

XML Standards Development Project Electronic Court Filing Query and Response Standard [Draft]

Document Number

Current Version

October 22, 2002

Previous Version(s)

August 20, 2002

August 5, 2002

July 1, 2002

February 10, 2002

December 10, 2001

November 30, 2001

Workgroup Information

Workgroup Name: OASIS LegalXML Court Filing Technical Committee

Workgroup Co-Chairs: John Greacen, Mary Campbell McQueen

Sub-Workgroup Name: LegalXML CMS API Subgroup

Sub-Workgroup Chair: Moira Rowley

Workgroup Mailing List: legalxml-courtfiling-cms-api@lists.oasis-open.org

Workgroup Mailing List Archive:

Document Author(s)

Shane Durham (shane.durham@lexisnexis.com)

Contributing Author(s)

Dwight R. Daniels (drdaniels@kpmg.com)

Marty Halvorson (martyh@nmcourts.com)

Document Editor(s)

Roger Winters (roger.winters@metrokc.gov)

Short Statement of Status

Draft standard for approval by Technical Committee workgroup.

Abstract

This Draft Standard provides the XML DTD required for Court Filing Query and Response.

Status of Document

This is a Court Filing Technical Committee Draft Standard for review.

Abstract	1
Status of Document.....	1
1 Introduction	4
1.1 Conventions	4
1.2 Document Description	5
1.3 Assumptions and Requirements	5
1.4 Terminology.....	5
1.5 Date and Time Format	5
1.6 White Space Treatment.....	5
1.7 Extensions	5
2 The Document Type Definitions	6
2.1 Query	6
3 Element Specification	8
3.1 query.....	8
3.1.1 authentication.....	8
3.1.2 queryName.....	8
3.1.3 queryDescription.....	8
3.1.4 inputParameters	8
3.1.5 queryIdentification	9
3.2 response.....	10
3.2.1 respondsTo	10
3.2.2 responseRow.....	10
3.2.3 errorMessage.....	10
3.3 New CMS Entities	10
3.3.1 CaseActorStatus.....	10
3.3.2 docketEntry	10
4 The Standard Queries	12
4.1 getCaseActorList.....	12
4.2 getCaseList.....	12
4.3 getCaseCalendar.....	12
4.4 getCaseDocument.....	13
4.5 getCaseHistory	13
4.6 getCaseInformation.....	13
4.7 Special Concepts	13
4.7.1 ActorIdentifier	13
4.7.2 Identification of Cases	14
4.8 Summary Table of Standard Queries	14
5 Examples.....	16
5.1 getCaseActorList.....	16
5.1.1 Query.....	16
5.1.2 Response	16
5.2 getCaseList.....	17
5.2.1 Query.....	17
5.2.2 Response	17

5.3	getCaseCalendar.....	18
5.3.1	Query.....	18
5.3.2	Response	18
5.4	getCaseDocument.....	19
5.4.1	Query.....	19
5.4.2	Response	19
5.5	getCaseHistory	20
5.5.1	Query.....	20
5.5.2	Response	20
5.6	getCaseInformation.....	20
5.6.1	Query.....	20
5.6.2	Response	21
6	Revision History.....	22

1 Introduction

This document is a Draft Standard developed by the OASIS Legal XML Court Filing Technical Committee workgroup. It is intended to describe the metadata that would be required for electronic retrieval of information available from a court that complies with this standard and to detail the structure that information would have. No information regarding the content of the information returned is included in the scope of this standard other than that which is required to accomplish the task.

The Query and Response DTD is meant to be generic and flexible. A court using it may implement any query it agrees to support¹. With the DTD, the Court Filing Technical Committee has described a set of standard queries that it highly recommends courts support to facilitate electronic filing.² These queries are discussed in §4 and form the basis of the examples in §5 of this document.

This specification is the product of a consensus process. The workgroup received valuable input on many items, from participants representing multiple viewpoints. The positions and views were often not identical. When discussed items needed to be closed, this was usually done when the question “Is there anyone who cannot live with this?” met with silence. On some occasions, decisions were made based on an overwhelming majority.

1.1 Conventions

Within this document the terms “shall” and “must” are used to describe mandatory items. The term “may” is used to describe optional items.

This draft standard conforms to the XML 1.0 Specification (<http://www.w3.org/TR/REC-xml.html>).

Courier New font is used for the Document Type Definition or portions thereof.

Ariel font is used for elements or attributes from a DTD when referred to in the body of the text.

“Times New Roman” font set in quotation marks and italicized is used to indicate a non-literal textual representation, e.g. of a transmitted file.

¹ It is anticipated that a court will express and publish the queries it supports in its Court Policy XML, for which a standard is yet to be defined. A proposed approach for expressing query definitions has been submitted to the LegalXML policy workgroup.

² These are listed in *EFM-CMS Interface Requirements*, v. 7, §6.1, though the list is based on an earlier version of this Query and Response specification. It should be noted that the *EFM-CMS Interface Requirements* document is *not* a requirements document for the Query and Response DTD. On Query and Response, it offers “a preliminary attempt to identify predefined, or normative, queries” and refers to those that it identifies as “candidate predefined queries.” Therefore, while cognizant of the *EFM-CMS Interface Requirements* document, this specification does not feel obligated to adhere to all of its statements regarding Query and Response.

1.2 Document Description

This document includes a DTD that is to be used to validate the syntax of XML documents used to retrieve information from a court. It also contains a DTD that is to be used to validate the syntax of XML documents used to return information to an inquirer. Any annotations appearing inside a DTD, which add further definition and specification, shall be binding.

The examples provided in this document are non-normative. Where there is a conflict between an example and the DTDs or the body of this document, or between the body of this document and the underlying DTD, the DTD shall be considered normative and ruling.

1.3 Assumptions and Requirements

All assumptions and requirements from Court Filing apply.

1.4 Terminology

All terms defined in Court Filing apply.

1.5 Date and Time Format

All date and time formats from Court Filing apply.

1.6 White Space Treatment

It is often convenient to use “white space” (spaces, tabs, and blank lines) to set apart the markup for greater readability.

Court Filing Query and Response XML processors may:

1. Discard leading and trailing white space contained within any element content returned to the sender in a response message.
2. Convert strings of white space characters into a single space character (#x20) contained within any element or attribute content returned to the sender in a response message.

It is expected that Court Filing XML processors shall discard leading and trailing white space contained within any element or attribute content returned to the sender in a response message.

1.7 Extensions

Extension rules from Court Filing apply.

2 The Document Type Definitions

The Document Type Definitions that follow do not contain any content defined in Court Filing apart from the **query** and **response** elements, as well as the **authentication** element and its children. In Court Filing, the **query** and **response** elements have a content model of **ANY**. This is a placeholder content model, and the query and response DTDs in this document can be used to replace the content model for both of these elements.

2.1 Query

```
<!-- This portion of the DTD defines the basic LegalXML
      query/response syntax -->
<!ELEMENT query (authentication?, queryName, queryDescription?,
      inputParameters, queryIdentification?)>

<!ELEMENT queryName (#PCDATA)>
<!ELEMENT queryDescription (#PCDATA)>
<!ELEMENT queryIdentification (#PCDATA)>

<!ELEMENT inputParameters (parameter*)>
<!ELEMENT parameter (parameterName, parameterDescription?,
      parameterValue)>
<!ELEMENT parameterName (#PCDATA)>
<!ELEMENT parameterDescription (#PCDATA)>
<!ELEMENT parameterValue (#PCDATA)>

<!ELEMENT response (respondsTo, (responseRow* | errorMessage))>
<!ELEMENT respondsTo (#PCDATA)>
<!ELEMENT responseRow ANY>
<!ELEMENT errorMessage (#PCDATA)>

<!-- This portion of the DTD define new compound CMS structures that
      do not yet exist in a prior dictionary (ie CourtFiling). They
      are specifically related to CMS data and the LegalXML
      'standard' query set. -->
<!ELEMENT caseActorStatus (#PCDATA)>

<!ELEMENT docketEntry (dateTime, actor*, docketType?, docketText,
      docketDocument?)>
```

Query and Response Standard [Draft]

```
<!ELEMENT docketDocument ( (courtDocumentReference |  
    documentContent), docketDocument* )>
```

```
<!ELEMENT docketType (#PCDATA)>
```

```
<!ELEMENT docketText (#PCDATA)>
```

3 Element Specification

3.1 query

```
<!ELEMENT query (authentication?, queryName,  
  queryDescription?, inputParameters, queryIdentification?)>
```

query provides a mechanism for submitting requests for information to a court. Within the elements available for a query, a requestor may specify their identity (authentication), the query it is executing, and the arguments for that query.

3.1.1 authentication

```
<!ELEMENT authentication ((userIdentification, password?) |  
  signature)>
```

On authentication and its child elements, see the Court Filing specification.

3.1.2 queryName

```
<!ELEMENT queryName (#PCDATA)>
```

The **queryName** element identifies the query that is being submitted. The data submitted in the **queryName** element must provide an exact match of one of the court's supported queries. If the court does not support the query, an **errorMessage** shall be returned (see §3.2 below).

3.1.3 queryDescription

```
<!ELEMENT queryDescription (#PCDATA)>
```

The **queryDescription** is used to provide an explanation of the purpose of the query and what information it provides to the user. The **queryDescription** is intended to assist with human readability and it is not necessary to include with a query. If it is submitted, the queried application may ignore it.

3.1.4 inputParameters

```
<!ELEMENT inputParameters (parameter*)>
```

The **inputParameters** element contains the parameters needed to fulfill a query.

3.1.4.1 parameter

```
<!ELEMENT parameter (parameterName, parameterDescription?,  
    parameterValue)>
```

In a query, the **parameter** element is used to convey the individual data parameters, if any, being submitted with the query. Named parameters are supported by the **parameterName** element, and each parameter shall have a unique name in the query. Wherever possible, the **parameterName** should coincide with the established elements and attributes of the LegalXML data dictionary and contain the same information.

```
<!ELEMENT parameterName (#PCDATA)>
```

The **parameterName** element contains the name by which each parameter is known. All parameters within a query shall have unique names.

```
<!ELEMENT parameterDescription (#PCDATA)>
```

The **parameterDescription** element contains provides an explanation of the nature and purpose of the parameter for presentation to the user, e.g., as a mouse over help feature. The **parameterDescription** is intended to be human readable and does not need to be submitted with a query. If it is submitted, the queried application may ignore it.

```
<!ELEMENT parameterValue (#PCDATA)>
```

The **parameterValue** element contains the value of the parameter being submitted with the query.

3.1.5 queryIdentification

```
<!ELEMENT queryIdentification (#PCDATA)>
```

The **queryIdentification** element must contain a value, generated by the submitter, that uniquely identifies the query to the submitter. When provided in a query, this value will be returned in the response to indicate the query to which the response is replying.

3.2 response

```
<!ELEMENT response (respondsTo, (responseRow* | errorMessage))>
```

The **response** element returns information requested by a query. An empty **response** element shall indicate that the submitted query was successfully processed, but that no data was found when the query was executed.

3.2.1 respondsTo

```
<!ELEMENT respondsTo (#PCDATA)>
```

The **respondsTo** element returns the **queryIdentification** value submitted with the query.

3.2.2 responseRow

```
<!ELEMENT responseRow ANY>
```

The **responseRow** element contains the individual “rows” returned by the query.

3.2.3 errorMessage

```
<!ELEMENT errorMessage (#PCDATA)>
```

The **errorMessage** element is used to return information to the requestor when an error condition occurred while processing a query.

3.3 New CMS Entities

3.3.1 CaseActorStatus

```
<!ELEMENT caseActorStatus (#PCDATA)>
```

The **caseActorStatus** element is used to express the status of a case participant (case actor).

3.3.2 docketEntry

```
<!ELEMENT docketEntry (dateTime, actor*, docketType?,  
docketText, docketDocument?)>
```

The **docketEntry** element is used to express an item of case history (aka registry of actions, or case docket).

DateTime allows the court to express the timestamp of when the docket entry was recorded.

Actor allows the court to express actors related to the docket entry, such as filer.

DocketType allows the court to express a docket entry code or categorization.

DocketText allows the court to express the description or the content of the docket entry.

3.3.2.1 docketDocument

```
<!ELEMENT docketDocument ( (courtDocumentReference |  
documentContent), docketDocument* )>
```

docketDocument allows the court to optionally express a document related to a docket entry. It is a recursive element (containing a child **docketDocument** list) to allow the court to express attachments to the lead **docketDocument**.

CourtDocumentReference allows the court to express a reference to a document that can be retrieved via url or subsequent query. Alternatively, **DocumentContent** can be used by the court to directly express the content of a document associated with the docket entry.

4 The Standard Queries

The Court Filing Technical Committee and its subcommittees have identified a list of standard queries that it highly recommends courts support to facilitate the electronic filing of documents. Given the differences in the data structures underlying the various case management systems utilized by courts, it is not possible to define normative input parameters and return values for the standard queries with exact precision. Different case management systems will require different input parameters to process a given query. Similarly, depending on the extent and nature of the data maintained in the different case management systems, as well as divergent court policies concerning the electronic dissemination of information, different courts may elect to return different information in response to the same query.

Nevertheless, these queries are intended to provide a degree of standardization so that practitioners will know which query to submit to obtain a desired piece of information. To accomplish this, a standard set of input parameters and response elements is defined for each of the standard queries, some of which are required and some of which are optional. Required elements have been limited to those that are inherent in the definition of the query and its purpose. Input parameters that are listed as optional in this specification may be defined as required by a given court in its Court Policy XML owing to the nature of its case management system or its policy concerning the dissemination of electronic information. Therefore, applications that support this specification must be capable of handling any and all of the input parameters and response elements listed for each of the standard queries, regardless of the specifics of a particular implementation. Courts are also free to return more information than is listed in the standard response elements. Wherever possible, the additional information should be returned using the established elements and attributes of the LegalXML data dictionary intended to convey that information.

4.1 getCaseActorList

The `getCaseActorList` query is used to retrieve the **name** and **role** of actors involved in the specified case, e.g., a party or an attorney, and the status of the actor on the case, e.g. active or inactive, and/or an identifier of that actor within a court's case management system.

4.2 getCaseList

The `getCaseList` query is used to retrieve the **fullCaseNumber** and other information pertaining to the cases in which an actor is involved.

4.3 getCaseCalendar

The `getCaseCalendar` query is used to retrieve the events scheduled by the court regarding a case, e.g., hearing dates or submission deadlines. This will be comprised of the **date**, **time**, **courtName**, **courtEventType** and, where appropriate, **courtEventReason** of the event.

4.4 **getCaseDocument**

The `getCaseDocument` query is used to retrieve a single lead document and any of its attachments. The court may return the requested `documentContent` in a format it supports or it may choose to return only a hyperlink to the document. If the court's practice is to supply hyperlinks, the requestor must then use the hyperlink to retrieve a copy of the document.

4.5 **getCaseHistory**

The `getCaseHistory` query is used to retrieve the contents of the case "docket," i.e., the recorded history of actions in a particular case. Each action will entail the `docketEntry` and the `date` on which it was made. The court may optionally include a `courtDocumentReference`, or `documentContent`, if applicable.

4.6 **getCaseInformation**

The `getCaseInformation` query is used to retrieve information about a case, e.g., the `caseTitle` (also known as the case's "caption"), the `caseCategory` (also called case "type"), the `caseYear`, and/or any applicable `lineageCaseNumber` values.³

This query differs from `getCaseList`, in that `getCaseInformation` represents a request and response for a specific case that the requestor believes to exist, whereas `getCaseList` is a search for a *possible* list of cases that may or may not exist.

A court may wish to populate the `caseInformation` response to `getCaseInformation` a little more robustly than the same `caseInformation` as a response to `getCaseList`. In the search context, a lightly populated list of case numbers and titles might be appropriate and desirable whereas in the more specific request context, the court may wish to provide additional case details.

4.7 **Special Concepts**

4.7.1 **ActorIdentifier**

In the standard queries, a concept of `actorIdentifier` has been introduced. As a query parameter, an `actorIdentifier` expresses a previously obtained identifier of an actor in the case management system. This may be either an identifier of the actor on the case, e.g., a case participant ID, or a unique identifier of the actor within the case management system as a whole. To express an `actorIdentifier` in a query response, the CMS populates the `actor id` attribute of `actor`.

³ A lineage case number is a previous case number. It may be used, for example, to supply the lower court case number of a case on appeal.

4.7.2 Identification of Cases

Except for the `getCaseList` and `getDocument` queries, each query has parameters to express the identification of a case. These parameters include `fullCaseNumber` and, optionally, `name` or `actorIdentifier`, to identify a participant of the case. The additional optional actor parameters are intended to assist with case identification to support those courts or case management systems that wish to require this additional criteria to identify and retrieve case information.

These additional actor-related arguments should not be used to restrict any actors within the query response. The actor-related arguments are to help identify a case; they are not intended as a filter of case participants within the case. If a participant filter is desirable, the court is encouraged to define and accept additional query parameters for that purpose.

4.8 Summary Table of Standard Queries

The following table summarizes the standard queries, their input parameters, and the corresponding responses.

Query and Response Standard [Draft]

Query	Input Parameters		Response	
	Element Name	Req	Element Name	Required Sub-Elements or Attributes
GetCaseActorList	fullCaseNumber lastName OR entityName firstName middleName actorIdentifier	Y N N N N	caseActorStatus actor	caseActorStatus actor name personName / lastName (or) agencyID / entityName (or) organizationID / entityName role roleName
GetCaseList	LastName OR entityName firstName middleName actorIdentifier	Y N N N	caseInformation	caseInformation fullCaseNumber
GetCaseCalendar	fullCaseNumber lastName OR entityName firstName middleName actorIdentifier	Y N N N N	courtEvent	courtEvent courtEventType courtEventReason courtEventSession dateTime date time ** actor ** courtInformation courtName courtType ** physicalLocation **
GetCaseDocument	CourtDocumentReference	Y	documentContent	documentContent
GetCaseHistory	fullCaseNumber lastName OR entityName firstName middleName actorIdentifier	Y N N N N	docketEntry	docketEntry dateTime date time ** docketText
GetCaseInformation	fullCaseNumber lastName OR entityName firstName middleName actorIdentifier	Y N N N N	caseInformation	caseInformation fullCaseNumber

5 Examples

In this section, an example is given for each of the standard queries. These examples are offered to illustrate the usage of the Query and Response DTDs and of the standard query set. These examples do not include the standard legalEnvelope header information that is to be included with any LegalXML message (see LegalXML CourtFiling).

5.1 getCaseActorList

5.1.1 Query

```
<query>
  <queryName>getCaseActorList</queryName>
  <inputParameters>
    <parameter>
      <parameterName>fullCaseNumber</parameterName>
      <parameterValue>02F12345</parameterValue>
    </parameter>
  </inputParameters>
</query>
```

5.1.2 Response

```
<response>
  <responseRow>
    <caseActorStatus>Active</caseActorStatus>
    <actor>
      <name>
        <personName>
          <firstName>Arthur</firstName><lastName>Perkins</lastName>
        </personName>
      </name>
      <role><roleName>Plaintiff</roleName></role>
    </actor>
  </responseRow>
  <responseRow>
    <caseActorStatus>Active</caseActorStatus>
```



```
<actor>
  <name>
    <personName>
      <firstName>Maria</firstName><lastName>Perkins</lastName>
    </personName>
  </name>
  <role><roleName>Defendant</roleName></role>
</actor>
</responseRow>
</response>
```

5.2 getCaseList

5.2.1 Query

```
<query>
  <queryName>getCaseList</queryName>
  <inputParameters>
    <parameter>
      <parameterName>entityName</parameterName>
      <parameterValue>Easy Credit Agency</parameterValue>
    </parameter>
  </inputParameters>
</query>
```

5.2.2 Response

```
<response>
  <responseRow>
    <caseInformation>
      <fullCaseNumber>99SC09876</fullCaseNumber>
      <caseTitle>Smith vs Easy Credit Agency</caseTitle>
      <caseYear>1999</caseYear>
    </caseInformation>
  </responseRow>
```

```
<responseRow>
  <caseInformation>
    <fullCaseNumber>02F12345</fullCaseNumber>
    <caseTitle>People vs Easy Credit Agency</caseTitle>
    <caseYear>2002</caseYear>
  </caseInformation>
</responseRow>
</response>
```

5.3 getCaseCalendar

5.3.1 Query

```
<query>
  <queryName>getCaseCalendar</queryName>
  <inputParameters>
    <parameter>
      <parameterName>fullCaseNumber</parameterName>
      <parameterValue>02F12345</parameterValue>
    </parameter>
  </inputParameters>
</query>
```

5.3.2 Response

```
<response>
  <responseRow>
    <courtEvent>
      <courtEventType>PTHEAR</courtEventType>
      <courtEventReason>Pre-Trial Hearing</courtEventReason>
      <courtEventSession>
        <dateTime><date>2002-07-15</date><time/></dateTime>
        <actor/>
      </courtEventSession>
      <courtInformation>
        <courtName>Nearby Justice Center</courtName>
      </courtInformation>
    </courtEvent>
  </responseRow>
</response>
```

```
        <courtType/><physicalLocation/>
      </courtEventSession>
    </courtEvent>
  </responseRow>
  <responseRow>
    ...
  </responseRow>
</response>
```

5.4 getCaseDocument

5.4.1 Query

```
<query>
  <queryName>getCaseDocument</queryName>
  <inputParameters>
    <parameter>
      <parameterName>documentReference</parameterName>
      <parameterValue>238491</parameterValue>
    </parameter>
  </inputParameters>
</query>
```

5.4.2 Response

```
<response>
  <responseRow>
    <documentContent mimeType = "application/pdf"> "a pdf document"
  </documentContent>
  </responseRow>
</response>
```

5.5 getCaseHistory

5.5.1 Query

```
<query>
  <queryName>getCaseHistory</queryName>
  <inputParameters>
    <parameter>
      <parameterName>fullCaseNumber</parameterName>
      <parameterValue>02F12345</parameterValue>
    </parameter>
  </inputParameters>
</query>
```

5.5.2 Response

```
<response>
  <responseRow>
    <docketEntry>
      <dateTime><date>2002-02-11</date><time/></dateTime>
      <docketEntryText>case initiated.</docketEntryText>
    </docketEntry>
  </responseRow>
  <responseRow>
    <docketEntry>
      <dateTime><date>2002-02-11</date><time/></dateTime>
      <docketEntryText>Complaint filed.</docketEntryText>
    </docketEntry>
  </responseRow>
</response>
```

5.6 getCaseInformation

5.6.1 Query

```
<query>
  <queryName>getCaseInformation</queryName>
```

```
<inputParameters>
  <parameter>
    <parameterName>fullCaseNumber</parameterName>
    <parameterValue>99SC09876</parameterValue>
  </parameter>
</inputParameters>
</query>
```

5.6.2 Response

```
<response>
  <responseRow>
    <caseInformation>
      <fullCaseNumber>99SC09876</fullCaseNumber>
      <caseTitle>Smith vs Easy Credit Agency</caseTitle>
      <caseCategory>Small Claims</caseCategory>
      <caseYear>1999</caseYear>
    </caseInformation>
  </responseRow>
</response>
```

6 Revision History

Date	Editor	Major Change(s)
1/7/02	Dwight R. Daniels	Major revision of entire document. Change of author to Dwight R. Daniels.
1/24/02	Roger Winters, Marty Halvorson	Editing of revised document.
1/28/02	Dwight R. Daniels	Incorporation of comments from 1/24/02 editing; further corrections and restructuring of document; addition of Query and Response Summary Table.
2/10/02	Dwight R. Daniels	Change of eventType to courtEventType; modification of dateFilter content model. Other minor revisions and release for comment.
7/1/02	Dwight R. Daniels	Fundamental change in approach and major revision of entire document.
8/5/02	Dwight R. Daniels	Normalized the standard queries and placed them within their own section of the document.
8/20/02	Dwight R. Daniels	Added authentication, queryIdentification and respondsTo elements. Added label attribute to parameterName element. Renamed getActorRole query to getCaseActorRole and revised input parameters and response elements to support getting a list of actors on a case. Added actorIdentifier element to standard queries with name parameters. Eliminated getCourtPolicy from list of standard queries.
10/22/2002	Shane Durham	<p>Removed material describing or implementing proposed Query Policy DTD. (to be included in separate policy proposal document.) Removed references to privilege level (to be included as example of how a court might assist an external system with managing security.)</p> <p>Revised standard query responses to reuse compound data structures previously defined for LegalXML Filing. Translated the previously identified list of required response elements into a list of required sub-elements of data structures.</p> <p>Renamed GetActorRole to GetCaseActorList for retrieval of a case's participants.</p> <p>Added formal structure for CaseActorStatus to supplement response to getCaseActorList. Added formal structural definition of docketEntry.</p>