



Electronic Court Filing JSON Representation Version 5.0

Committee Note Draft 01

14 January 2026

This version

- <http://docs.oasis-open.org/legalxml-courtfilling/ecf-json/v5.0/cnd01/ecf-json-v5.0-cnd01.docx> (Authoritative)
- <http://docs.oasis-open.org/legalxml-courtfilling/ecf-json/v5.0/cnd01/ecf-json-v5.0-cnd01.html>
- <http://docs.oasis-open.org/legalxml-courtfilling/ecf-json/v5.0/cnd01/ecf-json-v5.0-cnd01.pdf>

Previous version

- N/A

Latest version

- <http://docs.oasis-open.org/legalxml-courtfilling/ecf-json/v5.0/ecf-json-v5.0.docx> (Authoritative)
- <http://docs.oasis-open.org/legalxml-courtfilling/ecf-json/v5.0/ecf-json-v5.0.html>
- <http://docs.oasis-open.org/legalxml-courtfilling/ecf-json/v5.0/ecf-json-v5.0.pdf>

Technical Committee

OASIS LegalXML Electronic Court Filing TC

Chairs

- James Cabral (jjim@cabral.org), Individual Member

Secretaries

- Gary Graham (GGraham@courts.az.gov), Arizona Supreme Court
- Cassidy Stokes (cstokes@tybera.com), Tybera Development Group

Editors

- James Cabral (jjim@cabral.org), Individual Member
- Gary Graham (GGraham@courts.az.gov), Arizona Supreme Court

Abstract

The *Electronic Court Filing Version 5.0* (ECF v5.0) specification consists of a set of non-proprietary XML and Web services specifications developed to promote interoperability among electronic court filing vendors and systems. ECF 5.0 JSON is a non-normative, alternative representation of ECF 5.0 in JSON format.

Citation Format

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

[ECF-v5.0 JSON]

- *Electronic Court Filing Version 5.0 JSON*. Edited by James Cabral and Gary Graham. 14 January 2026. OASIS Committee Note Draft 01. <http://docs.oasis-open.org/legalxml-courtfilling/ecf-json/v5.0/cnd01/ecf-json-v5.0-cnd01.html>. Latest version: <http://docs.oasis-open.org/legalxml-courtfilling/ecf-json/v5.0/ecf-json-v5.0.html>.

Related Work

This document is related to:

[ECF-v5.01]

Electronic Court Filing Version 5.01. Edited by James Cabral, Gary Graham, and Philip Baughman. 29 September 2023. OASIS Committee Specification 01. <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v5.01/cs01/ecf-v5.01-cs01.html>. Latest version: <https://docs.oasis-open.org/legalxml-courtfilling/ecf/v5.01/ecf-v5.01.html>.

License, Document Status, and Notices

Copyright © OASIS Open 2026. All Rights Reserved. For license and copyright information, and complete status, please see Annex A which contains the License, Document Status and Notices.

Table of Contents

1 Scope.....	4
2 Definitions and Acronyms.....	4
2.1 Definitions.....	4
2.1.1 Terms Defined Elsewhere.....	4
2.1.2 Terms Defined in this Document.....	5
2.2 Abbreviations and Acronyms.....	5
3 Document Conventions.....	6
3.1 Typographical Conventions.....	6
4 Introduction.....	6
4.1 Changes From the Previous Version.....	6
5 JSON Information Model.....	7
5.1 Messages.....	7
5.2 Code Lists.....	9
5.2.1 Normative Code Lists.....	9
5.2.2 Court-specific Code Lists.....	10
5.3 Attachments.....	11
6 JSON Service Definitions.....	11
Annex A License, Document Status and Notices.....	14
A.1 Document Status.....	14
A.2 License and Notices.....	14
Annex B References.....	16
B.1 Informative References.....	16
Appendix 1 Acknowledgments.....	17
Participants.....	17
Appendix 2 Changes From Previous Version.....	18
Revision History.....	18
Appendix 3 Release Notes.....	19
Appendix 4 Development Approach and Artifacts.....	20
Appendix 5 Message Formats.....	21
Appendix 6 Example Instances.....	30

1 Scope

This committee note is a non-normative, alternative representation of the OASIS Open LegalXML Electronic Court Filing version 5.01 Committee Specification 01 ([**ECF-v5.01**]) in JSON format.

2 Definitions and Acronyms

2.1 Definitions

2.1.1 Terms Defined Elsewhere

This document uses the following terms defined in the [**ECF-v5.01**] specification:

- Attachment: See definition in Section 4.4 of the [**ECF-v5.01**] specification.
- 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 intended to protect document integrity, 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.
- File format: A file representation of a document (e.g. PDF).
- Filer: An attorney, judicial official or a pro se (self-represented) litigant who electronically provides filings (combinations of data and documents) for acceptance and filing by a court, or who has successfully filed filings with a court.
- Filing: An electronic submission (with any associated data, one or many lead and connected documents, and the like) that has been assembled for the purpose of being filed, either into a specified court case, or to initiate a new court case.
- Filing Identifier: A unique value assigned as a tracking identifier for a 'Filing' (e.g. an e-filing submission). The filing identifier is carried by messages that are involved in an e-filing transaction that begins with the submittal of a filing:ReviewFiling message, and culminates with the final NotifyFilingReviewComplete operation call for the original filing:ReviewFiling message. Upon receipt of the final reviewfilingcallback:NotifyFilingReviewCompleteMessage by the originating FilingAssembly MDE, all filing lead and connected documents in the original filing:ReviewFiling message will have been reviewed and dispositioned (e.g. accepted and docketed, or rejected, etc.) or the filing will have been cancelled. Even after the conclusion of the e-filing episode, the filing identifier continues to be useful for GetFilingStatus requests.
- 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 5.01. Each MDE operation receives one or more messages, returning a synchronous response message (a reaction to a message received) and returning an OPTIONAL asynchronous (later) response message to the originating message sender. An MDE in ECF is comparable to a UML "Component", "Port" or "Class" with the "implementationClass" stereotype.

- Message: See definition in Section 4.1 of the **[ECF-v5.01]** specification. A Message in ECF is comparable to a UML “Parameter” or “Class” with the “Type” stereotype.
- Message Identifier: A unique value assigned to a message, either as a unique reference to the message, or as a correlation value to reference a prior message.
- 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 5.01 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. An Operation in ECF is comparable to a UML “Operation”.
- 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 5.01 specification.
- Party: A litigant in a case. A party MAY be a person, organization or property (e.g. “in rem” property).
- Participant: An entity (person, organization or thing) that plays some role in the context of an e-filing submission. Participants include parties, attorneys, clerks, judicial officials, other entities receiving service, etc.
- Submitter: The person or organization that tenders an ECF message to an operation hosted by a MDE. In the case of a filing, the submitter MAY or MAY NOT be the filing attorney or party.
- Synchronous response: A message transmission returned immediately (synchronously) as the result of an operation. Every operation has a synchronous response.
- Transaction Identifier: A unique value that identifies a set of messages which collectively belong to or relate to a single purpose, episode, or outcome. Filing Identifier is an example of a specific type of transaction identifier. A transaction identifier MAY also be used to relate messages collectively involved in the ‘Scheduling Process’, such as requestdaterequest:RequestCourtDateRequestMessage, requestdateresponse:RequestDateResponseMessage, reservedate:ReserveCourtDateMessage, datecallback:NotifyCourtDateMessage, and allocatedate:AllocateCourtDateMessage.

2.1.2 Terms Defined in this Document

This document defines the following terms:

- None

2.2 Abbreviations and Acronyms

This document uses the following abbreviations and acronyms:

- CMF: NIEMOpen Common Model Format
- CSV: Comma Separated Values
- ECF 5.01: Electronic Court Filing 5.01
- HTML: Hypertext Markup Language
- JSON: JavaScript Object Notation
- MDE: Major Design Element
- MPD: Model Package Description
- NIEM: NIEMOpen OASIS Open Project
- OASIS: Organization for the Advancement of Structured Information Standards
- OpenAPI: OpenAPI Specification (previously known as Swagger Specification)
- XML: eXtensible Markup Language

3 Document Conventions

3.1 Typographical Conventions

None

4 Introduction

This committee note is a non-normative, alternative representation of the OASIS Open LegalXML Electronic Court Filing version 5.01 Committee Specification 01 (**[ECF-v5.01]**) in JSON format. This alternative representation is intended for implementations that prefer a RESTful web services. This representation is also a demonstration of the use of the NIEMOpen Common Model Format (CMF) and related tooling to support multiple formats, including normative XML schemas, simplified XML schemas, and JSON schemas from a common semantic model.

4.1 Changes From the Previous Version

No previous version

5 JSON Information Model

The information model describes the data content exchanged between MDEs in each operation as a set of JSON messages, JSON schema and **[Genericcode]** code lists and binary attachments.

5.1 Messages

A message is a JSON document that is a well-formed JSON data structure with a root element that is valid as defined by a normative JSON schema provided with this committee note. Each message MAY reference one or more binary attachments. The transmission format of messages and attachments is defined in a service interaction profile.

The following table lists each ECF 5.01 operation, the MDEs that MUST provide and MUST consume the operation if the operation is either required or OPTIONAL and enabled by Court Policy, and the input and output JSON messages that define the data content exchanged. Other MDEs MAY also consume the operation. Each provided operation definition includes the input (parameter) and output messages and the required message cardinality in the format: (minimum occurrences, maximum occurrences). The JSON schema provided with this committee note at [json/schema/ecf.schema.json](https://www.ecf.gov/json/schema/ecf.schema.json) is a non-normative representation of ECF 5.01 messages.

TABLE 1. MESSAGES

Providing MDE	Consuming MDE	Operation	Input Message JSON element(s)	Output Message JSON element
Court Policy	Filing Assembly	GetPolicy	policyrequest:GetPolicyRequestMessage (1,1)	policyresponse:GetPolicyResponseMessage (1,1)
Court Record	Court Scheduling	AllocateCourtDate	allocatedate:AllocateCourtDateMessage (1,1)	cbrn:MessageStatus (1,1)
		GetCase	caserequest:GetCaseRequestMessage (1,1)	caseresponse:GetCaseResponseMessage (1,1)
	GetCaseList	caselistrequest:GetCaseListRequestMessage (1,1)	caselistresponse:GetCaseListResponseMessage (1,1)	
	GetDocument	documentrequest:GetDocumentRequestMessage (1,1)	documentresponse:GetDocumentResponseMessage (1,1)	
	GetServiceInformation	serviceinformationrequest:GetServiceInformationRequestMessage (1,1)	serviceinformationresponse:GetServiceInformationResponseMessage (1,1)	
	Filing Review	DocumentStampInformation	stampinformation:DocumentStampInformationMessage (1,1)	cbrn:MessageStatus (1,1)
		RecordDocketing	docket:RecordDocketingMessage (1,unbounded) payment:PaymentMessage (0,1)	cbrn:MessageStatus (1,1)
Court Scheduling	Filing Assembly	GetCourtSchedule	schedulerequest:GetCourtScheduleRequestMessage (1,1)	schedulereponse:GetCourtScheduleResponseMessage (1,1)
		RequestCourtDate	requestdaterequest:RequestCourtDateRequestMessage (1,1)	requestdatereponse:RequestCourtDateResponseMessage (1,1)

Providing MDE	Consuming MDE	Operation	Input Message JSON element(s)	Output Message JSON element
		ReserveCourtDate	reservedate:ReserveCourtDateMessage (1,1)	cbrn:MessageStatus (1,1)
	Court Record	NotifyCourtDate	datecallback:NotifyCourtDateMessage (1,1)	cbrn:MessageStatus (1,1)
Filing Assembly	Court Scheduling			
	Filing Review	NotifyFilingReviewComplete	reviewfilingcallback:NotifyFilingReviewCompleteMessage (1,1)	cbrn:MessageStatus (1,1)
Filing Review	Filing Assembly	CancelFiling	cancel:CancelFilingMessage (1,1)	cbrn:MessageStatus (1,1)
		GetFeesCalculation	feesrequest:GetFeesCalculationRequestMessage (1,1) payment:PaymentMessage (0,1)	feesresponse:GetFeesCalculationResponseMessage (1,1)
		GetFilingList	filinglistrequest:GetFilingListRequestMessage (1,1)	filinglistresponse:GetFilingListResponseMessage (1,1)
		GetFilingStatus	filingstatusrequest:GetFilingStatusRequestMessage (1,1)	filingstatusresponse:GetFilingStatusResponseMessage (1,1)
		ReviewFiling	filing:FilingMessage (1,1) payment:PaymentMessage (0,1)	cbrn:MessageStatus (1,1)
	Court Record	NotifyDocketingComplete	docketcallback:NotifyDocketingCompleteMessage (1,1) payment:PaymentMessage (0,1)	cbrn:MessageStatus (1,1)
		NotifyDocumentStampInformation	stampinformationcallback:NotifyDocumentStampInformationMessage (1,1)	cbrn:MessageStatus (1,1)
Service	Filing Assembly	ServeFiling	filing:FilingMessage (1,1)	cbrn:MessageStatus (1,1)
		ServeProcess	serveprocess:ServeProcessMessage (1,1)	cbrn:MessageStatus (1,1)

Each ECF 5.01 JSON message contains the following information:

- Filing metadata including identifiers for the sender and receiver, the sending and receiving MDEs, and the submission date and time.
- Information about the court case, including identifiers for the court and case.
- Circumstantially, information about one or more lead documents that will be placed on the court's register of actions (docketed, indexed) as a result of the filing. 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. The message includes the document 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, as defined in [Attachment Identifiers](#).

- Optionally, one or more supporting document(s), which are present to supplement the lead document(s) in some way. The message includes the same document metadata for lead and supporting documents.

5.2 Code Lists

Code Lists are used to constrain the allowable values for certain information in a message.

5.2.1 Normative Code Lists

The allowable values for the following JSON elements are normative for all ECF 5.01 implementations, are defined in ECF **[Genericcode]** code lists or NIEM or UBL JSON. The ECF **[Genericcode]** code lists are provided in this committee note for convenience.

TABLE 2. CODE LISTS

JSON element	Code List or JSON Schema
ecf:DocumentDocketingStatusCode	DocumentDocketingStatusCode.gc
ecf:DocumentReviewStatusCode	DocumentReviewStatusCode.gc
ecf:FilingDocketingStatusCode	FilingDocketingStatusCode.gc
ecf:FilingReviewStatusCode	FilingReviewStatusCode.gc
ecf:ServiceStatusCode	ServiceStatusCode.gc
policyresponse:MajorDesignElementTypeCode	MajorDesignElementTypeCode.gc
policyresponse:OperationNameCode	OperationNameCode.gc
biom:BiometricClassificationCategoryCode	ecf.schema.json
hs:ParentChildKinshipCategoryCode	
hs:PlacementCategoryCode	
j:ConveyanceColorPrimaryCode	
j:CrashDrivingRestrictionCode	
j:DriverAccidentSeverityCode	
j:DrivingIncidentHazMatCode	
j:DriverLicenseCommericalClassCode	
j:JurisdictionNCICLISCode	
j:JurisdictionNCICLSTACode	
j:OrganizationAlternateNameCategoryCode	
j:PersonEthnicityCode	
j:PersonEyeColorCode	
j:PersonHairColorCode	
j:PersonNameCategoryCode	
j:PersonRaceCode	
j:PersonSexCode	
j:PersonUnionCategoryCode	
j:ProtectionOrderConditionCode	
j:VehicleMakeCode	
j:VehicleModelCode	
j:VehicleStyleCode	
j:WarrantExtraditionLimitationCode	
nc>ContactInformationAvailabilityCode	

JSON element	Code List or JSON Schema
nc:CurrencyCode	
nc:DocumentLanguageCode	
nc:LanguageCode	
nc:LengthUnitCode	
nc:LocationStateUSPostalServiceCode	
nc:PersonCitizenshipFIPS10-4Code	
nc:SpeedUnitCode	
nc:WeightUnitCode	
cbc:PaymentMeansCode	PaymentMeansCode-2.1.gc

5.2.2 Court-specific Code Lists

Courts SHOULD publish **[Genericode]** 1.0 code lists that define the allowable values in that court for each of the following JSON elements in the following table.

TABLE 3. COURT-SPECIFIC CODE LISTS

JSON element	[Genericode] code list	Default values
civil:FiduciaryTypeCode	FiduciaryTypeCode.gc	Yes
civil:JurisdictionalGroundsCode	JurisdictionalGroundsCode.gc	
civil:ReliefTypeCode	ReliefTypeCode.gc	
cbrn:ErrorCodeText	ErrorCodeText.gc	Yes
ecf:CaseCategoryCode	CaseCategoryCode.gc	
ecf:CaseParticipantRoleCode	CaseParticipantRoleCode.gc	Yes
ecf:CaseTypeCode	CaseTypeCode.gc	Yes
ecf:CauseOfActionCode	CauseOfActionCode.gc	
ecf:CourtEventTypeCode	CourtEventTypeCode.gc	
ecf:DocumentRelatedCode	DocumentRelatedCode.gc	
ecf:DocumentTypeCode	DocumentTypeCode.gc	
ecf:EntityAssociationTypeCode	EntityAssociationTypeCode.gc	
ecf:FeeExceptionReasonCode	FeeExceptionReasonCode.gc	
ecf:PersonIdentificationCategoryCode	PersonIdentificationCategoryCode.gc	Yes
ecf:RelatedCaseAssociationTypeCode	RelatedCaseAssociationTypeCode.gc	Yes
ecf:ServiceInteractionProfileCode	ServiceInteractionProfileCode.gc	Yes
ecf:SignatureProfileCode	SignatureProfileCode.gc	Yes
hs:AbuseNeglectAllegationCategoryText	AbuseNeglectAllegationCategoryText.gc	
j:ChargeDegreeText	ChargeDegreeText.gc	
j:ChargeEnhancingFactorText	ChargeEnhancingFactorText.gc	
j:ChargeSpecialAllegationText	ChargeSpecialAllegationText.gc	
j:IncidentLevelCode	IncidentLevelCode.gc	Yes

JSON element	[Genericcode] code list	Default values
j:PersonIdentificationCategoryCode	PersonIdentificationCategoryCode.gc	
j:RegisterActionDescriptionText	RegisterActionDescriptionText.gc	
juvenile:DelinquentActCategoryCode	DelinquentActCategoryCode.gc	
nc:BinaryFormatText	BinaryFormatText.gc	Yes
nc:IdentificationCategoryDescriptionText	IdentificationCategoryDescriptionText.gc	Yes
nc:LocationCountryName	LocationCountryName.gc	
nc:SensitivityText	SensitivityText.gc	Yes

The **[ECF 5.01]** specification provides non-normative **[Genericcode]** code lists for each of the XML elements in the following table. The **[Genericcode]** templates for each code list are provided in this committee note for convenience. The specification-provided code lists in the table above that are marked as “Yes” for “Default Values” have specification-provided values.

5.3 Attachments

The binary content of an electronic document SHOULD be transmitted as one or more attachments. A document MAY be split into several attachments to satisfy a court requirement regarding maximum document size. Each attachment MUST include a content identifier unique to the specific message exchange and referenced in the message using a `nc:BinaryURI` element. The assignment of content identifiers to attachments and the order of transmission of messages and attachments is defined in the service interaction profile.

JSON Example 1: reference to a binary document attachment

```
```json
{
 "FilingLeadDocument": {
 "ecf:DocumentRendition": [{
 "nc:DocumentBinary": {
 "nc:BinaryURI": "cid://Payload2</nc:BinaryURI>"
 }
 }]
 }
}
...
```
```

Sample messages input and output message formats for both synchronous and asynchronous operations are provided in [Message Formats](#).

6 JSON Service Definitions

Implementation of ECF 5.01 JSON messages MAY be described in a web services based on the following non-normative OpenAPI 3.1 documents corresponding to each MDE:

- [CourtPolicyMDE.openapi.json](#)
- [CourtRecordMDE.openapi.json](#)
- [CourtSchedulingMDE.openapi.json](#)

- [FilingAssemblyMDE.openapi.json](#)
- [FilingReviewMDE.openapi.json](#)
- [ServiceMDE.openapi.json](#)

Annex A License, Document Status and Notices

(This annex forms an integral part of this document.)

A.1 Document Status

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

TC members should send comments on this document to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the comment instructions at the TC's comments list web page at https://www.oasis-open.org/committees/comments/index.php?wg_abbrev=legalxml-courtfilling.

A.2 License and Notices

Copyright © OASIS Open 2026. All Rights Reserved.

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

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

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns, as provided in the OASIS IPR Policy.

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

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, its documents, while reserving the right to enforce its marks against misleading uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark/> for guidance.

Annex B References

(This annex forms an integral part of this document.)

This section contains the informative references that are used in this document.

Informative references are either specific or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies. While any hyperlinks included in this section were valid at the time of publication, OASIS cannot guarantee their long term validity.

B.1 Informative References

The following referenced documents are not required for the application of this document but may assist the reader with regard to a particular subject area.

- [Genericode]** ***Code List Representation (genericode) Version 1.0***. Edited by G. Ken Holman. 31 January 2023. OASIS Standard. <https://docs.oasis-open.org/codelist/genericode/v1.0/os/genericode-v1.0-os.html>. Latest stage: <https://docs.oasis-open.org/codelist/genericode/v1.0/genericode-v1.0.html>.
- [NIEM]** *National Information Exchange Model 4.1*, 2018, NIEM Business Architecture Committee, <http://niemopen.org/>.
- [NIEM Code Lists]** ***NIEM Code Lists Specification 4.0.1***, 28 September 2022, NIEM Technical Architecture Committee, <https://github.com/niemopen/code-lists-specification/blob/dev/code-list-spec-4.0.1-export.md>.
- [NIEM Conformance]** ***NIEM Conformance Specification 5.0***, 19 March 2021, NIEM Technical Architecture Committee, <https://github.com/niemopen/conformance-specification/blob/dev/conformance-spec-5.0-export.md>
- [NIEM NDR]** ***NIEM Naming and Design Rules 4.0***, 7 November 2017, NIEM Technical Architecture Committee, <https://github.com/NIEM/NIEM-NDR/tree/dev-niem-ndr-4.0>.
- [UBL]** Universal Business Language Version 2.2, 9 July 2018, OASIS Standard, <http://docs.oasis-open.org/ubl/os-UBL-2.2/>

Appendix 1 Acknowledgments

(This appendix does not form an integral part of this document and is informational.)

Participants

The following individuals were members of this committee during the creation of this document, not just this version of the document, and their contributions are gratefully acknowledged:

- James Cabral, Individual Member
- Eric Eastman, InfoTrack US
- Ryan Foley, i3 ImageSoft
- Gary Graham, Arizona Supreme Court
- George Knecht, InfoTrack US
- Mark Leong, Arizona Supreme Court
- Dallas Powell, Tybera Development Group
- Tyler Powell, Tybera Development Group
- Jim Price, Arizona Supreme Court
- Brock Rogers, File & ServeXpress
- Cassidy Stokes, Tybera Development Group
- David Sturdivant, Tyler Technologies, Inc
- Patrick Wallace, Tyler Technologies, Inc.
- Charlie Wilson, Tyler Technologies, Inc..

Appendix 2 Changes From Previous Version

(This appendix does not form an integral part of this document and is informational.)

The changes from previous versions of this document as listed below:

- None

Revision History

- 2026-1-14, CND01

Appendix 3 Release Notes

(This appendix does not form an integral part of this document and is informational.)

Availability

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

Package Structure

The ECF 5.01 JSON specification is also published as a ZIP archive accompanying this document in the OASIS Library. Unzipping this archive creates a directory named `ecf5-json/` 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 1.0 release. A description of each subdirectory is given below.

[cmf](#)

NIEMOpen Common Model Format (CMF)

[codelists](#)

Genericcode codelists – as provided in the ECF 5.01 specification and included for convenience

[json/examples/](#)

Example instances; see [Example Instances](#)

[json/openapi/](#)

OpenAPI 3.1 documents

[json/schema/](#)

JSON schemas

[model](#)

HTML documentation of the UML model and NIEMOpen mapping spreadsheet

[uml](#)

BoUML project files describing the UML model

Date, Time and Duration Formats

The **[ECF 5.01]** specification provides additional guidance on the interpretation of date, time and duration formats.

Appendix 4 Development Approach and Artifacts

(This appendix does not form an integral part of this document and is informational.)

This appendix describes the approach used to develop the JSON representation of ECF 5.01.

Approach

As described in the ECF 5.01 specification, the ECF 5.01 message schemas were developed as a **[NIEM]** Message Specification (formerly called IEPD) as defined by the NIEM Naming and Design Rules (**[NIEM NDR]**)

Specifically, the sequence of tools used to develop the artifacts described below is listed here:

- [BoUML](#) was used to model the data requirements for ECF 5.01.
- The [niem-tools plugout](#) was used to map the requirements from UML to the NIEMOpen reference model, and generate a NIEMOpen Common Model Format (CMF) document, Genericcode definitions for each codelists as well as OpenAPI 3.1 documents for each MDE.
- [cmftool](#) was then used to generate complete XML schemas, simplified XML schemas and JSON schema representations from the CMF.
- [Oxygen XML editor](#) was then used to generate example JSON message for each ECF message.

UML Models

The UML models provided in the [uml](#) folder and exported in HTML format in the [model](#) folder describe the use cases, components, services, interfaces, messages and data content required for ECF 5.0. The [index.html](#) file provides a starting point for navigating the models. The models are the result of a detailed analysis of the process and data requirements to support the ECF 5.01 use cases.

Spreadsheet Model

A spreadsheet model maps the UML objects and attributes to **[NIEM]**, **[UBL]** and ECF 5.0 specific types and elements. The ECF 5.0 spreadsheet model is provided in both [CSV](#) and [HTML](#) formats.

Common Model Format (CMF)

The [cmf/ecf5.cmf](#) file is an abstract representation of the model conforming with the [Common Model Format \(CMF\) specification](#).

JSON Message Schemas

The structure of all ECF 5.01 messages is defined in JSON schema format in the [json/schema/ecf5.schema.json](#) file. This file uses a [JSON-LD context](#) to manage multiple namespaces. Each coded JSON element has a corresponding Genericcode definition in the [codelists](#) folder.

Example JSON Messages

The [json/examples/](#) folder includes [Example Instances](#) of all ECF 5.01 messages in JSON format.

OpenAPI 3.1 documents

The [json/openapi/](#) folder provides OpenAPI 3.1 documents defining the API of each MDE.

Appendix 5 Message Formats

(This appendix does not form an integral part of this document and is informational.)

The following JSON fragments illustrates the typical structure for input and output messages in asynchronous and synchronous ECF 5.0 operations.

Asynchronous operation input message

The following JSON fragment illustrates the typical structure for an input message to an asynchronous operation. `nc:BinaryURI` references the MIME ID of the attached document

JSON Example 2.1: Asynchronous operation input message

```
```json
{
 "$schema": "../schema/ecf5.schema.json",
 "@context": "../schema/ecf5.context.jsonld",
 "filing:FilingMessage": {
 "nc:DocumentIdentification": [{
 "nc:IdentificationID": "123456ABC",
 "nc:IdentificationCategoryDescriptionText": "messageID",
 "nc:IdentificationSourceText": "FilingAssembly"
 }],
 "ecf:SendingMDELocationID": {"nc:IdentificationID": "http://example.com/efsp1"},
 "ecf:ServiceInteractionProfileCode": "urn:oasis:names:tc:legalxml-
courtfiling:schema:xsd:WebServicesMessaging-5.0",
 "j:CaseCourt": {
 "nc:OrganizationIdentification": {"nc:IdentificationID": "gregg:dc"}
 },
 "nc:DocumentPostDate": {"nc:DateTime": "2008-07-07T13:47:42.0Z"},
 "filing:FilingLeadDocument": [{
 "@id": "#Document1",
 "nc:metadataRef": { "$ref": ["#Document1Metadata"] },
 "nc:DocumentDescriptionText": "Appearance",
 "nc:DocumentIdentification": [],
 "nc:DocumentSequenceID": "0",
 "ecf:DocumentRendition": [{
 "nc:DocumentIdentification": [],
 "nc:Attachment": {
 "nc:Base64BinaryObject": "",
 "nc:BinaryDescriptionText": "Appearance Document",
 "nc:BinaryFormatText": "332"
 }
 }],
 }],
 "ecf:RegisterActionDescriptionCode": "661",
 "nc:Metadata": [{
 "@id": "#Document1Metadata",
 "nc:SensitivityText": "3200"
 }],
 }],
 "nc:Case": {
 "j:CaseOfficial": [{
 "nc:RoleOfPerson": {
 "nc:PersonOtherIdentification": [{"nc:IdentificationID": "037BAEC1-F6E8-
43ED-A102-B8C5E76748EB"}]
 }
 }],
 }
}
```

```

 },
 "ecf:CaseRepresentedParty": [{"@id": "#Party1"}]
 }],
 "ecf:CaseCategoryCode": "7",
 "ecf:CaseNewIndicator": true,
 "ecf:CaseParty": [{
 "nc:EntityPerson": {
 "@id": "#Party1",
 "nc:PersonBirthDate": {"nc:Date": "1980-01-01T00:00:00Z"},
 "nc:PersonName": {
 "nc:PersonGivenName": "John",
 "nc:PersonMiddleName": "",
 "nc:PersonSurName": "Smith"
 },
 },
 "nc:PersonOtherIdentification": [{
 "nc:IdentificationID": "000-00-0000",
 "ecf:PersonIdentificationCategoryCode": "SSN"
 }],
 "ecf:CaseParticipantRoleCode": ["Plaintiff"],
 "ecf:ParticipantID": {},
 "nc:ContactInformation": [{
 "nc:ContactTelephoneNumber": [{
 "nc:FullTelephoneNumber": {"nc:TelephoneNumberFullID":
"9727133770"}
 }],
 "nc:ContactEmailID": ["john.doe@someemail.com"],
 "nc:ContactMailingAddress": [{
 "nc:LocationStreet": {
 "nc:StreetFullText": "100 Circle Road",
 "nc:StreetExtensionText": "Apt 2"
 },
 "nc:LocationCityName": "San Juan",
 "nc:LocationState": {"nc:LocationStateName": "IN"},
 "nc:LocationCountry": {"nc:LocationCountryName": "RQ"},
 "nc:LocationPostalCode": "75093"
 }
 }
]
}
}],
"ecf:CaseTypeCode": "civil",
"civil:CivilClassActionIndicator": false,
"civil:JuryDemandIndicator": false,
"civil:ReliefTypeCode": [],
"ecf:CauseOfActionCode": ""
}
}
}
...

```

## Asynchronous operation output message

The following JSON fragment illustrates the typical structure for an output message to an asynchronous operation.

### JSON Example 2.2: Asynchronous operation output message

```
```json
{
  "$schema": "../schema/ecf5.schema.json",
  "@context": "../schema/ecf5.context.jsonld",
  "cbrn:MessageStatus": {
    "cbrn:systemSimulatedIndicator": false,
    "cbrn:SystemEventDateTime": "2017-01-07T13:47:42.0Z",
    "cbrn:SystemOperatingModeCode": "Ops",
    "cbrn:CredentialsAuthenticatedCode": "Authenticated",
    "cbrn:MessageStatusCode": "DataError",
    "cbrn:MessageContentError": [{
      "cbrn:ErrorNodeName": "filing:FilingMessage",
      "cbrn:ErrorDescription": {"cbrn:ErrorCodeText": "1"}
    }],
    "cbrn:MessageHandlingError": {"cbrn:ErrorCodeText": "1"},
    "cbrn:ResendRequestIndicator": false,

    "ecf:ServiceRecipientID": {"nc:IdentificationID": "1"},
    "nc:DocumentIdentification": [
      {
        "nc:IdentificationID": "1",
        "nc:IdentificationCategoryDescriptionText": "messageID",
        "nc:IdentificationSourceText": "CourtRecord"
      },
      {
        "nc:IdentificationID": "1065XYZ9786",
        "nc:IdentificationCategoryDescriptionText": "messageID",
        "nc:IdentificationSourceText": "FilingReview"
      },
      {
        "nc:IdentificationID": "123456ABC",
        "nc:IdentificationCategoryDescriptionText": "filingID",
        "nc:IdentificationSourceText": "FilingReview"
      }
    ]
  }
}
```
```

## Synchronous operation input message

The following JSON fragment illustrates the typical structure for an input message to a synchronous operation.

### JSON Example 2.3: Synchronous operation input message

```
```json
{
  "$schema": "../schema/ecf5.schema.json",
  "@context": "../schema/ecf5.context.jsonld",
  "filinglistrequest:GetFilingListRequestMessage": {
    "nc:DocumentIdentification": [{
      "nc:IdentificationID": "123456ABC",
      "nc:IdentificationCategoryDescriptionText": "messageID",
      "nc:IdentificationSourceText": "FilingAssembly"
    }],
    "ecf:DocumentFiler": [{
      "nc:EntityPerson": {
        "ecf:ParticipantID": {"nc:IdentificationID": "10"}
      }
    }],
    "ecf:SendingMDELocationID": {"nc:IdentificationID": "http://example.com/efsp1"},
    "ecf:ServiceInteractionProfileCode": "urn:oasis:names:tc:legalxml-
courtfiling:schema:xsd:WebServicesMessaging-5.0",
    "j:CaseCourt": {
      "nc:OrganizationIdentification": {"nc:IdentificationID": "10"},
      "nc:OrganizationUnitName": "Municipal Court"
    },
    "nc:DocumentPostDate": {"nc:DateTime": "2008-07-07T13:47:42.0Z"},
    "filinglistrequest:FilingListQueryCriteria": {
      "ecf:CaseTrackingID": [{"nc:IdentificationID": "123456ABC"}],
      "j:CaseNumberText": "123456ABC",
      "nc:DateRange": {
        "nc:StartDate": {"nc:Date": "2008-07-06T00:00:00Z"},
        "nc:EndDate": {"nc:Date": "2008-07-07T00:00:00Z"}
      },
      "nc:DocumentSubmitter": {
        "nc:EntityPerson": {
          "nc:PersonName": {
            "nc:PersonGivenName": "John",
            "nc:PersonMiddleName": "W.",
            "nc:PersonSurName": "Doe"
          }
        }
      }
    }
  }
}
```
```

## Synchronous operation output message

The following JSON fragment illustrates the typical structure for an output message to a synchronous operation.

### JSON Example 2.3: Synchronous operation output message

```
```json
{
  "$schema": "../schema/ecf5.schema.json",
  "@context": "../schema/ecf5.context.jsonld",
  "filinglistresponse:GetFilingListResponseMessage": {
    "nc:DocumentIdentification": [
      {
        "nc:IdentificationID": "10",
        "nc:IdentificationCategoryDescriptionText": "messageID",
        "nc:IdentificationSourceText": "FilingReview"
      },
      {
        "nc:IdentificationID": "123456ABC",
        "nc:IdentificationCategoryDescriptionText": "messageID",
        "nc:IdentificationSourceText": "FilingAssembly"
      }
    ],
    "ecf:DocumentFiler": [{
      "nc:EntityPerson": {
        "@id": "#Person2",
        "ecf:ParticipantID": {"nc:IdentificationID": "10"}
      }
    }],
    "ecf:SendingMDELocationID": {"nc:IdentificationID":
"https://efilingmanager.com:8000"},
    "ecf:ServiceInteractionProfileCode": "urn:oasis:names:tc:legalxml-
courtfiling:schema:xsd:WebServicesMessaging-5.0",
    "j:CaseCourt": {
      "nc:OrganizationIdentification": {"nc:IdentificationID": "01"},
      "j:CourtName": "King County Superior Court"
    },
    "nc:DocumentPostDate": {"nc:DateTime": "2007-06-06T14:20:47.0Z"},
    "ecf:MatchingFiling": [{
      "nc:DocumentIdentification": [{"nc:IdentificationID": "10"}],
      "ecf:DocumentFiler": [{
        "nc:EntityPerson": {
          "@id": "#Person2"
        }
      }
    ]],
    "ecf:SendingMDELocationID": {"nc:IdentificationID":
"https://efilingmanager.com:8000"},
    "ecf:ServiceInteractionProfileCode": "urn:oasis:names:tc:legalxml-
courtfiling:schema:xsd:WebServicesMessaging-5.0",
    "j:CaseCourt": {
      "nc:OrganizationIdentification": {"nc:IdentificationID": "01"},
      "j:CourtName": "King County Superior Court"
    },
    "nc:DocumentPostDate": {"nc:DateTime": "2007-06-06T14:20:47.0Z"},
    "ecf:ConnectedDocumentReview": [{
      "ecf:DocumentReviewDisposition": {
        "ecf:DocumentReviewStatus": {
          "nc:StatusText": "accepted",
          "nc:StatusDate": {"nc:DateTime": "2008-07-07T13:47:42.0Z"}
        }
      }
    ]
  }
}
```

```

    }
  },
  "ecf:ReviewedDocument": {
    "@id": "#Document2",
    "nc:metadataRef": { "$ref": ["#Document1Metadata"] },
    "nc:DocumentCategoryText": ["Information"],
    "nc:DocumentSoftwareName": "Microsoft Word",
    "nc:DocumentDescriptionText": "Petition",
    "nc:DocumentEffectiveDate": {"nc:Date": "2008-07-07T00:00:00Z"},
    "nc:DocumentFileControlID": "1",
    "nc:DocumentIdentification": [{"nc:IdentificationID": "1"}],
    "nc:DocumentSequenceID": "1",
    "nc:DocumentSubmitter": {
      "nc:EntityPerson": {
        "@id": "#Person2"
      }
    }
  },
  "ecf:DocumentFiler": [{
    "nc:EntityPerson": {
      "@id": "#Person2"
    }
  }],
  "ecf:DocumentRendition": [{
    "nc:DocumentIdentification": [],
    "nc:Attachment": {
      "@id": "#Attachment2",
      "nc:BinaryDescriptionText": "Information",
      "nc:BinaryFormatText": "application/pdf",
      "nc:BinaryURI": "cid://Payload2",
      "nc:BinarySizeValue": 32000
    }
  }],
  "ecf:RedactionRequiredIndicator": false,
  "ecf:RegisterActionDescriptionCode": "Information",
  "ecf:SpecialHandlingInstructionsText": "Please verify TCN",
  "nc:DocumentAssociation": {
    "nc:PrimaryDocument": {
      "@id": "#Document1"
    }
  },
  "ecf:DocumentRelatedCode": "parent"
}
}],
"ecf:FilingStatus": {"ecf:FilingReviewStatusCode": "accepted"},
"ecf:LeadDocumentReview": [{
  "ecf:DocumentReviewDisposition": {
    "ecf:DocumentReviewStatus": {
      "nc:StatusText": "accepted",
      "nc:StatusDate": {"nc:DateTime": "2008-07-07T13:47:42.0Z"}
    }
  }
}],
"ecf:ReviewedDocument": {
  "@id": "#Document1",
  "nc:metadataRef": { "$ref": ["#Document1Metadata"] },
  "nc:DocumentCategoryText": ["Apperance"],
  "nc:DocumentSoftwareName": "Microsoft Word",
  "nc:DocumentDescriptionText": "Appearance",
  "nc:DocumentEffectiveDate": {"nc:Date": "2008-07-07T00:00:00Z"},

```

```

"nc:DocumentFileControlID": "2",
"nc:DocumentIdentification": [{"nc:IdentificationID": "2"}],
"nc:DocumentSequenceID": "2",
"nc:DocumentSubmitter": {
  "nc:EntityPerson": {
    "@id": "#Person2"
  }
},
{
  "ecf:DocumentFiler": [{
    "nc:EntityPerson": {
      "@id": "#Person2"
    }
  }],
  "ecf:DocumentRendition": [{
    "nc:DocumentIdentification": [],
    "nc:Attachment": {
      "@id": "#Attachment1",
      "nc:BinaryDescriptionText": "Appearance",
      "nc:BinaryFormatText": "application/pdf",
      "nc:BinaryURI": "cid://Payload1",
      "nc:BinarySizeValue": 32000
    }
  }],
  "ecf:RedactionRequiredIndicator": false,
  "ecf:RegisterActionDescriptionCode": "Apperance",
  "ecf:SpecialHandlingInstructionsText": "Please verify TCN",
  "nc:Metadata": [{
    "@id": "#Document1Metadata",
    "nc:SensitivityText": "public",
    "nc:LanguageCode": "eng"
  }
]
},
}],
"nc:Case": {
  "nc:CaseTitleText": "Jane Doe vs. John Doe ",
  "j:CaseCourt": {
    "nc:OrganizationIdentification": {"nc:IdentificationID": "1"},
    "j:CourtName": "King County Circuit Court"
  },
  "j:CaseOfficial": [{
    "nc:RoleOfPerson": {
      "@id": "#Person3",
      "nc:PersonName": {"nc:PersonFullName": "Jane, Doe, JD"},
      "ecf:ParticipantID": {"nc:IdentificationID": "30"}
    },
    "j:JudicialOfficialBarMembership": {
      "j:JudicialOfficialBarIdentification": {"nc:IdentificationID": "100"}
    }
  }],
  "ecf:CaseCategoryCode": "Estate",
  "ecf:CaseParty": [
    {
      "nc:EntityPerson": {
        "@id": "#Person1",

```

```

"nc:PersonBirthDate": {"nc:Date": "1983-01-01T00:00:00Z"},
"nc:PersonName": {
  "nc:PersonGivenName": "John",
  "nc:PersonMiddleName": "W.",
  "nc:PersonSurName": "Doe"
},
"nc:PersonOtherIdentification": [{
  "nc:IdentificationID": "1234-56-789",
  "ecf:PersonIdentificationCategoryCode": "DriverLicense",
  "nc:IdentificationSourceText": "source"
}],
"nc:PersonRaceText": "W",
"j:PersonSexCode": "M",
"nc:PersonTaxIdentification": {"nc:IdentificationID": "123-45-
67890"},
  "ecf:CaseParticipantRoleCode": ["Defendant"],
  "ecf:ParticipantID": {"nc:IdentificationID": "10"},
  "nc:ContactInformation": [{
    "nc:ContactMailingAddress": [{
      "nc:LocationStreet": {"nc:StreetFullText": "123 Main St."},
      "nc:LocationCityName": "Anytown",
      "nc:LocationState": {"nc:LocationStateName": "IN"},
      "nc:LocationPostalCode": "12345"
    }]
  }]
}
},
{
  "nc:EntityPerson": {
    "@id": "#Person4",
    "nc:PersonBirthDate": {"nc:Date": "1983-01-01T00:00:00Z"},
    "nc:PersonName": {
      "nc:PersonGivenName": "Jane",
      "nc:PersonMiddleName": "Q",
      "nc:PersonSurName": "Doe",
      "nc:PersonMaidenName": "Smith"
    },
    "nc:PersonOtherIdentification": [{
      "nc:IdentificationID": "1234-56-789",
      "ecf:PersonIdentificationCategoryCode": "DriverLicense",
      "nc:IdentificationSourceText": "source"
    }],
    "j:PersonSexCode": "M",
    "nc:PersonTaxIdentification": {"nc:IdentificationID": "123-45-
67890"},
    "ecf:CaseParticipantRoleCode": ["Defendant"],
    "ecf:ParticipantID": {"nc:IdentificationID": "20"},
    "nc:ContactInformation": [{
      "nc:ContactMailingAddress": [{
        "nc:LocationStreet": {"nc:StreetFullText": "123 Main St."},
        "nc:LocationCityName": "Anytown",
        "nc:LocationState": {"nc:LocationStateName": "IN"},
        "nc:LocationPostalCode": "12345"
      }]
    }]
  }
}
}

```

```

],
"ecf:CaseTrackingID": [{"nc:IdentificationID": "123456ABC"}],
"j:CaseNumberText": "123456ABC",
"nc:Metadata": [{
  "@id": "#Metadata1",
  "nc:SensitivityText": "public",
  "nc:LanguageCode": "eng"
}],
"nc:PersonAssociation": [{
  "nc:Person": [
    {
      "@id": "#Person1"
    },
    {
      "@id": "#Person2"
    }
  ],
  "ecf:EntityAssociationTypeCode": "spouse"
}],
"civil:AmountInControversy": {"nc:Amount": 100},
"civil:CivilClassActionIndicator": false,
"civil:DecedentEstateCase": {
  "civil:Decedent": {
    "nc:PersonBirthDate": {"nc:Date": "1900-01-01T00:00:00Z"},
    "nc:PersonName": {
      "nc:PersonGivenName": "Janice",
      "nc:PersonMiddleName": "H",
      "nc:PersonFullName": "Doe"
    }
  },
  "nc:PersonDeathDate": {"nc:Date": "2017-01-01T00:00:00Z"}
},
"civil:JurisdictionalGroundsCode": "Statewide",
"civil:JuryDemandIndicator": false,
"civil:ReliefTypeCode": ["Monetary"],
"ecf:CauseOfActionCode": "100"
}
}]
}
}
...

```

Appendix 6 Example Instances

(This appendix does not form an integral part of this document and is informational.)

Example instances of each ECF 5.0 JSON message are provided in the [json/examples/](#) folder.

Example Messages

Examples of each JSON message are listed below.

TABLE 4. EXAMPLE MESSAGES

ECF message	Example JSON instance(s)
allocatedate:AllocateCourtDateMessage	allocatedate.json
cancel:CancelFilingMessage	cancel.json
caselistrequest:GetCaseListRequestMessage	caselistrequest.json
caselistresponse:GetCaseListResponseMessage	caselistresponse.json
caserequest:GetCaseRequestMessage	caserequest.json
caseresponse:GetCaseResponseMessage	caseresponse.json
cbrn:MessageStatus	messagestatus.json
datecallback:NotifyCourtDateMessage	datecallback.json
docket:RecordDocketingMessage	docket.json
docketcallback:NotifyDocketingCompleteMessage	docketcallback.json
documentrequest:GetDocumentRequestMessage	documentrequest.json
documentresponse:GetDocumentResponseMessage	documentresponse.json
feesrequest:GetFeesCalculationRequestMessage	feesrequest.json
feesresponse:GetFeesCalculationResponseMessage	feesresponse.json
filing:FilingMessage	civil.json
filinglistrequest:GetFilingListRequestMessage	filinglistrequest.json
filinglistresponse:GetFilingListResponseMessage	filinglistresponse.json
filingstatusrequest:GetFilingStatusRequestMessage	filingstatusrequest.json
filingstatusresponse:GetFilingStatusResponse	filingstatusresponse.json
payment:PaymentMessage	payment.json
policyrequest:GetPolicyRequestMessage	policyrequest.json
policyresponse:GetPolicyResponseMessage	policyresponse.json
reservedate:ReserveCourtDateMessage	reservedate.json
reviewfilingcallback:NotifyFilingReviewCompleteMessage	reviewfilingcallback.json
schedulesrequest:GetCourtScheduleRequestMessage	schedulesrequest.json
schedulesresponse:GetCourtScheduleResponseMessage	schedulesresponse.json
serveprocess:ServeProcessMessage	serveprocess.json

ECF message	Example JSON instance(s)
<u>serviceinformationrequest:GetServiceInformationRequestMessage</u>	<u>serviceinformationrequest.json</u>
<u>serviceinformationresponse:GetServiceInformationResponseMessage</u>	<u>serviceinformationresponse.json</u>
<u>stampinformation:DocumentStampInformationMessage</u>	<u>stampinformation.json</u>
<u>stampinformationcallback:NotifyDocumentStampInformationMessage</u>	<u>stampinformationcallback.json</u>
