

# Yadis Specification

Version 0.92

27 February 2006

Joaquin Miller, editor

This is a minor revision of Yadis Specification Version 0.91.  
There are changes for clarity or to correct editorial errors.  
Thanks to Larry Drebes of JanRain for all his work on this.

There was a name change in 0.91:  
'X-XRDS-Location' is the name of the HTTP response-header, replacing 'X-YADIS-Location.'

There is one substantive change:  
If there are X-XRDS-Locations both in an HTTP response-header and in an HTML element, use the location in the HTTP response-header. See 6.2.6, third paragraph.

These constraints on a Yadis XRD are removed:  
7.3.3: XRD element MUST contain one or more Service elements.  
7.3.4: Service element XRDS MUST contain at least one Type element.

These are rewritten:  
6.2.1  
6.2.3

Notes are removed in:  
6.2.5  
7.4.3

## **Implementors:**

Now it the time to tell us what's unclear, what can't be implemented, what's just dumb.

Send your reports to the Yadis mailing list or post on the Yadis wiki.

Thanks!

## Table of Contents

0 Introduction.....	3
1 Scope.....	3
2 Normative References.....	4
2.1 Requests for Comments.....	4
2.2 W3C Recommendations.....	4
2.3 OASIS Specifications.....	4
3 Terms and Definitions.....	5
3.1 Implementor Terms.....	5
3.2 Definitions from other specifications.....	6
4 Abbreviations.....	7
5 Conventions.....	7
6 The Yadis Protocol.....	8
6.1 Overview of the Yadis Protocol.....	8
6.2 Protocol Specification.....	8
7. The Yadis document.....	12
7.1 Overview of the Yadis document.....	12
7.2 A simple Yadis document.....	12
7.3. Element definitions.....	13
7.4 Other parts of a Yadis Resource Descriptor.....	14
7.5 Schemas for the Yadis document.....	17

## 0 Introduction

Yadis uses URLs as identifiers. It provides a mechanism for determining the services that are available with a given identifier.

Section 1 describes the scope of this Specification.

Sections 2 through 5 provide references, definitions, abbreviations and conventions.

Section 6 specifies the Yadis protocol.

Section 7 specifies the Yadis Resource Descriptor document.

This draft Version 0.92 has been prepared by the Yadis project. For more information on the Yadis project, see the [Yadis wiki](#) and [Yadis Overview](#).)

We are publishing this draft for implementors.

If you are implementing Yadis, please add a page to the Yadis wiki by adding a link at [Yadis Implementations](#). When you discover problems with this Version 0.92, please add a page to the Yadis wiki by adding a link at [Change Proposals](#).

During implementation you may find

- ambiguity, lack of clarity or specificity, or errors in the specification,
- provisions of the specification that are difficult to implement,
- holes in the specification where we have missed something or
- other problems.

Please let us know; send your comments on this text to the editor, [Joaquin Miller](#).

Based on the feedback from implementors, we will prepare Yadis Specification Version 1.0 for general release.

This text is made available under [Attribution-ShareAlike 2.5](#) when attributed to the Yadis project.

## 1 Scope

This Yadis specification provides:

- A general purpose identifier for persons and any other entities, which can be used in a variety of services.
- A syntax for a resource description document identifying services available using that identifier and an interpretation of the elements of that document.
- A protocol for obtaining that resource description document, given that identifier.

Together these enable coexistence and interoperation of a rich variety of services using a single identifier. The identifier uses a standard syntax and a well-established namespace; it requires no additional namespace administration infrastructure.

This specification is intended to be used to implement software for requesting a resource description document and to implement software for providing a resource description document or a locator for that document. The document syntax and interpretation and the protocol are specified to a level of detail sufficient to enable the determination of the compliance of any implementation to this Specification.

## 2 Normative References

The following specifications contain provisions which, through reference in this text, constitute provisions of this Yadis Specification. At the time of publication, the editions indicated were valid, with the exception of those in Clause 2.3 (see the discussion there). All specifications are subject to revision, and parties to agreements based on this Specification are encouraged to investigate the possibility of applying the most recent edition of the specifications listed below.

### 2.1 Requests for Comments

<a href="#">RFC 2046</a>	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, November 1996.
<a href="#">RFC 2119</a>	Key words for use in RFCs to Indicate Requirement Levels, March 1997.
<a href="#">RFC 2616</a>	Hypertext Transfer Protocol -- HTTP/1.1, June 1999.
<a href="#">RFC 3986</a>	Uniform Resource Identifier (URI): Generic Syntax, January 2005.
<a href="#">RFC 3987</a>	Internationalized Resource Identifiers (IRIs), January 2005.

### 2.2 W3C Recommendations

<a href="#">HTML 4.01</a>	HTML 4.01 Specification, 24 December 1999.
<a href="#">XHTML 1.0</a>	The Extensible HyperText Markup Language (Second Edition), 1 August 2002.
<a href="#">XML</a>	Extensible Markup Language (XML) 1.0 (Third Edition), 4 February 2004
<a href="#">XML Schema Part 1</a>	XML Schema Part 1: Structures Second Edition, 28 October 2004
<a href="#">XML Schema Part 2</a>	XML Schema Part 2: Datatypes Second Edition, 28 October 2004

### 2.3 OASIS Specifications

The following documents have not been adopted by OASIS. They are listed here for two reasons:

- These documents were used in preparing this Yadis Specification.

- When and if OASIS adopts such specifications, it is expected that this Yadis specification will conform to them and will at that time reference them here as normative references.

[XRI Syntax](#)

Extensible Resource Identifier (XRI) Syntax V2.0, Committee Specification, 14 November 2005.

[XRI Resolution](#)

Extensible Resource Identifier (XRI) Resolution V2.0, Working Draft 09, 10 November 2005.  
(See the [Yadis wiki](#) for the current draft.)

## 3 Terms and Definitions

### 3.1 Implementor Terms

Yadis User	An entity using a Yadis ID as an identifier.
Yadis ID	A identifier used with one or more Yadis Services. A Yadis ID may be a URL; it may be any other identifier, such as an XRI, that can be resolved to a URL.
Yadis URL	A Yadis ID, if it is a URL, otherwise the URL to which that Yadis ID resolves. A Yadis URL may be used in the Yadis Protocol to obtain a Yadis Resource Descriptor.
Yadis Resource	A computer software process (or system of processes) that provides one or more services located using the Yadis Protocol.
Yadis Service	A service provided by a Yadis Resource.
Yadis document	A document containing a Yadis Resource Descriptor
Yadis Resource Descriptor	An element of a Yadis document identifying the services that are available using a Yadis ID.
Resource Descriptor URL	A URL that locates a Yadis document.
Agent	A computer software process (or system of processes) acting on behalf of a party.
Yadis User Agent	An agent acting on behalf of a Yadis User (for example, a regular web browser).
Relying Party	A party responsible for a Relying Party Agent and on whose behalf that Agent acts. A Relying Party is relying on the services provided by a Yadis Resource, in particular on data provided by that service concerning the person identified by a Yadis ID.

Relying Party Agent	A role to be fulfilled by an agent that uses a Yadis ID (and the data accessible using that Yadis ID) provided by a Yadis User Agent. The Relying Party Agent discovers the services available for a Yadis ID using the Yadis Protocol, and may modify its own behavior accordingly.
RESTful	Having the REST architectural style.

Note: Yadis IDs are intended to be used not only by persons and other parties, such as clubs, crowds, businesses and governments, but also to identify agents of any such parties, such as RFID tags and software processes.

The Yadis Protocol is intended to be used by any software agent wishing to discover the services available for use with any Yadis ID.

### 3.2 Definitions from other specifications

This specification makes use of the following terms as defined in the indicated specifications:

absolute URL	<a href="#">RFC 3986</a> 4.3
Accept request-header	<a href="#">RFC 2616</a> 14.1
agent	<a href="#">X.911</a> 6.5.7
authentication	<a href="#">X.903</a> 15.4
failure	<a href="#">X.902</a> 13.5.1
GET request	<a href="#">RFC 2616</a> 9.3
<head> element	<a href="#">HTML 4.01</a> 7.4.1
HEAD request	<a href="#">RFC 2616</a> 9.4
HTTP	<a href="#">RFC 2616</a>
identifier	<a href="#">X.902</a> 12.2
IRI	<a href="#">RFC 3987</a>
<meta> element	<a href="#">HTML 4.01</a> 7.4.4
<meta> element http-equiv attribute	<a href="#">HTML 4.01</a> 7.4.4
MIME media type	<a href="#">RFC 2616</a> 14.1 and <a href="#">RFC 2046</a> 3
party	<a href="#">X.911</a> 6.5.1
relative reference	<a href="#">RFC 3986</a> 4.2
request-header	<a href="#">RFC 2616</a> 5.3
response-header	<a href="#">RFC 2616</a> 6.2
REST architectural style	<a href="#">REST</a> 5
scheme	<a href="#">RFC 3986</a> 3.1

URI	<a href="#">RFC 3986</a>
URL	<a href="#">RFC 3986</a> 1.1.3
Vary: Accept response-header	<a href="#">RFC 2616</a> 14.44
XRI	<a href="#">XRI Syntax</a>

## 4 Abbreviations

These abbreviations are use in this Specification:

HTTPS	The HTTP protocol when that protocol is used, in accordance with current conventions, so that the session data is encrypted using a version of the Transport Layer Security or Secure Socket Layer protocols
-------	--

## 5 Conventions

The key words and word pairs, '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](#).

Notes and examples in this document are not normative.

## 6 The Yadis Protocol

### 6.1 Overview of the Yadis Protocol

The purpose of the Yadis protocol is to enable a Relying Party to obtain a Yadis Resource Descriptor that describes the services available using a Yadis ID.

#### 6.1.1 Obtaining the Yadis Resource Descriptor

When a User offers a Yadis ID to a Relying Party, that Relying Party will want to discover which services are available using that Yadis ID.

Examples: Is it an OpenID URL or a LID? What authentication methods are available?

To do that, the Relying Party Agent makes an HTTP request. This request may take any one of several forms, specified in Clause 6.2.3 of this document.

In response to the request, the Relying Party Agent obtains either:

1. A Yadis document.
2. A URL that locates a Yadis document.

If the Relying Party Agent obtains a URL, the Relying Party Agent uses that URL to obtain the Yadis document.

The Yadis document contains a Yadis Resource Descriptor, which identifies the services available using that Yadis ID, including services that can authenticate the User.

For a description of the Yadis document, see Clause 7, Yadis document.

#### 6.1.2 Authentication services

The Yadis protocol was originally intended to be used to discover authentication services that can be used with a Yadis ID. This specification enables discovery of other services.

This Specification does not prescribe the operation of authentication services. Authentication is performed using one or more discovered services. The Relying Party Agent uses the information in the Yadis Resource Descriptor to choose a service suitable to its purposes, and uses that service to authenticate the user.

#### 6.1.3 Other services

Yadis resources offer other services, in addition to authentication. These services are identified in the Yadis Resource Descriptor and each service operates according to the specification of that particular Yadis service.

Example: LID defines a RESTful protocol that allows the structured query of data about the Yadis User. LID Profile Exchange can be offered as a Yadis Service.

### 6.2 Protocol Specification

The HTTP protocol MUST be used for all interactions of the Yadis protocol.

If the scheme name of a Yadis URL is 'https', HTTP must be used, in accordance with current conventions, so that the session data is encrypted using a version of the Transport Layer Security or Secure Socket Layer protocols.

### **6.2.1 Yadis ID**

A Yadis ID is a identifier used by one or more Yadis Services. A Yadis ID MAY be a URL; it MUST be an identifier that is resolvable to a URL.

Note: This specification does not require a Relying Party Agent to resolve an XRI or other identifier. Some Relying Party Agents may work only when the Yadis ID is a URL.

### **6.2.2 Yadis URL**

If a Yadis ID is a URL, it is a Yadis URL; if it is not a URL, the URL to which it resolves is the corresponding Yadis URL. The scheme name of a Yadis URL must be 'http' or 'https'.

### **6.2.3 Alternatives**

A Relying Party Agent MAY use the YADIS URL to make an HTTP GET request. The response MAY contain, in an HTTP response-header or in an HTML document, a Resource Descriptor URL giving the location of the YADIS document. If it does, the Relying Party Agent then obtains the YADIS document using that URL.

The Yadis Protocol also includes two alternatives:

The Relying Party Agent MAY first issue an HTTP HEAD request. In that case, the Resource Descriptor URL MAY be returned in an HTTP response-header.

The Relying Party Agent MAY include in the HTTP GET request an Accept request-header asking for the Yadis document to be returned. In that case the Yadis document MAY be returned in response to that request, instead of an HTML document.

Note: If the server supports content negotiation for the Yadis URL, the response needs to include a Vary: Accept header to allow caches to properly interpret future requests. This header needs to be present even in the case where the HTML page is returned (instead of a Yadis document).

The following Clauses specify the steps of the Yadis Protocol.

### **6.2.4 Initiation**

The Yadis Protocol is initiated by the Relying Party Agent with an initial HTTP request using the Yadis URL.

This request MUST be either a GET or a HEAD request.

A GET or HEAD request MAY include an HTTP Accept request-header (HTTP 14.1) specifying MIME media type, application/xrds+xml.

### **6.2.5 Response**

The response MUST be one of:

1. An HTML document with a <head> element that includes a <meta> element with http-equiv attribute, X-XRDS-Location,
2. HTTP response-headers that include an X-XRDS-Location response-header, together with a document
3. HTTP response-headers only, which MAY include an X-XRDS-Location response-header, a content-type response-header specifying MIME media type, application/xrds+xml, or both.
4. A document of MIME media type, application/xrds+xml.

### **6.2.6 Resource Descriptor URL**

The response MAY include an X-XRDS-Location HTTP response-header; the value of that header MUST be a Yadis Resource Descriptor URL.

The response MAY include an HTML document with a <head> element containing a <meta> element whose http-equiv attribute is X-XRDS-Location; the value of that attribute MUST be a Yadis Resource Descriptor URL.

If the response includes Yadis Resource Descriptor URLs in both an HTTP response-header and an HTML document <head> element, the Yadis Resource Descriptor URL in the HTTP response-header must be used.

A Yadis Resource Descriptor URL MUST be an absolute URL; it MUST locate a Yadis document (see Clause 7).

If the response includes both a content-type response-header specifying MIME media type, application/xrds+xml and a Yadis Resource Descriptor URL, then the Yadis document is located by that Yadis Resource Descriptor URL.

### **6.2.7 Document**

If the response does not include a Yadis Resource Descriptor URL and the response is a document of MIME media type, application/xrds+xml, then that document MUST be a Yadis document (see Clause 7).

### **6.2.8 Second Request**

If the response includes a Yadis Resource Descriptor URL, the Relying Party Agent MUST request the document located by that URL.

If the response to an HTTP HEAD request does not contain a Yadis Resource Descriptor URL, the Relying Party Agent MUST then issue an HTTP GET request to the Yadis URL.

### **6.2.9 Third Request**

The response to an HTTP GET request that follows an HTTP HEAD request MUST be handled as prescribed in 6.2.6 for an initial HTTP GET request. This MAY result in a third request using a Yadis Resource Descriptor URL.

### **6.2.10 Termination**

When a Yadis document is returned to the Relying Party Agent the Yadis Protocol terminates.

Note: Further steps depend on the services identified in the Yadis Resource Descriptor and the intent of the Relying Party Agent; they are outside of the scope of the current Yadis Specification.

Example: A Relying Party Agent may, in the next step, authenticate the user agent through OpenID and then perform a LID Profile Exchange.

#### **6.2.11 Failure**

If none of the requests succeed in obtaining a Yadis document then the URL used in the initial request is not a Yadis URL or there has been a failure.

## 7. The Yadis document

### 7.1 Overview of the Yadis document

The Yadis document is based on a simple, extensible XML document called an Extensible Resource Descriptor (abbreviated 'XRD'). The format of XRD documents is being specified by the XRI Technical Committee of OASIS (see the [XRI Resolution 2.0 specification](#).) The XML schemas for the Yadis document are specified in Clause 7.5 of this Specification.

Note: The Yadis Project considered providing both a plain text format and an XML format, in order to make it as easy as possible for Relying Party Agents to use service information available through Yadis IDs. However, we came to the conclusion that requiring both a plain text format and an XML format provides only small additional value, while using an XML format enables use of standard parsing tools and allows service specifications to include additional information without upsetting parsers that do not understand those extensions.

The Yadis document contains a Yadis Resource Descriptor, which provides a list of identifiers of services. These are the services that know the User identified by the Yadis ID used to obtain the Yadis document. In the case of some services, additional data is included in the Yadis Resource Descriptor for use by the Relying Party Agent in making a request to that service. Such additional data is not specified in the Yadis Specification but is specified in the definition of that service.

The Yadis Resource Descriptor also enables the User to specify which services it prefers be used.

### 7.2 A simple Yadis document

Here is an example of a small Yadis document:

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS xmlns:xrds="xri://$xrds" xmlns="xri://$xrd*($v*2.0)">
  <XRD>
    <Service>
      <Type> http://lid.netmesh.org/sso/2.0 </Type>
    </Service>
    <Service>
      <Type> http://lid.netmesh.org/sso/1.0 </Type>
    </Service>
  </XRD>
</xrds:XRDS>
```

This document specifies two services.

Note: The XML declaration and the XRDS start-tag appear in all Yadis documents, with the attributes and values shown. The XRD element is the Yadis Resource Descriptor.

### 7.3. Element definitions

These definitions of the elements of a Yadis document:

are constraints on the XRDS and XRD schemas specified in Clause 7.5 and specify how the elements specified in those schemas are to be interpreted when used in a Yadis document.

#### 7.3.1 XRDS

A Yadis document consists of an XRD container (an XRDS element) enclosing a Yadis Resource Descriptor (an XRD element).

A Yadis document MAY contain more than one XRD in the XRDS and MAY contain other elements in the XRDS, in addition to XRD elements.

If a Yadis XRDS includes more than one XRD element, the Yadis Resource Descriptor is the *last* XRD element. A Relying Party Agent MAY ignore other XRD elements.

A Relying Party Agent MAY ignore all other elements in an XRDS.

#### 7.3.2 XRD

A Yadis Resource Descriptor is an XRD element containing a sequence of Service elements. The order of the Service elements is not significant.

#### 7.3.3 Service

A Yadis Resource Descriptor MAY contain Service elements, each describing a Yadis service.

Note: When an XRDS document returned when using the Yadis protocol contains no Service elements describing a Yadis service this indicates that the URL is not intended for use with any Yadis service.

#### 7.3.4 Type

Each Service element MAY contain one or more Type elements.

Note: A Service element containing no Type elements does not describe a Yadis service.

Each Type element MUST contain an identifier of some version of some service. This service identifier MUST be a URI or XRI.

Note: It is our intention that this element may contain an IRI, as specified by [RFC 3987](#). This will be specified in a later version of this Specification.

For each service identified by a Type element there SHOULD be a service specification document. It is RECOMMENDED that the service identifier be a URL that can be used in an ordinary web browser to display that service specification document.

It is RECOMMENDED that each service identifier include an explicit version identifier, in order to assist the evolution of the service in the future.

## 7.4 Other parts of a Yadis Resource Descriptor

Here is an example of a larger Yadis document:

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS xmlns:xrds="xri://$xrds" xmlns="xri://$xrd*($v*2.0)"
  xmlns:openid="http://openid.net/xmlns/1.0">
  <XRD>
    <Service priority="10">
      <Type>http://openid.net/signon/1.0</Type>
      <URI>http://www.myopenid.com/server</URI>
      <openid:Delegate>http://smoker.myopenid.com/</openid:Delegate>
    </Service>
    <Service priority="50">
      <Type>http://openid.net/signon/1.0</Type>
      <URI>http://www.livejournal.com/openid/server.bml</URI>
      <openid:Delegate>
        http://www.livejournal.com/users/frank/
      </openid:Delegate>
    </Service>
    <Service priority="20">
      <Type>http://lid.netmesh.org/sso/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/sso/1.0</Type>
    </Service>
  </XRD>
</xrds:XRDS>
```

### 7.4.1 URI element

A Yadis Service element MAY contain one or more URI elements.

Each URI element MUST contain a URI that resolves to a resource providing the service (or services) specified by the Type element(s) of that Service element.

That URI MUST be an absolute URL, not a relative reference or other URI.

Note: It is our intention that this element may contain any absolute URI that identifies a resource providing the service(s) or may contain an absolute IRI, as specified by [RFC 3987](#). This will be specified in a later version of this Specification.

If there is more than one URI element, the URIs in those elements MUST be equivalent for the purpose of using the identified service(s). A Relying Party Agent MAY attempt to use any one or all of the URIs.

If one or more URI elements has a priority attribute, a Relying Party Agent MAY use the priority values as specified in Clause 2.4.3 of the [XRI Resolution 2.0 specification](#).

The order of URI elements is not significant.

The URI element is OPTIONAL. If a URI element is provided, the service determines the meaning of that element and the protocol to use with it.

#### **7.4.2 Priority attribute**

The OPTIONAL priority attribute MAY be used with the Service element, allowing the User to specify preferences for the service to be used.

Example: The example document at 7.4 above indicates that the User prefers the OpenID protocol using the server, <http://www.myopenid.com/server>, and that the last choice is LID version 1.0.

The OPTIONAL priority attribute MAY be used with the URI element, allowing the User to specify preferences for the URI to be used.

Note: The LID SSO service elements in the document at 7.4 above do not contain a URI element because in LID SSO the LID URL itself is used to obtain LID services.

In keeping with the goal of ease of implementation, a Relying Party Agent MAY ignore priority attributes.

A Relying Party Agent that recognizes and uses the priority attribute in one or more Service or URI elements MUST follow the specification of priority attributes in Clause 2.4.3 of the [XRI Resolution 2.0 specification](#).

Note: The XRI Resolution 2.0 specification provides that the User prefers first the element with the smallest number in the value of the priority attribute, then those with higher numbers in the order of the number, and prefers last those elements with no priority attribute.

#### **7.4.3 Other elements in a Service element**

Yadis does not specify any other elements for use inside a Service element. A service may specify additional service-specific elements for use inside a Service element. The use of these elements is dictated by the service specification of that service.

Note: This allows a Service element to indicate information that is specific to that service.

Example: The OpenID specification defines the optional Delegate element, which specifies a URL by which the OpenID server knows the User.

Note: We encourage protocol designers to be brief.

#### **7.4.4 Other elements in an XRD**

A Relying Party Agent MAY ignore all elements in an XRD element other than the Service elements.

A Relying Party Agent MAY recognize and use all elements found in a Service element that are specified by the XRD schema for inclusion in a Service element.

A Yadis resource MUST NOT require the use of any elements of an XRD other than the Service elements that identify the services provided by that Yadis resource.

### 7.4.5 Other services

A Yadis resource MAY offer services other than authentication services. A Yadis resource is NOT REQUIRED to offer an authentication service.

Example: The following Resource Descriptor specifies a number of LID services available at the resource identified by the user's Yadis ID:

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS xmlns:xrds="xri://$xrds" xmlns="xri://$xrd*(v*2.0)">
  <XRD>
    <Service priority="10">
      <Type>http://lid.netmesh.org/sso/2.0</Type>
    </Service>
    <Service priority="20">
      <Type>http://lid.netmesh.org/sso/1.0</Type>
    </Service>
    <Service priority="30" xmlns:openid="http://openid.net/xmlns/1.0">
      <Type>http://openid.net/signon/1.0</Type>
      <URI>http://www.livejournal.com/openid/server.bml</URI>
      <openid:Delegate>
        http://www.livejournal.com/users/frank/
      </openid:Delegate>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/post/sender/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/post/receiver/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/relying-party/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/traversal/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/format-negotiation/2.0</Type>
    </Service>
  </XRD>
</xrds:XRDS>
```

## 7.5 Schemas for the Yadis document

The schemas of the Yadis document are the XRDS and XRD schemas contained in this clause.

A Yadis document MUST be a valid XML document and MUST conform to the XRDS schema. A Yadis Resource Descriptor is an XRD element; it MUST be contained in an XRDS element and MUST conform to the XRD schema.

Note: It is our intention to incorporate the XRDS and XRD standards by reference when and if they have been adopted by OASIS. In the meantime, we reproduce the schemas here. For the latest information on the OASIS work, see [XRI Resolution 2.0 specification](#).

### 7.5.1 XRDS Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="xri://$xrds" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xrds="xri://$xrds">
  <!-- Utility patterns -->
  <xs:attributeGroup name="otherattribute">
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:attributeGroup>
  <xs:group name="otherelement">
    <xs:choice>
      <xs:any namespace="##other" processContents="lax"/>
      <xs:any namespace="##local" processContents="lax"/>
    </xs:choice>
  </xs:group>
  <!-- Patterns for elements -->
  <xs:element name="XRDS">
    <xs:complexType>
      <xs:sequence>
        <xs:group ref="xrds:otherelement" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="xrds:otherattribute"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## 7.5.2 XRD Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="xri://$xrd*($v*2.0)" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xrd="xri://$xrd*($v*2.0)"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <!-- Utility patterns -->
  <xs:attributeGroup name="otherattribute">
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:attributeGroup>
  <xs:group name="otherelement">
    <xs:choice>
      <xs:any namespace="##other" processContents="lax"/>
      <xs:any namespace="##local" processContents="lax"/>
    </xs:choice>
  </xs:group>
  <xs:complexType name="URIpattern">
    <xs:simpleContent>
      <xs:extension base="xs:anyURI">
        <xs:attributeGroup ref="xrd:otherattribute"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="Stringpattern">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attributeGroup ref="xrd:otherattribute"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <!-- Patterns for elements -->
  <xs:element name="XRD">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="xrd:Query" minOccurs="0"/>
        <xs:element ref="xrd:Expires" minOccurs="0"/>
        <xs:element ref="xrd:Synonym" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="xrd:XSynonym" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="xrd:ProviderID"/>
        <xs:element ref="xrd:Service" minOccurs="0" maxOccurs="unbounded"/>
        <xs:group ref="xrd:otherelement" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute ref="xrd:id"/>
      <xs:attribute ref="xrd:version"/>
      <xs:attributeGroup ref="xrd:otherattribute"/>
    </xs:complexType>

```

## Yadis Specification 0.92

```
</xs:element>
<xs:element name="Query" type="xrd:Stringpattern"/>
<xs:element name="Expires">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:dateTime">
        <xs:attributeGroup ref="xrd:otherattribute"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="Synonym" type="xrd:URIpattern"/>
<xs:element name="XSynonym" type="xrd:URIpattern"/>
<xs:element name="ProviderID" type="xrd:URIpattern"/>
<xs:element name="Service">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="xrd:Type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="xrd:MediaType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="xrd:Pattern" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="xrd:URI" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="xrd:ProviderID" minOccurs="0"/>
      <xs:group ref="xrd:otherelement" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="xrd:otherattribute"/>
  </xs:complexType>
</xs:element>
<xs:element name="Type" type="xrd:URIpattern"/>
<xs:element name="MediaType" type="xrd:Stringpattern"/>
<xs:element name="Pattern" type="xrd:Stringpattern"/>
<xs:element name="URI" type="xrd:URIpattern"/>
<xs:attribute name="priority" type="xs:nonNegativeInteger"/>
<xs:attribute name="version" type="xs:string" fixed="2.0"/>
<xs:attribute name="id" type="xs:ID"/>
</xs:schema>
```