XPointer element() Scheme

W3C Recommendation 25 March 2003

This version:

http://www.w3.org/TR/2003/REC-xptr-element-20030325/

Latest version:

http://www.w3.org/TR/xptr-element/

Previous version:

http://www.w3.org/TR/2002/PR-xptr-element-20021113/

Authors and Contributors:

Paul Grosso (Arbortext, Inc.) <paul@arbortext.com>
Eve Maler (Sun Microsystems) <eve.maler@sun.com>
Jonathan Marsh (Microsoft) <jmarsh@microsoft.com>
Norman Walsh (Sun Microsystems) <Norman.Walsh@Sun.COM>

Abstract

The XPointer `element()` scheme is intended to be used with the XPointer Framework [XPtrFrame] to allow basic addressing of XML elements.

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 Recommendation (REC) of the W3C. It has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

This document has been produced by the W3C XML Linking Working Group as part of the XML Activity. It is intended to address a subset of the original XPointer requirements, and to serve, along with the accompanying XPointer Framework and XPointer xmlns() Scheme specifications, as a part of a fragment identifier syntax for the XML Media types.

Public comments on this Recommendation are welcome. Please send them to the public mailing list www-xml-linking-comments@w3.org (archive).

Information about implementations relevant to this specification and the accompanying XPointer element() Scheme and XPointer xmlns() Scheme can be found in the Implementation Report.

There are patent disclosures and license commitments associated with this Recommendation, which may be found on the XPointer IPR Statement page in conformance with W3C policy.

A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR/. W3C publications may be updated, replaced, or obsoleted by other documents at any time.

Table of Contents

Appendices

This page is intentionally left blank.

1. Introduction

The XPointer `element()` scheme is intended to be used with the XPointer Framework [XPtrFrame] to allow basic addressing of XML elements.

The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*, and *optional* in this specification are to be interpreted as described in [RFC 2119].

The terms pointer part, scheme, XPointer processor, error, and namespace binding context are used in this specification as defined in the XPointer Framework specification.

The formal grammar for the `element()` scheme is given using simple Extended Backus-Naur Form (EBNF) notation, as described in the XML Recommendation [XML].

2. Conformance

This specification normatively depends on the XPointer Framework [XPtrFrame] specification.

XPointer processors supporting the `element()` scheme depend on the ability of applications to expose an XML resource as at least the XML Information Set [Infoset] and XML Schema [XMLSchema] information items and properties listed in the XPointer Framework specification.

Conforming XPointer processors claiming to support the `element()` scheme must conform to the behavior defined in this specification and may conform to additional XPointer scheme specifications.

3. Language and Processing

This section describes the syntax and semantics of the `element()` scheme and the behavior of XPointer processors with respect to this scheme.

The scheme name is “element”. The scheme data syntax is as follows; if scheme data in a pointer part with the `element()` scheme does not conform to the syntax defined in this section the pointer part does not identify a subresource.

element() Scheme Syntax

[1]	ElementSchemeData	::=	(NCName **ChildSequence**?) | **ChildSequence**
[2]	ChildSequence	::=	('/' [1-9] [0-9]*)+

The scheme data consists of either a NCName (as defined in the XML namespaces specification [XML-Names]) or a child sequence, or both.

A NCName appearing alone identifies a single element exactly as it would in a shorthand pointer, as defined in the XPointer Framework [XPtrFrame] specification, except that failure to identify an element results simply in no subresource being identified by this pointer part rather than an XPointer Framework error.

For example, the following pointer part identifies the element with an ID (as defined in XPointer Framework) of “intro”:

```
element(intro)
```

A child sequence appearing alone identifies an element by means of stepwise navigation, which is directed by a sequence of integers separated by slashes (/); each integer *n* locates the *n*th child element of the previously located element. The integer *n* following the first slash locates the *n*th top-level element: either the unique document element (the `[document element]` property, if the resource is an XML document, in which case the integer is always 1) or one of potentially several root elements (the entity root element(s), if the resource is an external parsed entity). For example, assuming that the XML resource is a whole XML document, the following pointer part identifies the second child element inside the root element of the document:

```
element(/1/2)
```

A child sequence appearing after an NCName identifies an element by means of stepwise navigation, starting from the element located by the given name. For example, the following pointer part identifies an element by first locating the element identified by the value intro, then locating that element's third child element, then finally identifying that element's first child element:

```
element(intro/3/1)
```

If either the NCName or the child sequence does not locate an element, no element is identified by the pointer part as a whole.

The `element()` scheme does not use the namespace binding context because it does not support qualified names.

Appendix A. Normative References

Infoset

John Cowan and Richard Tobin, editors. *XML Information Set*. World Wide Web Consortium, 2001.

RFC 2119

Scott Bradner, *RFC 2119: Key words for use in RFCs to Indicate Requirement Levels*. Internet Engineering Task Force, 1997.

XML

Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, and Eve Maler, editors. *Extensible Markup Language (XML) 1.0 (Second Edition).* World Wide Web Consortium, 2000.

XML-Names

Tim Bray, Dave Hollander, and Andrew Layman, editors. *Namespaces in XML*. World Wide Web Consortium, 1999.

XPtrFrame

Paul Grosso, Eve Maler, Jonathan Marsh, and Norman Walsh, editors. *XPointer Framework*. World Wide Web Consortium, 2002.

XMLSchema

Henry S. Thompson et al., editors. *XML Schema Part 1.* World Wide Web Consortium, 2001.