profile specification as a “Committee Draft” through the OASIS
1164 standards approval process.

1165 From time to time, it may be necessary to revise or update a service interaction profile to bring it into
1166 compliance with changes in network and messaging protocols, or to support additional non-functional
1167 requirements. Any revision(s) to previously approved service interaction profiles will be considered a new
1168 service interaction profile and MUST meet the requirements of a new service interaction profile, including
1169 sponsorship by a voting member of the ECF TC and review and approval by the ECF TC. There will be
1170 no guarantees that future versions of a service interaction profile will be backwardly compatible with the
1171 current version.

1172 ~~5.3~~6.3 Supported Service Interaction Profiles

1173 The following ECF 4.1 service interaction profile ~~specifications are~~specification is for use in conjunction
1174 with implementations of the ECF 4.1 specification:

- 1175 • **Web Services Service Interaction Profile 4.1 Specification** – This specification defines a
1176 transmission system using the specifications described in the Web Services Interoperability (WS-I)
1177 Basic Profile 1.1, W3C SOAP 1.1 Binding for MTOM 1.0, WS-I Basic Security Profile 1.0 and OASIS
1178 WS-Reliable Messaging 1.1.

1179 Additional service interaction profiles, or revisions to these service interaction profiles, may be approved
1180 by the ECF TC for use in conjunction with implementations of the ECF 4.1 specification according to the
1181 process described in Section 6.2 (“Service Interaction Profile Approval and Revision Processes”) above.

1182

1183 The following service interaction profile was defined for previous versions of ECF. Their use is
1184 deprecated for use in conjunction with the ECF 4.1 specification:

- 1185 • **Portable Media Service Interaction Profile 1.01 Specification** – This specification defines a
1186 transmission system in which the sending MDE stores message transmissions on portable media
1187 (e.g., a compact disc), which is then physically transported to the receiving MDE where it is
1188 connected for retrieval of the message transmissions. This specification may be needed in the
1189 absence of an active network between the sending and receiving MDEs.

1190

1191 ~~Additional service interaction profiles, or revisions to these service interaction profiles, may be approved~~
1192 ~~by the ECF TC for use in conjunction with implementations of the ECF 4.1 specification according to the~~
1193 ~~process described in Section (“”) above.~~

1194 **6.7 Document Signature Profiles**

1195 An ECF document signature profile defines a mechanism for asserting that a person signed a single
1196 electronic or imaged document, which is an attachment to a message transmission. The signing of an
1197 entire message transmission is described in a service interaction profile and is not supported by a
1198 document signature profile.

1199 **6.1.7.1 Document Signature Profile Requirements**

1200 Each document signature profile will define standard conventions and configuration details to support
1201 interoperability in the creation and verification of document signatures between and among ECF ~~4.1~~ ECF
1202 implementations that support the same document signature profile. However, compliance with these
1203 requirements will not necessarily guarantee interoperability.

1204 Except for the Null Document Signature Profile, to be compliant with the ECF 4.1 specification, a
1205 document signature profile **MUST** satisfy the following non-functional requirements:

- 1206 1. **Signer name assertion** – A document signature profile **MUST** make an assertion regarding the
1207 name of the person who signed a document.
- 1208 2. **Signed date assertion** – A document signature profile **MUST** make an assertion regarding the date
1209 the person signed a document.
- 1210 3. **Multiple signatures** – A document signature profile **MUST** allow multiple signatures to be associated
1211 with the same document.

1212 A signature profile **SHOULD** provide the following non-functional features:

- 1213 1. **Signer and date non-repudiation** – A document signature profile **SHOULD** provide a mechanism so
1214 that the receiving MDE is provided with verifiable evidence that demonstrates:
 - 1215 a. the unique identity of the person who signed the document
 - 1216 b. the date the person signed a document
- 1217 2. **Document integrity** – A document signature profile **SHOULD** provide a mechanism so that the
1218 receiving MDE is able to determine if the document was modified since the person signed the
1219 document.
- 1220 3. **Document signature auditing** – A document signature profile **SHOULD** provide a mechanism for
1221 the MDE to receive both the document and signatures for auditing purposes.

1222 **6.2.7.2 Document Signature Profile Approval and Revision Processes**

1223 The ECF Technical Committee will recommend certain document signature profiles for use in
1224 implementations of the ECF 4.1 specification. The TC will consider a document signature profile for
1225 recommendation for use in ECF 4.1 implementations provided the profile meets the following
1226 requirements:

- 1227 1. The document signature profile **MUST** be described in a document in the format of an OASIS
1228 specification.
- 1229 2. The document signature profile specification **MUST** identify a unique URI to identify the document
1230 signature profile and version.
- 1231 3. If the document signature is not embedded in the document, the document signature profile
1232 specification **MUST** include an XML structure for describing precisely how the document signature is
1233 represented.
- 1234 4. The document signature profile specification **MUST** demonstrate that the document signature profile
1235 satisfies the non-functional requirements described in Section 7.1 (“Document Signature Profile
1236 Requirements”) of this specification.
- 1237 5. The document signature profile specification **MUST** include samples that demonstrate how the
1238 document signature information and “payload” content are combined into message transmissions.

1239 6. At least one voting member of the ECF TC MUST agree to sponsor the document signature profile
1240 and submit the document signature profile specification to the TC for review as a candidate for
1241 approval as an ECF document signature profile.

1242 Certifying that a candidate document signature profile meets certain document signature profile
1243 requirements will necessarily involve some subjectivity, since document signature profile requirements
1244 cannot be expressed algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to
1245 assess whether the proposed profile's description is adequate to the requirements before approving the
1246 profile specification as a Committee Draft through the OASIS standards approval process.

1247 From time to time, it may be necessary to revise or update a document signature profile to bring it into
1248 compliance with changes in authentication and encryption protocols, or to support additional non-
1249 functional requirements. Any revision(s) to previously approved document signature profiles will be
1250 considered a new document signature profile and MUST meet the requirements of a new document
1251 signature profile, including sponsorship by a voting member of the ECF TC and review and approval by
1252 the ECF TC. There will be no guarantees that future versions of document signature profiles will be
1253 backwardly compatible with the current version.

1254 ~~6.3~~7.3 Supported Document Signature Profiles

1255 The following ECF document signature profile specifications are candidate Committee Drafts for use in
1256 conjunction with implementations of the ECF 4.1 specification:

- 1257 • **Null Document Signature Profile 1.0 Specification** – This specification defines a default
1258 mechanism to describe documents that do not have any associated signatures.
- 1259 • **XML Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1260 associating a W3C XML Signature with a document.
- 1261 • **Application-Specific Document Signature Profile 1.0 Specification** – This specification defines a
1262 mechanism for embedding an application-specific binary signature with a document. This profile
1263 supports the native capabilities in document formats such as Microsoft Word and the Adobe Portable
1264 Document Format (PDF) for describing and embedding signatures.
- 1265 • **Proxy Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1266 indicating documents that are digitally signed by a court filing infrastructure component on behalf of
1267 an authenticated signer.
- 1268 • **Symmetric Key Document Signature Profile 1.0 Specification** – This specification defines a
1269 mechanism for indicating documents that are digitally signed by a trusted entity on behalf of the
1270 signer using a symmetric key known only to the trusted entity.

1271 Additional document signature profiles, or revisions to these document signature profiles, may be
1272 approved by the ECF TC for use in conjunction with implementation of the ECF 4.1 specification
1273 according to the process described in Section 7.2 (“Document Signature Profile Approval and Revision
1274 Processes”) above.

1275

78 Conformance

1276

An implementation conforms with the Electronic Court Filing Version 4.1 if the implementation meets the requirements in Sections 1-6 including conformance with the XSD schemas and [Genericcode] code lists

1277

referenced in Section 3 and 4.

1278

1279 Appendix A. (Informative) Release Notes

1280 A.1 Availability

1281 Online and downloadable versions of this release are available from the locations specified at the top of
1282 this document.

1283 A.2 Package Structure

1284 The ECF ~~4.1~~ specification is published as a ZIP archive ~~named ecf-v4.1.zip~~. Unzipping this archive
1285 creates a directory ~~named ecf-4.1/~~ containing this specification document and a number of
1286 subdirectories. The files in these subdirectories, linked to the specification document, contain the various
1287 normative and informational pieces of the ~~1.0~~ release. A description of each subdirectory is given below.

1288 gc/

1289 [Genericcode] 1.0 code lists

1290 model/

1291 ~~ECF 4.1~~ UML exchange content model diagrams and spreadsheet models; see Appendix B.3 and
1292 B4

1293 xml/

1294 Example instances; see Appendix ~~D~~C

1295 xsd/

1296 XSD schemas; see Section 4

1297 A.3 Recursive Structures

1298 Certain components in the [NIEM] version 2.0 schemas allow recursive nesting. For example, a
1299 `nc:Case` may be related to another `nc:Case`, etc. These are legitimate business data structures. Most
1300 real-world applications will limit the depth of recursion in such structures, but XSD schemas are incapable
1301 of expressing this constraint. Implementers should be aware of this and may wish to set limits on the
1302 depth of recursive structures in their applications. If so, these limits should be described in human-
1303 readable court policy.

1304 A.4 Date and Time Formats

1305 The date and time elements contained in the messages defined by the ECF 4.1 XSD schemas should be
1306 formatted according to the documentation in the [NIEM] version 2.0. The [NIEM] documentation
1307 indicates the following:

- 1308 • Calendar date values should be expressed as “CCYY-MM-DD”, with an optional time zone qualifier
1309 designated by appending -hh:00, where hh represent the number of hours the local time zone is
1310 behind Coordinated Universal Time (UTC).
- 1311 • Time values should be expressed as “hh:mm:ss.sss”, with an optional time zone qualifier designated
1312 by appending -hh:00, where hh represent the number of hours the local time zone is behind
1313 Coordinated Universal Time (UTC).
- 1314 • Date and time values should be expressed as “CCYY-MM-DDThh:mm:ss.sss” with an optional time
1315 zone designated by appending -hh:00, where hh represent the number of hours the local time zone is
1316 behind Coordinated Universal Time (UTC).qualifier.

1317 These formats are documented in, but not enforced by, the XSD schema at
1318 `xsd/constraint/niem/proxy/xsd/2.0/xsd.xsd`.

1319 **A.5 Known Errata**

1320 Known errors in the ECF 4.1 specification will be identified in an errata document available at:

1321 <http://www.oasis-open.org/committees/legalxml-courtfilling/>.

1322 Appendix B. (Informative) ECF 4.1 Development 1323 Approach and Artifacts

1324 This appendix describes the approach used to develop ECF 4.1 and the modeling artifacts.

1325 B.1 Principles

1326 The key principles that guided the design of the ECF 4.1 message structures were:

- 1327 • **Interoperability** – The ECF 4.1 message structures should provide a means for exchanging court
1328 filings among all types of court information systems.
- 1329 • **Completeness** – The ECF Filing 4.0 message structures format should provide for all the elements
1330 of an electronic filing system.
- 1331 • **Simple implementation** – The design should foster rapid implementation.
- 1332 • **Simple XML and portable structure** – The core messages in an ECF 4.1 exchange will be
1333 formatted as XML documents.
- 1334 • **Familiarity** – The data elements and code values should be meaningful to the legal community and
1335 non-expert recipients alike.
- 1336 • **Interdisciplinary and international utility** – The design should be usable by a broad range of court-
1337 related applications and should be applicable internationally.

1338 B.2 Approach

1339 The ECF 4.1 message schemas were developed as a **[NIEM]** Information Exchange Package Definition
1340 (IEPD). A **[NIEM IEPD]** is a collection of artifacts that describe the structure and content of a set of data
1341 that is transmitted for a specific business purpose. It does not specify other interface layers (such as Web
1342 services).

1343 The NIEM Naming and Design Rules (MNDR) **[NIEM NDR]** describe best practices for the development
1344 of NIEM-conformant Information Exchange Packages and documentation. The Design Rules set forth:

- 1345 • A methodology for the construction of **[NIEM]**-conformant exchange documents
- 1346 • Naming and design rules for the artifacts called for by the methodology
- 1347 • Guidelines for the customization of **[NIEM]** schema structures

1348 B.3 ECF 4.1 Exchange Content Models

1349 The ECF 4.1 exchange content models describe the information components used in all of the messages
1350 defined by ECF 4.1.

1351 The exchange content models are the result of a detailed analysis of the data requirements to support the
1352 ECF 4.1 Process Model (see Section 3). During the modeling process, common items of data were
1353 identified by a process of normalization to identify aggregates based on functional dependency. Where
1354 appropriate, these were generalized so that they could be re-used to support the various messages.

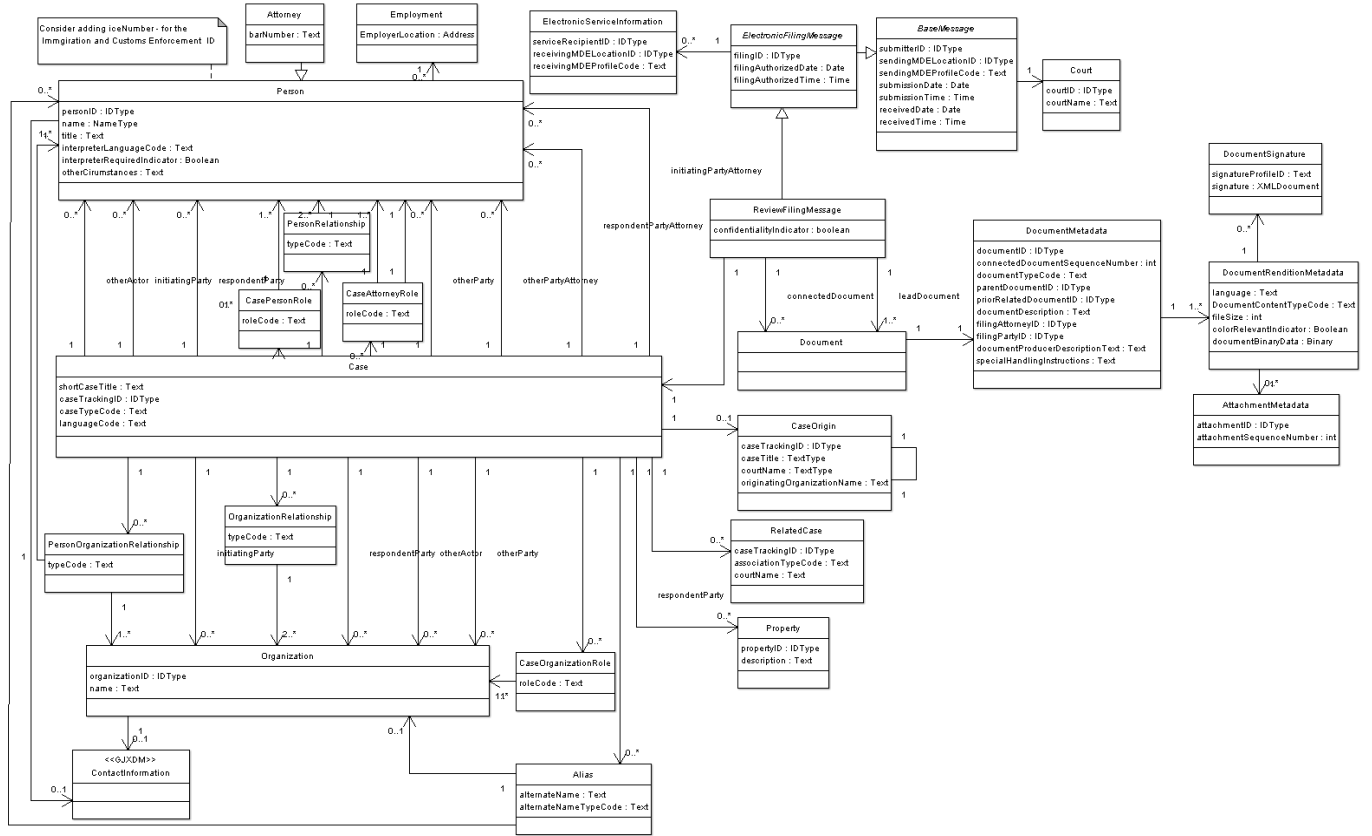
1355 The exchange content models are used for the following purposes:

- 1356 • They facilitate the identification of the reusable components, i.e., the data structures that are common
1357 across the ECF 4.1 messages.
- 1358 • They aid in understanding the information requirements of the total scenario.
- 1359 • They are the source from which the object classes are derived and documented in the ECF 4.1
1360 schemas (see Section 4).

1361 To facilitate comprehension, the ECF 4.1 is composed of several exchange content model diagrams.
1362 Each diagram represents a logical grouping of components and displays both the attributes and object

1363 classes belonging to the components in this grouping. The scope of each diagram is arbitrary and does
1364 not hold any significance beyond these diagrams.

1365 For example, the ECF 4.1 Review Filing Model diagram is shown below:



1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385

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

<model/uml/html/DomesticFiling.png>

Get Calculated Fees Query Model

- 1386 <model/uml/html/GetFeesCalculationQuery.png>
- 1387 **Get Case List Query Model**
- 1388 <model/uml/html/GetCaseListQuery.png>
- 1389 **Get Policy Query Model**
- 1390 <model/uml/html/CourtPolicy.png>
- 1391 **Get Document Query Model**
- 1392 <model/uml/html/GetDocumentQuery.png>
- 1393 **Get Filing List Query Model**
- 1394 <model/uml/html/GetFilingListQuery.png>
- 1395 **Get Filing Status Query Model**
- 1396 <model/uml/html/GetFilingStatusQuery.png>
- 1397 **Get Service Information Query Model**
- 1398 <model/uml/html/GetServiceInformationQuery.png>
- 1399 **Major Design Elements Model**
- 1400 <model/uml/html/MajorDesignElements.png>
- 1401 **Juvenile Filing Model**
- 1402 <model/uml/html/JuvenileFiling.png>
- 1403 **Record Docketing Model**
- 1404 <model/uml/html/RecordDocketing.png>
- 1405 **Review Filing Model**

1406

1407 No specific directions are defined for the associations in these models; they can be navigated in either
1408 direction. The specific navigation path for each association is defined when documents are assembled.

1409 **B.4 Spreadsheet Models**

1410 ECF 4.1 uses spreadsheet models to describe the mapping of objects and attributes to **[NIEM]** and ECF
1411 4.1 elements. The spreadsheet models use rows to define components. Components are either simple
1412 data types or associations. Columns define the metadata associated with each component type.

1413 The ECF 4.0 spreadsheet model is located at <model\ECF-4.0-NIEM2-mapping.xls>.

1414

81 (Informative) MDE Operations

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

8.11.1 C.1 Filing Assembly MDE

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

8.1.11.1.1 C.1.1 Provided Operations

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	xsd/message/ECF-4.1-ReviewFilingCallbackMessage.xsd: ReviewFilingCallbackMessage xsd/message/ECF-4.1-PaymentReceiptMessage.xsd: PaymentReceiptMessage

8.1.21.1.1 C.1.2 Consumed Operations

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

Operation	Provided-By	Return-Type
GetPolicy	Filing-Review MDE	xsd/message/ECF-4.1-CourtPolicyQueryMessage.xsd: CourtPolicyResponseMessage
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

1430 **~~8.21.1 C.2 Filing Review MDE~~**

1431 ~~The Filing Review MDE receives, presents and manages the filings. The Filing Review MDE receives~~
 1432 ~~filings in a standard format and presents those filings to a Clerk for review, where they may be accepted~~
 1433 ~~or rejected. The Filing Review MDE transmits data and documents to the Filing Assembly MDE to inform~~
 1434 ~~the filer that the filing has been accepted or rejected. The Filing Review MDE transmits data and~~
 1435 ~~documents for accepted filings to the Court Record MDE for docketing and recording. While there will~~
 1436 ~~generally be one Filing Review MDE per court, there is no physical barrier to having more than one,~~
 1437 ~~particularly if a court wants to support different Filing Review MDEs for particular case types.~~

1438 **~~8.2.11.1.1 C.2.1 Provided Operations~~**

1439 ~~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	xsd/message/ECF-4.1-CoreFilingMessage.xsd : CoreFilingMessage
			xsd/message/ECF-4.1-PaymentMessage.xsd : PaymentMessage
NotifyDocketingComplete	Court Docketing MDE	xsd/message/ECF-4.1-MessageReceiptMessage.xsd : MessageReceiptMessage	xsd/message/ECF-4.1-RecordDocketingCallbackMessage.xsd : RecordDocketingCallbackMessage
GetFeesCalculation	Filing Assembly MDE	xsd/message/ECF-4.1-FeesCalculationResponseMessage.xsd : FeesCalculationResponseMessage	xsd/message/ECF-4.1-FeesCalculationQueryMessage.xsd : FeesCalculationQueryMessage
GetFilingList	Filing Assembly MDE	xsd/message/ECF-4.1-FilingListResponseMessage.xsd : FilingListResponseMessage	xsd/message/ECF-4.1-FilingListQueryMessage.xsd : FilingListQueryMessage
GetFilingStatus	Filing Assembly MDE	xsd/message/ECF-4.1-FilingStatusResponseMessage.xsd : FilingStatusResponseMessage	xsd/message/ECF-4.1-FilingStatusQueryMessage.xsd : FilingStatusQueryMessage
GetPolicy	Filing Assembly MDE	xsd/message/ECF-4.1-CourtPolicyQueryMessage.xsd : CourtPolicyReponseMessage	xsd/message/ECF-4.1-CourtPolicyQueryMessage.xsd : CourtPolicyQueryMessage

1440 **~~8.2.21.1.1 C.2.2 Consumed Operations~~**

1441 ~~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

1442 **~~8.31.1 C.3 Court Record MDE~~**

1443 ~~The Court Record MDE receives the filed documents from the Filing Review MDE and enters them into~~
 1444 ~~the official case record of the court. The Court Record MDE notifies the Filing Review MDE that the filing~~
 1445 ~~has been filed.~~

1446 **~~8.3.11.1.1 C.3.1 Provided Operations~~**

1447 ~~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	xsd/message/ECF-4.1-RecordDocketingMessage.xsd : RecordDocketingMessage
			xsd/message/ECF-4.14.0-CoreFilingMessage.xsd : CoreFilingMessage
GetCase	Filing Assembly MDE	xsd/message/ECF-4.1-CaseResponseMessage.xsd : CaseResponseMessage	xsd/message/ECF-4.1-CaseQueryMessage.xsd : CaseQueryMessage
GetCaseList	Filing Assembly MDE	xsd/message/ECF-4.1-CaseListResponseMessage.xsd : CaseListResponseMessage	xsd/message/ECF-4.1-CaseListQueryMessage.xsd : CaseListQueryMessage
GetServiceInformation	Filing Assembly MDE	xsd/message/ECF-4.1-ServiceInformationResponseMessage.xsd : ServiceInformationResponseMessage	xsd/message/ECF-4.1-ServiceInformationQueryMessage.xsd : ServiceInformationQueryMessage
GetDocument	Filing Assembly MDE	xsd/message/ECF-4.1-DocumentResponseMessage.xsd : DocumentResponseMessage	xsd/message/ECF-4.1-DocumentQueryMessage.xsd : DocumentQueryMessage

1448 **~~8.3.21.1.1 C.3.2 Consumed Operations~~**

1449 ~~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

1450 **~~8.41.1 C.4 Legal Service MDE~~**

1451 ~~The Legal Service MDE enables a filer or a court to electronically transmit copies of, or links to,~~
1452 ~~electronically filed documents to other parties who are participating in the case and who are entitled to be~~
1453 ~~promptly served with the electronically filed documents. The Filing Assembly MDE transmits data and~~
1454 ~~documents to the Legal Service MDE to inform the case participant that an electronic filing has been~~
1455 ~~submitted to the court clerk. The Legal Service MDE transmits a callback message to the Filing~~
1456 ~~Assembly MDE requesting a notification to confirm receipt of the served document.~~

1457 **~~8.4.11.1.1 C.4.1 Provided Operations~~**

1458 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	xsd/message/ECF-4.1-CoreFilingMessage.xsd: CoreFilingMessage

1459 **~~8.4.21.1.1 C.4.2 Consumed Operations~~**

1460 ~~The Legal Service MDE does not call operations in other MDEs~~

1461 Appendix C. (Informative) Example Instances

1462 Example instances of each ECF 4.1 message are provided in the xml/ subdirectory, as listed below:

1463

1464 **FeesCalculationQueryMessage**

1465 [xml/ECF-4.1-FeesCalculationQueryMessage.xml](#)

1466 **FeesCalculationResponseMessage**

1467 [xml/ECF-4.1-FeesCalculationResponseMessage.xml](#)

1468 **CaseListQueryMessage**

1469 [xml/ECF-4.1-CaseListQueryMessage.xml](#)

1470 **CaseListResponseMessage**

1471 [xml/ECF-4.1-CaseListResponseMessage.xml](#)

1472 **CaseQueryMessage**

1473 [xml/ECF-4.1-CaseQueryMessage.xml](#)

1474 **CaseResponseMessage**

1475 [xml/ECF-4.1-CaseResponseMessage.xml](#)

1476 **CoreFilingMessage (Appellate case type)**

1477 [xml/ECF-4.1-CoreFilingMessage-Appellate.xml](#)

1478 **CoreFilingMessage (Criminal case type)**

1479 [xml/ECF-4.1-CoreFilingMessage-Criminal.xml](#)

1480 **CourtPolicyQueryMessage**

1481 [xml/ECF-4.1-CourtPolicyQueryMessage.xml](#)

1482 **CourtPolicyReponseMessage**

1483 [xml/ECF-4.1-CourtPolicyResponseMessage.xml](#)

1484 **DocumentQueryMessage**

1485 [xml/ECF-4.1-DocumentQueryMessage.xml](#)

1486 **DocumentResponseMessage**

1487 [xml/ECF-4.1-DocumentResponseMessage.xml](#)

1488 **FilingListQueryMessage**

1489 [xml/ECF-4.1-FilingListQueryMessage.xml](#)

1490 **FilingListResponseMessage**

1491 [xml/ECF-4.1-FilingListResponseMessage.xml](#)

1492 **FilingPaymentMessage**

1493 [xml/ECF-4.1-PaymentMessage.xml](#)

1494 **FilingStatusQueryMessage**

1495 [xml/ECF-4.1-FilingStatusQueryMessage.xml](#)

1496 **FilingStatusResponseMessage**

1497 [xml/ECF-4.1-FilingStatusResponseMessage.xml](#)

1498 **MessageReceiptMessage**

1499 [xml/ECF-4.1-MessageReceiptMessage.xml](#)

1500 **PaymentReceiptMessage**

1501 [xml/ECF-4.1-PaymentReceiptMessage.xml](#)

- 1502 **RecordDocketingCallbackMessage**
- 1503 [xml/ECF-4.1-RecordDocketingCallbackMessage.xml](#)
- 1504 **RecordDocketingMessage**
- 1505 [xml/ECF-4.1-RecordDocketingMessage.xml](#)
- 1506 **ReviewFilingCallbackMessage**
- 1507 [xml/ECF-4.1-ReviewFilingCallbackMessage.xml](#)
- 1508 **ServiceInformationQueryMessage**
- 1509 [xml/ECF-4.1-ServiceInformationQueryMessage.xml](#)
- 1510 **ServiceInformationResponseMessage**
- 1511 [xml/ECF-4.1-ServiceInformationResponseMessage.xml](#)
- 1512 **ServiceReceiptMessage**
- 1513 [xml/ECF-4.1-ServiceReceiptMessage.xml](#)

1514

Appendix D. (Informative) Ongoing Work Items

1515 The Electronic Court Filing TC plans to continue to revise and expand this specification through future
1516 versions. Future versions of ECF will:

- 1517 • Address filings in administrative tribunals
- 1518 • Address primary service (the delivery of documents such as summonses, subpoenas and warrants
1519 that establish a court's jurisdiction over a party)
- 1520 • Consider how the specifications for filing of documents intended for filing with a court relate to
1521 specifications for filing other documents, e.g., property records, in the offices of elected clerks of
1522 courts
- 1523 • Incorporate feedback from ECF implementations
- 1524 • Support future releases of the **[NIEM]**
- 1525 • Support future **[Court Document]** specifications and integration optimizations
- 1526 • Support non-case related filings into a court clerk's office

1527

Appendix E. (Informative) Acknowledgments

1528
1529

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

1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544

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

1545

1546
1547

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

1548

Participants:

1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559

Philip Baughman, Tyler Technologies, Inc.
James Cabral, InfoTrack US
Eric Eastman, InfoTrack US
Ryan Foley, i3-ImageSoft, LLC
Gary Graham, Arizona Supreme Court
Barbara Holmes, National Center for State Courts
George Knecht, InfoTrack US
James McMillan, National Center for State Courts
Enrique Othon, Tyler Technologies, Inc.
Jim Price, Arizona Supreme Court
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 <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> in <ecf:CaseFilingType> to enable MDEs to send messages without requiring an asynchronous message. Added <DevelopmentPolicyParametersType>/<RequireAsynchronousResponsesIndicator> to CourtPolicyResponseMessage to indicate whether all MDEs MUST support asynchronous responses to messages they send. Relaxed the cardinality of <nc:ItemType>/<nc:ItemOtherIdentification>, <nc:ObligationType>/ <nc:ObligationEntity> and <nc:OrganizationType>/<nc:OrganizationIdentification> to allow multiples. Added <PersonCitizenshipISO3166Alpha2Code> as an alternative to <PersonCitizenshipFIPS10-4Code> and <LocationCountryISO3166Alpha2Code> as an alternative to <LocationCountryFIPS10-4Code> due to the retirement of the FIPS10-4 code list. Added xsd/wrappers.xsd 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 xsd/wrappers.xsd, fixed consistency of message names and changed docket:RecordDocketingMessage to docketcb:RecordDocketingCallbackMessage in NotifyDocketingCompleteRequestType.
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.

Appendix HCS01	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.

			<p>Clarified in Section 2.4.2 that some form of machine-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.</p>
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.

Appendix G. Notices

1563

1564 Copyright © OASIS Open 2022³. All Rights Reserved.

1565 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
1566 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website:
1567 [<https://www.oasis-open.org/policies-guidelines/ipr/>].

1568 This document and translations of it may be copied and furnished to others, and derivative works that
1569 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
1570 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
1571 and this section are included on all such copies and derivative works. However, this document itself may
1572 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
1573 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
1574 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must
1575 be followed) or as required to translate it into languages other than English.

1576 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
1577 or assigns.

1578 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1579 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
1580 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
1581 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1582 PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT,
1583 INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS
1584 DOCUMENT OR ANY PART THEREOF.

1585 As stated in the OASIS IPR Policy, the following three paragraphs in brackets apply to OASIS Standards
1586 Final Deliverable documents (Committee Specifications, OASIS Standards, or Approved Errata).

1587 [OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
1588 necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS
1589 TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims
1590 in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this
1591 deliverable.]

1592 [OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of
1593 any patent claims that would necessarily be infringed by implementations of this OASIS Standards Final
1594 Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner
1595 consistent with the IPR Mode of the OASIS Technical Committee that produced this OASIS Standards
1596 Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so.]

1597 [OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
1598 might be claimed to pertain to the implementation or use of the technology described in this OASIS
1599 Standards Final Deliverable or the extent to which any license under such rights might or might not be
1600 available; neither does it represent that it has made any effort to identify any such rights. Information on
1601 OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS
1602 Technical Committee can be found on the OASIS website. Copies of claims of rights made available for
1603 publication and any assurances of licenses to be made available, or the result of an attempt made to
1604 obtain a general license or permission for the use of such proprietary rights by implementers or users of
1605 this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS
1606 makes no representation that any information or list of intellectual property rights will at any time be
1607 complete, or that any claims in such list are, in fact, Essential Claims.]

1608 The name "OASIS" is a trademark of OASIS, the owner and developer of this document, and should be
1609 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
1610 implementation and use of, documents, while reserving the right to enforce its marks against misleading
1611 uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark/> for above guidance.