

Web Service Enumeration (WS-Enumeration)

September 2004

Authors

Jan Alexander, Systinet
Don Box, Microsoft
Luis Felipe Cabrera, Microsoft
Dave Chappell, Sonic Software
Glen Daniels, Sonic Software
Alan Geller, Microsoft (editor)
Chris Kaler, Microsoft
David Orchard, BEA
Igor Sedukhin, Computer Associates
Miroslav Simek, Systinet
Marvin Theimer, Microsoft

Copyright Notice

(c) 2004 [BEA Systems Inc.](#), [Computer Associates](#), [Microsoft Corporation, Inc.](#), [Sonic Software](#), and [Systinet Corporation](#). All rights reserved.

Permission to copy and display the WS-Enumeration (the "Specification", which includes WSDL and schema documents), in any medium without fee or royalty is hereby granted, provided that you include the following on ALL copies of the Specification that you make:

1. A link or URL to the Specification at one of the Co-Developers' websites.
2. The copyright notice as shown in the Specification.

BEA Systems, Computer Associates, Microsoft, Sonic Software, and Systinet (collectively, the "Co-Developers") each agree to grant you a license, under royalty-free and otherwise reasonable, non-discriminatory terms and conditions, to their respective essential patent claims that they deem necessary to implement the Specification.

THE SPECIFICATION IS PROVIDED "AS IS," AND THE CO-DEVELOPERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE CO-DEVELOPERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE SPECIFICATIONS.

The name and trademarks of the Co-Developers may NOT be used in any manner, including advertising or publicity pertaining to the Specifications or their contents without specific, written prior permission. Title to copyright in the Specifications will at all times remain with the Co-Developers.

No other rights are granted by implication, estoppel or otherwise.

Abstract

This specification describes a general SOAP-based protocol for enumerating a sequence of XML elements that is suitable for traversing logs, message queues, or other linear information models.

Status

WS-Enumeration and related specifications are provided as-is and for review and evaluation only. BEA Systems, Computer Associates, Microsoft, Sonic Software, and Systinet make no warranties or representations regarding the specifications in any manner whatsoever.

Table of Contents

1. Introduction

1.1 Requirements

2. Notations and Terminology

2.1 Notational Conventions

2.2 XML Namespaces

2.3 Terminology

2.4 Compliance

3. Enumeration Messages

3.1 Enumerate

3.2 Pull

3.3 Renew

3.4 GetStatus

3.5 Release

3.6 EnumerationEnd

4. Security Considerations

5. Acknowledgements

6. References

Appendix I – XML Schema

Appendix II – WSDL

1. Introduction

There are numerous applications for which a simple single-request/single-reply metaphor is insufficient for transferring large data sets over SOAP. Applications that do not fit into this simple paradigm include streaming, traversal, query, and enumeration.

This specification defines a simple SOAP-based protocol for enumeration that allows the data source to provide a session abstraction, called an enumeration context, to a consumer that represents a logical cursor through a sequence of data items. The consumer can then request XML element information items using this enumeration context over the span of one or more SOAP messages.

Somewhere, state must be maintained regarding the progress of the iteration. This state may be maintained between requests by the data source being enumerated or by the data consumer. WS-Enumeration allows the data source to decide, on a request-by-request basis, which party will be responsible for maintaining this state for the next request.

In its simplest form, WS-Enumeration defines a single operation, Pull, which allows a data source, in the context of a specific enumeration, to produce a sequence of XML elements in the body of a SOAP message. Each subsequent Pull operation returns the next N elements in the aggregate sequence.

A data source may provide a custom mechanism for starting a new enumeration. For instance, a data source that provides access to a SQL database may support a SELECT operation that performs a database query and uses an explicit database cursor to iterate through the returned rows. In general, however, it is simpler if all data sources support a single, standard operation to start an enumeration. This specification defines such an operation, Enumerate, that data sources may implement for starting a new enumeration of a data source. The Enumerate operation is used to create new enumeration contexts for subsequent traversal/retrieval. Each Enumerate operation results in a distinct enumeration context, each with its own logical cursor/position.

It should be emphasized that different enumerations of the same data source may produce different results; this may happen even for two enumeration contexts created concurrently by a single consumer using identical Enumerate requests. In general, the consumer of an enumeration should not make any assumptions about the ordering or completeness of the enumeration; the returned data items represent a selection by the data source of items it wishes to present to that consumer at that time in that order, with no guarantee that every available item is returned or that the order in which items is returned has any semantic meaning whatsoever (of course, any specific data source may provide strong guarantees, if so desired). In particular, it should be noted that the very act of enumerating the contents of a data source may modify the contents of the data source; for instance, a queue might be represented as a data source such that items that are returned in a Pull response are removed from the queue.

1.1 Requirements

This specification intends to meet the following requirements:

- Support enumeration of data sources that cannot practically fit into a single SOAP message.
- Support both server-side and client-side enumeration state maintenance.
- Minimize additional mechanism beyond the current web service architecture.

2. Notations and Terminology

This section specifies the notations, namespaces, and terminology used in this specification.

2.1 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [[RFC 2119](#)].

This specification uses the following syntax to define normative outlines for messages:

- The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- Characters are appended to elements and attributes to indicate cardinality:
 - "?" (0 or 1)
 - "*" (0 or more)
 - "+" (1 or more)

- The character "|" is used to indicate a choice between alternatives.
- The characters "[" and "]" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content. Additional children and/or attributes MAY be added at the indicated extension points but MUST NOT contradict the semantics of the parent and/or owner, respectively. If a receiver does not recognize an extension, the receiver SHOULD ignore it.
- XML namespace prefixes (see Table 1) are used to indicate the namespace of the element being defined.

2.2 XML Namespaces

The XML namespace URI that MUST be used by implementations of this specification is:

<http://schemas.xmlsoap.org/ws/2004/09/enumeration>

Table 1 lists XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Table 1: Prefixes and XML namespaces used in this specification

Prefix	XML Namespace	Specification(s)
wsen	http://schemas.xmlsoap.org/ws/2004/09/enumeration	This specification
s	http://www.w3.org/2003/05/soap-envelope	SOAP 1.2 [SOAP 1.2]
s11	http://schemas.xmlsoap.org/soap/envelope/	SOAP 1.1 [SOAP 1.1]
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing	WS-Addressing [WS-Addressing]
xs	http://www.w3.org/2001/XMLSchema	XML Schema [Part 1 , 2]
wSDL	http://schemas.xmlsoap.org/wSDL	WSDL/1.1 [WSDL 1.1]

2.3 Terminology

Consumer

The Web service that is requesting the data enumeration from the data source

Data source

A Web service that supports traversal using enumeration contexts via the Enumerate operation defined in this specification

Enumeration context

A session context that represents a specific traversal through a logical sequence of XML element information items using the Pull operation defined in this specification

2.4 Compliance

An implementation is not compliant with this specification if it fails to satisfy one or more of the MUST or REQUIRED level requirements defined herein. A SOAP Node MUST NOT use the XML namespace identifier for this specification (listed in [Section 2.2](#)) within SOAP Envelopes unless it is compliant with this specification.

Normative text within this specification takes precedence over normative outlines, which in turn takes precedence over the XML Schema and WSDL descriptions.

3. Enumeration Messages

Enumeration contexts represent a specific traversal through a sequence of XML information items. An Enumerate operation may be used to establish an enumeration context from a data source. A Pull operation is used to fetch information items from a data source according to a specific enumeration context. A Release operation is used to tell a data source that the consumer is abandoning an enumeration context before it has completed the enumeration.

Enumeration contexts are represented as XML data that is opaque to the consumer. Initially, the consumer gets an enumeration context from the data source by means of an Enumerate operation. The consumer then passes that XML data back to the data source in the Pull request. Optionally, the data source may return an updated enumeration context in the Pull response; when present, this new enumeration context should replace the old one on the consumer, and should be passed to the data source in all future responses until and unless the data source again returns an updated enumeration context.

Consumers should not reuse old enumeration contexts that have been replaced by the data source. Using a replaced enumeration context in a Pull response MAY yield undefined results, including being ignored or generating a fault.

Once the last element in a sequence has been returned, or the enumeration context has expired, the enumeration context is considered invalid and the result of subsequent operations referencing that context is undefined.

Callers MAY issue a Release operation against a valid enumeration context at any time, which causes the enumeration context to become invalid and allows the data source to free up any resources it may have allocated to the enumeration. Issuing a Release operation prior to reaching the end of the sequence of elements is explicitly allowed; however, no further operations should be issued after a Release.

In addition, the data source MAY invalidate an enumeration context at any time, as necessary.

3.1 Enumerate

All data sources MUST support some operation that allows an enumeration to be started. A data source MAY support the Enumerate operation, or it may provide some other mechanism for starting an enumeration and receiving an enumeration context.

The Enumerate operation is initiated by sending an Enumerate request message to the data source. The Enumerate request message MUST be of the following form:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:Enumerate ...>
      <wsen:EndTo>endpoint-reference</wsen:EndTo> ?
      <wsen:Expires>[xs:dateTime | xs:duration]</wsen:Expires> ?
      <wsen:Filter Dialect="xs:anyURI"?> xs:any </wsen:Filter> ?
      ...
    </wsen:Enumerate ...>
  </s:Body ...>
</s:Envelope ...>
```

```
</wsen:Enumerate>
</s:Body>
</s:Envelope>
```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Header/wsa:Action

This required element MUST contain the value

<http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate>. If a SOAP Action URI is also present in the underlying transport, its value MUST convey the same value.

/s:Envelope/s:Body/*wsen:EndTo

This optional element denotes where to send an EnumerationEnd message if the enumeration is terminated unexpectedly (see section 3.6 below). If present, this element MUST be of type `wsa:EndpointReferenceType`. Default is to not send this message.

/s:Envelope/s:Body/*wsen:Expires

Requested expiration time for the enumeration. (No implied value.) The data source defines the actual expiration and is not constrained to use a time less or greater than the requested expiration. The expiration time may be a specific time or a duration from the enumeration's creation time. Both specific times and durations are interpreted based on the data source's clock.

If this element does not appear, then the request is for an enumeration that will not expire. That is, the consumer is requesting the data source to create an enumeration with an indefinite lifetime. If the data source grants such an enumeration, it will terminate when the end of the enumeration is reached, or if the consumer sends a Release request, or by the data source at any time for reasons such as connection termination, resource constraints, or system shut-down.

If the expiration time is either a zero duration or a specific time that occurs in the past according to the data source, then the request MUST fail, and the data source MAY generate a fault indicating that an invalid expiration time was requested.

The SOAP 1.1 binding of this fault is:

- faultcode = `s11:Client`
- faultstring = e.g., "invalid expiration time"

The SOAP 1.2 binding of this fault is:

- `s12:Code/s12:Value` = `s12:Sender`
- `s12:Code/s12:Subcode/s12:Value` = `wsen:InvalidExpirationTime`
- `s12:Reason/s12:Text` = e.g., "invalid expiration time"

Some data sources may not have a "wall time" clock available, and so are able only to accept durations as expirations. If such a source receives an Enumerate request containing a specific time expiration, then the request MUST fail; if so, the data source SHOULD generate a fault indicating that an unsupported expiration type was requested.

The SOAP 1.1 binding of this fault is:

- faultcode = `s11:Client`
- faultstring = e.g., "unsupported expiration type"

The SOAP 1.2 binding of this fault is:

- `s12:Code/s12:Value` = `s12:Sender`
- `s12:Code/s12:Subcode/s12:Value` = `wsen:UnsupportedExpirationType`

- s12:Reason/s12:Text = e.g., "unsupported expiration type"

/s:Envelope/s:Body/wsen:Enumerate/wsen:Filter

This optional element contains a Boolean predicate in some dialect (see /s:Envelope/s:Body/*/wsen:Filter/@Dialect) that all elements of interest must satisfy. The resultant enumeration context MUST NOT return elements for which this predicate expression evaluates to the value false. If this element is absent, then the implied value is the expression true(), indicating that no filtering is desired.

If the data source does not support filtering, the request MUST fail, and the data source MAY generate a SOAP fault as follows:

SOAP 1.1:

- faultcode = s11:Client
- faultstring = e.g., "filtering not supported"

SOAP 1.2:

- s12:Code/s12:Value = s12:Sender
- s12:Code/s12:Subcode/s12:Value = wsen:FilteringNotSupported
- s12:Reason/s12:Text = e.g., "filtering not supported"

If the data source supports filtering but cannot honor the requested filter dialect, the request MUST fail, and the data source MAY generate a SOAP fault as follows:

SOAP 1.1:

- faultcode = s11:Client
- faultstring = e.g., "filter dialect requested unavailable"
- [optional] detail/wsen:SupportedDialect = repeating; one per filter dialect supported by the container

SOAP 1.2:

- s12:Code/s12:Value = s12:Sender
- s12:Code/s12:Subcode/s12:Value = wsen:FilterDialectRequestedUnavailable
- s12:Reason/s12:Text = e.g., "filter dialect requested unavailable"
- [optional] s12:Detail/wsen:SupportedDialect = repeating; one per filter dialect supported by the container

If the data source supports filtering and the requested dialect but cannot process the requested filter content, the request MUST fail, and the data source MAY generate a SOAP fault as follows:

SOAP 1.1:

- faultcode = s11:Client
- faultstring = e.g., "cannot filter as requested"

SOAP 1.2:

- s12:Code/s12:Value = s12:Sender
- s12:Code/s12:Subcode/s12:Value = wsen:CannotProcessFilter
- s12:Reason/s12:Text = e.g., "cannot filter as requested"

/s:Envelope/s:Body/*/wsen:Filter/@Dialect

Implied value is 'http://www.w3.org/TR/1999/REC-xpath-19991116'.

/s:Envelope/s:Body/*/wsen:Filter/@Dialect=' <http://www.w3.org/TR/1999/REC-xpath-19991116>'

Value of `/s:Envelope/s:Body/*/wsen:Filter` is an XPath [[XPath 1.0](#)] predicate expression (PredicateExpr); the context of the expression is:

- Context Node: any XML element that could be returned as a direct child of the `wsen:Items` element.
- Context Position: 1.
- Context Size: 1.
- Variable Bindings: None.
- Function Libraries: Core Function Library [[XPath 1.0](#)].
- Namespace Declarations: The [in-scope namespaces] property [[XML Infoset](#)] of `/s:Envelope/s:Body/*/wsen:Filter`.

Other components of the outline above are not further constrained by this specification.

Upon successful processing of an Enumerate request message, a data source is expected to create an enumeration context and return that context in an Enumerate response message, which MUST adhere to the following form:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse
    </wsa:Action>
    <wsa:ReplyTo>endpoint-reference</wsa:ReplyTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:EnumerateResponse ...>
      <wsen:Expires>[xs:dateTime | xs:duration]</wsen:Expires> ?
      <wsen:EnumerationContext>...</wsen:EnumerationContext>
      ...
    </wsen:EnumerateResponse>
  </s:Body>
</s:Envelope>
```

The following describes additional, normative constraints on the outline listed above:

`/s:Envelope/s:Header/wsa:Action`

This required element MUST contain the value `http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse`. If a SOAP Action URI is also present in the underlying transport, its value MUST convey the same value.

`/s:Envelope/s:Body/*/wsen:Expires`

The expiration time assigned by the data source. The expiration time MAY be either an absolute time or a duration but SHOULD be of the same type as the requested expiration (if any).

If this element does not appear, then the enumeration will not expire. That is, the enumeration has an indefinite lifetime. It will terminate when the end of the enumeration is reached, or if the consumer sends a Release request, or by the data source at any time for reasons such as connection termination, resource constraints, or system shut-down.

`/s:Envelope/s:Body/wsen:EnumerateResponse /wsen:EnumerationContext`

The required EnumerationContext element contains the XML representation of the new enumeration context. The consumer is required to pass this XML data in Pull requests for this enumeration context, until and unless a PullResponse message updates the enumeration context.

Table 4 lists a sample Enumerate request.

Table 2: Enumerate request

```
(01) <s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope '  
(02)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing '  
(03)     xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/enumeration">  
(04)   <s:Header>  
(05)     <wsa:Action>  
(06)       http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate  
(07)     </wsa:Action>  
(08)     <wsa:MessageID>  
(09)       uuid:e7c5726b-de29-4313-b4d4-b3425b200839  
(10)     </wsa:MessageID>  
(11)     <wsa:To>http://www.example.com/relayAgent/enum19</wsa:To>  
(12)     <wsa:ReplyTo>  
(13)       <wsa:Address>  
(14)         http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous  
(15)       </wsa:Address>  
(16)     </wsa:ReplyTo>  
(17)   </s:Header>  
(18)   <s:Body>  
(19)     <wsen:Enumerate>  
(20)       <wsen:Expires> PT10M </wsen:Expires>  
(21)     </wsen:Enumerate>  
(22)   </s:Body>  
(23) </s:Envelope>
```

Lines (05-07) in Table 4 indicate this message is an Enumerate request and that the data source is expected to respond with an Enumerate response message. The wsen:Expires element on line (20) indicates that the consumer would like an enumeration context that will be good for at least 10 minutes; that is, it expects to complete its enumeration within a 10 minute period. No wsen:Filter element is present, so the resultant enumeration context is expected to return all available elements.

Table 5 lists a response to the request in Table 4.

Table 3: Response to Enumerate request

```
(01) <s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope '  
(02)     xmlns:wxf='http://schemas.xmlsoap.org/ws/2004/09/enumeration '  
(03)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing '  
(04) >  
(05)   <s:Header>  
(06)     <wsa:Action>  
(07)       http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse  
(08)     </wsa:Action>  
(09)     <wsa:RelatesTo>  
(10)       uuid:e7c5726b-de29-4313-b4d4-b3425b200839  
(11)     </wsa:RelatesTo>  
(12)     <wsa:To>  
(13)       http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous  
(14)     </wsa:To>  
(15)   </s:Header>  
(16)   <s:Body>
```

```

(17)    <wsen:EnumerateResponse>
(18)        <wsen:Expires> PT15M </wsen:Expires>
(19)        <wsen:EnumerationContext>
(20)            123
(21)        </wsen:EnumerationContext>
(22)    </wsen:EnumerateResponse>
(23)    </s:Body>
(24) </s:Envelope>

```

Lines (06-08) in Table 5 indicate this message is an Enumerate response message. Line (18) indicates that the data source has actually created an enumeration context with a lifetime of 15 minutes. Lines (19-21) are the XML representation of the enumeration context that supports the Pull operation defined below.

3.2 Pull

The Pull operation is initiated by sending a Pull request message to the data source. The Pull request message MUST be of the following form:

```

<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:ReplyTo>wsa:EndpointReference</wsa:ReplyTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:Pull ...>
      <wsen:EnumerationContext>...</wsen:EnumerationContext>
      <wsen:MaxTime>xsd:duration</wsen:MaxTime> ?
      <wsen:MaxElements>xsd:long</wsen:MaxElements> ?
      <wsen:MaxCharacters>xsd:long</wsen:MaxCharacters> ?
      ...
    </wsen:Pull>
  </s:Body>
</s:Envelope>

```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Header/wsa:Action

This required element MUST contain the value

http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull. If a SOAP Action URI is also present in the underlying transport, its value MUST convey the same value.

/s:Envelope/s:Body/wsen:Pull/wsen:EnumerationContext

This required element contains the XML data that represents the current enumeration context.

If the enumeration context is not valid, either because it has been replaced in the response to another Pull request, or because it has completed (EndOfSequence has been returned in a Pull response), or because it has been Released, or because it has expired, or because the data source has had to invalidate the context, then the data source SHOULD fail the request, and MAY generate the following fault:

SOAP 1.1:

- faultcode = s11:Server

- faultstring = *text explaining why the enumeration context is no longer valid, or "Invalid enumeration context" if no additional information is available*

SOAP 1.2:

- s:Code/s:Value = s:Receiver
- s:Code/s:Subcode/s:Value = wsen:InvalidEnumerationContext
- s:Reason/s:Text = *text explaining why the enumeration context is no longer valid, or "Invalid enumeration context" if no additional information is available*

Note that the data source may not be able to determine that an enumeration context is not valid, especially if all of the state associated with the enumeration is kept in the enumeration context and refreshed on every PullResponse.

/s:Envelope/s:Body/wsen:Pull/wsen:MaxTime

This optional element (of type xsd:duration) indicates the maximum amount of time the initiator is willing to allow the data source to assemble the Pull response. When this element is absent, the data source is not required to limit the amount of time it takes to assemble the Pull response.

This is useful with data sources that accumulate elements over time and package them into a single Pull response.

/s:Envelope/s:Body/wsen:Pull/wsen:MaxElements

This optional element (of type xsd:long) indicates the number of items (child elements of wsen:Items in the Pull response) the consumer is willing to accept. When this element is absent, its implied value is 1. Implementations **MUST NOT** return more than this number of elements in the Pull response message. Implementations **MAY** return fewer than this number based on either the wsen:MaxTime timeout, the wsen:MaxCharacters size limit, or implementation-specific constraints.

/s:Envelope/s:Body/wsen:Pull/wsen:MaxCharacters

This optional element (of type xsd:long) indicates the maximum size of the returned elements, in Unicode characters, that the initiator is willing to accept. When this element is absent, the data source is not required to limit the number of characters in the Pull response. Implementations **MUST NOT** return a Pull response message whose wsen:Items element is larger than MaxCharacters. Implementations **MAY** return a smaller message based on the wsen:MaxTime timeout, the wsen:MaxElements limit, or implementation-specific constraints.

Even if a Pull request contains a MaxCharacters element, the consumer **MUST** be prepared to receive a Pull response that contains more data characters than specified, as XML canonicalization or alternate XML serialization algorithms may change the size of the representation.

It may happen that the next item the data source would return to the consumer is larger than MaxCharacters. In this case, the data source **MAY** skip the item, or **MAY** return an abbreviated representation of the item that fits inside MaxCharacters. If the data source skips the item, it **MAY** return it as part of the response to a future Pull request with a larger value of MaxCharacters, or it **MAY** omit it entirely from the enumeration. If the oversize item is the last item to be returned for this enumeration context and the data source skips it, it **MUST** include the wsen:EndOfSequence item in the Pull response and invalidate the enumeration context; that is, it may not return zero items but not consider the enumeration completed. See the discussion of wsen:EndOfSequence below.

Other components of the outline above are not further constrained by this specification.

Upon receipt of a Pull request message, the data source may wait as long as it deems necessary (but not longer than the value of the wsen:MaxTime element, if present) to produce a message for delivery to the consumer. The data source MUST recognize the wsen:MaxTime element and return the following fault if no elements are available prior to the request message's deadline:

SOAP 1.1:

- faultcode = s11:Server
- faultstring = e.g., "timeout"

SOAP 1.2:

- s:Code/s:Value = s:Receiver
- s:Code/s:Subcode/s:Value = wsen:TimedOut
- s:Reason/s:Text = e.g., "timeout"

Note, however, that this fault SHOULD NOT cause the enumeration context to become invalid (of course, the data source may invalidate the enumeration context for other reasons). That is, the requestor should be able to issue additional Pull requests using this enumeration context after receiving this fault.

Upon successful processing of a Pull request message, a data source is expected to return a Pull response message, which MUST adhere to the following form:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse
    </wsa:Action>
    <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:PullResponse ...>
      <wsen:EnumerationContext>...</wsen:EnumerationContext> ?
      <wsen:Items> ?
        <xs:any> enumeration-specific element </xs:any> +
      </wsen:Items>
      <wsen:EndOfSequence/> ?
      ...
    </wsen:PullResponse>
  </s:Body>
</s:Envelope>
```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Header/wsa:Action

This required element MUST contain the value `http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse`. If a SOAP Action URI is also present in the underlying transport, its value MUST convey the same value.

/s:Envelope/s:Body/wsen:PullResponse/wsen:EnumerationContext

The optional EnumerationContext element, if present, contains a new XML representation of the current enumeration context. The consumer is required to replace the prior representation with the contents of this element.

/s:Envelope/s:Body/wsen:PullResponse/wsen:Items/*any*

The optional Items element contains one or more enumeration-specific elements, one for each element being returned.

/s:Envelope/s:Body/wsen:PullResponse/wsen:EndOfSequence

This optional element indicates that no more elements are available from this enumeration. Additionally, once this element is returned in a Pull response message, subsequent Pull requests using that enumeration context SHOULD generate a wsen:InvalidEnumerationContext fault message; in any case, they MUST NOT return a valid PullResponse.

Note that at least one of wsen:Items or wsen:EndOfSequence MUST appear. It is possible for both to appear if items are returned and the sequence is exhausted. Similarly, wsen:EnumerationContext and wsen:EndOfSequence MUST NOT both appear; neither may appear, or one without the other, but not both in the same PullResponse.

Table 2 lists a Pull request.

Table 4: Pull request

```
(01) <s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope'  
(02)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'  
(03)     xmlns:wsen='http://schemas.xmlsoap.org/ws/2004/09/enumeration'>  
(04)   <s:Header>  
(05)     <wsa:Action>  
(06)       http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull  
(07)     </wsa:Action>  
(08)     <wsa:MessageID>  
(09)       uuid:e7c5726b-de29-4313-b4d4-b3425b200839  
(10)     </wsa:MessageID>  
(11)     <wsa:To>http://www.example.com/relayAgent</wsa:To>  
(12)     <wsa:ReplyTo>  
(13)       <wsa:Address>  
(14)         http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous  
(15)       </wsa:Address>  
(16)     </wsa:ReplyTo>  
(17)   </s:Header>  
(18)   <s:Body>  
(19)     <wsen:Pull>  
(20)       <wsen:EnumerationContext>123</wsen:EnumerationContext>  
(21)       <wsen:MaxTime>P30S</wsen:MaxTime>  
(22)       <wsen:MaxElements>10</wsen:MaxElements>  
(23)     </wsen:Pull>  
(24)   </s:Body>  
(25) </s:Envelope>
```

Lines (05-07) in Table 2 indicate this message is a Pull request and that the data source is expected to respond with a Pull response message. Line (21) indicates that the response message should be generated no more than 30 seconds after receipt of the Pull request message. Line (22) indicates that no more than 10 elements should be returned in the body of the Pull response message.

Table 3 lists a response to the request in Table 2.

Table 5: Response to Pull request

```
(01) <s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope'  
(02)     xmlns:wsen='http://schemas.xmlsoap.org/ws/2004/09/enumeration'  
(03)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'  
(04) >  
(05)   <s:Header>
```

```

(06) <wsa:Action>
(07)   http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse
(08) </wsa:Action>
(09) <wsa:RelatesTo>
(10)   uuid:e7c5726b-de29-4313-b4d4-b3425b200839
(11) </wsa:RelatesTo>
(12) <wsa:To>
(13)   http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
(14) </wsa:To>
(15) </s:Header>
(16) <s:Body>
(17)   <wsen:PullResponse>
(18)     <wsen:Items xmlns:xx="http://fabrikam123.com/schema/log">
(19)       <xx:LogEntry id="1">System booted</xx:LogEntry>
(20)       <xx:LogEntry id="2">AppX started</xx:LogEntry>
(21)       <xx:LogEntry id="3">John Smith logged on</xx:LogEntry>
(22)       <xx:LogEntry id="4">AppY started</xx:LogEntry>
(23)       <xx:LogEntry id="5">AppX crashed</xx:LogEntry>
(24)     </wsen:Items>
(25)     <wsen:EndOfSequence/>
(26)   </wsen:PullResponse>
(27) </s:Body>
(28) </s:Envelope>

```

Lines (06-08) in Table 3 indicate this message is a Pull response message. Lines (19-23) are the five elements returned by this Pull request. The presence of a `wsen:EndOfSequence` element (line (25)) indicates that no more elements are available and that the enumeration context is now invalid.

The consumer SHOULD NOT issue additional Pull request messages after a Pull response containing a `wsen:EndOfSequence` element has been returned. Similarly, upon receipt of a Pull response containing a `wsen:EndOfSequence` element, the consumer SHOULD NOT issue a Release operation to signal that the enumeration context is no longer needed.

If the consumer does issue a Pull or Release on an invalid enumeration context, the result is undefined: the data source MAY ignore the request or MAY return a `wsen:InvalidEnumerationContext` fault, as described above, or MAY take some other action.

3.3 Renew

Data sources MUST support requests to renew enumerations.

To renew an enumeration, the consumer sends a request of the following form to the data source:

```

<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/Renew
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:FaultTo>endpoint-reference</wsa:FaultTo> ?
    <wsa:ReplyTo>endpoint-reference</wsa:ReplyTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:Renew ...>
      <wsen:EnumerationContext>...</wsen:EnumerationContext>

```

```

    <wsen:Expires>[xs:dateTime | xs:duration]</wsen:Expires> ?
    ...
  </wsen:Renew>
</s:Body>
</s:Envelope>

```

Components of the outline listed above are additionally constrained as for a request to create an enumeration (see Section 3.1 Subscribe) with the following addition(s):

/s:Envelope/s:Body/*wsen:EnumerationContext

This required element contains the XML data that represents the current enumeration context.

If the enumeration context is not valid, either because it has been replaced in the response to another Pull request, or because it has completed (EndOfSequence has been returned in a Pull response), or because it has been Released, or because it has expired, or because the data source has had to invalidate the context, then the data source SHOULD fail the request, and MAY generate the following fault:

SOAP 1.1:

- faultcode = s11:Server
- faultstring = *text explaining why the enumeration context is no longer valid, or "Invalid enumeration context" if no additional information is available*

SOAP 1.2:

- s:Code/s:Value = s:Receiver
- s:Code/s:Subcode/s:Value = wsen:InvalidEnumerationContext
- s:Reason/s:Text = *text explaining why the enumeration context is no longer valid, or "Invalid enumeration context" if no additional information is available*

Note that the data source may not be able to determine that an enumeration context is not valid, especially if all of the state associated with the enumeration is kept in the enumeration context and refreshed on every PullResponse.

Other components of the outline above are not further constrained by this specification.

If the data source accepts a request to renew an enumeration, it MUST reply with a response of the following form:

```

<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/RenewResponse
    </wsa:Action>
    <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:RenewResponse ...>
      <wsen:Expires>[xs:dateTime | xs:duration]</wsen:Expires> ?
      <wsen:EnumerationContext>...</wsen:EnumerationContext> ?
      ...
    </wsen:RenewResponse>
  </s:Body>
</s:Envelope>

```

Components of the outline listed above are constrained as for a response to an Enumerate request (see section 3.1 above) with the following addition:

/s:Envelope/s:Body/wsen:RenewResponse/wsen:Expires

If the requested expiration is a duration, then the implied start of that duration is the time when the data source starts processing the Renew request.

/s:Envelope/s:Body/wsen:RenewResponse/wsen:EnumerationContext

This element is optional in this response.

If the data source chooses not to renew this enumeration, the request MUST fail, and the data source SHOULD generate a fault indicating that the renewal was not accepted.

The SOAP 1.1 binding of this type of fault is:

- faultcode = s11:Server
- faultstring = *text explaining why the enumeration context cannot be renewed, or "data source unable to renew" if no additional information is available*

The SOAP 1.2 binding of this type of fault is:

- s12:Code/s12:Value = s12:Receiver
- s12:Code/s12:Subcode/s12:Value = wsen:UnableToRenew
- s12:Reason/s12:Text = *text explaining why the enumeration context cannot be renewed, or "data source unable to renew" if no additional information is available*

Other components of the outline above are not further constrained by this specification.

3.4 GetStatus

To get the status of an enumeration, the subscriber sends a request of the following form to the data source:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatus
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:FaultTo>endpoint-reference</wsa:FaultTo> ?
    <wsa:ReplyTo>endpoint-reference</wsa:ReplyTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:GetStatus ...>
      <wsen:EnumerationContext>...</wsen:EnumerationContext> ?
      ...
    </wsen:GetStatus>
  </s:Body>
</s:Envelope>
```

Components of the outline listed above are additionally constrained as for a request to renew an enumeration (see section 3.3 above). Other components of the outline above are not further constrained by this specification.

If the enumeration is valid and has not expired, the data source MUST reply with a response of the following form:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatusResponse
    </wsa:Action>
```

```

    <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
</s:Header>
<s:Body ...>
  <wsen:GetStatusResponse ...>
    <wsen:Expires>[xs:dateTime | xs:duration]</wsen:Expires> ?
    ...
  </wsen:GetStatusResponse>
</s:Body>
</s:Envelope>

```

Components of the outline listed above are constrained as for a response to a renew request (see section 3.3 above). Other components of the outline above are not further constrained by this specification.

3.5 Release

The Release operation is initiated by sending a Release request message to the data source. The Release request message MUST be of the following form:

```

<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:ReplyTo>wsa:EndpointReference</wsa:ReplyTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:Release ...>
      <wsen:EnumerationContext>...</wsen:EnumerationContext>
      ...
    </wsen:Release>
  </s:Body>
</s:Envelope>

```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Header/wsa:Action

This required element MUST contain the value `http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release`. If a SOAP Action URI is also present in the underlying transport, its value MUST convey the same value.

/s:Envelope/s:Body/wsen:Release/wsen:EnumerationContext

This required element contains the XML data that represents the enumeration context being abandoned.

Other components of the outline above are not further constrained by this specification.

Upon successful processing of a Release request message, a data source is expected to return a Release response message, which MUST adhere to the following form:

```

<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/ReleaseResponse
    </wsa:Action>
    <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>

```

```

    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body />
</s:Envelope>

```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Header/wsa:Action

This required element MUST contain the value

http://schemas.xmlsoap.org/ws/2004/09/enumeration/ReleaseResponse. If a SOAP Action URI is also present in the underlying transport, its value MUST convey the same value.

Table 6 lists a Release request.

Table 6: Release request

```

(01) <s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope'
(02)   xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
(03)   xmlns:wsen='http://schemas.xmlsoap.org/ws/2004/09/enumeration'>
(04)   <s:Header>
(05)     <wsa:Action>
(06)       http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release
(07)     </wsa:Action>
(08)     <wsa:MessageID>
(09)       uuid:e7c5726b-de29-4313-b4d4-b3425b200839
(10)     </wsa:MessageID>
(11)     <wsa:To>http://www.example.com/relayAgent</wsa:To>
(12)     <wsa:ReplyTo>
(13)       <wsa:Address>
(14)         http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
(15)       </wsa:Address>
(16)     </wsa:ReplyTo>
(17)   </s:Header>
(18)   <s:Body>
(19)     <wsen:Release>
(20)       <wsen:EnumerationContext>123</wsen:EnumerationContext>
(21)     </wsen:Release>
(22)   </s:Body>
(23) </s:Envelope>

```

Lines (05-07) in Table 6 indicate this message is a Release request and that the data source is expected to respond with a Release response message. Line (20) identifies the enumeration context to be released.

Table 7 lists a response to the request in Table 6.

Table 7: Response to Release request

```

(01) <s:Envelope xmlns:S='http://www.w3.org/2003/05/soap-envelope'
(02)   xmlns:wsen='http://schemas.xmlsoap.org/ws/2004/09/enumeration'
(03)   xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
(04) >
(05)   <s:Header>
(06)     <wsa:Action>
(07)       http://schemas.xmlsoap.org/ws/2004/09/enumeration/ReleaseResponse
(08)     </wsa:Action>
(09)     <wsa:RelatesTo>
(10)       uuid:e7c5726b-de29-4313-b4d4-b3425b200839
(11)     </wsa:RelatesTo>

```

```

(12)    <wsa:To>
(13)      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
(14)    <wsa:To>
(15)  </s:Header>
(16)  <s:Body />
(17) </s:Envelope>

```

Lines (06-08) in Table 7 indicate this message is a Release response message.

3.6 EnumerationEnd

If the data source terminates an enumeration unexpectedly, the data source SHOULD send an EnumerationEnd SOAP message to the endpoint reference indicated when the enumeration was created (see section 3.1 above). The message MUST be of the following form:

```

<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerationEnd
    </wsa:Action>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body ...>
    <wsen:EnumerationEnd ...>
      <wsen:EnumerationContext>...</wsen:EnumerationContext>
      <wsen:Code>
        [wsen:SourceShuttingDown|wsen:SourceCanceling]
      </wsen:Code>
      <wsen:Reason xml:lang="language identifier" >xs:string</wsen:Reason> ?
      ...
    </wsen:EnumerationEnd>
  </s:Body>
</s:Envelope>

```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Body/wsen:Release/wsen:EnumerationContext

This required element contains the XML data that represents the enumeration context being terminated.

/s:Envelope/s:Body/wsen:EnumerationEnd/wsen:Code =

"http://schemas.xmlsoap.org/ws/2004/09/enumeration/SourceShuttingDown"

This value MUST be used if the data source terminated the enumeration because the source is being shut down in a controlled manner; that is, if the data source is being shut down but has the opportunity to send an EnumerationEnd message before it exits.

/s:Envelope/s:Body/wsen:EnumerationEnd/wsen:Code =

"http://schemas.xmlsoap.org/ws/2004/09/enumeration/SourceCanceling"

This value MUST be used if the data source terminated the enumeration for some other reason before it expired.

/s:Envelope/s:Body/wsen:EnumerationEnd/wsen:Reason

This optional element contains text, in the language specified by the @xml:lang attribute, describing the reason for the unexpected enumeration termination.

Other components of the outline above are not further constrained by this specification.

4. Security Considerations

It is strongly recommended that the communication between services be secured using the mechanisms described in WS-Security.

In order to properly secure messages, the body (even if empty) and all relevant headers need to be included in the signature. Specifically, the WS-Addressing header blocks, WS-Security timestamp, and any header blocks resulting from a `<wsa:ReferenceProperties>` in references need to be signed along with the body in order to "bind" them together and prevent certain types of attacks.

If a requestor is issuing multiple messages to a Web service, such as when a consumer is enumerating a data source, it is recommended that a security context be established using the mechanisms described in [\[WS-SecureConversation\]](#). It is often appropriate to establish a security context that is used both for the initiation of enumeration (i.e., the Enumerate request or an equivalent service-specific request) and the actual enumeration itself (i.e., the Pull requests). It is further recommended that if shared secrets are used, message-specific derived keys should be used to protect the secret from crypto attacks.

The access control semantics of data sources is out-of-scope of this specification and are specific to each data source. Similarly, any protection mechanisms on data source independent of their transfer (e.g. embedded signatures and encryption) are also out-of-scope.

It is recommended that the security considerations of WS-Security also be considered.

While a comprehensive set of attacks is not feasible, the following list summarizes common classes of attacks that apply to this protocol and identifies the mechanism(s) to prevent/mitigate the attacks.

- **Replay** – Messages, or portions of messages, can be replayed in an attempt to gain access or disrupt services. Freshness checks such as timestamps, digests, and sequences can be used to detect duplicate messages.
- **Invalid tokens** – There are a number of token attacks including certificate authorities, false signatures, and PKI attacks. Care should be taken to ensure each token is valid (usage window, digest, signing authority, revocation, ...), and that the appropriate delegation policies are in compliance.
- **Man-in-the-middle** – The message exchanges in this specification could be subject to man-in-the-middle attacks so care should be taken to reduce possibilities here such as establishing a secure channel and verifying that the security tokens user represent identities authorized to speak for, or on behalf of, the desired resource reference.
- **Message alteration** – Alteration is prevented by including signatures of the message information using WS-Security. Care should be taken to review message part references to ensure they haven't been forged (e.g. ID duplication).
- **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using WS-Security.
- **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing secured policies – see [\[WS-Policy\]](#) and [\[WS-SecurityPolicy\]](#)) and by using derived keys ([\[WS-SecureConversation\]](#)).
- **Authentication** – Authentication is established using the mechanisms described in WS-Security and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.

- **Accountability** – Accountability is a function of the type of and string of the key and algorithms being used. In many cases, a strong symmetric key provides sufficient accountability. However, in some environments, strong PKI signatures are required.
- **Availability** – All reliable messaging services are subject to a variety of availability attacks. Replay detection is a common attack and it is recommended that this be addressed by the mechanisms described in WS-Security. Other attacks, such as network-level denial of service attacks are harder to avoid and are outside the scope of this specification. That said, care should be taken to ensure that minimal state is saved prior to any authenticating sequences.

5. Acknowledgements

This specification has been developed as a result of joint work with many individuals and teams, including:

Suwat Chitphakdibodin, Microsoft Corporation
 Omri Gazitt, Microsoft
 Gopal Kakivaya, Microsoft
 Brad Lovering, Microsoft
 Ray McCollum, Microsoft
 Jeffrey Schlimmer, Microsoft
 John Shewchuk, Microsoft

6. References

[RFC 2119]

S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," [RFC 2119](#), Harvard University, March 1997.

[SOAP 1.1]

D. Box, et al, "[Simple Object Access Protocol \(SOAP\) 1.1](#)," May 2000.

[SOAP 1.2]

M. Gudgin, et al, "[SOAP Version 1.2 Part 1: Messaging Framework](#)," June 2003.

[WS-Addressing]

D. Box, et al, "[Web Services Addressing \(WS-Addressing\)](#)," August 2004.

[WS-Policy]

S. Bajaj, et al, "[Web Services Policy Framework \(WS-Policy\)](#)," September 2004.

[WS-SecureConversation]

G. Della-Libera et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," May 2004.

[WS-Security]

A. Nadalin, et al, "[Web Services Security: SOAP Message Security V1.0](#)," March 2004.

[WS-SecurityPolicy]

G. Della-Libera, et al, "[Web Services Security Policy Language \(WS-SecurityPolicy\)](#)," December 2002.

[WSDL 1.1]

E. Christensen, et al, "[Web Services Description Language \(WSDL\) 1.1](#)," March 2001.

[XML Infoset]

J. Cowan, et al, "[XML Information Set](#)," October 2001.

[XML Schema, Part 1]

H. Thompson, et al, "[XML Schema Part 1: Structures](#)," May 2001.

[XML Schema, Part 2]

P. Biron, et al, "[XML Schema Part 2: Datatypes](#)," May 2001.

[XPath 1.0]

J. Clark, et al, "[XML Path Language \(XPath\) Version 1.0](#)," November 1999.

Appendix I – XML Schema

A normative copy of the XML Schema [[XML Schema Part 1, Part 2](#)] for this specification may be retrieved by resolving the XML namespace URI for this specification (listed in Section 2.2 XML Namespaces).

A non-normative copy of the XML schema is listed below for convenience.

```
<xs:schema
  targetNamespace="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
  xmlns:tns="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  blockDefault="#all">

  <xs:import namespace='http://www.w3.org/XML/1998/namespace' />
  <xs:import namespace='http://schemas.xmlsoap.org/ws/2004/08/addressing' />

  <!-- Types and global elements -->
  <xs:complexType name="FilterType" mixed="true">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="Dialect" type="xs:anyURI" />
    <xs:anyAttribute namespace="##other" processContents="lax" />
  </xs:complexType>

  <xs:simpleType name="PositiveDurationType">
    <xs:restriction base="xs:duration">
      <xs:minExclusive value="P0Y0M0DT0H0M0S" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="NonNegativeDurationType">
    <xs:restriction base="xs:duration">
      <xs:minInclusive value="P0Y0M0DT0H0M0S" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ExpirationType">
    <xs:union memberTypes="xs:dateTime tns:NonNegativeDurationType" />
  </xs:simpleType>

  <xs:complexType name="EnumerationContextType">
    <xs:complexContent mixed="true">
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:any namespace="##other" processContents="lax"
            minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
```

```

        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:restriction>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="ItemListType">
    <xs:sequence maxOccurs="unbounded">
        <xs:any namespace="##other" processContents="lax"
            minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="LanguageSpecificStringType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute ref="xml:lang" />
            <xs:anyAttribute namespace="##other" processContents="lax" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<!-- Enumerate request -->
<xs:element name="Enumerate">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="EndTo" type="wsa:EndpointReferenceType"
                minOccurs="0" />
            <xs:element name="Expires" type="tns:ExpirationType"
                minOccurs="0" />
            <xs:element name="Filter" type="tns:FilterType"
                minOccurs="0" />
            <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>
</xs:element>

<!-- Used for a fault response -->
<xs:element name="SupportedDialect" type="xs:anyURI" />

<!-- Enumerate response -->
<xs:element name="EnumerateResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Expires" type="tns:ExpirationType"
                minOccurs="0" />
            <xs:element name="EnumerationContext"
                type="tns:EnumerationContextType" />
            <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>
</xs:element>

```

```

<!-- Pull request -->
<xs:element name="Pull">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EnumerationContext"
        type="tns:EnumerationContextType" />
      <xs:element name="MaxTime" type="tns:PositiveDurationType"
        minOccurs="0" />
      <xs:element name="MaxElements" type="xs:positiveInteger"
        minOccurs="0" />
      <xs:element name="MaxCharacters" type="xs:positiveInteger"
        minOccurs="0" />
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax" />
  </xs:complexType>
</xs:element>

<!-- Pull response -->
<xs:element name="PullResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EnumerationContext"
        type="tns:EnumerationContextType" minOccurs="0" />
      <xs:element name="Items" type="tns:ItemListType" minOccurs="0" />
      <xs:element name="EndOfSequence" minOccurs="0" />
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax" />
  </xs:complexType>
</xs:element>

<!-- Renew request -->
<xs:element name="Renew">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EnumerationContext"
        type="tns:EnumerationContextType" />
      <xs:element name="Expires" type="tns:ExpirationType"
        minOccurs="0" />
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax" />
  </xs:complexType>
</xs:element>

<!-- Renew response -->
<xs:element name="RenewResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Expires" type="tns:ExpirationType"
        minOccurs="0" />
      <xs:element name="EnumerationContext"
        type="tns:EnumerationContextType" minOccurs="0" />
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>
</xs:element>

<!-- GetStatus request -->
<xs:element name="GetStatus">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="EnumerationContext"
                type="tns:EnumerationContextType" />
            <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>
</xs:element>

<!-- GetStatus response -->
<xs:element name="GetStatusResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Expires" type="tns:ExpirationType" minOccurs="0" />
            <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>
</xs:element>

<!-- Release request -->
<xs:element name="Release">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="EnumerationContext"
                type="tns:EnumerationContextType" />
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>
</xs:element>

<!-- Release response has an empty body -->

<!-- EnumerationEnd message -->
<xs:element name="EnumerationEnd">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="EnumerationContext"
                type="tns:EnumerationContextType" />
            <xs:element name="Code" type="tns:OpenEnumerationEndCodeType" />
            <xs:element name="Reason" type="tns:LanguageSpecificStringType"
                minOccurs="0" maxOccurs="unbounded" />
            <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>

```

```

</xs:element>

<xs:simpleType name="EnumerationEndCodeType">
  <xs:restriction base="xs:anyURI">
    <xs:enumeration value=
"http://schemas.xmlsoap.org/ws/2004/09/enumeration/SourceShuttingDown" />
    <xs:enumeration value=
"http://schemas.xmlsoap.org/ws/2004/09/enumeration/SourceCancelling" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="OpenEnumerationEndCodeType">
  <xs:union memberTypes="tns:EnumerationEndCodeType xs:anyURI" />
</xs:simpleType>
</xs:schema>

```

Appendix II – WSDL

A normative copy of the WSDL [\[WSDL 1.1\]](http://schemas.xmlsoap.org/ws/2004/09/enumeration/enumeration.wsdl) description for this specification may be retrieved from the following address:

```
http://schemas.xmlsoap.org/ws/2004/09/enumeration/enumeration.wsdl
```

A non-normative copy of the WSDL description is listed below for convenience.

```

<wsdl:definitions
  targetNamespace="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
  xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
  xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" >

  <wsdl:types>
    <xs:schema
      targetNamespace="http://schemas.xmlsoap.org/ws/2004/09/enumeration">
      <xs:include schemaLocation="enumeration.xsd" />
    </xs:schema>
  </wsdl:types>

  <wsdl:message name="EnumerateMsg">
    <wsdl:part name="body" element="wsen:Enumerate" />
  </wsdl:message>
  <wsdl:message name="EnumerateResponseMsg">
    <wsdl:part name="body" element="wsen:EnumerateResponse" />
  </wsdl:message>
  <wsdl:message name="PullMsg">
    <wsdl:part name="body" element="wsen:Pull" />
  </wsdl:message>
  <wsdl:message name="PullResponseMsg">
    <wsdl:part name="body" element="wsen:PullResponse" />
  </wsdl:message>
  <wsdl:message name="RenewMsg" >
    <wsdl:part name="body" element="wsen:Renew" />
  </wsdl:message>
  <wsdl:message name="RenewResponseMsg" >
    <wsdl:part name="body" element="wsen:RenewResponse" />
  </wsdl:message>
  <wsdl:message name="GetStatusMsg" >
    <wsdl:part name="body" element="wsen:GetStatus" />
  </wsdl:message>

```

```
<wsdl:message name="GetStatusResponseMsg" >
  <wsdl:part name="body" element="wsen:GetStatusResponse" />
</wsdl:message>
<wsdl:message name="ReleaseMsg">
  <wsdl:part name="body" element="wsen:Release" />
</wsdl:message>
<wsdl:message name="ReleaseResponseMsg" />
<wsdl:message name="EnumerationEndMsg" >
  <wsdl:part name="body" element="EnumerationEnd" />
</wsdl:message>

<wsdl:portType name="DataSource">
  <wsdl:operation name="EnumerateOp">
    <wsdl:input message="wsen:EnumerateMsg" />
    <wsdl:output message="wsen:EnumerateResponseMsg" />
  </wsdl:operation>
  <wsdl:operation name="PullOp">
    <wsdl:input message="wsen:PullMsg" />
    <wsdl:output message="wsen:PullResponseMsg" />
  </wsdl:operation>
  <wsdl:operation name="RenewOp" >
    <wsdl:input message="wsen:RenewMsg" />
    <wsdl:output message="wsen:RenewResponseMsg" />
  </wsdl:operation>
  <wsdl:operation name="GetStatusOp" >
    <wsdl:input message="wsen:GetStatusMsg" />
    <wsdl:output message="wsen:GetStatusResponseMsg" />
  </wsdl:operation>
  <wsdl:operation name="ReleaseOp">
    <wsdl:input message="wsen:ReleaseMsg" />
    <wsdl:output message="wsen:ReleaseResponseMsg" />
  </wsdl:operation>
  <wsdl:operation name="EnumerationEndOp" >
    <wsdl:output message="wsen:EnumerationEnd" />
  </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```