Electronic Court Filing Version 4.1

Committee Specification 01

29 September 2023

This stage:
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/cs01/ecf-v4.1-cs01.docx (Authoritative)
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/cs01/ecf-v4.1-cs01.html
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/cs01/ecf-v4.1-cs01.pdf

Previous stage:
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/csd02/ecf-v4.1-csd02.docx (Authoritative)
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/csd02/ecf-v4.1-csd02.html
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/csd02/ecf-v4.1-csd02.pdf

Latest stage:
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/ecf-v4.1.docx (Authoritative)
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/ecf-v4.1.html
https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/ecf-v4.1.pdf

Technical Committee:
OASIS LegalXML Electronic Court Filing TC

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

Editors:
James Cabral (jim.cabral@infotrack.com), InfoTrack US
Gary Graham (GGraham@courts.az.gov), Arizona Supreme Court
Philip Baughman (Philip.Baughman@tylertech.com), Tyler Technologies, Inc.

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

- XML schemas: https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/cs01/xsd/.
- XML sample messages: https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/cs01/xml/.
- Model and documentation: https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/cs01/model/.
- Genericode code lists: https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/cs01/gc/.
- Specification metadata: https://docs.oasis-open.org/legalxml-courtfiling/ecf/v4.1/cs01/xsd/metadata.xml.

Related work:
This specification replaces or supersedes:

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

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

This specification is related to:

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

Declared XML namespaces:

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

Abstract:

This document defines the LegalXML Electronic Court Filing 4.1 (ECF 4.1) specification, which consists of a set of non-proprietary XML and Web services specifications, along with clarifying explanations and amendments to those specifications, that have been added for the purpose of promoting interoperability among electronic court filing vendors and systems. ECF Version 4.1 is a maintenance release to address several minor schema and definition issues identified by implementers of the ECF 4.0 and 4.01 specifications.

Status:

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

This specification is provided under the RF on Limited Terms Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/legalxml-courtfiling/ipr.php).

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

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

[ECF-v4.1]

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

Notices:

Table of Contents

1 Introduction

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

1.1 Scope

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

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

This specification supports the following automated information exchanges:

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

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

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

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

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

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

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

1.2 Relationship to Prior Specifications

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

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

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

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

1.3 ECF Version 4.1

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

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

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

1.3.1 National Information Exchange Model (NIEM)

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

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

1.3.2 OASIS Universal Business Language

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

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

`<AllowanceCharge>`

Information about a charge or discount price component.

`<Address>`

Information about a structured address.

`<Payment>`

Information directly relating to a specific payment.

1.3.3 W3C *XML-Signature Syntax and Processing*

The W3C XML Signature Syntax and Processing (**[XMLSIG]**) specification describes a mechanism for 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

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

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

1.3.4 OASIS Reference Model for Service Oriented Architecture

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

1.3.5 OASIS Code List Representation (Genericode)

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

1.4 Terms and Definitions

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

This section defines key terms used in this specification.

Attachment

See definition in Section 2.3.2.

Callback message

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

Document

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

Document hash

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

Docketing

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

Filer

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

[2] http://xml.coverpages.org/xmlSig.html

Filing

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

Hub Service MDE

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

Major Design Element (MDE)

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

Message

See definition in Section 2.3.1.

Message Transmission

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

Operation (or MDE Operation)

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

Operation signature

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

Synchronous response

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

1.5 Symbols and Abbreviations

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

ECF 4.1

Electronic Court Filing 4.1

IEPD

Information Exchange Package Documentation

MDE

Major Design Element

NIEM

National Information Exchange Model

OASIS

Organization for the Advancement of Structured Information Standards

XML

eXtensible Markup Language

W3C

World Wide Web Consortium

WS-I

Web Services Interoperability Organization

1.6 Normative References

[FIPS 180-4]

Secure Hash Standard, http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf, National Institute for Standards and Technology, March 2012.

[Genericode]

G. K. Holman, Code List Representation (Genericode) 1.0, http://docs.oasis-open.org/codelist/ns/genericode/1.0/, OASIS Committee Specification 02, April 06, 2022.

[NIEM]

National Information Exchange Model 2.0, http://niem.gov*, US DOJ and DHS, 2007.*

[NIEM Guide]

NIEM Implementation Guidelines, http://www.niem.gov/implementationguide.php, US DOJ and DHS, 2007.

[NIEM Techniques]

*Techniques for Building and Extending NIEM***,** http://reference.niem.gov/niem/guidance/techniques-for-building-and-extending/2.0.1/techniques-for-building-and-extending-niem-2.0.1.pdf, Georgia Tech Research Institute, August 2007.

[Namespaces]

T. Bray, Namespaces in XML, http://www.w3.org/TR/1999/REC-xml-names-19990114, January 14, 1999.

[RFC2046]

N. Freed, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, http://www.ietf.org/rfc/rfc2046.txt, IETF RFC 2046, November 1996.

[RFC2119]

S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[RFC4122]

Leach, et al., A Universally Unique IDentifier (UUID) URN Namespace, http://www.ietf.org/rfc/rfc4122.txt, IETF RFC 4112, July 2005.

[Schema Part 1]

H. S. Thompson, D. Beech. M. Maloney, N. Mendelsohn, XML Schema Part 1: Structures Second Edition, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/, W3C Recommendation, October 28, 2004.

[Schema Part 2]

P. Biron, A. Malhotra, XML Schema Part 2: Datatypes Second Edition, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/, W3C Recommendation, October 28, 2004

[UBL] *Universal Business Language Version 2.1. 04 November 2013. OASIS Standard. http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html.*

[XML 1.0]

T. Bray, Extensible Markup Language (XML) 1.0 (Third Edition), http://www.w3.org/TR/REC-xml/REC-XML-20040204, W3C Recommendation, February 4, 2004.

[XMLENC]

D. Eastlake, J. Reagle, XML Encryption Syntax and Processing, http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/, *W3C Recommendation, December 2002.*

[XMLSIG]

D. Eastlake, J. Reagle, D. Solo, XML-Signature Syntax and Processing, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/, *W3C Recommendation, February 2002.*

1.7 Non-Normative References

[Court Document]

*OASIS LegalXML Court Document Subcommittee,Charter***,** July 2006**,** *http://www.oasis-open.org/committees/download.php/19120/DocumentSC-Charter-Rev6-1.txt.*

[NIEM MNDR]

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

[Juvenile XML]

S. Rondendell, et. al., Juvenile Justice XML Report, https://web.archive.org/web/20050910223309/http://www.ijis.org/db/share/public/Library/Publications/juvenile_justice_xml_final_report_20050630.pdf, IJIS Institute, July 2005.

[NIEM]

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

[NCSC Guide]

State Court Guide to Statistical Reporting, https://www.courtstatistics.org/pub-and-def-second-row-cards/guide-to-statistical-reporting#:~:text=The%20State%20Court%20Guide%20to%20Statistical%20Reporting%20was%20updated%20in,court%20caseload%20statistics%20in%201975., National Center for State Courts, January 2023.

[Rap Sheet]

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

[SOA-RM]

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

[Traffic IEPD]

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

2 ECF 4.1 Architecture

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

2.1 Core vs. Profiles

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

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

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

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

2.2 Major Design Elements

ECF 4.1 defines four MDEs. They are:

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

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

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

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

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

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

2.3 Information Model

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

2.3.1 Messages

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

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

```
<FilingLeadDocument> (or <FilingConnectedDocument>)
   <ecf:DocumentRendition>
         <DocumentRenditionMetadata>
               <DocumentAttachment>
                     <BinaryBase64Object>2345klj345h…<BinaryBase64Objec
               t>
               </DocumentAttachment>
         </DocumentRenditionMetadata>
   </ecf:DocumentRendition
```

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

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

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

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

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

2.3.2 Attachment

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

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

2.3.3 Sample Message Streams

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

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.

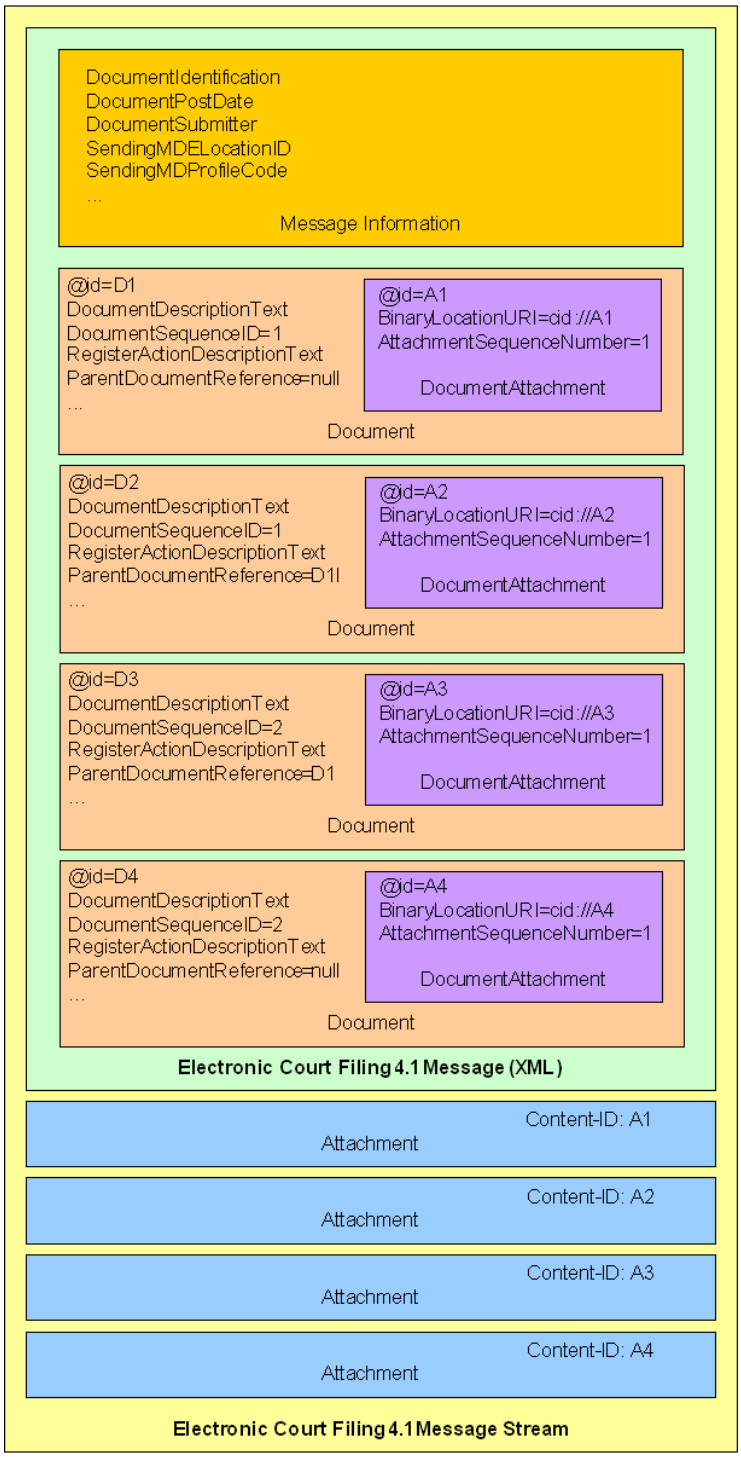


Figure 1. Simple Message Stream

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.

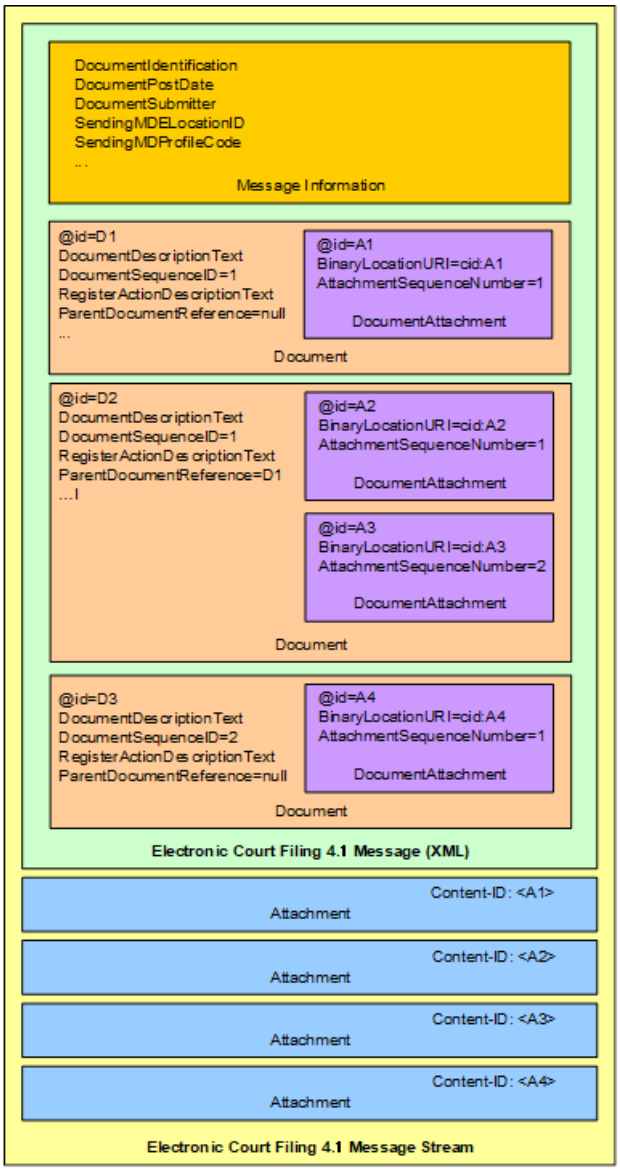


Figure 2. Message Stream with a Document in Multiple Attachments

2.4 Court Policy

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

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

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

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

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

2.4.1 Human-Readable Court Policy

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

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

2.4.2 Machine-Readable Court Policy

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

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

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

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

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

2.4.3 Case-Type and Court Extensions

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

2.4.4 Court-Specific Code Lists

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

- ECF Code Lists
  - Civil Case Type
    - `<FiduciaryTypeCode>*`
    - `<JurisdictionalGroundsCode>`
    - `<ReliefTypeCode>`
  - Domestic Case Type
    - `<NoContactCode>*`
    - `<RequestToVacateCode>`
  - Common Types
    - `<AliasAlternateNameTypeCode>*`
    - `<CaseAssociationTypeCode>*`
    - `<CaseOfficialRoleText>*`

- <CaseParticipantRoleCode>*
- <CauseOfActionCode>
- <CourtEventTypeCode>
- <EntityAssociationTypeCode>
- <ErrorCode>*

- Juvenile Case Type
  - <DelinquentActApplicabilityCode>
  - <DelinquentActDegreeCode>
  - <DelinquentActSeverityCode>
  - <DelinquentActSpecialAllegationCode>
  - <DependencyAllegationCode>
  - <GuardianAssociationTypeCode>*
  - <PlacementTypeCode>

- NIEM Code Lists
  - JXDM
    - <ChargeEnhancingFactorText>
    - <CourtLocationCode>
    - <RegisterActionDescriptionText>
    - <StatuteCodeIdentification>
    - <StatuteCodeSectionIdentification>
    - <StatuteOffenseIdentification>
    - <StatusOffenseCodeIdentification>
  - NIEM Core
    - <BinaryDescriptionText>*
    - <CaseCategoryText>
    - <DriverLicenseCommercialClassCode>
    - <FamilyKinshipCode>*

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

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

2.4.5 Court-Specific Constraint Schemas

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

3 ECF 4.1 Process Model

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

3.1 The Filing-Preparation-to-Docketing Process Model

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

- GetPolicy
- GetServiceInformation
- GetFeesCalculation
- **ReviewFiling**
- ServeFiling
- **RecordFiling**
- NotifyDocketingComplete
- NotifyFilingReviewComplete

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

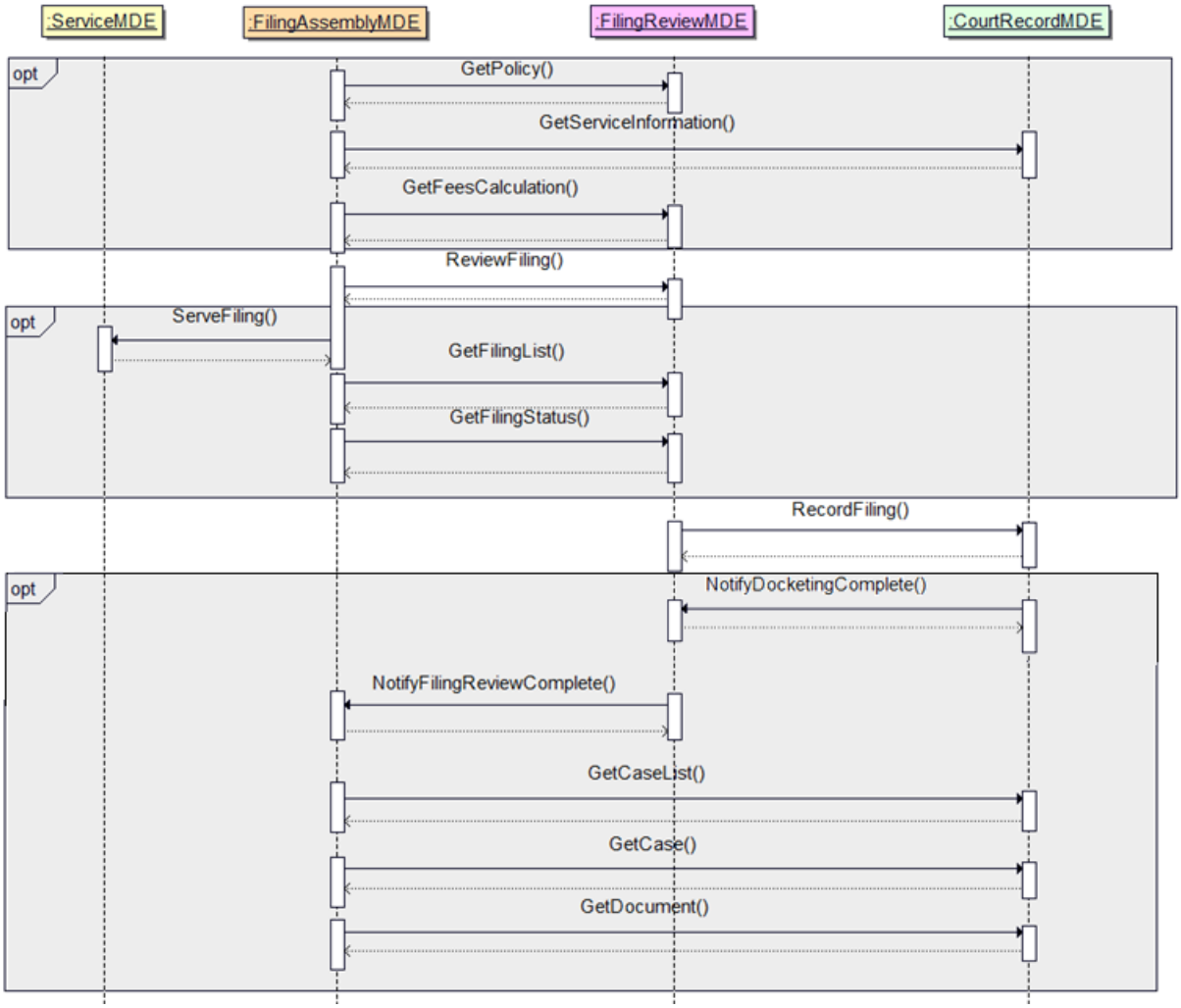
- GetFilingList
- GetFilingStatus

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

- GetCaseList
- GetCase
- GetDocument

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

Figure 4. Filing Preparation to Docketing Process Model



3.2 Business Rules

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

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

3.2.1 GetPolicy

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

3.2.2 GetServiceInformation

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

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

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

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

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

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

3.2.3 GetFeesCalculation

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

3.2.4 ReviewFiling

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

3.2.5 ServeFiling

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

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

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

3.2.6 RecordFiling

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

3.2.7 NotifyDocketingComplete

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

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

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

3.2.8 NotifyFilingReviewComplete

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

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

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

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

3.2.9 GetFilingList

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

3.2.10 GetFilingStatus

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

3.2.11 GetCaseList

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

3.2.12 GetCase

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

3.2.13 GetDocument

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

3.3 Message Business Rules

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

3.3.1 Identifiers

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

3.3.1.1 Attachment Identifiers

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

3.3.1.2 Case Identifiers

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

3.3.1.3 Court Identifiers

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

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

Examples of conformant court identifiers include:

- courts.wa.gov:superior.king
- nmcourts.com:albd.civil
- uscourts.gov:100
- courts.gov.bc.ca:appeal

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

3.3.1.4 Document Identifiers

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

3.3.1.5 Filing Identifiers

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

3.3.1.6 MDE Identifiers

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

3.3.1.7 Asynchronous responses

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

3.3.1.8 Filer and Party Identifiers

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

```
<nc:PersonOtherIdentification>
      <nc:IdentificationID>100<nc:IdentificationID>
      <nc:IdentificationCategoryText>ECFFilerID</nc:IdentificationCategoryTex
t>
</nc:PersonOtherIdentification>
```

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

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

3.3.2 Code Lists

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

- ECF Code Lists
  - Bankruptcy Case Type
    - `<DebtorTypeCode>`*
    - `<EstimatedAssetsValueLevelCode>`*
    - `<EstimatedDebtsValueLevelCode>`*
    - `<NatureOfDebtCode>`*
    - `<NumberOfCreditorsValueLevelCode>`*
  - Common Types
    - `<FilingStatusCode>`*
  - Court Policy Response Message
    - `<MajorDesignElementNameCode>`

794 - <OperationNameCode>
795 - Service Receipt Message
796 - <ServiceStatusCode>*
797 - NIEM Code Lists
798 - ANSI NIST
799 - <FingerPositionCode>
800 - JXDM
801 - <ChargeNCICCode>
802 - <DrivingIncidentHazMatCode>
803 - <DrivingJurisdictionAuthorityNCICLSTACode>
804 - <IdentificationJurisdictionNCICLISCode>
805 - <WarrantExtraditionLimitationCode>
806 - NIEM Core
807 - <DocumentLangageCode>
808 - <DriverLicenseCommercialClassCode>
809 - <DrivingRestrictionCode>
810 - <LanguageCode>
811 - <LengthUnitCode>
812 - <LocationCountryFIPS10-4Code>
813 - <LocationCountryISO3166Alpha2Code>
814 - <LocationCountyCode>
815 - <LocationStateUSPostalServiceCode>
816 - <PersonCitizenshipFIPS10-4Code>
817 - <PersonCitizenshipISO3166Alpha2Code>
818 - <PersonEthnicityCode>
819 - <PersonEyeColorCode>
820 - <PersonHairColorCode>
821 - <PersonRaceCode>
822 - <PersonSexCode>
823 - <PersonUnionCategoryCode>
824 - <PhysicalFeatureCategoryCode>
825 - <VehicleColorPrimaryCode>
826 - <VehicleMakeCode>
827 - <VehicleModelCode>
828 - <VehicleStyleCode>
829 - <WeightUnitCode>
830

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

3.3.3 Message-Specific Business Rules

The following business rules apply to specific messages:

3.3.3.1 CoreFilingMessage

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

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

3.3.3.2 PaymentMessage

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

3.3.3.3 RecordDocketingMessage

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

3.4 Filing the Record on Appeal

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

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

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

- The supporting documents that comprise the ROA transaction MAY also have additional attached documents.

- All ROA documents being submitted, including the Index of Record document and each document within the record, MUST have at least one court-defined document type that indicates the type of transaction to be performed on the document, and whether the document is being added to or stricken from the record.

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

- The Index of Record document and each document within the ROA transaction MAY also have an additional document type or types, which characterize the document for the Court Record MDE.

- When a document within the ROA transaction is being stricken from the court record, the document MUST be identified by the unique document identifier, which was provided by the Court Record MDE when the document was initially filed (See section 3.3.1.4).

- A hierarchical structure of case lineage elements MUST be used to express the target case's predecessor cases at prior courts. Each predecessor case MAY also have its own predecessor case, as necessary to express the full lineage of an appellate case.[4]

- When the ROA transaction is electronically transferred from one court to another, the target case number in the destination court and the case lineage, which includes the predecessor case number in the sending court, MUST be provided.

- If the ROA transaction is a case initiating filing in the destination court, then the `<FilingCase>` object MUST be present and the `<CaseTrackingID>` MUST be absent.

- Each predecessor case identified in the target case's case lineage may include case type-specific and court-specific extensions. The case type and the case type-specific extensions for each predecessor case MUST be consistent throughout the case lineage.

- When a ROA amendment transaction is sent, the Index of Record document MUST reflect the status of the record assuming that the transaction will be accepted. If however the transaction is rejected, there will be ramifications for other pending amendment transactions for the same ROA in the same target case. [5]

- While an ROA transaction is awaiting acceptance or rejection in the destination court, and when the target case consists of multiple records, courts SHOULD NOT send additional amendment 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.

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

4 ECF 4.1 Schemas

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

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

4.1 ECF 4.1 Case Type Schemas

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

AppellateCase

xsd/casetype/ECF-4.1-AppellateCase.xsd

BankruptcyCase

xsd/casetype/ECF-4.1-BankruptcyCase.xsd

CitationCase

xsd/casetype/ECF-4.1-CitationCase.xsd

CivilCase

xsd/casetype/ECF-4.1-CivilCase.xsd

CriminalCase

xsd/casetype/ECF-4.1-CriminalCase.xsd

DomesticCase

xsd/casetype/ECF-4.1-DomesticCase.xsd

JuvenileCase

xsd/casetype/ECF-4.1-JuvenileCase.xsd

4.2 ECF 4.1 Common Schemas

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

AppInfo

xsd/common/ECF-4.1-AppInfo.xsd

CommonTypes

xsd/common/ECF-4.1-CommonTypes.xsd

DigitalSignature

xsd/common/xmldsig-core-schema.xsd

Genericode

xsd/common/genericode.xsd

4.3 ECF 4.1 Constraint and Subset Schemas

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

4.4 ECF 4.1 Message Schemas

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

CaseListQueryMessage

xsd/message/ECF-4.1-CaseListQueryMessage.xsd

CaseListResponseMessage

xsd/message/ECF-4.1-CaseListResponseMessage.xsd

CaseQueryMessage

xsd/message/ECF-4.1-CaseQueryMessage.xsd

CaseResponseMessage

xsd/message/ECF-4.1-CaseResponseMessage.xsd

CoreFilingMessage

xsd/message/ECF-4.1-CoreFilingMessage.xsd

CourtPolicyQueryMessage

xsd/message/ECF-4.1-CourtPolicyQueryMessage.xsd

CourtPolicyReponseMessage

xsd/message/ECF-4.1-CourtPolicyResponseMessage.xsd

DocumentQueryMessage

xsd/message/ECF-4.1-DocumentQueryMessage.xsd

DocumentResponseMessage

xsd/message/ECF-4.1-DocumentResponseMessage.xsd

FeesCalculationQueryMessage

xsd/message/ECF-4.1-FeesCalculationQueryMessage.xsd

FeesCalculationResponseMessage

xsd/message/ECF-4.1-FeesCalculationResponseMessage.xsd

FilingListQueryMessage

xsd/message/ECF-4.1-FilingListQueryMessage.xsd

FilingListResponseMessage

xsd/message/ECF-4.1-FilingListResponseMessage.xsd

FilingStatusQueryMessage

xsd/message/ECF-4.1-FilingStatusQueryMessage.xsd

FilingStatusResponseMessage

xsd/message/ECF-4.1-FilingStatusResponseMessage.xsd

MessageReceiptMessage

xsd/message/ECF-4.1-MessageReceiptMessage.xsd

PaymentMessage

xsd/message/ECF-4.1-PaymentMessage.xsd

PaymentReceiptMessage

xsd/message/ECF-4.1-PaymentReceiptMessage.xsd

RecordDocketingCallbackMessage

xsd/message/ECF-4.1-RecordDocketingCallbackMessage.xsd

RecordDocketingMessage

xsd/message/ECF-4.1-RecordDocketingMessage.xsd

ReviewFilingCallbackMessage

xsd/message/ECF-4.1-ReviewFilingCallbackMessage.xsd

ServiceInformationQueryMessage

xsd/message/ECF-4.1-ServiceInformationQueryMessage.xsd

ServiceInformationResponseMessage

xsd/message/ECF-4.1-ServiceInformationResponseMessage.xsd

ServiceReceiptMessage

xsd/message/ECF-4.1-ServiceReceiptMessage.xsd

5 MDE Operations

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

5.1 Filing Assembly MDE

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

5.1.1 Provided Operations

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

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

5.1.2 Consumed Operations

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

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

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

5.2 Filing Review MDE

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

5.2.1 Provided Operations

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

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

5.2.2 Consumed Operations

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

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

5.3 Court Record MDE

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

5.3.1 Provided Operations

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

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

5.3.2 Consumed Operations

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

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

5.4 Legal Service MDE

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

5.4.1 Provided Operations

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

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

5.4.2 Consumed Operations

The Legal Service MDE does not call operations in other MDEs

6 Service Interaction Profiles

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

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

6.1 Service Interaction Profile Requirements

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

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

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

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

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

c. the date and time of the message transmission

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

6.2 Service Interaction Profile Approval and Revision Processes

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

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

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

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

6.3 Supported Service Interaction Profiles

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

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

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

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

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

7 Document Signature Profiles

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

7.1 Document Signature Profile Requirements

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

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

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

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

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

7.2 Document Signature Profile Approval and Revision Processes

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

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

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

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

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

7.3 Supported Document Signature Profiles

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

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

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

8 Conformance

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

Appendix A. (Informative) Release Notes

A.1 Availability

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

A.2 Package Structure

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

`gc/`

[Genericode] 1.0 code lists

`model/`

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

`xml/`

Example instances; see Appendix C

`xsd/`

XSD schemas; see Section 4

A.3 Recursive Structures

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

A.4 Date and Time Formats

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

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

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

A.5 Known Errata

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

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

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

B.1 Principles

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

- **Interoperability** – The ECF 4.1 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 4.1 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 4.1 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 4.1 Exchange Content Models

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

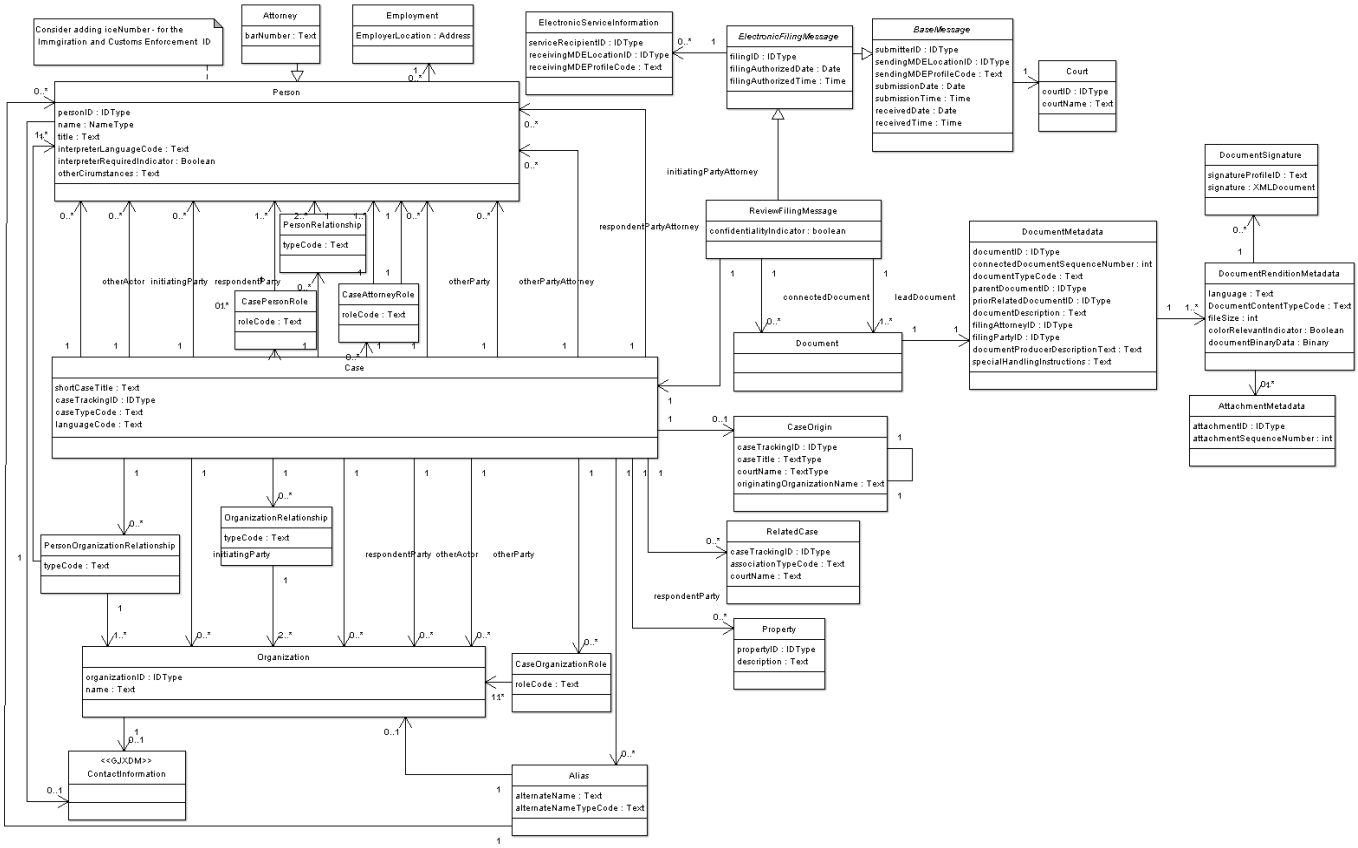
The exchange content models are the result of a detailed analysis of the data requirements to support the ECF 4.1 Process Model (see Section 3). During the modeling process, common items of data were identified by a process of normalization to identify aggregates based on functional dependency. Where 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 4.1 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 4.1 schemas (see Section 4).

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

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



The complete set of exchange content models for all the ECF 4.1 components is listed below:

Appellate Filing Model

model/uml/html/AppellateFiling.png

Bankruptcy Filing Model

model/uml/html/BankruptcyFiling.png

Base Message Model

model/uml/html/BaseMessage.png

Civil Filing Model

model/uml/html/CivilFiling.png

Citation Filing Model

model\uml\html\CitationFiling.png

Criminal Filing Model

model/uml/html/CriminalFiling.png

Domestic Filing Model

model/uml/html/DomesticFiling.png

Get Calculated Fees Query Model

model/uml/html/GetFeesCalculationQuery.png

Get Case List Query Model

model/uml/html/GetCaseListQuery.png

Get Policy Query Model

model/uml/html/CourtPolicy.png

Get Document Query Model

model/uml/html/GetDocumentQuery.png

Get Filing List Query Model

model/uml/html/GetFilingListQuery.png

Get Filing Status Query Model

model/uml/html/GetFilingStatusQuery.png

Get Service Information Query Model

model/uml/html/GetServiceInformationQuery.png

Major Design Elements Model

model/uml/html/MajorDesignElements.png

Juvenile Filing Model

model/uml/html/JuvenileFiling.png

Record Docketing Model

model/uml/html/RecordDocketing.png

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

B.4 Spreadsheet Models

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

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

Appendix C. (Informative) Example Instances

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

FeesCalculationQueryMessage
xml/ECF-4.1-FeesCalculationQueryMessage.xml
FeesCalculationResponseMessage
xml/ECF-4.1-FeesCalculationResponseMessage.xml
CaseListQueryMessage
xml/ECF-4.1-CaseListQueryMessage.xml
CaseListResponseMessage
xml/ECF-4.1-CaseListResponseMessage.xml
CaseQueryMessage
xml/ECF-4.1-CaseQueryMessage.xml
CaseResponseMessage
xml/ECF-4.1-CaseResponseMessage.xml
CoreFilingMessage (Appellate case type)
xml/ECF-4.1-CoreFilingMessage-Appellate.xml
CoreFilingMessage (Criminal case type)
xml/ECF-4.1-CoreFilingMessage-Criminal.xml
CourtPolicyQueryMessage
xml/ECF-4.1-CourtPolicyQueryMessage.xml
CourtPolicyReponseMessage
xml/ECF-4.1-CourtPolicyResponseMessage.xml
DocumentQueryMessage
xml/ECF-4.1-DocumentQueryMessage.xml
DocumentResponseMessage
xml/ECF-4.1-DocumentResponseMessage.xml
FilingListQueryMessage
xml/ECF-4.1-FilingListQueryMessage.xml
FilingListResponseMessage
xml/ECF-4.1-FilingListResponseMessage.xml
FilingPaymentMessage
xml/ECF-4.1-PaymentMessage.xml
FilingStatusQueryMessage
xml/ECF-4.1-FilingStatusQueryMessage.xml
FilingStatusResponseMessage
xml/ECF-4.1-FilingStatusResponseMessage.xml
MessageReceiptMessage
xml/ECF-4.1-MessageReceiptMessage.xml
PaymentReceiptMessage

xml/ECF-4.1-PaymentReceiptMessage.xml

RecordDocketingCallbackMessage

xml/ECF-4.1-RecordDocketingCallbackMessage.xml

RecordDocketingMessage

xml/ECF-4.1-RecordDocketingMessage.xml

ReviewFilingCallbackMessage

xml/ECF-4.1-ReviewFilingCallbackMessage.xml

ServiceInformationQueryMessage

xml/ECF-4.1-ServiceInformationQueryMessage.xml

ServiceInformationResponseMessage

xml/ECF-4.1-ServiceInformationResponseMessage.xml

ServiceReceiptMessage

xml/ECF-4.1-ServiceReceiptMessage.xml

Appendix D. (Informative) Ongoing Work Items

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

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

Appendix E. (Informative) Acknowledgments

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

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

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

Participants:

Philip Baughman, Tyler Technologies, Inc.

James Cabral, InfoTrack US

Eric Eastman, InfoTrack US

Ryan Foley, i3-ImageSoft, LLC

Gary Graham, Arizona Supreme Court

Barbara Holmes, National Center for State Courts

George Knecht, InfoTrack US

James McMillan, National Center for State Courts

Enrique Othon, Tyler Technologies, Inc.

Jim Price, Arizona Supreme Court

Brock Rogers, File & ServeXpress

Appendix F. (Informative) Revision History

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

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

1499

Appendix G. Notices

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: [https://www.oasis-open.org/policies-guidelines/ipr/].

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

[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 Standards Final Deliverable, 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 deliverable.]

[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 OASIS Standards Final Deliverable 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 OASIS Standards Final Deliverable. 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 OASIS Standards Final Deliverable 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 Standards Final Deliverable, 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 document, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, documents, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark/ for above guidance.