Web Services for Remote Portlets Specification

Working Draft 0.85, 26 November 2002

Document identifier:

WSRP_Specification-v0.85 (Word)

Location:

http://www.oasis-open.org/committees/wsia

http://www.oasis-open.org/committees/wsrp

Editors:

Alan Kropp, Epicentric, Inc. <akropp@epicentric.com>
Carsten Leue, IBM Corporation <cleue@de.ibm.com>
Rich Thompson, IBM Corporation <richt2@us.ibm.com>

Contributors:

Chris Braun, Novell <cbraun@silverstream.com>
Jeff Broberg, Novell <jbroberg@silverstream.com>
Mark Cassidy, Netegrity <mcassidy@Netegrity.com>
Michael Freedman, Oracle Corporation <Michael.Freedman@oracle.com>
Timothy N. Jones, CrossWeave <tim@crossweave.com>
Thomas Schaeck, IBM Corporation <schaeck@de.ibm.com>
Gil Tayar, WebCollage <Gil.Tayar@webcollage.com>

Abstract:

Integration of remote content and application logic into an End-User presentation has been a task requiring significant custom programming effort. Typically, vendors of aggregating applications, such as a portal, had to write special adapters for applications and content providers to accommodate the variety of different interfaces and protocols those providers used. The goal of this specification is to enable an application designer or administrator to pick from a rich choice of compliant remote content and application providers, and integrate them with just a few mouse clicks and no programming effort.

This specification is a joint effort of two OASIS technical committees. Web Services for Interactive Applications (WSIA) and Web Services for Remote Portlets (WSRP) aim to simplify the integration effort through a standard set of web service interfaces allowing integrating applications to quickly exploit new web services as they become available. The joint authoring of these interfaces by WSRP and WSIA allows maximum reuse of user-facing, interactive web services while allowing the consuming applications to access a much richer set of standardized web services.

This joint standard layers on top of the existing web services stack, utilizing existing web services standards and will leverage emerging web service standards (such as security) as they become available. The interfaces are defined using the Web Services Description Language (WSDL).

Status:

This draft is an early version of the public spec. Various concepts continue to be debated. Points needing clarification as this evolves into the final specification are much appreciated and may be emailed to *Rich Thompson*.

If you are on the wsia@lists.oasis-open.org or wsrp@lists.oasis-open.org list for committee members, send comments there. If you are not on that list, subscribe to the wsia-comment@lists.oasis-open.org or wsrp-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to wsia-comment-request@lists.oasis-open.org or wsrp-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

The errata page for this specification is at http://www.oasis-open.org/committees/wsrp/specification_v1_errata.html.

Formatted: Bullets and Numbering

[Excerpt Only!]

Note: This unofficial document represents an excerpt from a provisional working draft, used to illustrate a point about name and address information; it clips from sections 10 and 11 of the version 0.85 OASIS draft. See:

http://lists.oasis-open.org/archives/wsia/200211/msg00009.html

for the reference to the complete document, itself at:

http://lists.oasis-open.org/archives/wsia/200211/doc00001.doc

Formatted: Bullets and Numbering

10 User Information

This specification provides a mechanism for entities to use End-User information as a means for personalizing behavior to the current user [A600][A606]. A standard set of user attributes has been derived from P3P User Data and is defined in Section 11. Extensibility is supported in both directions; the Consumer indicates to the Producer during registration what set of user profile extensions it supports, and an entity's metadata declares what user profile items it uses (including any extended user profile items). The following table maps the nested profile structures to profileNames:

Profile Name	Structure 1	Structure 2	Field Name
name/prefix	name		prefix
name/given	name		given
name/family	name		family
name/middle	name		middle
name/suffix	name		suffix
name/nickName	name		nickName
birthDate			birthDate
gender			gender
employerInfo/employer	employerInfo		employer
employerInfo/department	employerInfo		department
employerInfo/jobTitle	employerInfo		jobTitle
homeInfo/address/name	homeInfo	address	name
homeInfo/address/street	homeInfo	address	street
homeInfo/address/city	homeInfo	address	city
homeInfo/address/stateprov	homeInfo	address	stateprov
homeInfo/address/country	homeInfo	address	country
homeInfo/address/org	homeInfo	address	org
homeInfo/telephone	homeInfo		telephone
homeInfo/email	homeInfo		email
homeInfo/online	homeInfo		online
workInfo/address/name	workInfo	address	name

workInfo/address/street	workInfo	address	Street
workInfo/address/city	workInfo	address	City
workInfo/address/stateprov	workInfo	address	stateprov
workInfo/address/country	workInfo	address	country
workInfo/address/org	workInfo	address	org
workInfo/telephone	workInfo		telephone
workInfo/email	workInfo		email
workInfo/online	workInfo		online

Entities that need access to user information MUST declare in its metadata the specific user profile fields it needs using the names specified above.

Consumers supplying additional custom profile fields are encourage to publish a similar mapping between profileNames and the custom fields.

Formatted: Bullets and Numbering

1.1 Passing User Information

User information MAY be passed to the Producer when a Consumer invokes certain operations. A Consumer SHOULD provide the specific fields the entity declared it needs, unless the information is not available or is restricted by policy (e.g. privacy policy).

Formatted: Bullets and Numbering

1.2 User Identity

Mechanisms that support federation of user identity between web services systems are defined in other specifications, such as WS-Security and SAML. If a Consumer and Producer need to share a common identity for an End-User, it is recommended that compliance with these standards be the means to passing the required information.

It is anticipated that some entities will interact with one or more back-end applications that require a user identity for the End-User. If the user identity required by the back-end application is not the same as that authenticated or otherwise supplied by the Consumer, the entity MAY request the End-User to provide the necessary information (preferably using secure transport) for use with the back-end application via markup interactions (e.g. display a form that prompts for a user identity and any security tokens (such as a password) for the back-end system).

Formatted: Bullets and Numbering

User Profile Types

The `UserProfile` structure is used to carry information about the End-User. The entity uses the `userProfileItems` in its metadata to describe the fields it uses to generate markup from this set and any others the Consumer indicated were available when it registered. See section 0 for a complete description of this portion of the protocol.

```
UserProfile
    [O] UserName        name
    [O] DateTime        birthdate
    [O] string          gender
    [O] EmployerInfo    employer
    [O] LocationInfo    homeInfo
    [O] LocationInfo    workInfo
```

```
    [O] Extension    extensions[]
```

Members:

- `name`: A structure containing the various fields for the End-User's name.
- `birthdate`: The End-User's birthdate. This uses the schema-defined datatype for DateTime rather than Date as not all web stacks serialize / deserialize Date properly.
- `gender`: The End-User's gender ("M" = male, "F" = female).
- `employer`: A structure containing various fields for the End-User employer's information.
- `homeInfo`: The End-User's home location information.
- `workInfo`: The End-User's work location information.
- `extensions`: A mechanism implementations MAY choose to use for extending this structure provided those extensions come from a different namespace.

Formatted: Bullets and Numbering

UserName Type

The `UserName` structure carries the detailed fields for the parts of an End-User's name.

```
UserName
    [O] string      prefix
    [O] string      given
    [O] string      family
    [O] string      middle
    [O] string      suffix
    [O] string      nickName
    [O] Extension   extensions[]
```

Members:

- `prefix`: Examples include Mr, Mrs, Ms, Dr, etc.
- `given`: The End-User's first or given name.
- `family`: The End-User's last or family name.
- `middle`: The End-User's middle name(s) or initial(s).
- `suffix`: Examples include Sr, Jr, III, etc.
- `nickName`: The End-User's preferred nick name.
- `extensions`: A mechanism implementations MAY choose to use for extending this structure provided those extensions come from a different namespace.

Formatted: Bullets and Numbering

EmployerInfo Type

The `EmployerInfo` structure contains the detailed fields concerning the End-User's employer.

```
Employerinfo
    [O] string      employer
    [O] string      department
    [O] string      jobTitle
    [O] Extension   extensions[]
```

Members:

- `employer`: The name of the employer.
- `department`: The name of the department the End-User works within.
- `jobTitle`: The title of the End-User's job.

- `extensions`: A mechanism implementations MAY choose to use for extending this structure provided those extensions come from a different namespace.

Formatted: Bullets and Numbering

LocationInfo Type

The `LocationInfo` structure is used to describe a location for the End-User.

```
LocationInfo
    [O] Address     address
    [O] string      telephone[]
    [O] string      email[]
    [O] string      online[]
    [O] Extension   extensions[]
```

Members:

- `address`: A structure for various fields holding portions of the postal address.
- `telephone`: An array of telephone numbers for the End-User.
- `email`: An array of email addresses for the End-User.
- `online`: An array of URIs for the End-User (usually web-sites).
- `extensions`: A mechanism implementations MAY choose to use for extending this structure provided those extensions come from a different namespace.

Formatted: Bullets and Numbering

Address Type

The `Address` structure carries the detailed fields describing a particular address.

```
Address
    [O] string      name
    [O] string      street[]
    [O] string      city
    [O] string      stateprov
    [O] string      country
    [O] string      org
    [O] Extension   extensions[]
```

Members:

- `name`: The name to which items should be addressed.
- `street`: The street portion of the addess. This may involve multiple lines of an address.
- `city`: The city portion of the address.
- `stateprov`: The state or province portion of the address.
- `country`: The country portion of the address.
- `org`: Any organization needing to be specified in the address.
- `extensions`: A mechanism implementations MAY choose to use for extending this structure provided those extensions come from a different namespace.