



Electronic Court Filing Version 4.01 **Plus** **Errata 02**

OASIS Standard **incorporating Draft 01 of Errata 02**

14 April 2015

Specification URIs

This version:

<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/ecf-v4.01-spec-errata02-csd01-complete.doc> (Authoritative)
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/ecf-v4.01-spec-errata02-csd01-complete.html>
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/ecf-v4.01-spec-errata02-csd01-complete.pdf>

Previous version:

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

Latest version:

<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/ecf-v4.01-spec-errata02-complete.doc> (Authoritative)
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/ecf-v4.01-spec-errata02-complete.html>
<http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/ecf-v4.01-spec-errata02-complete.pdf>

Technical Committee:

OASIS LegalXML Electronic Court Filing TC

Chairs:

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

Editors:

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

Additional artifacts:

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

- *Electronic Court Filing Version 4.01 Errata 02*. Edited by James Cabral and Gary Graham. 14 April 2015. OASIS Committee Specification Draft 01. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/ecf-v4.01-spec-errata02-csd01.html>.
- XML schemas: <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/xsd/>.

- XML sample messages: <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/xml/>.
- Model: <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/model/>.
- Generic code lists: <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/gc/>.

Related work:

This specification replaces or supersedes:

- *OASIS LegalXML Electronic Court Filing Version 3.0*. Edited by Roger Winters. 15 November 2005. Committee Specification Draft. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v3.0/>.
- *OASIS Electronic Court Filing Version 4.0*. Edited by Adam Angione and Roger Winters. 21 September 2008. Committee Specification Draft. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.0/ecf-v4.0-spec/>.

This specification is related to:

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

Declared XML namespaces:

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

Abstract:

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

is a maintenance release to address several minor schema and definition issues identified by implementers of the ECF 4.0 specification.

Status:

This document was last revised or approved by the OASIS LegalXML Electronic Court Filing TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=legalxml-courtfilling#technical.

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

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

Citation format:

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

[ECF v4.01]

Electronic Court Filing Version 4.01 Plus Errata 02. Edited by Adam Angione and James Cabral. 14 April 2015. OASIS Standard incorporating Draft 01 of Errata 02. <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/csd01/ecf-v4.01-spec-errata02-csd01-complete.html>. Latest version: <http://docs.oasis-open.org/legalxml-courtfilling/specs/ecf/v4.01/ecf-v4.01-spec/errata02/ecf-v4.01-spec-errata02-complete.html>.

Notices

Copyright © OASIS Open 2015. All Rights Reserved.

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

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

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

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

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

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

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

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

Table of Contents

1	Introduction.....	8
1.1	Scope.....	8
1.2	Relationship to Prior Specifications	9
1.3	ECF Version 4.01	9
1.3.1	National Information Exchange Model (NIEM).....	10
1.3.2	OASIS Universal Business Language.....	10
1.3.3	W3C XML-Signature Syntax and Processing	10
1.3.4	OASIS Reference Model for Service Oriented Architecture	11
1.3.5	OASIS Code List Representation (Genericode).....	11
1.4	Terms and Definitions.....	11
1.5	Symbols and Abbreviations	12
1.6	Normative References	13
1.7	Non-Normative References	14
2	ECF 4.0 Architecture	15
2.1	Core vs. Profiles	15
2.2	Major Design Elements.....	15
2.3	Information Model	16
2.3.1	Messages	16
2.3.2	Attachment	17
2.3.3	Sample Message Streams	17
2.4	Court Policy.....	20
2.4.1	Human-Readable Court Policy.....	20
2.4.2	Machine-Readable Court Policy.....	20
2.4.3	Case-Type and Court Extensions	21
2.4.4	Court-Specific Code Lists.....	21
2.4.5	Court-Specific Constraint Schemas	22
3	ECF 4.0 Process Model	23
3.1	The Filing-Preparation-to-Docketing Process Model.....	23
3.2	Business Rules	25
3.2.1	GetPolicy	25
3.2.2	GetServiceInformation.....	25
3.2.3	GetFeesCalculation	25
3.2.4	ReviewFiling	25
3.2.5	ServeFiling.....	25
3.2.6	RecordFiling	26
3.2.7	NotifyDocketingComplete	26
3.2.8	NotifyFilingReviewComplete	26
3.2.9	GetFilingList.....	26
3.2.10	GetFilingStatus	26
3.2.11	GetCaseList.....	27
3.2.12	GetCase	27
3.2.13	GetDocument	27
3.3	Message Business Rules	27

3.3.1 Identifiers	27
3.3.2 Code Lists.....	28
3.3.3 Message-Specific Business Rules	30
3.4 Filing the Record on Appeal	30
4 ECF 4.0 Schemas	33
4.1 ECF 4.0 Case Type Schemas	33
4.2 ECF 4.0 Common Schemas.....	33
4.3 ECF 4.0 Constraint and Subset Schemas.....	34
4.4 ECF 4.0 Message Schemas.....	34
5 Service Interaction Profiles.....	36
5.1 Service Interaction Profile Requirements	36
5.2 Service Interaction Profile Approval and Revision Processes.....	37
5.3 Supported Service Interaction Profiles	38
6 Document Signature Profiles.....	39
6.1 Document Signature Profile Requirements	39
6.2 Document Signature Profile Approval and Revision Processes.....	39
6.3 Supported Document Signature Profiles	40
7 Conformance	41
Appendix A. (Informative) Release Notes	42
A.1 Availability.....	42
A.2 Package Structure.....	42
A.3 Recursive Structures	42
A.4 Date and Time Formats.....	42
A.5 Known Errata.....	43
Appendix B. (Informative) ECF 4.0 Development Approach and Artifacts	44
B.1 Principles.....	44
B.2 Approach	44
B.3 ECF 4.0 Exchange Content Models	44
B.4 Spreadsheet Models	46
Appendix C. (Informative) MDE Operations.....	47
C.1 Filing Assembly MDE	47
C.1.1 Provided Operations	47
C.1.2 Consumed Operations.....	47
C.2 Filing Review MDE	48
C.2.1 Provided Operations	48
C.2.2 Consumed Operations.....	48
C.3 Court Record MDE	49
C.3.1 Provided Operations	49
C.3.2 Consumed Operations.....	49
C.4 Legal Service MDE	49
C.4.1 Provided Operations	50
C.4.2 Consumed Operations	50
Appendix D. (Informative) Example Instances.....	51
Appendix E. (Informative) Ongoing Work Items.....	53
Appendix F. (Informative) Acknowledgments	54

Appendix G. (Informative) Revision History 55

1 Introduction

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

1.1 Scope

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

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

This specification supports the following automated information exchanges:

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

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

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

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

52

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

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

63

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

67 1.2 Relationship to Prior Specifications

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

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

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

79

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

84 1.3 ECF Version 4.01

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

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

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

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

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

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

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

115 **1.3.2 OASIS Universal Business Language**

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

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

122 <AllowanceCharge>

123 Information about a charge or discount price component.

124 <Address>

125 Information about a structured address.

126 <Payment>

127 Information directly relating to a specific payment.

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

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

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

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

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

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

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

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

145

146 **1.4 Terms and Definitions**

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

150

151 This section defines key terms used in this specification.

152

153 **Attachment**

154 See definition in Section 2.3.2.

155 **Callback message**

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

158 **Document**

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

161 **Document hash**

162 A condensed representation of a document intended to protect document integrity, calculated
163 according to the FIPS 180-24 SHA 256 algorithm.

164 **Docketing**

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

167 **Filer**

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

170 **Filing**

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

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

173 **Hub Service MDE**

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

177 **Major Design Element (MDE)**

178 A logical grouping of operations representing a significant business process supported by ECF
179 4.0. Each MDE operation receives one or more messages, returning a synchronous response
180 message (a reaction to a message received) and, optionally, returning an asynchronous (later)
181 response message to the originating message sender.

182 **Message**

183 See definition in Section 2.3.1.

184 **Message Transmission**

185 The sending of one or more messages and associated attachments to an MDE. Each
186 transmission must invoke or respond to an operation on the receiving MDE, as defined in the
187 ECF 4.0 specification.

188 **Operation (or MDE Operation)**

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

193 **Operation signature**

194 A definition of the input message and synchronous response message associated with an
195 operation. Each message is given a name and a type by the operation. The type is defined by a
196 single one of the message structures defined in the ECF 4.0 specification.

197 **Synchronous response**

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

200 **1.5 Symbols and Abbreviations**

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

202

203 **ECF 4.0**

204 Electronic Court Filing 4.0

205 **IEPD**

206 Information Exchange Package Documentation

207 **MDE**

208 Major Design Element

209 **NIEM**

210 National Information Exchange Model

211 **OASIS**

212 Organization for the Advancement of Structured Information Standards

213 **XML**

214 eXtensible Markup Language

- 215 **W3C**
216 World Wide Web Consortium
217 **WS-I**
218 Web Services Interoperability Organization
219

220 **1.6 Normative References**

- 221 **[FIPS 180-2]** ———
222 ~~Secure Hash Standard, [http://csrc.nist.gov/publications/fips/fips180-2/fips180-](http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf)~~
223 ~~[2withchangenotice.pdf](http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf), National Institute for Standards and Technology, August~~
224 ~~2002.~~
- 225 **[FIPS 180-4]**
226 *Secure Hash Standard*, [http://csrc.nist.gov/publications/fips/fips180-4/fips-180-](http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf)
227 [4.pdf](http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf), National Institute for Standards and Technology, March 2012.
- 228
- 229 **[Genericode]**
230 A. B. Coates, *Code List Representation (Genericode) 1.0*, [http://docs.oasis-](http://docs.oasis-open.org/codelist/ns/genericode/1.0/)
231 [open.org/codelist/ns/genericode/1.0/](http://docs.oasis-open.org/codelist/ns/genericode/1.0/), OASIS Committee Specification, December
232 28, 2007
- 233 **[NIEM]**
234 *National Information Exchange Model 2.0*, <http://niem.gov>, US DOJ and DHS,
235 2007.
- 236 **[NIEM Guide]**
237 *NIEM Implementation Guidelines*, <http://www.niem.gov/implementationguide.php>,
238 US DOJ and DHS, 2007.
- 239 **[NIEM Techniques]**
240 *Techniques for Building and Extending NIEM*,
241 <http://www.niem.gov/topicIndex.php?topic=techPDF>, Georgia Tech Research
242 Institute, August 2007.
- 243 **[Namespaces]**
244 T. Bray, *Namespaces in XML*, [http://www.w3.org/TR/1999/REC-xml-names-](http://www.w3.org/TR/1999/REC-xml-names-19990114)
245 [19990114](http://www.w3.org/TR/1999/REC-xml-names-19990114), January 14, 1999.
- 246 **[RFC2046]**
247 N. Freed, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*,
248 <http://www.ietf.org/rfc/rfc2046.txt>, IETF RFC 2046, November 1996.
- 249 **[RFC2119]**
250 S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
251 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 252 **[RFC4122]**
253 Leach, et al., *A Universally Unique Identifier (UUID) URN Namespace*,
254 <http://www.ietf.org/rfc/rfc4122.txt>, IETF RFC 4122, July 2005.
- 255 **[Schema Part 1]**
256 H. S. Thompson, D. Beech. M. Maloney, N. Mendelsohn, *XML Schema Part 1:*
257 *Structures Second Edition*, [http://www.w3.org/TR/2004/REC-xmlschema-1-](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/)
258 [20041028/](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/), W3C Recommendation, October 28, 2004.
- 259 **[Schema Part 2]**
260 P. Biron, A. Malhotra, *XML Schema Part 2: Datatypes Second Edition*,
261 <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>, W3C
262 Recommendation, October 28, 2004

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

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

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

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

280 **1.7 Non-Normative References**

281 **[Court Document]**
 282 *OASIS LegalXML Court Document Subcommittee, Charter, July 2006*,
 283 [http://www.oasis-open.org/committees/download.php/19120/DocumentSC-](http://www.oasis-open.org/committees/download.php/19120/DocumentSC-Charter-Rev6-1.txt)
 284 [Charter-Rev6-1.txt](http://www.oasis-open.org/committees/download.php/19120/DocumentSC-Charter-Rev6-1.txt).

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

289 **[Juvenile XML]**
 290 S. Rondendell, et. al., *Juvenile Justice XML Report*,
 291 [http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml](http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml%5ffinal%5freport%5f20050630.pdf)
 292 [%5ffinal%5freport%5f20050630.pdf](http://www.ijis.org/db/share/public/Library/Publications/juvenile%5fjustice%5fxml%5ffinal%5freport%5f20050630.pdf), IJIS Institute, July 2005.

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

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

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

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

307 **[Traffic IEPD]**
 308 *Traffic Citation IEPD*,
 309 http://www.ncsconline.org/d_tech/gjxdm/iepd/citation/Traffic_Citation_08_08_2005.zip,
 310 National Center for State Courts, August 8, 2005.

311 2 ECF 4.0 Architecture

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

320 2.1 Core vs. Profiles

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

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

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

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

333 2.2 Major Design Elements

334 ECF 4.0 defines four MDEs. They are:

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

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

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

356 court, vendor or application, the application MUST maintain the ECF 4.0 specified operations between
357 each MDE so that other applications will be able to interoperate with it.

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

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

367 2.3 Information Model

368 The ECF information model describes the messages that may be exchanged between MDEs. All ECF
369 4.0 operations use the same core message stream structure, which is implemented in the service
370 interaction profiles. Each ECF core message stream is a stream of bytes that contains at least one
371 message and may also contain attachments.

372 2.3.1 Messages

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

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

```
397     <FilingLeadDocument> (or <FilingConnectedDocument>)  
398         <ecf:DocumentRendition>  
399             <DocumentRenditionMetadata>  
400                 <DocumentAttachment>  
401                     <BinaryBase64Object>2345klj345h...<BinaryBase64Objec  
402                     t>  
403                 </DocumentAttachment>
```


404 </DocumentRenditionMetadata>
405 </ecf:DocumentRendition
406 </FilingLeadDocument> (or </FilingConnectedDocument>)

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

418
419 The scope of the ECF core messages and extensions is limited by the following criteria:

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

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

427 **2.3.2 Attachment**

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

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

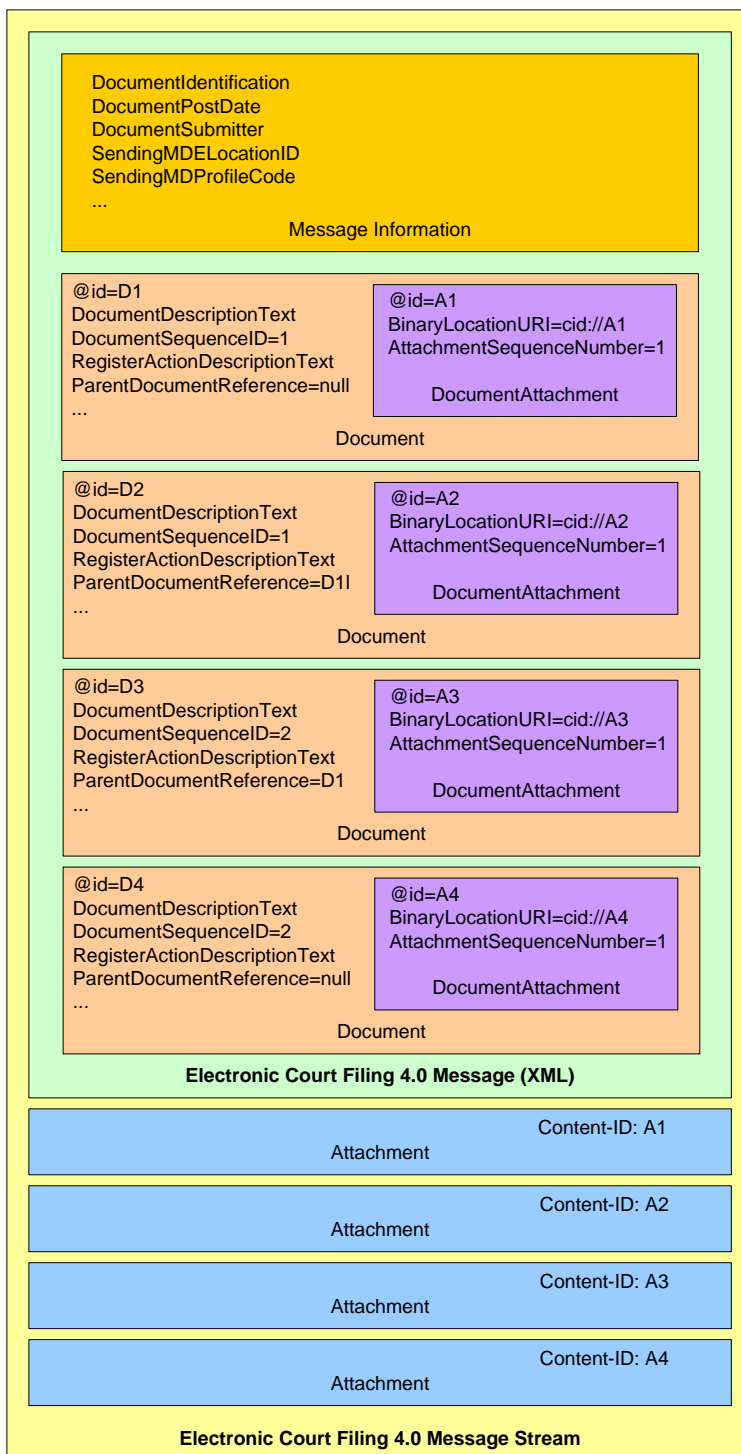
438 **2.3.3 Sample Message Streams**

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

440
441

442
 443
 444
 445
 446

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

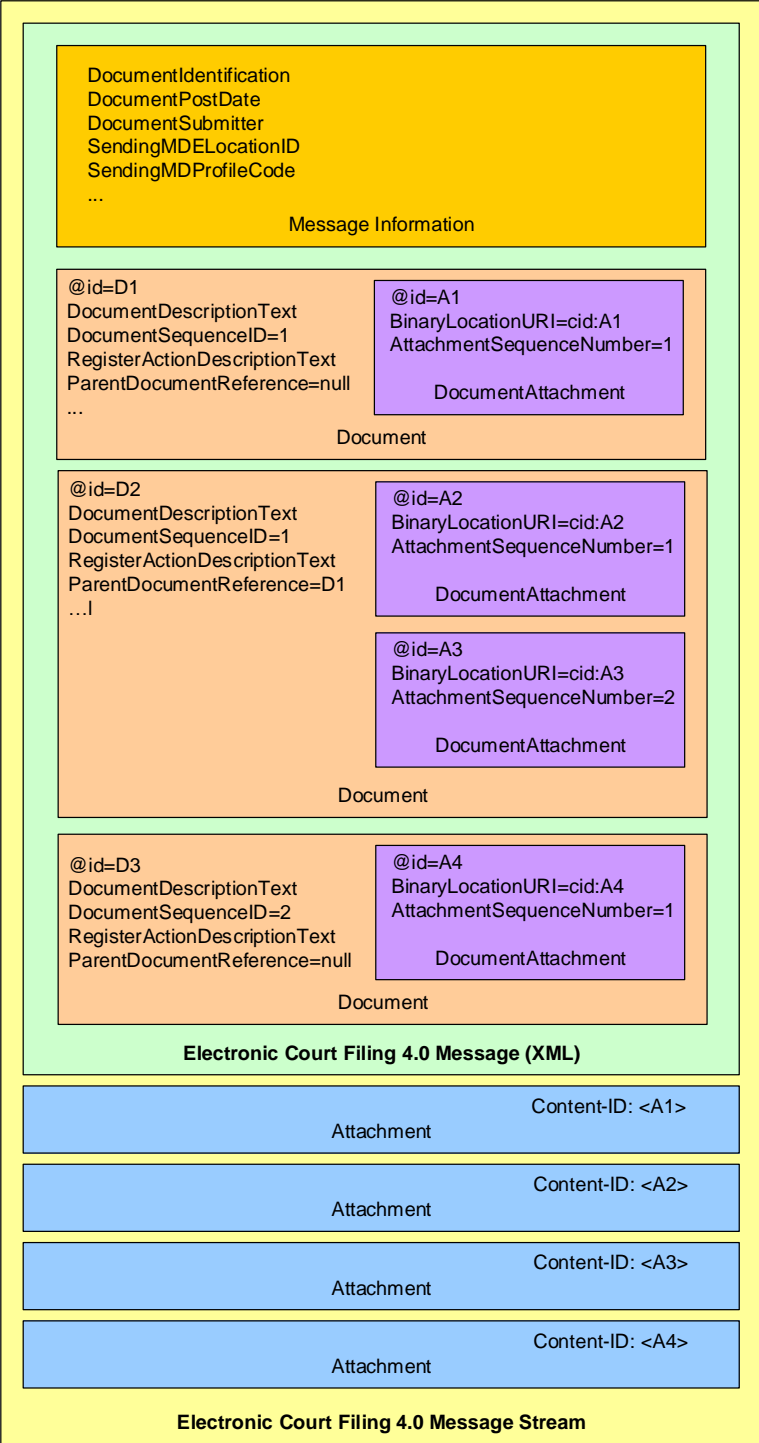


447
 448
 449

Figure 1. Simple Message Stream

450
451
452
453
454
455

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



456
457
458

Figure 2. Message Stream with a Document in Multiple Attachments

459 2.4 Court Policy

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

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

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

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

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

486 2.4.1 Human-Readable Court Policy

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

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

500 2.4.2 Machine-Readable Court Policy

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

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

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

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

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

519 **2.4.3 Case-Type and Court Extensions**

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

530

531 **2.4.4 Court-Specific Code Lists**

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

535

- 536 • ECF Code Lists
 - 537 • **Civil Case Type**
 - 538 • `<FiduciaryTypeCode>*`
 - 539 • `<JurisdictionalGroundsCode>`
 - 540 • `<ReliefTypeCode>`
 - 541 • **Domestic Case Type**
 - 542 • `<NoContactCode>*`
 - 543 • `<RequestToVacateCode>`
 - 544 • **Common Types**
 - 545 • `<AliasAlternateNameTypeCode>*`
 - 546 • `<CaseAssociationTypeCode>*`
 - 547 • `<CaseOfficialRoleText>*`
 - 548 • `<CaseParticipantRoleCode>*`

- 549 • <CauseOfActionCode>
- 550 • <CourtEventTypeCode>
- 551 • <EntityAssociationTypeCode>
- 552 • <ErrorCode>*
- 553 • **Juvenile Case Type**
- 554 • <DelinquentActApplicabilityCode>
- 555 • <DelinquentActDegreeCode>
- 556 • <DelinquentActSeverityCode>
- 557 • <DelinquentActSpecialAllegationCode>
- 558 • <DependencyAllegationCode>
- 559 • <GuardianAssociationTypeCode>*
- 560 • <PlacementTypeCode>
- 561 • **NIEM Code Lists**
- 562 • **JXDM**
- 563 • <ChargeEnhancingFactorText>
- 564 • <CourtLocationCode>
- 565 • <RegisterActionDescriptionText>
- 566 • <StatuteCodeIdentification>
- 567 • <StatuteCodeSectionIdentification>
- 568 • <StatuteOffenseIdentification>
- 569 • <StatusOffenseCodeIdentification>
- 570 • **NIEM Core**
- 571 • <BinaryDescriptionText>*
- 572 • <CaseCategoryText>
- 573 • <DriverLicenseCommercialClassCode>
- 574 • <FamilyKinshipCode>*

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

578
579 If a court does not define allowable values for any of the above code lists in court policy, then any value
580 MUST be considered acceptable for that code.

582 **2.4.5 Court-Specific Constraint Schemas**

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

589 3 ECF 4.0 Process Model

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

593 3.1 The Filing-Preparation-to-Docketing Process Model

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

- 601 • GetPolicy
- 602 • GetServiceInformation
- 603 • GetFeesCalculation
- 604 • **ReviewFiling**
- 605 • ServeFiling
- 606 • **RecordFiling**
- 607 • **NotifyDocketingComplete**
- 608 • **NotifyFilingReviewComplete**

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

- 611 • GetFilingList
- 612 • GetFilingStatus

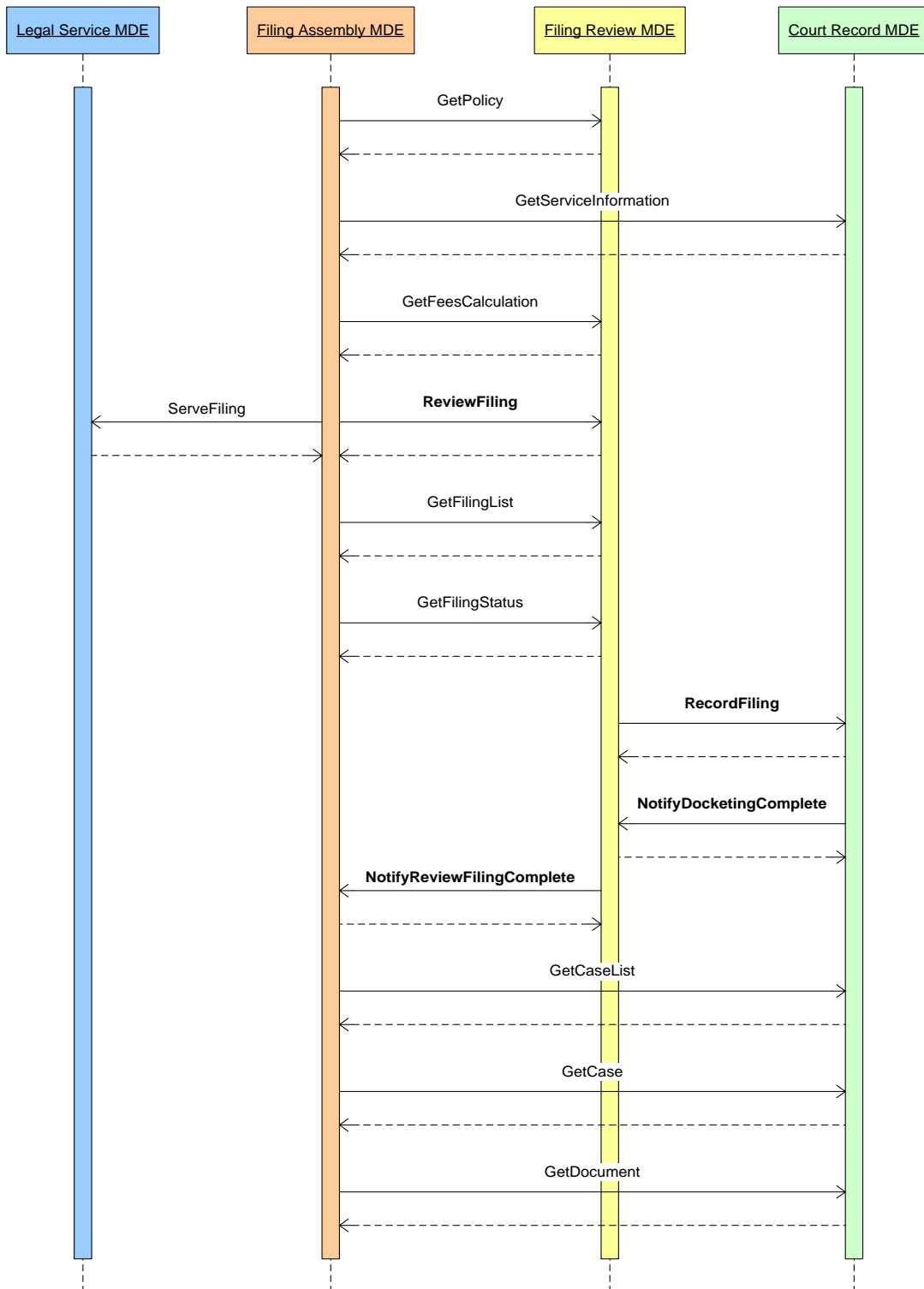
613 At any point after the NotifyFilingReviewComplete operation, if the case is accessible, a party MAY
614 access information through the following operations:

- 615 • GetCaseList
- 616 • GetCase
- 617 • GetDocument

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

620

Figure 4. Filing Preparation to Docketing Process Model



624 **3.2 Business Rules**

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

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

631 **3.2.1 GetPolicy**

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

637 **3.2.2 GetServiceInformation**

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

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

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

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

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

655 **3.2.3 GetFeesCalculation**

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

661 **3.2.4 ReviewFiling**

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

666 **3.2.5 ServeFiling**

667 At approximately the same time the Filing Assembly MDE submits the filing to the court, the Filing
668 Assembly MDE MAY serve the entire filing, to other parties in the case by invoking the ServeFiling

669 operation on the ServiceMDE associated with the service recipient. This operation MUST NOT be used
670 to serve parties in a new case or to persons or organizations that have not yet been made party to the
671 case. The Legal Service MDE responds synchronously with an acknowledgement that the message will
672 be delivered to the service recipient or with an error.

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

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

680 **3.2.6 RecordFiling**

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

685 **3.2.7 NotifyDocketingComplete**

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

693 **3.2.8 NotifyFilingReviewComplete**

694 If the clerk rejects the filings or the Filing Review MDE receives the Notify Docketing Complete message,
695 the Filing Review MDE MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly
696 MDE as a callback message to the ReviewFiling operation to indicate whether the filing was accepted
697 and docketed by the clerk and court record system. The operation MAY return the filed documents or
698 links to the documents, but MUST include the **[FIPS 180-24]** SHA 256 document hash, a condensed
699 representation of a document intended to protect document integrity.

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

703 **3.2.9 GetFilingList**

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

708 **3.2.10 GetFilingStatus**

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

712 **3.2.11 GetCaseList**

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

716 **3.2.12 GetCase**

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

722 **3.2.13 GetDocument**

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

726 **3.3 Message Business Rules**

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

729 **3.3.1 Identifiers**

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

732 **3.3.1.1 Attachment Identifiers**

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

736 **3.3.1.2 Case Identifiers**

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

739 **3.3.1.3 Court Identifiers**

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

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

744 Examples of conformant court identifiers include:

- 745 • courts.wa.gov:superior.king
- 746 • nmcourts.com:albd.civil
- 747 • uscourts.gov:100
- 748 • courts.gov.bc.ca:appeal

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

750 3.3.1.4 Document Identifiers

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

752 3.3.1.5 Filing Identifiers

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

755 3.3.1.6 MDE Identifiers

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

758 3.3.1.7 Filer and Party Identifiers

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

```
762  
763 <nc:PersonOtherIdentification>  
764     <nc:IdentificationID>100<nc:IdentificationID>  
765     <nc:IdentificationCategoryText>ECFFilerID</nc:IdentificationCategoryTex  
766 t>  
767 </nc:PersonOtherIdentification>
```

768
769 In addition to <nc:PersonOtherIdentification>, other elements that may contain a filer identifier
770 include <nc:OrganizationOtherIdentification>, <ecf:FilingPartyID> and
771 <ecf:FilingAttorneyID>.

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

776 3.3.2 Code Lists

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

- 780
781 • ECF Code Lists
 - 782 • [Bankruptcy Case Type](#)
 - 783 • <DebtorTypeCode>*
 - 784 • <EstimatedAssetsValueLevelCode>*
 - 785 • <EstimatedDebtsValueLevelCode>*
 - 786 • <NatureOfDebtCode>*
 - 787 • <NumberOfCreditorsValueLevelCode>*
 - 788 • [Common Types](#)
 - 789 • <FilingStatusCode>*
 - 790 • [Court Policy Response Message](#)

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

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

829

830 3.3.3 Message-Specific Business Rules

831 The following business rules apply to specific messages:

832 3.3.3.1 CoreFilingMessage

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

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

844 3.3.3.2 FilingPaymentMessage

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

850 3.3.3.3 RecordDocketingMessage

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

853 3.4 Filing the Record on Appeal

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

- 856
- 857 • All ROA transactions, either the original filing or subsequent amendments, MUST contain, as the
858 lead document, an Index of Record document that itemizes the content of the record on appeal.³
859
 - 860 • The documents that comprise the ROA transaction will be identified as supporting documents.
861
 - 862 • The supporting documents that comprise the ROA transaction MAY also have additional attached
863 documents.
864
 - 865 • All ROA documents being submitted, including the Index of Record document and each
866 document within the record, MUST have at least one court-defined document type that indicates
867 the type of transaction to be performed on the document, and whether the document is being
868 added to or stricken from the record.
869

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

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

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

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

901
902
903
904
905

- Individual documents within the ROA transaction **MUST** not be individually accepted or rejected. All documents within the ROA transaction **MUST** have the same acceptance or rejection disposition.

906 4 ECF 4.0 Schemas

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

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

913

914 4.1 ECF 4.0 Case Type Schemas

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

917

918 **AppellateCase**

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

920 **BankruptcyCase**

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

922 **CitationCase**

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

924 **CivilCase**

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

926 **CriminalCase**

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

928 **DomesticCase**

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

930 **JuvenileCase**

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

932

933 4.2 ECF 4.0 Common Schemas

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

936

937 **AppInfo**

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

939 **CommonTypes**

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

941 **DigitalSignature**

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

943 **Genericcode**

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

945 **4.3 ECF 4.0 Constraint and Subset Schemas**

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

954 **4.4 ECF 4.0 Message Schemas**

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

957

958 **CaseListQueryMessage**

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

960 **CaseListResponseMessage**

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

962 **CaseQueryMessage**

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

964 **CaseResponseMessage**

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

966 **CoreFilingMessage**

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

968 **CourtPolicyQueryMessage**

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

970 **CourtPolicyResponseMessage**

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

972 **DocumentQueryMessage**

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

974 **DocumentResponseMessage**

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

976 **FeesCalculationQueryMessage**

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

978 **FeesCalculationResponseMessage**

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

980 **FilingListQueryMessage**

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

982 **FilingListResponseMessage**

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

984 **FilingStatusQueryMessage**

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

986 **FilingStatusResponseMessage**

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

988 **MessageReceiptMessage**
989 [xsd/message/ECF-4.0-MessageReceiptMessage.xsd](#)
990 **PaymentMessage**
991 [xsd/message/ECF-4.0-PaymentMessage.xsd](#)
992 **PaymentReceiptMessage**
993 [xsd/message/ECF-4.0-PaymentReceiptMessage.xsd](#)
994 **RecordDocketingCallbackMessage**
995 [xsd/message/ECF-4.0-RecordDocketingCallbackMessage.xsd](#)
996 **RecordDocketingMessage**
997 [xsd/message/ECF-4.0-RecordDocketingMessage.xsd](#)
998 **ReviewFilingCallbackMessage**
999 [xsd/message/ECF-4.0-ReviewFilingCallbackMessage.xsd](#)
1000 **ServiceInformationQueryMessage**
1001 [xsd/message/ECF-4.0-ServiceInformationQueryMessage.xsd](#)
1002 **ServiceInformationResponseMessage**
1003 [xsd/message/ECF-4.0-ServiceInformationResponseMessage.xsd](#)
1004 **ServiceReceiptMessage**
1005 [xsd/message/ECF-4.0-ServiceReceiptMessage.xsd](#)
1006

1007 5 Service Interaction Profiles

1008 An ECF 4.0 service interaction profile defines a transmission system that supports the functional
1009 requirements of electronic filing, along with the MDE operations and message structures, and implements
1010 certain non-functional requirements. A service interaction profile does not govern the content of
1011 messages – message content is described in Sections 2 and 3 of this specification. A service interaction
1012 profile will define how a message gets from the sending MDE to the receiving MDE in a given messaging
1013 framework.

1014 5.1 Service Interaction Profile Requirements

1015 Each service interaction profile will define standard conventions and configuration details to support
1016 interoperability between and among ECF 4.0 implementations that support the same service interaction
1017 profile. However, compliance with these requirements will not necessarily guarantee interoperability.

1018 To be compliant with the ECF 4.0 specification, a service interaction profile **MUST** satisfy the following
1019 non-functional requirements:

- 1020 1. **Transport protocol** – A service interaction profile **MUST** define how messages are physically
1021 transported from a sending MDE to a receiving MDE. In so doing, a profile may identify factors that
1022 restrict the range of environments in which the profile is applicable.
- 1023 2. **MDE addressing** – A service interaction profile **MUST** include a convention for uniquely addressing
1024 each MDE.
- 1025 3. **Operation addressing** – A service interaction profile **MUST** describe a convention for uniquely
1026 addressing each MDE operation.
- 1027 4. **Request and operation invocation** – A service interaction profile **MUST** describe a mechanism for a
1028 sending MDE to invoke an operation on the receiving MDE.
- 1029 5. **Synchronous mode response** – A service interaction profile **MUST** support synchronous operations
1030 in which the response to an operation is always returned immediately, typically within a matter of
1031 seconds, to the invoking MDE.
- 1032 6. **Asynchronous mode response** – A service interaction profile **MUST** support asynchronous
1033 operations in which the response to an operation may not necessarily be returned immediately to the
1034 invoking MDE. Instead, the response may be returned at some later time through a callback from the
1035 MDE that received the operations to the invoking MDE. The callback **MUST** include a reference to
1036 the invoking message transmission.
- 1037 7. **Message/attachment delimiters** – A service interaction profile **MUST** define how the receiving MDE
1038 distinguishes messages from attachments within a message transmission.
- 1039 8. **Message identifiers** – A service interaction profile **MUST** provide a means for a sending MDE to
1040 assign a unique identifier to each message (including any attachments) within a message
1041 transmission.

1042 In addition, there are some non-functional features that a service interaction profile **SHOULD** provide,
1043 including:

- 1044 1. **Message non-repudiation** – A service interaction profile **SHOULD** provide a mechanism so that the
1045 receiving MDE is provided with evidence that demonstrates:
 - 1046 a. the identity of the sending MDE
 - 1047 b. the content of the message(s) transmitted
 - 1048 c. the date and time of the message transmission
- 1049 2. **Message integrity** – A service interaction profile **SHOULD** provide a mechanism so that the
1050 receiving MDE is able to determine whether the message(s) transmitted (including any attachments)
1051 was (were) modified during the message transmission.

- 1052 3. **Message confidentiality** – A service interaction profile SHOULD provide a mechanism, such as
 1053 encryption, that can be used with a sending MDE to ensure that the message(s) in a transmission
 1054 (including any attachments) can be processed only by the receiving MDE.
- 1055 4. **Message authentication** – A service interaction profile SHOULD provide a mechanism, such that a
 1056 sending MDE is required to include, to display credentials that demonstrate its identity to the receiving
 1057 MDE in each message transmission.
- 1058 5. **Message transmission reliability** – A service interaction profile SHOULD provide a mechanism,
 1059 such that a sending MDE is required to include, to guarantee that a message transmission will be
 1060 delivered to the receiving MDE within a specified period of time, or else the sending MDE will receive
 1061 notification at the end of that period of time that the message transmission was not deliverable to the
 1062 receiving MDE.
- 1063 6. **Message splitting and assembly** – A service interaction profile SHOULD provide a mechanism by
 1064 which a large message and attachments MAY be split into multiple pieces that are transmitted
 1065 separately by the sending MDE and reassembled into the complete message by the receiving MDE.
 1066 In the HTTP 1.1 protocol, this is called “chunking.”
- 1067 7. **Transmission auditing** – A service interaction profile SHOULD provide a mechanism for the MDE to
 1068 receive message transmissions in their entirety (both messaging and “payload” content) for auditing
 1069 purposes.

1070 **5.2 Service Interaction Profile Approval and Revision Processes**

1071 The ECF Technical Committee (TC) will recommend certain service interaction profiles for use in
 1072 implementations of the ECF 4.0 specification. The TC will consider a service interaction profile for
 1073 recommendation for use in ECF 4.0 implementations provided the profile meets the following
 1074 requirements:

- 1075 1. The service interaction profile MUST be described in a document in the format of an OASIS
 1076 specification.
- 1077 2. The service interaction profile specification MUST identify a unique URI to identify the service
 1078 interaction profile and version.
- 1079 3. The service interaction profile specification MUST describe the binding of MDE operations to the
 1080 service interaction profile that satisfies the functional requirements described in Section 3 (“ECF 4.0
 1081 Process Model”) and Section 4 (“ECF 4.0 Schema”) of this specification.
- 1082 4. The service interaction profile specification MUST demonstrate that the service interaction profile
 1083 satisfies the non-functional service interaction profile requirements described in Section 5.1 (“Service
 1084 Interaction Profile Requirements”) of this specification.
- 1085 5. The service interaction profile specification MUST include samples that demonstrate how the
 1086 messaging information and “payload” content are combined into message transmissions. These
 1087 samples MUST include samples that demonstrate both synchronous and asynchronous mode
 1088 operations.
- 1089 6. At least one voting member of the ECF TC MUST agree to sponsor the service interaction profile and
 1090 submit the service interaction profile specification to the TC for review as a candidate for approval as
 1091 an ECF 4.0 compliant service interaction profile.

1092 Certifying that a candidate service interaction profile meets certain service interaction profile requirements
 1093 will necessarily involve some subjectivity since service interaction profile requirements cannot be
 1094 expressed algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to assess
 1095 whether the proposed profile’s description is adequate in meeting the requirements of ECF 4.0 before
 1096 approving the service interaction profile specification as a “Committee Draft” through the OASIS
 1097 standards approval process.

1098 From time to time, it may be necessary to revise or update a service interaction profile to bring it into
 1099 compliance with changes in network and messaging protocols, or to support additional non-functional
 1100 requirements. Any revision(s) to previously approved service interaction profiles will be considered a new
 1101 service interaction profile and MUST meet the requirements of a new service interaction profile, including
 1102 sponsorship by a voting member of the ECF TC and review and approval by the ECF TC. There will be

1103 no guarantees that future versions of a service interaction profile will be backwardly compatible with the
1104 current version.

1105 **5.3 Supported Service Interaction Profiles**

1106 The following ECF 4.0 service interaction profile specifications are for use in conjunction with
1107 implementations of the ECF 4.0 specification:

- 1108 • **Web Services Service Interaction Profile 2.0 Specification** – This specification defines a
1109 transmission system using the specifications described in the Web Services Interoperability (WS-I)
1110 Basic Profile 1.1, W3C SOAP 1.1 Binding for MTOM 1.0, WS-I Basic Security Profile 1.0 and OASIS
1111 WS-Reliable Messaging 1.1.
- 1112 • **Web Services Service Interaction Profile 2.1 Specification** – This specification defines a
1113 transmission system using the specifications described in the Web Services Interoperability (WS-I)
1114 Basic Profile 1.1, W3C SOAP 1.1 Binding for MTOM 1.0 and WS-I Basic Security Profile 1.1 and
1115 OASIS WS-Reliable Messaging 1.1.
- 1116 • **Portable Media Service Interaction Profile 1.01 Specification** – This specification defines a
1117 transmission system in which the sending MDE stores message transmissions on portable media
1118 (e.g., a compact disc), which is then physically transported to the receiving MDE where it is
1119 connected for retrieval of the message transmissions. This specification may be needed in the
1120 absence of an active network between the sending and receiving MDEs.

1121 Additional service interaction profiles, or revisions to these service interaction profiles, may be approved
1122 by the ECF TC for use in conjunction with implementations of the ECF 4.0 specification according to the
1123 process described in Section 5.2 (“Service Interaction Profile Approval and Revision Processes”) above.

1124 6 Document Signature Profiles

1125 An ~~ECF 4.0~~ ECF document signature profile defines a mechanism for asserting that a person signed a
1126 single electronic or imaged document, which is an attachment to a message transmission. The signing of
1127 an entire message transmission is described in a service interaction profile and is not supported by a
1128 document signature profile.

1129 6.1 Document Signature Profile Requirements

1130 Each document signature profile will define standard conventions and configuration details to support
1131 interoperability in the creation and verification of document signatures between and among ~~ECF 4.0~~ ECF
1132 implementations that support the same document signature profile. However, compliance with these
1133 requirements will not necessarily guarantee interoperability.

1134 Except for the Null Document Signature Profile, to be compliant with the ECF 4.0 specification, a
1135 document signature profile MUST satisfy the following non-functional requirements:

- 1136 1. **Signer name assertion** – A document signature profile MUST make an assertion regarding the
1137 name of the person who signed a document.
- 1138 2. **Signed date assertion** – A document signature profile MUST make an assertion regarding the date
1139 the person signed a document.
- 1140 3. **Multiple signatures** – A document signature profile MUST allow multiple signatures to be associated
1141 with the same document.

1142 A signature profile SHOULD provide the following non-functional features:

- 1143 1. **Signer and date non-repudiation** – A document signature profile SHOULD provide a mechanism so
1144 that the receiving MDE is provided with verifiable evidence that demonstrates:
 - 1145 a. the unique identity of the person who signed the document
 - 1146 b. the date the person signed a document
- 1147 2. **Document integrity** – A document signature profile SHOULD provide a mechanism so that the
1148 receiving MDE is able to determine if the document was modified since the person signed the
1149 document.
- 1150 3. **Document signature auditing** – A document signature profile SHOULD provide a mechanism for
1151 the MDE to receive both the document and signatures for auditing purposes.

1152 6.2 Document Signature Profile Approval and Revision Processes

1153 The ECF Technical Committee will recommend certain document signature profiles for use in
1154 implementations of the ECF 4.0 specification. The TC will consider a document signature profile for
1155 recommendation for use in ECF 4.0 implementations provided the profile meets the following
1156 requirements:

- 1157 1. The document signature profile MUST be described in a document in the format of an OASIS
1158 specification.
- 1159 2. The document signature profile specification MUST identify a unique URI to identify the document
1160 signature profile and version.
- 1161 3. If the document signature is not embedded in the document, the document signature profile
1162 specification MUST include an XML structure for describing precisely how the document signature is
1163 represented.
- 1164 4. The document signature profile specification MUST demonstrate that the document signature profile
1165 satisfies the non-functional requirements described in Section 6.1 (“Document Signature Profile
1166 Requirements”) of this specification.

- 1167 5. The document signature profile specification MUST include samples that demonstrate how the
1168 document signature information and “payload” content are combined into message transmissions.
- 1169 6. At least one voting member of the ECF TC MUST agree to sponsor the document signature profile
1170 and submit the document signature profile specification to the TC for review as a candidate for
1171 approval as an ECF 4.0 ECF document signature profile.

1172 Certifying that a candidate document signature profile meets certain document signature profile
1173 requirements will necessarily involve some subjectivity, since document signature profile requirements
1174 cannot be expressed algebraically, in the manner of XML Schemas. Therefore, it will be up to the TC to
1175 assess whether the proposed profile’s description is adequate to the requirements before approving the
1176 profile specification as a Committee Draft through the OASIS standards approval process.

1177 From time to time, it may be necessary to revise or update a document signature profile to bring it into
1178 compliance with changes in authentication and encryption protocols, or to support additional non-
1179 functional requirements. Any revision(s) to previously approved document signature profiles will be
1180 considered a new document signature profile and MUST meet the requirements of a new document
1181 signature profile, including sponsorship by a voting member of the ECF TC and review and approval by
1182 the ECF TC. There will be no guarantees that future versions of document signature profiles will be
1183 backwardly compatible with the current version.

1184 6.3 Supported Document Signature Profiles

1185 The following ECF 4.0 ECF document signature profile specifications are candidate Committee Drafts for
1186 use in conjunction with implementations of the ECF 4.0 specification:

- 1187 • **Null Document Signature Profile 1.0 Specification** – This specification defines a default
1188 mechanism to describe documents that do not have any associated signatures.
- 1189 • **XML Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1190 associating a W3C XML Signature with a document.
- 1191 • **Application-Specific Document Signature Profile 1.0 Specification** – This specification defines a
1192 mechanism for embedding an application-specific binary signature with a document. This profile
1193 supports the native capabilities in document formats such as Microsoft Word and the Adobe Portable
1194 Document Format (PDF) for describing and embedding signatures.
- 1195 • **Proxy Document Signature Profile 1.0 Specification** – This specification defines a mechanism for
1196 indicating documents that are digitally signed by a court filing infrastructure component on behalf of
1197 an authenticated signer.
- 1198 • **Symmetric Key Document Signature Profile 1.0 Specification** – This specification defines a
1199 mechanism for indicating documents that are digitally signed by a trusted entity on behalf of the
1200 signer using a symmetric key known only to the trusted entity.

1201 Additional document signature profiles, or revisions to these document signatures profiles, may be
1202 approved by the ECF TC for use in conjunction with implementation of the ECF 4.0 specification
1203 according to the process described in Section 6.2 (“Document Signature Profile Approval and Revision
1204 Processes”) above.

1205 **7 Conformance**

1206 *An implementation conforms with the Electronic Court Filing Version 4.01 if the implementation meets the*
1207 *requirements in Sections 1-6 including conformance with the XSD schemas and [Genericcode] code lists*
1208 *referenced in Section 3 and 4.*

1209 Appendix A. (Informative) Release Notes

1210 A.1 Availability

1211 Online and downloadable versions of this release are available from the locations specified at the top of
1212 this document.

1213 A.2 Package Structure

1214 The ECF 4.0 specification is published as a ZIP archive named `ecf-v4.0.zip`. Unzipping this archive
1215 creates a directory named `ecf-4.0/` containing this specification document and a number of
1216 subdirectories. The files in these subdirectories, linked to the specification document, contain the various
1217 normative and informational pieces of the 1.0 release. A description of each subdirectory is given below.

1218 `gc/`

1219 **[Genericcode]** 1.0 code lists

1220 `model/`

1221 ECF 4.0 UML exchange content model diagrams and spreadsheet models; see Appendix B.3 and

1222 B4

1223 `xml/`

1224 Example instances; see Appendix D

1225 `xsd/`

1226 XSD schemas; see Section 4

1227 A.3 Recursive Structures

1228 Certain components in the **[NIEM]** version 2.0 schemas allow recursive nesting. For example, a
1229 `nc:Case` may be related to another `nc:Case`, etc. These are legitimate business data structures. Most
1230 real-world applications will limit the depth of recursion in such structures, but XSD schemas are incapable
1231 of expressing this constraint. Implementers should be aware of this and may wish to set limits on the
1232 depth of recursive structures in their applications.

1233 A.4 Date and Time Formats

1234 The date and time elements contained in the messages defined by the ECF 4.0 XSD schemas should be
1235 formatted according to the documentation in the **[NIEM]** version 2.0. The **[NIEM]** documentation
1236 indicates the following:

- 1237 • Calendar date values should be expressed as “CCYY-MM-DD”, with an optional time zone qualifier
1238 designated by appending `-hh:00`, where `hh` represent the number of hours the local time zone is
1239 behind Coordinated Universal Time (UTC).
- 1240 • Time values should be expressed as “hh:mm:ss.sss”, with an optional time zone qualifier designated
1241 by appending `-hh:00`, where `hh` represent the number of hours the local time zone is behind
1242 Coordinated Universal Time (UTC).
- 1243 • Date and time values should be expressed as “CCYY-MM-DDThh:mm:ss.sss” with an optional time
1244 zone designated by appending `-hh:00`, where `hh` represent the number of hours the local time zone is
1245 behind Coordinated Universal Time (UTC).qualifier.

1246 These formats are documented in, but not enforced by, the XSD schema at
1247 `xsd/constraint/niem/proxy/xsd/2.0/xsd.xsd`.

1248 **A.5 Known Errata**

1249 Known errors in the ECF 4.0 specification will be identified in an errata document available at:
1250 <http://www.oasis-open.org/committees/legalxml-courtfilling/>.

1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289

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

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

B.1 Principles

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

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

B.2 Approach

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

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

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

B.3 ECF 4.0 Exchange Content Models

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

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

The exchange content models are used for the following purposes:

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

- 1313 <model/uml/html/DomesticFiling.png>
- 1314 **Extended Person Information Model**
- 1315 <model/uml/html/ExtendedPersonInformation.png>
- 1316 **Get Calculated Fees Query Model**
- 1317 <model/uml/html/GetFeesCalculationQuery.png>
- 1318 **Get Case List Query Model**
- 1319 <model/uml/html/GetCaseListQuery.png>
- 1320 **Get Document Query Model**
- 1321 <model/uml/html/GetDocumentQuery.png>
- 1322 **Get Filing List Query Model**
- 1323 <model/uml/html/GetFilingListQuery.png>
- 1324 **Get Filing Status Query Model**
- 1325 <model/uml/html/GetFilingStatusQuery.png>
- 1326 **Get Service Information Query Model**
- 1327 <model/uml/html/GetServiceInformationQuery.png>
- 1328 **Major Design Elements Model**
- 1329 <model/uml/html/MajorDesignElements.png>
- 1330 **Juvenile Filing Model**
- 1331 <model/uml/html/JuvenileFiling.png>
- 1332 **Record Docketing Model**
- 1333 <model/uml/html/RecordDocketing.png>
- 1334 **Review Filing Model**
- 1335 <model/uml/html/ReviewFiling.png>

1336
1337 No specific directions are defined for the associations in these models; they can be navigated in either
1338 direction. The specific navigation path for each association is defined when documents are assembled.

1339 **B.4 Spreadsheet Models**

1340 ECF 4.0 uses spreadsheet models to describe the mapping of objects and attributes to **[NIEM]** and ECF
1341 4.0 elements. The spreadsheet models use rows to define components. Components are either simple
1342 data types or associations. Columns define the metadata associated with each component type.

1343 The ECF 4.0 spreadsheet model is located at <model\ECF-4.0-NIEM2-mapping.xls>.

1344

1345 **Appendix C. (Informative) MDE Operations**

1346 This appendix details the operations that are provided by each Major Design Element (MDE) and the
 1347 operations, provided by other MDEs that each MDE “consumes.” Implementation of an MDE requires
 1348 both that the MDE provide certain functionality and that the MDE use particular operations provided by
 1349 other MDEs.

1350 **C.1 Filing Assembly MDE**

1351 The Filing Assembly MDE supports the preparation and submission of filed documents to a court for
 1352 review, and can receive the results of that process. The Filing Assembly MDE also conveys filings to the
 1353 Legal Service MDE for service on other case participants. The Filing Assembly MDE calls operations in
 1354 other MDEs and provides a single operation for notifying the submitter that the filing has been reviewed
 1355 by a court. A Filing Assembly MDE may be provided by a court or by a third party.

1356 **C.1.1 Provided Operations**

1357 The Filing Assembly MDE provides the following operations to other MDEs:

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

1358 **C.1.2 Consumed Operations**

1359 The Filing Assembly MDE calls the following operations in other MDEs:

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

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

1360 C.2 Filing Review MDE

1361 The Filing Review MDE receives, presents and manages the filings. The Filing Review MDE receives
 1362 filings in a standard format and presents those filings to a Clerk for review, where they may be accepted
 1363 or rejected. The Filing Review MDE transmits data and documents to the Filing Assembly MDE to inform
 1364 the filer that the filing has been accepted or rejected. The Filing Review MDE transmits data and
 1365 documents for accepted filings to the Court Record MDE for docketing and recording. While there will
 1366 generally be one Filing Review MDE per court, there is no physical barrier to having more than one,
 1367 particularly if a court wants to support different Filing Review MDEs for particular case types.

1368 C.2.1 Provided Operations

1369 The Filing Review MDE provides the following operations to other MDEs:

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

1370 C.2.2 Consumed Operations

1371 The Filing Review MDE calls the following operations in other MDEs:

Operation	Provided By	Output
-----------	-------------	--------

RecordFiling	Court Record MDE	xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage
NotifyFilingReviewComplete	Filing Assembly MDE	xsd/message/ECF-4.0-MessageReceiptMessage.xsd : MessageReceiptMessage

1372 C.3 Court Record MDE

1373 The Court Record MDE receives the filed documents from the Filing Review MDE and enters them into
 1374 the official case record of the court. The Court Record MDE notifies the Filing Review MDE that the filing
 1375 has been filed.

1376 C.3.1 Provided Operations

1377 The Court Record MDE provides the following operations to other MDEs:

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

1378 C.3.2 Consumed Operations

1379 The Court Record MDE calls the following operations in other MDEs:

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

1380 C.4 Legal Service MDE

1381 The Legal Service MDE enables a filer or a court to electronically transmit copies of, or links to,
 1382 electronically filed documents to other parties who are participating in the case and who are entitled to be
 1383 promptly served with the electronically filed documents. The Filing Assembly MDE transmits data and
 1384 documents to the Legal Service MDE to inform the case participant that an electronic filing has been

1385 submitted to the court clerk. The Legal Service MDE transmits a callback message to the Filing
1386 Assembly MDE requesting a notification to confirm receipt of the served document.

1387 **C.4.1 Provided Operations**

1388 The Legal Service MDE provides the following operations to other MDEs:

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

1389 **C.4.2 Consumed Operations**

1390 The Legal Service MDE does not call operations in other MDEs

1391

Appendix D. (Informative) Example Instances

1392 Example instances of each ECF 4.0 message are provided in the xml/ subdirectory, as listed below:

1393

1394 **FeesCalculationQueryMessage**

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

1396 **FeesCalculationResponseMessage**

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

1398 **CaseListQueryMessage**

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

1400 **CaseListResponseMessage**

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

1402 **CaseQueryMessage**

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

1404 **CaseResponseMessage**

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

1406 **CoreFilingMessage (Appellate case type)**

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

1408 **CoreFilingMessage (Criminal case type)**

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

1410 **CourtPolicyQueryMessage**

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

1412 **CourtPolicyResponseMessage**

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

1414 **DocumentQueryMessage**

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

1416 **DocumentResponseMessage**

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

1418 **FilingListQueryMessage**

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

1420 **FilingListResponseMessage**

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

1422 **FilingPaymentMessage**

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

1424 **FilingStatusQueryMessage**

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

1426 **FilingStatusResponseMessage**

1427 [xml/ECF-4.0-FilingStatusResponseMessage.xml](#)

1428 **MessageReceiptMessage**

1429 [xml/ECF-4.0-MessageReceiptMessage.xml](#)

1430 **PaymentReceiptMessage**

- 1431 [xml/ECF-4.0-PaymentReceiptMessage.xml](#)
- 1432 **RecordDocketingCallbackMessage**
- 1433 [xml/ECF-4.0-RecordDocketingCallbackMessage.xml](#)
- 1434 **RecordDocketingMessage**
- 1435 [xml/ECF-4.0-RecordDocketingMessage.xml](#)
- 1436 **ReviewFilingCallbackMessage**
- 1437 [xml/ECF-4.0-ReviewFilingCallbackMessage.xml](#)
- 1438 **ServiceInformationQueryMessage**
- 1439 [xml/ECF-4.0-ServiceInformationQueryMessage.xml](#)
- 1440 **ServiceInformationResponseMessage**
- 1441 [xml/ECF-4.0-ServiceInformationResponseMessage.xml](#)
- 1442 **ServiceReceiptMessage**
- 1443 [xml/ECF-4.0-ServiceReceiptMessage.xml](#)

1444 **Appendix E. (Informative) Ongoing Work Items**

1445 The Electronic Court Filing TC plans to continue to revise and expand this specification through future
1446 versions. Future versions of ECF will:

- 1447 • Address filings in administrative tribunals
- 1448 • Address primary service (the delivery of documents such as summonses, subpoenas and warrants
1449 that establish a court's jurisdiction over a party)
- 1450 • Consider how the specifications for filing of documents intended for filing with a court relate to
1451 specifications for filing other documents, e.g., property records, in the offices of elected clerks of
1452 courts
- 1453 • Incorporate feedback from ECF implementations
- 1454 • Support future releases of the **[NIEM]**
- 1455 • Support future **[Court Document]** specifications and integration optimizations
- 1456 • Support non-case related filings into a court clerk's office

1457

Appendix F. (Informative) Acknowledgments

1458 The following court organizations provided lists of data elements required for initiating cases in their case
1459 management information systems:

- 1460 • Administrative Office of United States Courts
 - 1461 ○ Bankruptcy, Civil, Criminal
- 1462 • Arizona Administrative Office of the Courts
 - 1463 ○ Appellate
- 1464 • Fourth Judicial District Court, Hennepin County, Minneapolis
 - 1465 ○ Criminal
- 1466 • King County Superior Court, Washington
 - 1467 ○ Civil, Criminal
- 1468 • Missouri Office of State Courts Administrator
 - 1469 ○ Citation
- 1470 • Thirteenth Judicial District, Orange County, Florida (through vendor)
 - 1471 ○ Civil, Criminal, Domestic relations, Mental health, Juvenile delinquency/dependency,
 - 1472 Probate, Citation
- 1473 • Utah State Courts
 - 1474 ○ Civil, Criminal

1475

1476 The following individuals were members or voting members of the committee during the development of
1477 this specification:

- 1478 • Rolly Chambers, American Bar Association
- 1479 • John Messing, American Bar Association
- 1480 • Adam Angione, Courthouse News Service
- 1481 • Eric Eastman Doxpop, LLC
- 1482 • Robert DeFilippis, Associate
- 1483 • Chester Ensign, Associate
- 1484 • Gary Poindexter, Associate
- 1485 • Michael Alexandrou, Judicial Council of Georgia
- 1486 • Shawn Artrip, Judicial Council of Georgia,
- 1487 • Robbie Diaz, Judicial Council of Georgiar
- 1488 • Hui Ji, Judicial Council of Georgia
- 1489 • Morgan Medders, Judicial Council of Georgia
- 1490 • Scott Edson, LA County Information Systems Advisory Body
- 1491 • Ali Farahani, LA County Information Systems Advisory Body
- 1492 • John Ruegg, LA County Information Systems Advisory Body
- 1493 • CJ Allen, Maricopa County
- 1494 • Robin Gibson, Missouri Office of State Courts Admin
- 1495 • James Cabral, MTG Management Consultants, LLC
- 1496 • Thomas Clarke, National Center for State Courts
- 1497 • Diana Graski, National Center for State Courts
- 1498 • Jim Harris, National Center for State Courts
- 1499 • Jason Hill, New York State Office of Court Administration, DoT
- 1500 • Robert O'Brien, Ottawa Courts Administration Service
- 1501 • George Knecht, PC Intellect, LLC
- 1502 • Mark Ladd, Property Records Industry Assn.
- 1503 • Ron Bowmaster. Utah Administrative Office of the Courts

Appendix G. (Informative) Revision History

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