

XML Standards Development Project Proposal for Query and Response Policy

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)

Previous 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

Proposal for consideration by Technical Committee workgroup.

Abstract

This document describes a proposal to express Query and Response Policy, the court policy for LegalXML Query and Response system.

Status of Document

This is a Court Filing Technical Committee Proposal for consideration.

Abstract	1
Status of Document.....	1
1 Introduction	3
1.1 Conventions	3
1.2 Document Description	3
1.3 Assumptions and Requirements	4
1.4 Terminology.....	4
1.5 Date and Time Format	4
1.6 White Space Treatment.....	4
1.7 Extensions	4
2 The Document Type Definitions	5
3 Element Specification	6
3.1 supportedQueries	6
3.2 query.....	6
3.2.1 queryName.....	6
3.2.2 queryDescription.....	6
3.2.3 queryParameters	6
3.3 responseRecord	8
3.3.1 responseElement.....	8
4 Examples.....	9
4.1 getCaseActorList.....	9
4.2 getCaseList.....	10
4.3 getCaseCalendar.....	11
4.4 getCaseDocument.....	12
4.5 getCaseHistory	12
4.6 getCaseInformation.....	13
5 Revision History.....	15

1 Introduction

This Query and Response Policy proposal enables an organization to define the queries¹ it wishes to support.

It describes an XML solution for expressing extensions and restrictions to the LegalXML Query and Response standard. It enables a court to publish its supported query set in a manner that may be interpreted by remote systems, such that they can automatically present users with the court's query options.

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.

Finally, this proposal represents work almost entirely extracted from previous drafts of LegalXML Query and Response.

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 proposal 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.

`Arial` 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.

1.2 Document Description

This document includes a DTD that is to be used to validate the syntax of XML documents describing query and response policy. 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 The concept of Case Management System Data Configuration XML, or CDC XML, has also been introduced into the discussion and suggested as the location where a court would define its supported queries. The relationship between CDC XML and Court Policy XML has not been finalized, but it seems likely that CDC XML will be one segment within the larger Court Policy XML. For this reason, this specification refers to Court Policy XML without thereby intending to indicate a specific, or even preferred, resolution to the issue. On CDC XML see *EFM-CMS Interface Requirements*, v. 7, §1.

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 Policy 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

```
<!ELEMENT supportedQueries (supportedQuery*)>
<!ELEMENT supportedQuery (query, responseRecord)>

<!ELEMENT query (queryName, queryDescription?, queryParameters)
<!ELEMENT queryName (#PCDATA)>
<!ELEMENT queryDescription (#PCDATA)>
<!ELEMENT queryParameters (parameter*)>
<!ELEMENT parameter (parameterName, parameterDescription?,
parameterDatatype)>
<!ELEMENT parameterName (#PCDATA)>
    <!ATTLIST parameterName required (yes | no) #IMPLIED>
    <!ATTLIST parameterName label #IMPLIED>
<!ELEMENT parameterDescription (#PCDATA)>
<!ELEMENT parameterDatatype (#PCDATA)>

<!ELEMENT responseRecord (responseElement*)>
<!ELEMENT responseElement (elementName, elementDatatype,
    elementAttribute*)>
<!ELEMENT elementName (#PCDATA)>
    <!ATTLIST elementName label CDATA #IMPLIED>
<!ELEMENT elementDatatype (#PCDATA)>
<!ELEMENT elementAttribute (#PCDATA)>
    <!ATTLIST elementAttribute attributeType CDATA #REQUIRED>
    <!ATTLIST elementAttribute attributeValue (implied | required)
        "implied">
```

3 Element Specification

3.1 supportedQueries

```
<!ELEMENT supportedQueries (supportedQuery*)>
<!ELEMENT supportedQuery (query, responseRecord)>
```

The `supportedQueries` and `supportedQuery` element express a list of query definitions supported by a court.

3.2 query

```
<!ELEMENT query (queryName, queryDescription?,
  queryParameters)>
```

`query` provides a mechanism for describing a request for court information. Within the elements a court may specify the queries it will accept, and the required and optional input parameters.

3.2.1 queryName

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

The `queryName` element identifies a supported query. A submitted `queryName` must provide an exact match of one of the supported queries described by this Policy XML. If an exact match is not found, an `errorMessage` shall be returned (see §3.3 below).

3.2.2 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 does not affect the acceptability of a submitted query.

3.2.3 queryParameters

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

The `queryParameters` element contains the `parameters` needed to fulfill a query.

3.2.3.1 parameter

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

In a query, the `parameter` element is used to convey the individual data parameters, if any, that may be 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)>  
    <!ATTLIST parameterName required (yes | no) #IMPLIED>  
    <!ATTLIST parameterName label #IMPLIED>
```

The `parameterName` element contains the name by which each parameter is known. All parameters within a query shall have unique names. The `required` attribute indicates if a parameter must be provided in a submitted query. The `label` attribute contains an alternate designation, e.g., the local vernacular, for the parameter to be used as a label for the input field. The `label` is intended to assist with human readability and does not effect the acceptability of a submitted query.

```
<!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 assist with human readability and does not effect the acceptability of a submitted query.

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

The `parameterDatatype` element specifies the data type of the parameter. Valid `parameterDatatype` values are the built-in datatypes of the XML Schema Part 2: Datatypes recommendation.

3.3 responseRecord

```
<!ELEMENT responseRecord (responseElement*)>
```

The `responseRecord` element describes the information that will be returned when a system processes a given valid query.

3.3.1 responseElement

```
<!ELEMENT responseElement (elementName, elementDatatype?,  
    elementAttribute*)>
```

The `responseElement` describes the individual pieces (ie columns, cells, objects) of the `responseRecord`.

```
<!ELEMENT elementName (#PCDATA)>  
    <!ATTLIST elementName label #IMPLIED>
```

The `elementName` element contains the name by which each `responseElement` is known. All `responseElements` within a query shall have unique names. The `label` is intended to assist with human readability.

The `elementName` should coincide with the established elements and attributes of the LegalXML data dictionary and contain the same information. If the `elementName` does not coorespond with established LegalXML elements, then the following `elementDatatype` and `elementAttribute` should be used to describe the new `responseElement`.

```
<!ELEMENT elementDatatype (#PCDATA)>  
<!ELEMENT elementAttribute (#PCDATA)>  
    <!ATTLIST elementAttribute attributeType CDATA #REQUIRED>  
<!ATTLIST elementAttribute attributeValue (implied | required) "implied">
```

The `elementDatatype` and `elementAttribute` elements optionally express the XML definition of a previously undefined `responseElement`. It should not be included in the policy set if the cooresponding `responseElement` represents data defined elsewhere in the LegalXML dictionary (ie. CourtFiling 1.x, Query and Response 1.x, Document 1.x). These policy elements are to be used when the court needs to define result data not previously described by the LegalXML dictionary. Valid `elementDatatype` values are the built-in datatypes of the XML Schema Part 2: Datatypes recommendation.

4 Examples

In this section, an example of how Query and Response Policy can be used to describe the standard queries defined by the LegalXML Query and Response standard.

By adjusting these sample policies, an organization can express restrictions or extensions to the standard query set. To avoid confusion, query extensions or restrictions should be given new names that do not coincide with the 'standard' set defined by LegalXML Query and Response.

4.1 getCaseActorList

```
<supportedQuery>
  <query>
    <queryName>getCaseActorList</queryName>
    <queryDescription>
      >Use this query to find the list of case participants and their roles.
    </queryDescription>
    <queryParameters>
      <parameter>
        <parameterName required = "yes" label = "Case
          Number">fullCaseNumber</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
      <parameter>
        <parameterName required = "yes" label = "Family
          Name">lastName</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
      <parameter>
        <parameterName required = "no" label = "First
          Name">firstName</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
    </queryParameters>
  </query>
</supportedQuery>
```

```
<responseRecord>
  <responseElement>
    <elementName label = "Party Status">caseActorStatus</elementName>
  </responseElement>
  <responseElement>
    <elementName label = "Party">actor</elementName>
  </responseElement>
</responseRecord>
</supportedQuery>
```

4.2 getCaseList

```
<supportedQuery>
  <query>
    <queryName>getCaseList</queryName>
    <queryDescription> Use this query to get a listing of all currently active cases in
      which the person is involved. </queryDescription>
    <queryParameters>
      <parameter>
        <parameterName required = "yes" label = "Last
          Name">lastName</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
      <parameter>
        <parameterName required = "yes" label = "First
          Name">firstName</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
      <parameter>
        <parameterName required = "no" label = "Middle
          Name">middleName</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
    </queryParameters>
  </query>
</supportedQuery>
```

```
    </queryParameters>
  </query>
  <responseRecord>
    <responseElement>
      <elementName label = "Case">caseInformation</elementName>
    </responseElement>
  </responseRecord>
</supportedQuery>
```

4.3 getCaseCalendar

```
<supportedQuery>
  <query>
    <queryName>getCaseCalendar</queryName>
    <queryDescription> This query will give you a list of all hearings currently scheduled
      for the case. </queryDescription>
    <queryParameters>
      <parameter>
        <parameterName required = "yes" label = "Case
          Number">fullCaseNumber</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
    </queryParameters>
  </query>
  <responseRecord>
    <responseElement>
      <elementName label = "Calendar Event">courtEvent</elementName>
    </responseElement>
  </responseRecord>
</supportedQuery>
```

4.4 **getCaseDocument**

```
<supportedQuery>
  <query>
    <queryName>getCaseDocument</queryName>
    <queryParameters>
      <parameter>
        <parameterName required = "yes">
          courtDocumentReference
        </parameterName>
        <parameterDatatype>integer</parameterDatatype>
      </parameter>
    </queryParameters>
  </query>
  <responseRecord>
    <responseElement>
      <elementName>documentContent</elementName>
    </responseElement>
  </responseRecord>
</supportedQuery>
```

4.5 **getCaseHistory**

```
<supportedQuery>
  <query>
    <queryName>getCaseHistory</queryName>
    <queryDescription>View the Case Registry of Actions</queryDescription>
    <queryParameters>
      <parameter>
        <parameterName required = "yes" label = "Case
          Number">fullCaseNumber</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
    </queryParameters>
  </query>
</supportedQuery>
```

```
</parameter>
<parameter>
  <parameterName required = "yes" label = "Family
    Name">lastName</parameterName>
  <parameterDatatype>string</parameterDatatype>
</parameter>
<parameter>
  <parameterName required = "no" label = "First
    Name">firstName</parameterName>
  <parameterDatatype>string</parameterDatatype>
</parameter>
</queryParameters>
</query>
<responseRecord>
  <responseElement>
    <elementName label = "Case Action">docketEntry</elementName>
  </responseElement>
</responseRecord>
</supportedQuery>
```

4.6 getCaseInformation

```
<supportedQuery>
  <query>
    <queryName>getCaseInformation</queryName>
    <queryDescription> Provides information about the case. </queryDescription>
    <queryParameters>
      <parameter>
        <parameterName required = "yes" label = "Case
          Number">fullCaseNumber</parameterName>
        <parameterDatatype>string</parameterDatatype>
      </parameter>
    </queryParameters>
```

Query and Response Policy [Proposal]

```
</query>  
<responseRecord>  
  <responseElement>  
    <elementName label = "Case Information">caseInformation</elementName>  
  </responseElement>  
</responseRecord>  
</supportedQuery>
```

5 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.
10/22/02	Shane Durham	Extracted policy work from prior draft of Query and Response. Renamed and re-titled document to indicate the document's policy focus. Remove references to authentication and privilege levels to be consistent with current draft of Query and Response specification. Enabled responseElementType to be optional so that court's need not express the data types of compound elements referenced from existing LegalXML standards. Indicated where prior portions of policy DTD had not been defined; took a stab at providing some draft language for some of those undefined elements. Updated example policy set related to standard query set to reflect recent changes to current draft of Query and Response specification.