

## **WS-Iterator 1.0**

### Status of This Memo

This memo provides information to the Grid community on simple, content agnostic iteration through XML data in a manner which is both consistent with, and compliant with WSRF. Distribution is unlimited.

### Copyright Notice

Copyright © Open Grid Forum (2006-2009). All Rights Reserved.

### Trademarks

OGSA is a trademark of the Open Grid Forum.

### **Abstract**

A number of grid services aggregate data together in lists or maps as part of their inherent function. Consider RNS (the Resource Namespace Specification) [**RNS**] which provides the means of mapping human readable names to resource endpoints. Also consider a grid queue or cluster management system that might both group together target or backend BESs (Basic Execution Services) [**BES**] as well as provide mechanisms for querying or manipulating lists of queued jobs. Numerous other examples exist. In both of these cases, it is unreasonable and inefficient to expect communication where the entire content of such a group is transferred in a single SOAP [**SOAP1.1**] document. At the same time, given the potentially large and diverse spectrum of likely uses for which iteration might be ideal, a generic form of iteration is desirable – one for which iterable content is extensible.

A number of iteration service specifications exist already; in particular WS-Enumeration provides similar functionality. Unfortunately, WS-Enumeration [**WS-Enumeration**] is based off of an entirely different model of web service endpoint interaction than that of WSRF. While WSRF has adopted a notion of the “implied” target resource as identified by WS-Addressing [**WS-Addressing**] information included in the request message's SOAP headers, WS-Enumeration uses a more service oriented, “token-in-the-soap-body” protocol. As such, in cases where grid service design is heavily influenced by and modeled after a WSRF pattern, WS-Enumeration is confusing and ungainly. In those cases, WS-Iterator provides the same functionality in a way that is more consistent with intended WSRF practices. For a more detailed description of why the existing WS-Enumeration specification is not suited to this task of WSRF-based iteration, please reference [**CS-2008-9**].

## Table of Contents

1	Introduction.....	3
1.1	Outline for this Document .....	3
1.2	Terminology .....	3
1.3	Namespaces.....	4
2	WS-Iterator Port Type .....	4
2.1	WS-Iterator Interface .....	4
2.1.1	WS-Iterator elementCount Property .....	5
2.1.2	WS-Iterator preferredBlockSize Property.....	5
2.2	WS-Iterator iterate Operation.....	5
2.2.1	WS-Iterator iterate.....	5
2.2.2	Example SOAP Encoding of the iterate Message Exchange .....	6
3	WS-Iterator Properties .....	9
4	Faults and Failures .....	10
5	Static versus Dynamic Content .....	10
6	Security Considerations .....	10
7	Author Information .....	11
8	Glossary .....	11
9	Intellectual Property Statement.....	11
10	References.....	12
Appendix A:	WS-Iterator Schema .....	14
Appendix B:	WS-Iterator WSDL.....	16

## 1 Introduction

A number of grid services aggregate data together in lists or maps as part of their inherent function. Consider RNS (the Resource Namespace Specification) [RNS] which provides the means of mapping human readable names to resource endpoints. Also consider a grid queue or cluster management system that might both group together target or backend BESs (Basic Execution Services) [BES] as well as provide mechanisms for querying or manipulating lists of queued jobs. Numerous other examples exist. In both of these cases, it is unreasonable and inefficient to expect communication where the entire content of such a group is transferred in a single SOAP [SOAP1.1] document. At the same time, given the potentially large and diverse spectrum of likely uses for which iteration might be ideal, a generic form of iteration is desirable – one for which iterable content is extensible.

A number of iteration service specifications exist already; in particular WS-Enumeration provides similar functionality. Unfortunately, WS-Enumeration [WS-Enumeration] is based off of an entirely different model of web service endpoint interaction than that of WSRF. While WSRF has adopted a notion of the “implied” target resource as identified by WS-Addressing [WS-Addressing] information included in the request message's SOAP headers, WS-Enumeration uses a more service oriented, “token-in-the-soap-body” protocol. As such, in cases where grid service design is heavily influenced by and modeled after a WSRF pattern, WS-Enumeration is confusing and ungainly. In those cases, WS-Iterator provides the same functionality in a way that is more consistent with intended WSRF practices. For a more detailed description of why the existing WS-Enumeration specification is not suited to this task of WSRF-based iteration, please reference [CS-2008-9].

### 1.1 Outline for this Document

The remainder of this document will be organized as follows. First, we will present a high level overview of the port type we recommend for the WS-Iterator specification. We will follow this with sections that drill down into the details of the port type. Because this specification is specific to a WSRF style of rendering, it is possible to provide normative message examples and WSDL<sup>1</sup> and those components will be included. Finally, we will summarize the information in this document and wrap up with information about security considerations, author information, and glossary terms.

### 1.2 Terminology

The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

---

<sup>1</sup> This specification document is modeled after the OGSA ByteIO specification document. However, in the case of ByteIO, the port types described were described in a rendering agnostic way (they did not require WSRF or any other specific OGSA Basic Profile rendering). Rather, ByteIO expects that various “rendering” documents will be provided to augment and complete the rendering neutral specification.

In addition to the terms introduced in **[RFC2119]**, additional terms commonly used in this document are defined in the Glossary in the back.

When describing abstract data models, this specification uses the notational convention used by the **[XML Infoset]**.

When describing concrete XML schemas, this specification uses the notational convention of **[WS-Security]**. Specifically, each member of an element's [children] or [attributes] property is described using an Xpath-like **[XPath]** notation (e.g., /x:MyHeader/x:SomeProperty@value1). The use of {any} indicates the presence of an element wildcard (<xsd:any/>). The use of @{any} indicates the presence of an attribute wildcard (<xsd:anyAttribute/>).

### 1.3 Namespaces

The following namespaces are used in this document:

Prefix	Namespace
s11	<a href="http://schemas.xmlsoap.org/soap/envelope">http://schemas.xmlsoap.org/soap/envelope</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
wsa	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>
iterator	<a href="http://schemas.ogf.org/ws-iterator/2008/06/iterator">http://schemas.ogf.org/ws-iterator/2008/06/iterator</a>

## 2 WS-Iterator Port Type

The WS-Iterator port type is a very simple port type consisting of exactly one operation (for retrieving a range of contained elements) and a number of resource properties describing the contained elements. Further, as an OGSA Basic Profile compliant service port type, it requires the addition of a handful of other WSRF specifications. Specifically, all web services that implement the WS-Iterator port type **MUST** also be compliant with the OGSA WSRF Basic Profile 1.0 **[WSRFProfileDoc]** which requires the service to implement a number of other WSRF port types having to do with resource properties and service resource lifetime.

### 2.1 WS-Iterator Interface

The WS-Iterator port allows clients to retrieve small subsets of a stored set of arbitrary XML documents. It is up to other specifications that make use of WS-Iterator to define the exact contents of these contained documents. The WS-Iterator interface is conceptually defined as follows:

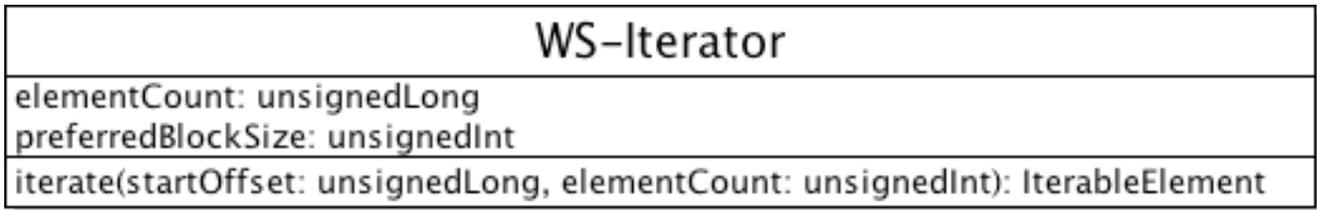


Figure 1: WS-Iterator Pseudo-UML

### 2.1.1 WS-Iterator elementCount Property

The elementCount property is a required WSRF Resource Property describing the total number of elements contained within this iterator. This WSRF Resource Property has a cardinality of exactly 1.

### 2.1.2 WS-Iterator preferredBlockSize Property

The preferredBlockSize property is an optional WSRF Resource Property describing the preferred number of elements that a client should ask for in any one request. This WSRF Resource Property has a cardinality of [0, 1].

## 2.2 WS-Iterator iterate Operation

The iterate operation is used by clients who wish to retrieve a subset of the elements stored within a given iterator. The WS-Iterator **MUST** respond to an iterate request message with an iterateResponse message. The iterator **MAY** respond with fewer elements than requested (for example, because the services determines that the SOAP message for the requested number of items would be too large), however it **MUST** not respond with zero elements unless the requested block of elements contains no elements (i.e., the request is for a block of elements past the end of the iterator's current list of elements). The iterateResponse message contains an element indicating the iterator's current size (total number of contained elements). This gives clients a convenient way of determining whether or not there are more elements available from the iterator although clients **SHOULD** consider this iterator size to be dynamic.

### 2.2.1 WS-Iterator iterate

The format of the iterate Message is:

```

...
<iterator:iterate>
  <iterator:start-offset> xsd:unsignedLong </iterator:start-offset>
  <iterator:element-count> xsd:unsignedInt </iterator:element-count>
</iterator:iterate>
...

```

The components of the iterate message are further described as follows:

/iterator:start-offset

This element gives the index of the first element (0-based) within the iterator resource for which the client wishes to obtain a subset.

/iterator:element-count

This element gives the maximum number of elements that the client wishes to retrieve from the target iterator resource. The iterator MAY return fewer elements than specified but may not return zero elements unless the start-offset is beyond the current end of the iteration contents list.

The response to the iterate message is a message of the form:

```
...
<iterator:iterateResponse>
  <iterator:iterator-size> xsd:unsignedLong </iterator:iterator-size>
  <iterator:iterable-element index="xsd:unsignedInt">
    {any} *
  </iterator:iterable-element> *
</iterator:iterateResponse>
```

The components of the iterateResponse message are further described as follows:

/iterator:iterator-size

This element represents the current size of an iterator (the current total number of elements stored in the iterator).

/iterator:iterable-element

This element represents a single element of data stored by the iterator. The contents of this element are the exact documents stored as a single element within the iterator.

/iterator:iterable-element@index

This property indicates the global index of the element represented by this iterable element. This index is 0-based and MUST be consecutively numbered within the response message.

## 2.2.2 Example SOAP Encoding of the iterate Message Exchange

The following is a non-normative example of an iterate request message using **[SOAP1.1]**:

```
<s11:Envelope
  xmlns:s11="http://www.w3.org/2003/05/soap-envelope"
  xmlns:iterator="http://schemas.org/ws-iterator/2008/06/iterator">
  <s11:Header>
    <wsa:Action>
      http://schemas.org/ws-iterator/2008/06/iterator/iterate
    </wsa:Action>
    <wsa:To s11:mustUnderstand="1">
      http://tempuri.org/iterator-source
    </wsa:To>
  </s11:Header>

  <s11:Body>
    <iterator:iterate>
      <iterator:start-offset>1000</iterator:start-offset>
      <iterator:element-count>5</iterator:element-count>
    </iterator:iterate>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of an iterate response message using **[SOAP1.1]**:

```

<s11:Envelope
  xmlns:s11="http://www.w3.org/2003/05/soap-envelope"
  xmlns:iterator="http://schemas.ogf.org/ws-iterator/2008/06/iterator">
  <s11:Header>
    <wsa:Action>
      http://schemas.ogf.org/ws-iterator/2008/06/iterator/iterateResponse
    </wsa:Action>
    <wsa:To s11:mustUnderstand="1">
      http://schemas.xmlsoap.org/ws/2004/03/addressing/role/anonymous
    </wsa:To>
  </s11:Header>

  <s11:Body>
    <iterator:iterateResponse>
      <iterator:iterator-size>1001</iterator:iterator-size>
      <iterator:iterable-element index="1000">
        <ns1:user-info xmlns:ns1="http://tempuri.org/some-
namespace">
          some unknown content
        </ns1:user-info>
      </iterator:iterable-element>
    </iterator:iterateResponse>
  </s11:Body>

```

### 3 WS-Iterator Properties<sup>2</sup>

The following table indicates the properties that WS-Iterator resources contain. The *Requirement Level* entry in the table describes whether a WS-Iterator resource MUST have the property, SHOULD have the property, or MAY have the property (as per [RFC2119]). This list is by no means exhaustive and implementers are free to add their own properties as they see fit.

Property	Requirement Level	Description
elementCount	MUST	The total number of elements contained in the target WS-Iterator resource.
preferredBlockSize	MAY	The preferred number of elements to retrieve in any given iterate request message.

<sup>2</sup> These resource properties are in addition to any resource properties required or specified by the OGSA WSRF Basic Profile 1.0 on which this specification is dependent.

These properties are further described normatively as follows:

```
...
targetNamespace="http://schemas.ogf.org/ws-iterator/2008/06/iterator"
...
<xsd:element name="elementCount" type="xsd:unsignedLong"
  minOccurs="1" maxOccurs="1"/>
<xsd:element name="preferredBlockSize" type="xsd:unsignedInt"
  minOccurs="0" maxOccurs="1"/>
```

#### 4 Faults and Failures

The WS-Iterator port type requires no faults or failure messages beyond those indicated by WSRF and by the OGSA WSRF Basic Profile 1.0. Any client requesting an invalid range of elements will simply receive an empty response message. Any client requesting more elements than the iterator has available will receive an appropriately truncated response message.

#### 5 Static versus Dynamic Content

While it is anticipated that most services or port types that make use of the WS-Iterator port type will instantiate static or “snapshot” instances (i.e., resources of WS-Iterators which can iterate over a static snapshot of another resources state or view of that state), this author sees no reason to outright prohibit the creation of a WS-Iterator with dynamic content. It is expected (though no normative declarations will be made here) that such a “dynamic” iterator would have the following properties:

- A client asking for a specific subset of the iterator's elements might receive different results from one invocation of the iterate operation to the next
- A client asking for a specific subset of the iterator's elements might receive different numbers of resultant elements from one invocation of the iterate operation to the next.
- The resource property describing a target iterator's contents might change over time
- The iterator-size element of an iterateResponse message might indicate a iterator-size that is out-of-date with the actual contents of the iterator.
- Some form of notification about an iterator's contents changing SHOULD be available

## 6 Security Considerations

Security is, of course, important for WS-Iterators. Considering that WS-Iterators will likely be used by service port type implementations to return sets or lists of data that might, in their own right, be considered sensitive, WS-Iterators will likely (though not required by this specification) inherit the security requirements of their source resources. However, as always, security is a cross-cutting concern here and as such specification thereof lies outside the purview of this document.

## 7 Author Information

Editor:  
Mark Morgan  
University of Virginia, Department of Computer Science  
151 Engineer's Way  
P.O. Box 400740  
Charlottesville, VA. 22904-4740  
Phone: +1 (434) 243-2175  
E-mail: [mmm2a@virginia.edu](mailto:mmm2a@virginia.edu)

## 8 Glossary

**Conceptual Interface** An interface which describes the conceptual behavior of a service but which doesn't necessarily reflect the actual parameters and methods that are being received and sent.

## 9 Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Directory.

## Full Copyright Notice

Copyright © Open Grid Forum (2006-2010). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assigns.

This document and the information contained herein is provided on an “AS IS” basis and THE OPEN GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## 10 References

- [WSRFProfileDoc] I. Foster, T. Maguire, D. Snelling, *OGSA WSRF Basic Profile 1.0*, <https://forge.gridforum.org/projects/ogsa-wg/document/draft-ggf-ogsa-wsrf-basic-profile/en/15>, GWS-R (draft-ggf-ogsa-wsrf-basic-profile-021), 6 July 2005.
- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2991.txt>, IETF RFC 2119, March 1997.
- [XML-Infoset] <http://www.w3.org/TR/xml-infoset/>
- [XPath] <http://www.w3.org/TR/xpath>
- [WS-Addressing] M. Gudgin, M. Hadley, and T. Rogers (ed.), *Web Services Addressing 1.0 – Core (WS-Addressing)*, 9 May 2006, <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509>
- [WS-Enumeration] <http://www.w3.org/Submission/WS-Enumeration/>
- [CS-2008-9] M. Morgan, *WS-Enumeration and OGSA: A Clash of Cultures*, <http://vcgr.cs.virginia.edu/documents/CS-2008-9.pdf>, CS-2008-9, July 2008
- [ByteIOSpec] M. Morgan (ed.), *ByteIO Specification 1.0*, <http://www.ggf.org/documents/GFD.88.pdf>, GFD.88, 31 October 2006.

- [ByteIOWSRFRend]** M. Morgan (ed.), *ByteIO OGSA WSRF Basic Profile Rendering 1.0*, <http://www.ggf.org/documents/GFD.87.pdf>, GFD.87, 31 October 2006.
- [BES]** <http://www.ggf.org/documents/GFD.108.pdf>
- [SOAP1.1]** <http://www.w3.org/TR/soap11>
- [RNS]** <http://www.ggf.org/documents/GFD.101.pdf>

## Appendix A: WS-Iterator Schema

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Directory.

Full Copyright Notice

Copyright © Open Grid Forum (2006-2008). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE OPEN GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

```
-->
```

```
<xsd:schema
```

```
  xmlns="http://schemas.ogf.org/ws-iterator/2008/06/iterator"
```

```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
  xmlns:iterator="http://schemas.ogf.org/ws-iterator/2008/06/iterator"
```

```
  xmlns:wsbf="http://docs.oasis-open.org/wsrf/bf-2"
```

```
  targetNamespace="http://schemas.ogf.org/ws-iterator/2008/06/iterator"
```

```
  elementFormDefault="qualified"
```

```
  attributeFormDefault="unqualified">
```

```
  <xsd:import
```

```
    namespace="http://docs.oasis-open.org/wsrf/bf-2"
```

```
    schemaLocation="http://docs.oasis-open.org/wsrf/bf-2.xsd"/>
```

```

<!-- Resource Property Related -->
  <xsd:element name="elementCount" type="xsd:unsignedLong"/>
  <xsd:element name="preferredBlockSize" type="xsd:unsignedInt"/>

<!-- Message Related -->
  <xsd:complexType name="IterableElementType">
    <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="index" type="xsd:unsignedInt"
      use="required"/>
  </xsd:complexType>

<!-- Message Types -->
  <xsd:complexType name="IterateRequestType">
    <xsd:sequence>
      <xsd:element name="start-offset" type="xsd:unsignedLong"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="element-count" type="xsd:unsignedInt"
        minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="IterateRequestType" type="iterator:IterateRequestType"/>

  <xsd:complexType name="IterateResponseType">
    <xsd:sequence>
      <xsd:element name="iterator-size" type="xsd:unsignedLong"
        minOccurs="1" maxOccurs="1" nillable="false"/>
      <xsd:element name="iterable-element" type="iterator:IterableElementType"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="IterateResponseType"
    type="iterator:IterateResponseType"/>
</xsd:schema>

```

## Appendix B: WS-Iterator WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Directory.

Full Copyright Notice

Copyright © Open Grid Forum (2006-2008). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assigns.

This document and the information contained herein is provided on an “AS IS” basis and THE OPEN GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

```
-->
```

```
<wsdl:definitions
```

```
  name="WSIterator"
```

```
  xmlns="http://schemas.ogf.org/ws-iterator/2008/06/iterator"
```

```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
  xmlns:iterator="http://schemas.ogf.org/ws-iterator/2008/06/iterator"
```

```
  xmlns:wsbf="http://docs.oasis-open.org/wsrf/bf-2"
```

```
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```

```
  xmlns:genii-common="http://vcgr.cs.virginia.edu/genii/2006/12/common"
```

```
  xmlns:rpw-2="http://docs.oasis-open.org/wsrf/rpw-2"
```

```
  xmlns:rw-2="http://docs.oasis-open.org/wsrf/rw-2"
```

```
  xmlns:wsrp="http://docs.oasis-open.org/wsrf/rp-2"
```

```
  xmlns:wsrpw="http://docs.oasis-open.org/wsrf/rpw-2"
```

```
xmlns:wsrf-r="http://docs.oasis-open.org/wsrf/r-2"
xmlns:wsrf-rw="http://docs.oasis-open.org/wsrf/rw-2"
xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
targetNamespace="http://schemas.ogf.org/ws-iterator/2008/06/iterator">
```

```
<wSDL:import
  namespace="http://docs.oasis-open.org/wsrf/rw-2"
  location="http://docs.oasis-open.org/wsrf/rw-2.wSDL"/>
```

```
<wSDL:import
  namespace="http://docs.oasis-open.org/wsrf/rpw-2"
  location="http://docs.oasis-open.org/wsrf/rpw-2.wSDL"/>
```

```
<wSDL:import
  namespace="http://docs.oasis-open.org/wsrf/rlw-2"
  location="http://docs.oasis-open.org/wsrf/rlw-2.wSDL"/>
```

```
<wSDL:types>
  <xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    attributeFormDefault="unqualified"
    elementFormDefault="qualified"
    targetNamespace="http://schemas.ogf.org/ws-iterator/2008/06/iterator">
```

```
  <xsd:import
    namespace="http://schemas.ogf.org/ws-iterator/2008/06/iterator"
    schemaLocation="./iterator.xsd"/>
```

```
  <xsd:import
    namespace="http://docs.oasis-open.org/wsrf/bf-2"
    schemaLocation="http://docs.oasis-open.org/wsrf/bf-2.xsd"/>
```

```
  <xsd:import
    namespace="http://docs.oasis-open.org/wsrf/rp-2"
    schemaLocation="http://docs.oasis-open.org/wsrf/rp-2.xsd"/>
```

```
  <xsd:import
    namespace="http://docs.oasis-open.org/wsrf/rl-2"
    schemaLocation="http://docs.oasis-open.org/wsrf/rl-2.xsd"/>
```

```
<!-- == Resource Property Related == -->
  <xsd:element name="WSIteratorRP">
    <xsd:complexType>
      <xsd:sequence>
```

```

        <xsd:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
        <xsd:element ref="iterator:elementCount"
            minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="iterator:preferredBlockSize"
            minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>

<wsdl:message name="IterateRequest">
    <wsdl:part name="IterateRequest"
        element="iterator:IterateRequestType"/>
</wsdl:message>

<wsdl:message name="IterateResponse">
    <wsdl:part name="IterateResponse"
        element="iterator:IterateResponseType"/>
</wsdl:message>

<wsdl:portType name="WSIteratorPortType"
    wsrp:ResourceProperties="iterator:WSIteratorRP">

<!-- Strictly speaking one would insert here operations that were being "imported" from other port
types (specifically, those required by the OGSA WSRF Base Profile 1.0). However, doing so would
tremendously bloat and obfuscate the material relevant to this document and as such I merely introduce
via comments the other port types that should be added. There are a few mechanisms for doing this
such as Globus' wsdlpp:extends element and the Genesis II genii-ext:extend element. However, since
neither of these mechanisms is standard and each grid provider currently has its own methods for
accomplishing this, we resort simply to comments here. -->
<!-- extends wsrpw:ResourceProperties-->
<!-- extends wsrlw:ImmediateTerminate-->
<!-- extends wsrlw:ScheduledTermination-->

    <wsdl:operation name="iterate">
        <wsdl:input message="iterator:IterateRequest"/>
        <wsdl:output message="iterator:IterateResponse"/>
    </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>

```