XML Query Requirements

W3C Working Draft 15 February 2001

This version:
http://www.w3.org/TR/2001/WD-xmlquery-req-20010215
Latest version:
http://www.w3.org/TR/xmlquery-req
Previous version:
http://www.w3.org/TR/2000/WD-xmlquery-req-20000815
Editors:
Don Chamberlin (IBM Almaden Research Center) <chamberlin@almaden.ibm.com >
Peter Fankhauser (GMD-IPSI) <fankhaus@darmstadt.gmd.de>
Massimo Marchiori (W3C/MIT/UNIVE) <massimo@w3.org>
Jonathan Robie (Software AG) <jonathan.robie@SoftwareAG-USA.com>

Abstract

This document specifies goals, requirements, and usage scenarios for the W3C XML Query data model, algebra, and query language.

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C. This document is a revision of the first public XML Query Requirements working draft that takes into account comments processed up to June 10, 2000.

This is a W3C Working Draft for review by W3C Members and other interested parties. It is a draft document and may be updated, replaced or made obsolete by other documents at any time. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress". This is work in progress and does not imply endorsement by the W3C membership.

This document has been produced as part of the W3C XML Activity, following the procedures set out for the W3C Process. The document has been written by the XML Query Working Group. The goals of the XML Query working group are discussed in the XML Query Working Group charter (W3C members only).

The XML Query Working Group feels that the contents of this Working Draft are relatively stable, and therefore encourages feedback on this version.

Comments on this document should be sent to the W3C mailing list www-xml-query-comments@w3.org (archived at http://lists.w3.org/Archives/Public/www-xml-query-comments/).

A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR/.

Table of contents

Appendices

---

1 Goals

The goal of the XML Query Working Group is to produce a data model for XML documents, a set of query operators on that data model, and a query language based on these query operators. The data model will be based on the W3C XML Infoset, and will include support for Namespaces.

Queries operate on single documents or fixed collections of documents. They can select whole documents or subtrees of documents that match conditions defined on document content and structure, and can construct new documents based on what is selected.

2 Usage Scenarios

The following usage scenarios describe how XML queries may be used in various environments, and represent a wide range of activities and needs that are representative of the problem space to be addressed. They are intended to be used as design cases during the development of XML Query, and should be reviewed when critical decisions are made. These usage scenarios should also prove useful in helping non-members of the XML Query Working Group understand the intent and goals of the project.

2.1 Human-readable documents

Perform queries on structured documents and collections of documents, such as technical manuals, to retrieve individual documents, to generate tables of contents, to search for information in structures found within a document, or to generate new documents as the result of a query.

2.2 Data-oriented documents

Perform queries on the XML representation of database data, object data, or other traditional data sources to extract data from these sources, to transform data into new XML representations, or to integrate data from multiple heterogeneous data sources. The XML representation of data sources may be either physical or virtual; that is, data may be physically encoded in XML, or an XML representation of the data may be produced.

2.3 Mixed-model documents

Perform both document-oriented and data-oriented queries on documents with embedded data, such as catalogs, patient health records, employment records, or business analysis documents.

2.4 Administrative data

Perform queries on configuration files, user profiles, or administrative logs represented in XML.

2.5 Filtering streams

Perform queries on streams of XML data to process the data in a manner analogous to UNIX filters. This might be used to process logs of email messages, network packets, stock market data, newswire feeds, EDI, or weather data to filter and route messages represented in XML, to extract data from XML streams, or to transform data in XML streams.

2.6 Document Object Model (DOM)

Perform queries on DOM structures to return sets of nodes that meet the specified criteria.

2.7 Native XML repositories and web servers

Perform queries on collections of documents managed by native XML repositories or web servers.

2.8 Catalog search

Perform queries to search catalogs that describe document servers, document types, XML schemas, or documents. Such catalogs may be combined to support search among multiple servers. A document-retrieval system could use queries to allow the user to select server catalogs, represented in XML, by the information provided by the servers, by access cost, or by authorization. Once a server is selected, a retrieval system could query the kinds of documents found on the server and allow the user to query those documents.

2.9 Multiple syntactic environments

Queries may be used in many environments. For example, a query might be embedded in a URL, an XML page, or a JSP or ASP page; represented by a string in a program written in a general-purpose programming language; provided as an argument on the command-line or standard input; or supported by a protocol, such as DASL or Z39.50.

3 Requirements

3.1 Terminology

The following key words are used throughout the document to specify the extent to which an item is a requirement for the work of the XML Query Working Group:

MUST

This word means that the item is an absolute requirement.

SHOULD

This word means that there may exist valid reasons not to treat this item as a requirement, but the full implications should be understood and the case carefully weighed before discarding this item.

MAY

This word means that an item deserves attention, but further study is needed to determine whether the item should be treated as a requirement.

When the words MUST, SHOULD, or MAY are used in this technical sense, they occur as a hyperlink to these definitions. These words will also be used with their conventional English meaning, in which case there is no hyperlink. For instance, the phrase "the full implications should be understood" uses the word "should" in its conventional English sense, and therefore occurs without the hyperlink.

3.2 General Requirements

3.2.1 Query Language Syntax

The XML Query Language MAY have more than one syntax binding. One query language syntax MUST be convenient for humans to read and write. One query language syntax MUST be expressed in XML in a way that reflects the underlying structure of the query.

3.2.2 Declarativity

The XML Query Language MUST be declarative. Notably, it MUST not enforce a particular evaluation strategy.

3.2.3 Protocol Independence

The XML Query Language MUST be defined independently of any protocols with which it is used. (Relationships to some specific protocols are discussed in **[4 Relationship to Other Activities]**.)

3.2.4 Error Conditions

The XML Query Language MUST define standard error conditions that can occur during the execution of a query, such as processing errors within expressions, unavailability of external functions to the query processor, or processing errors generated by external functions.

3.2.5 Updates

Version 1.0 of the XML Query Language MUST not preclude the ability to add update capabilities in future versions.

3.2.6 Defined for Finite Instances

The XML Query Language MUST be defined for finite instances of the data model. It MAY be defined for infinite instances.

3.3 XML Query Data Model

3.3.1 Reliance on XML Information Set

The XML Query Data Model relies on information provided by XML Processors and Schema Processors, and it MUST ensure that it does not require information that is not made available by such processors. For XML constructs found in XML 1.0 or the Namespaces Recommendation, the XML Query Data Model MUST show how the equivalent XML Query Data Model constructs are built from items in the XML Information Set. The XML Query Data Model SHOULD represent all information items, or provide justification for any information items omitted. For information found in the XML Schema, such as datatypes, the XML Query Working Group MUST coordinate with the XML Schema Working Group to ensure that schema processors may be relied on to provide the information needed to construct the Data Model.

3.3.2 Datatypes

The XML Query Data Model MUST represent both XML 1.0 character data and the simple and complex types of the XML Schema specification.

3.3.3 Collections

The XML Query Data Model MUST represent collections of documents and collections of simple and complex values. (Note that collections are not part of the current XML Infoset.)

3.3.4 References

The XML Query Data Model MUST include support for references, including both references within an XML document and references from one XML document to another.

3.3.5 Schema Availability

Queries MUST be possible whether or not a schema is available (in this document, the term "schema" may refer to either an XML Schema or a DTD). If a schema is available, the data model MUST represent any items that they define for their instances, such as default attributes, entity expansions, or data types. These items will not be present if a schema is not present.

3.3.6 Namespace Awareness

The XML Query Language and XML Query Language Data Model MUST be namespace aware.

3.4 XML Query Functionality

3.4.1 Supported Operations

The XML Query Language MUST support operations on all data types represented by the XML Query Data Model (see datatypes, collections, references) .

3.4.2 Text and Element Boundaries

Queries MUST be able to express simple conditions on text, including conditions on text that spans element boundaries.

3.4.3 Universal and Existential Quantifiers

Operations on collections MUST include support for universal and existential quantifiers.

3.4.4 Hierarchy and Sequence

Queries MUST support operations on hierarchy and sequence of document structures.

3.4.5 Combination

The XML Query Language MUST be able to combine related information from different parts of a given document or from multiple documents.

3.4.6 Aggregation

The XML Query Language MUST be able to compute summary information from a group of related document elements (this operation is sometimes called "aggregation.")

3.4.7 Sorting

The XML Query Language MUST be able to sort query results.

3.4.8 Composition of Operations

The XML Query Language MUST support expressions in which operations can be composed, including the use of queries as operands.

3.4.9 NULL Values

The XML Query Language MUST include support for NULL values. Therefore, all operators, including logical operators, MUST take NULL values into account.

3.4.10 Structural Preservation

Queries MUST be able to preserve the relative hierarchy and sequence of input document structures in query results.

3.4.11 Structural Transformation

Queries MUST be able to transform XML structures and MUST be able to create new structures.

3.4.12 References

Queries MUST be able to traverse intra- and inter-document references.

3.4.13 Identity Preservation

Queries MUST be able to preserve the identity of items in the XML Query Data Model.

3.4.14 Operations on Literal Data

Queries SHOULD be able operate on XML Query Data Model instances specified with the query ("literal" data).

3.4.15 Operations on Names

Queries MUST be able to perform simple operations on names, such as tests for equality in element names, attribute names, and processing instruction targets, and to perform simple operations on combinations of names and data. Queries MAY perform more powerful operations on names.

3.4.16 Operations on Schemas

Queries SHOULD provide access to the XML schema or DTD for a document, if there is one. If the schema is represented as a DTD, a mapping to an appropriate XML Schema representation MAY be required.

3.4.17 Operations on Schema PSV Infoset

Queries MUST be able to operate on information items provided by the post-schema-validation information set defined by XML Schema.

3.4.18 Extensibility

The XML Query Language SHOULD support the use of externally defined functions on all datatypes of the XML Query Data Model. The interface to such functions SHOULD be defined by the Query Language, and SHOULD distinguish these functions from functions defined in the Query Language. The implementation of externally defined functions is not part of the Query Language.

3.4.19 Environment Information

The XML Query Language MUST provide access to information derived from the environment in which the query is executed, such as the current date, time, locale, time zone, or user.

3.4.20 Closure

Queries MUST be closed with respect to the XML Query Data Model. Both the input to a query and the output of a query MUST be defined purely in terms of the XML Query Data Model. Non-XML sources such as traditional databases or objects may be queried if they are given an XML Query Data Model representation. Similarly, query results are defined purely in terms of the XML Query Data Model. In software systems these results may be instantiated in any convenient representation such as DOM nodes, hyperlinks, XML text, or various data formats.

4 Relationship to Other Activities

XML has become a strategic technology in W3C and in the global Web market. The deliverable of the

XML Query Working Group MUST satisfy the dependencies from the following Working Groups before it can advance to Proposed Recommendation. Some dependencies to and from the following W3C Working Groups will require close cooperation during the development process; the requirements posed for the Query work by these Working Groups may change during the development process, which means the interdependency of the Query work with these Working Groups must be managed actively:

W3C Document Object Model Working Group (DOM)

The XML Query Language must be able to return results in a form that can be used in DOM programs, such as DOM Nodes or the Iterators and TreeWalkers defined in the Traversal specification.

XSL and Linking Working Groups

Both XSLT and XPointer use the XML Path Language (XPath), which defines a location path syntax that can be used to search for matching parts of an XML document. The XML Query work will take into consideration the expressibility and search facilities of XPath when formulating its algebra and query syntax, and where desirable try to encompass those functionalities into its query language. The XML Query WG will also take into consideration the additional functionality in the XSLT and XPointer specifications.

XML Schema Working Group

It is a goal of the XML Query work to be compatible with the work of the XML Schema Working Group, including both Structures and Datatypes.

For example, it should be possible to base query predicates on the existing DTD or XSDL definition of the content of an XML document and on the new data types being defined as part of the XDTL.

W3C XML Core Working Group

The XML Query work will define a formal data model of XML documents. This model must be based on the model of the XML Infoset. In case incompatibilities arise, requirements must be posed to the W3C XML Core Working Group. In any case, the final model used by the XML Query working group will have to be based on, and totally compatible with, the model of the XML Infoset.

There are no requirements for co-development of features with the following Working Groups, but there are points of contact between their work and that of this Working Group, and thus logical dependency between their deliverables and those of this Working Group. Requirements from these Working Groups are expected to be well suited for communication via documents:

WAI Protocols & Formats Working Group

Reuse of common constructs greatly facilitates accessibility; the WAI PF Working Group will review work on the XML query facilities to be sure cost/benefit design decisions are informed of the benefits of accessibility.

Internationalization Working Group

The XML Query Working Group will solicit feedback from the Internationalization Working Group to ensure that it satisfies W3C goals for international access to the Web.

XML Fragments Working Group

It may be necessary for the XML Query Working Group to reference the XML Fragment specification if a valid query return type is an XML fragment.

IETF DASL Working Group

XML Query must strive for smooth interaction with the IETF DASL (DAV Searching & Locating) Working Group, in such a way that the XML query language can be easily incorporated into the DASL protocol.

Formal liaison between the XML Query Working Group and other W3C working groups, including the other XML working groups and the WAI (Web Accessibility Initiative) group, as well as organizations outside of the W3C, shall be accomplished by the exchange of documents (requirements, reviews, etc.) transmitted through the XML Coordination Group.

5 References (non-normative)

The following references are some of the works considered by the WG in deriving its requirements.

Maier98
Database Desiderata for an XML Query Language, David Maier, 1998. In Query Languages 98 (QL'98). Available at http://www.w3.org/TandS/QL/QL98/pp/maier.html.

Cotton98
Candidate Requirements for XML Query, Paul Cotton and Ashok Malhotra, 1998. In Query Languages 98 (QL'98). Available at http://www.w3.org/TandS/QL/QL98/pp/queryreq.html.

Fernandez
XML Query Languages: Experiences and Exemplars, Mary Fernandez, Jerome Simeon, Philip Wadler, 1999. Available at http://www.w3.org/1999/09/ql/docs/xquery.html.

Robie99
The Tree Structure of XML Queries, Jonathan Robie. Available at http://www.w3.org/XML/Group/1999/10/xquery-tree.html (W3C members only).

XML
Extensible Markup Language (XML), Version 1.0. W3C Recommendation. Available at http://www.w3.org/TR/1998/REC-xml-19980210.

XPath
XML Path Language (XPath), Version 1.0. W3C Recommendation. Available at http://www.w3.org/TR/xpath.

Namespaces
Namespaces in XML. W3C Recommendation. Available at http://www.w3.org/TR/1999/REC-xml-names-19990114/.

DOM
Document Object Model (DOM), Level 2 Specification. W3C Candidate Recommendation. Available at http://www.w3.org/TR/DOM-Level-2-Core/.

XSLT
XSL Transformations (XSLT), Version 1.0. W3C Recommendation. Available at http://www.w3.org/TR/xslt.

Infoset
XML Information Set, W3C Working Draft 20-December-1999, John Cowan, David Megginson (eds.), 1999. Available at http://www.w3.org/TR/xml-infoset/ .

XMLSchema0
XML Schema Part 0: Primer, David C. Fallside (ed.), 2000. Available at http://www.w3.org/TR/xmlschema-0/

XMLSchema1

XML Schema Part 1: Structures, Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelson (eds.), 2000. Available at http://www.w3.org/TR/xmlschema-1/

XMLSchema2

XML Schema Part 2: Datatypes, Paul V. Biron, Ashok Malhotra (eds.), 2000. Available at http://www.w3.org/TR/xmlschema-2/

A Glossary

Universal and Existential Quantifiers

A quantifier is a term denoting a constraint on the number of objects in a collection that satisfy an accompanying condition. The existential quantifier denotes that at least one object satisfies the condition. The universal quantifier denotes that all objects satisfy the condition.

Document

A document consists of the set of nodes and edges in the subtree descended from a Document node in the XML Query Data Model.

Inter-document references

References that refer to nodes that do not reside in the same XML document as the reference itself.

Intra-document references

References that reside in the same XML document as the nodes they reference.

Literal Data

Literal fragments of an XML document such as <name><first>Joe</first><last>Doe</last></name>, which may be used for comparison.