International

Virtual

Observatory

Alliance

IVOA Web Services Basic Profile

Version 0.24

IVOA Working Draft 2006 September 18

This version:
0.24 http://www.ivoa.net/internal/IVOA/VO-WS-Basic-Profile-0.24.pdf

Previous versions:
0.23 http://www.ivoa.net/internal/IVOA/VO-WS-Basic-Profile-0.23.pdf
0.22 http://www.ivoa.net/internal/IVOA/VO-WS-Basic-Profile-0.22.pdf
0.21 http://www.ivoa.net/internal/IVOA/VO-WS-Basic-Profile-0.21.pdf
0.2 http://www.ivoa.net/internal/IVOA/VO-WS-Basic-Profile-0.2.pdf
0.1 http://www.ivoa.net/internal/IVOA/VO-WS-Basic-Profile-0.1.pdf

Editors:
André Schaaff

Authors:
Web and Grid Services Working Group

Please send comments to: mailto:grid@ivoa.net

Abstract

This document describes rules to take into account when implementing Web Services.
It explains also how to check the conformance to these rules.
It can be resumed as a “Guideline for VO Web Services interoperability” or a “How to provide interoperable VO Web Services”.

Status of This Document

This is a Working Draft. The first release of this document was 2004 July 05.

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress".

A list of current IVOA Recommendations and other technical documents *can be found at http://www.ivoa.net/Documents/.*

Acknowledgements

This work is based on discussions [1] [9] [12] [13] and actions from the interoperability meetings in Cambridge MA, 24-28 May 2004, in Pune, 27-29 September 2004, in Kyoto, 16-20 May 2005 and in Madrid, 6-7 October 2005.

Contents

1. Introduction

The use of Web Services is increasing and it is foreseeable that many VO partners will provide services through this way in a near future.
VO Web Services providers need a guideline on how to use the existing specifications in the IVOA Web Services context.
This guideline should be an “interoperability guarantee” for the future.
Our goal is not to create this guideline from scratch but to base it on existing works.
We have to decide which part of existing profiles we want to use, which part we want to replace with our own work and which part we want to add.

2. WS-I [2] and the Basic Profile [3]

The Web Services Interoperability [2] organization is an open industry effort chartered to promote Web Services interoperability across platforms, applications, and programming languages. Its role is not to develop new specifications (like the W3C for example) but to interpret the existing ones and to explain how to make them work together in the best way.

The WS-I Basic Profile is a set of non-property Web Service specifications (SOAP, WSDL, UDDI, XML, XML Schema …)**.**

It provides clarifications because:

- Using a specification is very well but using it correctly and in the same way than others is better for a good interoperability
- Specifications are often ambiguous

WS-I Basic Profile is supported by the world major companies and working groups.

Examples:
On Microsoft web pages [7]: “Microsoft applauds the ratification of the Basic Profile 1.0…”

On Apache Axis web pages [8]: “For Axis 1.2, we are focusing on our document/literal support to better address the WS-I Basic Profile 1.0 …”

2.1 WS-I Basic Profile Goal

The WS-I “Basic Profile 1.1” describes:

- Messaging: exchange of Web service protocol elements
- Description: enumeration of the messages associated with a Web service, with implementation details
- Discovery: metadata which gives information about the Web Service
- Security: mechanism which provides integrity, confidentiality authentication

Remarks:
Concerning the Discovery topic, IVOA has not decided to adopt UDDI, so this part may be considered as replaced with IVOA own work.
Work about security is also undergoing at IVOA and it will be necessary to explore WS-I work in this domain.

2.2 WS-I Basic Profile content

In each part (HTTP, SOAP binding, etc.) the profile explains recommendations with the following format:

Rxxxx *statement text*

Examples:
R0001 *An Instance of a Web service MUST be defined by a WSDL service description*
R1140 *A message SHOULD be sent using HTTP/1.1*
R1141 *A message MUST be sent using either HTTP/1.1 or HTTP/1.0*

Before each rule or set of rules, the document explains the context and justifies the rule creation.
The rules are not all at the same level, the compliance to one rule can be mandatory and the compliance to another can be optional.

See Appendix about RFC 2119 for additional information about the use of “MUST”, “SHOULD”…

3. WS-I Simple SOAP Binding Profile [5]

WS-I Basic Profile 1.0 + errata is equivalent to WS-I Basic Profile 1.1 + WS-I Simple SOAP Binding Profile.

Simple SOAP Binding Profile 1.0 is a “subset” of the Basic Profile 1.0 requirements related to the serialization of the envelope and its representation in the message.
Web Services can be checked following this profile when they do not use attachments.

4. WS-I Attachment Profile [4]

Adds support for sending interoperable attachments with SOAP messages.
It defines a MIME (*Multipurpose Internet Mail Extensions*) multipart structure for packaging attachments with SOAP messages.
WS-I has chosen the most common solution and it is too restrictive.
But it is not mandatory to associate this profile to the WS-I Basic Profile 1.1
Possible solutions (VO-Ready attachments) must be defined and in any case, these solutions must be available on the most common Axis implementations. Service providers could implement other solution for internal exchange or in addition to VO-Ready attachments.
The door must be open for solutions like DIME (*Direct Internet Message Encapsulation*), MTOM (*Message Transmission Optimization Mechanism, inside attachments*) …
Decision taken at Kyoto: no attachment profile for the moment.

5. WS-I Basic Security Profile [11]

Decision taken at Kyoto: Not now.

6. WS-I Testing Tools [6]

6.1 Monitor and Analyzer

It is probably unattractive to check “by hand” every rule of the Basic Profile, so the WS-I has developed conformance testing tools. The first provided tool is a Monitor and Analyzer package.
These tools are based on configuration files which allow the user to enabled/disabled rules to tests (assertion files). It is possible to define a core assertion file to check in the context of IVOA Web Services.

6.2 Experiment

The conformance testing-tools have been experimented for Tomcat/Axis and .NET in the context of the VO. The result has been show at the Pune interoperability meeting in September 2004, based on the WS-I Basic Profile 1.0.
After the release of the WS-I conformance testing tools (for WS-I Basic Profile 1.1 and WS-I Simple SOAP Binding 1.0) in November 2004, the test has been done again.

6.3 Remark about use

Due to a very restrictive license, these tools and their related libraries cannot be integrated in other tools. But it is very easy to download and use these tools (available in Java and C#).

7. Toward a VO Web Service Basic Profile

7.1 Kickoff

At the Pune interoperability meeting it has been decided to create a VO Web Service Basic Profile based both on non IVOA recommendations and IVOA recommendations (like future VO Support Interfaces 1.0). This profile is intended to provide a guideline about how to implement a VO interoperable Web Service. It has also been decided to provide tools to check the conformance.

7.1 Aim

The aim is to define rules a VO Web Service should follow to be VO compliant.
New technologies are often published and it is very exciting to implement these in new or existing services but some of these technologies are not available at the same time for all the SOAP implementations. So, it is very important to define a basic common set of rules to maintain a high level of interoperability.

8. Conformance

8.1 Scope

Non IVOA Recommendations and IVOA own requirements (based on IVOA recommendations) must be detailed in this part.

8.2 Conformance to non IVOA recommendations

8.2.1 Rules

R0001 An IVOA Web Service **MUST** be compliant to the WS-I Basic Profile 1.1
R0002 An IVOA Web Service **MUST** be compliant to the WS-I Simple SOAP Binding 1.0

These rules are validated through the WS-I Testing tools [6].

Rules concerning attachments, ... have been deleted but will follow the decisions taken in this domain by the IVOA members.

8.3 IVOA Support Interfaces conformance [10]

8.3.1 Remarks

0.24 is the last version of VO Support Interfaces. Since this version, two new specifications are continuing the work :

- IVOA Support Interfaces : Mandatory Interfaces [14][15]
- IVOA Support Interfaces : Optional Logging Interfaces[16]

8.3.2 Rules concerning Mandatory Interfaces

R0100 All VO services **SHOULD** implement the "getRegistration" interface.

R0101 The "getRegistration" interface **SHALL** return a valid VOResource document describing the metadata of this service.

R0110 A VO service **SHOULD** implement the "registrationChangedOn" interface.

R0111 The "registrationChangedOn" interface **SHALL** return the date the metadata last changed.

R0120 A VO service **SHALL** implement the "getAvailability" interface.

R0121 The "getAvailability" interface **SHALL** return an XML document as defined in the schema availability

8.3.3 Rules concerning Non Mandatory Interfaces

R0200 A VO service **SHOULD** implement the "harvestWebLog" interface

R0201 The "harvestWebLog" interface **SHOULD** take three parameters "fromDate", "toDate", and "format", and should return a URL that points to a file containing a serialization of a set of WebLogEntry in the chosen format

R0202 A WebLogEntry **SHOULD** contain the following information:...

R0210 A VO services **SHOULD** implement the "harvestServiceLog" interface.

R0211 The "harvestServiceLog" interface **SHOULD** take three parameters "fromDate", "toDate", "notifyEmail" and "destination"

R0212 The "harvestServiceLog" interface **SHOULD** return a URL that point to a file containing a serialization of a set of ServiceLogEntry

R0213 A ServiceLogEntry **SHOULD** contain the following information:...

9. Conformance checking

9.1 Tools

The IVOA provides a tool to check the conformance to the profile (for IVOA recommendations). In any case, this tool will not be a debugger of Web Services.
Concerning the checking tools provided by the WS-I the license is very restrictive: it is not possible to integrate these tools in an IVOA tool (no distribution, no license transfer …). A guide about how to run quickly the WS-I conformance testing tool could be provided if needed.

9.2 Use cases

Use cases should be defined at the service provider level. The definition of this use cases and the problems due to bugs are out of the scope of the profile checking.

9.3 Assertion definition for IVOA recommendation checking

9.2.1 Support Interfaces case

```
<?xml version="1.0" encoding="UTF-8"?>
<vowsbasicprofile version="1.0">
<description>
    This document contains the test assertions for the
    IVOA WS Basic Profile
 </description>

<testAssertion id="R0100" type="required" enabled="true">
<context>Support Interfaces : Mandatory Interfaces</context>
<assertionDescription>The heartbeat interface is to tell us if the service is
in operation. It should do a good check on the underlying service to see if
it is still operational and not just be a simple return from a web server, e.g.,
if it relies on a database it should check that the database is still up (see
Support Interfaces Mandatory Interfaces).
</assertionDescription>
<failureMessage>The mandatory "getRegistration" interface is
missing</failureMessage>
<failureDetailDescription>...</failureDetailDescription>
<testToDo>
<InterfaceChecking>getRegistration</InterfaceChecking>
</testToDo>
</testAssertion>
```

```
<testAssertion id="R0101" type="required" enabled="true">
<context>Support Interfaces : Mandatory Interfaces</context>
<assertionDescription>All VO services should implement the
“getRegistration” interface. This shall return a valid VOResource
document describing the metadata of this service.</assertionDescription>
<failureMessage> getRegistration do not return an XML file conform to
VOResource schema</failureMessage>
<failureDetailDescription>...</failureDetailDescription>
<testToDo>
<OutputChecking type="XSD"
interface="getRegistration">http://www.ivoa.net/internal/IVOA/VOResourc
eV10/VOResource-v1.0.xsd</OutputChecking>
</testToDo>
</testAssertion>

<testAssertion id="R0120" type="required" enabled="true">
<context>Support Interfaces : Mandatory Interfaces</context>
<assertionDescription>The heartbeat interface is to tell us if the service is
in operation. It should do a good check on the underlying service to see if
it is still operational and not just be a simple return from a web server, e.g.,
if it relies on a database it should check that the database is still up (see
Support Interfaces Mandatory Interfaces).
</assertionDescription>
<failureMessage>The mandatory "getAvailability" interface is
missing</failureMessage>
<failureDetailDescription>...</failureDetailDescription>
<testToDo>
<InterfaceChecking>getAvailability</InterfaceChecking>
</testToDo>
</testAssertion>

<testAssertion id="R0111" type="required" enabled="true">
<context>Support Interfaces : Mandatory Interfaces</context>
<assertionDescription>All VO services should implement the
“registrationChangedOn” interface. This shall return the date the metadata
last changed.
</assertionDescription>
<failureMessage>The mandatory "registrationChangedOn" interface is
missing</failureMessage>
<failureDetailDescription>...</failureDetailDescription>
<testToDo>
<InterfaceChecking>registrationChangedOn</InterfaceChecking>
</testToDo>
</testAssertion>

<testAssertion id="R0121" type="required" enabled="true">
```

```xml
<context>Support Interfaces : Mandatory Interfaces</context>
<assertionDescription>The "getAvailability" interface SHALL return an
XML document as defined in the schema
availability</assertionDescription>
<failureMessage>getAvailability do not return an XML file conform to
availability schema</failureMessage>
<failureDetailDescription>...</failureDetailDescription>
<testToDo>
<OutputChecking type="XSD"
interface="getAvailability">http://www.ivoa.net/internal/IVOA/IvoaGridAnd
WebServices/availability-v0.2.xsd</OutputChecking>
</testToDo>
</testAssertion>

<testAssertion id="R0200" type="optional" enabled="true">
<context>Support Interfaces : Non Mandatory Interfaces</context>
<assertionDescription>A VO service should implement the
"harvestWebLog" interface.
</assertionDescription>
<failureMessage>The non mandatory "harvestWebLog” interface is
missing</failureMessage>
<failureDetailDescription>...</failureDetailDescription>
<testToDo>
<InterfaceChecking>harvestWebLog</InterfaceChecking>
</testToDo>
</testAssertion>

<testAssertion id="R0201" type="optional" enabled="true">
<context>Support Interfaces : Non Mandatory Interfaces</context>
<assertionDescription>The "harvestWebLog" interface should take four
parameters "fromDate", "toDate", "notifyEmail" and
"destination"</assertionDescription>
<failureMessage> harvestWebLog  do not take four parameters or
parameters name or type not valid </failureMessage>
<failureDetailDescription>...</failureDetailDescription>
<testToDo>
<InputChecking type="date"
interface="harvestWebLog">fromDate</InputChecking>
<InputChecking type="date"
interface="harvestWebLog">toDate</InputChecking>
<InputChecking type="email"
interface="harvestWebLog">notifyEmail</InputChecking>
<InputChecking type="vo:id"
interface="harvestWebLog">destination</InputChecking>
</testToDo>
</testAssertion>
```

```
...
</vowsbasicprofile>
```

Full schema will be published on the IVOA website.

10. Conformance claim

The conformance could be claimed after the conformance checking if all the assertions are verified.
“This Web Service is compliant to the IVOA Web Services Basic Profile 1.0 (or later)” if all the interfaces are compliant. This means that at least all the mandatory rules are true. It could be useful to provide the list of all the optional rules which are also verified.

11. Changes from previous versions

- Corrections.
- Take into account of the last support interfaces document version (splitting in two specifications).

Appendix A: RFC2119

A small extract from the RFC2119:

“MUST”: This word, or the terms “REQUIRED” or “SHALL”, means that the definition is an absolute requirement of the specification.
“MUST NOT”: This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the specification.
“SHOULD“: This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
“SHOULD NOT”: This phrase, or the phrase "NOT RECOMMENDED" means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
“MAY”: This word, or the adjective “OPTIONAL”, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option “MUST” be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option “MUST” be prepared to interoperate with another

implementation which does not include the option (except, of course, for the feature the option provides).

References

[1] Schaaff A., *A few words about WS-I Basic Profile*, http://www.ivoa.net/internal/IVOA/InterOpMay2004GridAndWebServices/WordsAbout-WS-I-BP.pdf

[2] WS-I, *WS-I Organization*, http://www.ws-i.org/

[3] WS-I, *Basic Profile 1.1 Final Material*, http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

[4] WS-I, *Attachment Profile 1.0 Final Material*, http://www.ws-i.org/Profiles/AttachmentsProfile-1.0-2004-08-24.html

[5] WS-I, *Simple SOAP Binding Profile 1.0 Final Material*, http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html

[6] WS-I, *WS-I Testing Tools*, http://www.ws-i.org/deliverables/workinggroup.aspx?wg=testingtools

[7] Microsoft, *WS-I Basic Profile is released*, http://msdn.microsoft.com/webservices/community/industrynews/default.aspx

[8] Apache, *Axis,* http://ws.apache.org/axis/

[9] Schaaff A., *VO Web Services Basic Profile*, http://www.ivoa.net/internal/IVOA/InterOpSep2004GridAndWebServices/VOWSBasicProfile_Pune.pdf

[10] O'Mullane W, Rixon G., Thakar A., *VO Support Interfaces 0.24*, http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/VOSupportInterfaces-0.24.pdf

[11] WS-I, *Basic Security Profile 1.0 Working Draft*, http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2005-01-20.html

[12] Schaaff A., *VO WS Basic Profile*, http://www.ivoa.net/internal/IVOA/InterOpMay2005GridAndWebServices/VOBasicProfile_Kyoto_021.pdf

[13] Schaaff A., *VO Web Service Basic Profile Status*, http://www.ivoa.net/internal/IVOA/InterOpOct2005GridAndWebServices/BasicProfile_Escorial.pdf

[14] Rixon G., *IVAO Support Interfaces : Mandatory Interfaces 0.25*, http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/VOSupportInterfacesMandatory-0.25.pdf

[15] Rixon G., *WSDL for mandatory standard-interfaces specification* , http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/VObsService.wsdl

[16] Rixon G., *WSDL for non-mandatory log harvesting support interfaces specification*, http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/VSupportServices.wsdl

[17] Schaaff A., *IVOA Web Services Basic Profile on line checking tool*, http://cdsws.u-strasbg.fr/ivoa/servlet/IVOAWSBasicProfileChecker