



*International  
Virtual  
Observatory  
Alliance*

## StandardsRegExt: a VOResource Schema Extension for Describing IVOA Standards Version 1.1

**IVOA Working Draft 2023-09-11**

Working Group  
Registry

This version  
<https://www.ivoa.net/documents/StandardsRegExt/20230911>

Latest version  
<https://www.ivoa.net/documents/StandardsRegExt>

Previous versions  
REC-1.0  
PR-20120217  
PR-20120213  
PR-20111017  
PR-20110921  
PR-20110316  
WD-20100519

Author(s)  
Paul Harrison, Douglas Burke, Ray Plante, Guy Rixon, Dave Morris, Renaud Savalle

Editor(s)  
Renaud Savalle

## Abstract

This document describes an XML encoding standard for metadata about IVOA standards themselves, referred to as StandardsRegExt. It is intended to allow for the discovery of a standard via an IVOA identifier that refers to the standard. It also allows one to define concepts that are defined by the standard which can themselves be referred to via an IVOA identifier (augmented with a URL fragment identifier). Finally, it can also provide a machine interpretable description of a standard service interface. We describe the general model for the schema and explain its intended use by interoperable registries for discovering resources.

## Status of this document

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 <https://www.ivoa.net/documents/>.

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Role within the VO Architecture . . . . .	5
1.2	Syntax Notation Using XML Schema . . . . .	5
<b>2</b>	<b>The StandardsRegExt Data Model</b>	<b>6</b>
2.1	The Schema Namespace and Location . . . . .	7
2.2	Summary of Metadata Concepts . . . . .	8
2.3	Defining Enumerations of Identifiers . . . . .	9
<b>3</b>	<b>The StandardsRegExt Metadata</b>	<b>11</b>
3.1	Resource Type Extensions . . . . .	11
3.1.1	Standard . . . . .	11
3.1.2	ServiceStandard . . . . .	16
3.2	Defining Keys: StandardKey and StandardKeyURI . . . . .	18
<b>4</b>	<b>Operational Aspects</b>	<b>19</b>
4.1	Management of Standards Records . . . . .	19
4.2	Management of Standard Keys . . . . .	20

<b>A</b>	<b>A Sample Record</b>	<b>20</b>
<b>B</b>	<b>Changes from Previous Versions</b>	<b>23</b>
B.1	Changes since Rec-1.0 . . . . .	23
B.2	Changes since PR-v1.0 20120217: . . . . .	24
B.3	Changes since PR-v1.0 20120213: . . . . .	24
B.4	Changes since PR-v1.0 20111017: . . . . .	24
B.5	Changes since PR-v1.0 20110921: . . . . .	25
B.6	Changes since PR-v1.0 20110316: . . . . .	25
B.7	Changes since WD-v1.0 20100519: . . . . .	25
B.8	Changes since WD-v1.0 20100514: . . . . .	25
B.9	Changes since WD-v0.4: . . . . .	25
	<b>References</b>	<b>25</b>

## Conformance-related definitions

The words “MUST”, “SHALL”, “SHOULD”, “MAY”, “RECOMMENDED”, and “OPTIONAL” (in upper or lower case) used in this document are to be interpreted as described in IETF standard RFC2119 (Bradner, 1997).

The *Virtual Observatory (VO)* is a general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach. The *International Virtual Observatory Alliance (IVOA)* is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications.

## Acknowledgements

The first versions of this document have been developed with support from the National Science Foundation’s<sup>1</sup> Information Technology Research Program under Cooperative Agreement AST0122449 with The Johns Hopkins University, from the UK Particle Physics and Astronomy Research Council (PPARC), and from the European Commission’s Seventh Framework Program<sup>2</sup>.

This document contains text lifted verbatim, with small changes, and with substantial changes from (old versions of) the VODataService specification (Demleitner and Plante et al., 2021). This has been done without specific attribution as a means for providing consistency across similar documents. We acknowledge the authors of that document for this text.

---

<sup>1</sup><http://www.nsf.gov/>

<sup>2</sup>[http://cordis.europa.eu/fp7/capacities/home\\_en.html](http://cordis.europa.eu/fp7/capacities/home_en.html)

## 1 Introduction

An important goal of the IVOA is to publish standards for services which can interoperate to create a Virtual Observatory (VO). Central to the coordination of these services is the concept of a Registry where resources can be described and thus discovered by users and applications in the VO. The standard Resource Metadata for the Virtual Observatory (Hanisch and IVOA Resource Registry Working Group et al., 2007) (hereafter referred to as **RM**) defines metadata terms for services and other discoverable resources. A specific XML encoding of these resources is described in the IVOA standard VOResource (Plante and Demleitner et al., 2018). In this schema, support for a standard service protocol is described as a service’s *capability*; the associated metadata is contained within the service resource description’s *capability* element. The specific standard protocol supported is uniquely identified via an attribute of the *capability* element called *standardID* whose value is a URI. VOResource does not place a formal validation requirement on the *standardID* other than it be a legal URI; however, it was intended that IVOA-endorsed standards would be represented via an IVOA identifier. As per the IVOA Identifier standard (Demleitner and Plante et al., 2016), an IVOA identifier must be registered as a resource in an IVOA-compliant registry.

This document defines a VOResource extension schema called StandardsRegExt which allows one to describe a standard and register it with an IVOA registry. By doing so, a unique IVOA identifier becomes “attached” to the standard which can be referred to in other resource descriptions, namely for services that support the standard. Not only does this aid in the unambiguous discovery of compliant service instances but also in validating their descriptions and support for the standard. Another benefit of associating an IVOA identifier with a standard is that it allows registry users who discover services that conform to a particular standard to also discover the document that describes that standard.

StandardsRegExt has two other purposes. First, it allows a service protocol description to communicate specifics about the standard input parameters and output formats specified by the standard. Such a machine-readable description of the interface can assist intelligent portals and applications to build GUI interfaces to standard services and manage workflows built around them. Second, it allows for the definition of small controlled sets of standardized names (referred to as *standard keys* or *keys* in this document) which might be used to identify, for example, specific features of a standard protocol (such as supported data transport protocols). By virtue of being defined within the context of a VOResource description, one can refer to the key using a globally unique URI by adding the key name as a URI fragment onto the IVOA identifier associated with the descriptions.

---

PDF fallback: A conversion from SVG to PDF failed. This is probably because inkscape is not installed. While SVG is not supported by the major TeX engines, it is recommended to commit built PDFs to the VCS.

---

*Figure 1: Architecture diagram for this document*

StandardsRegExt records that describe standards endorsed or otherwise in development by the IVOA are published in the IVOA Registry of Registries (Plante, 2007) using the authority identifier `ivoa.net` as discussed in Sect. 4. However, other standards, be they ad hoc or endorsed by another body, may be published in any compliant registry.

### 1.1 Role within the VO Architecture

Fig. 1 shows the role this document plays within the IVOA architecture (Dowler and Evans et al., 2021). The Registry enables applications in the User Layer to discover archives and services in the Resource Layer. The registry metadata model standards (in blue text and boxes) give structure to the information that enables that discovery. StandardsRegExt defines metadata for describing standards themselves (like those that define the Data Access Protocols).

In this architecture, users can leverage a variety of tools (from the User Layer) to discover archives and services of interest (represented in the Resource Layer); registries provide the means for this discovery. A registry is a repository of descriptions of resources that can be searched based on the metadata in those descriptions. In general, a resource can be more than just archives, data, or services; an IVOA standard, as represented by an IVOA document, can also be a resource. The Resource Metadata standard defines the core concepts used in the resource descriptions, and VOResource defines the XML format. StandardsRegExt is an extension of the VOResource standard that defines extra metadata for describing a standard.

### 1.2 Syntax Notation Using XML Schema

The eXtensible Markup Language, or XML, is a document syntax for marking textual information with named tags and is defined by the World Wide Web Consortium (W3C) Recommendation, XML 1.0 (Bray and Paoli et al.,

2008). The set of XML tag names and the syntax rules for their use is referred to as the document schema. One way to formally define a schema for XML documents is using the W3C standard known as XML Schema (Thompson and Beech et al., 2004).

This document defines the StandardsRegExt schema using XML Schema. Parts of the schema appear within the main sections of this document; however, documentation nodes have been left out for the sake of brevity. The full schema is available from the IVOA schema repository<sup>3</sup>. For documentation and development purposes, this document is accompanied by a copy of that schema<sup>4</sup>. In case of conflicts, the copy at the schema repository is normative.

Reference to specific elements and types defined in the VOResource schema include the namespaces prefix, `vr`, as in `vr:Resource` (a type defined in the VOResource schema). Reference to specific elements and types defined in the StandardsRegExt schema include the namespaces prefix, `vstd`, as in `vstd:ServiceStandard` (a type defined in the StandardsRegExt schema). Use of the `vstd` prefix in compliant instance documents is strongly recommended. It is required where the Registry Interfaces standard (Dower and Demleitner et al., 2018) applies.

## 2 The StandardsRegExt Data Model

The StandardsRegExt extension in general enables the description of two types of resources:

- a generic standard (specified by an external document)
- a standard specifically defining a service protocol

Each type of resource can have a set of related, standardized names called *keys*.

Here’s an example of defining the HiPS standard and its associated *keys*.

```
<ri:Resource xsi:type="vstd:Standard"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:vstd="http://www.ivoa.net/xml/StandardsRegExt/v1.0"
  xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
  created="2017-06-01T09:33:00Z"
  updated="2017-06-01T09:33:00Z"
  status="active">
  <title>HiPS -- Hierarchical Progressive Survey</title>
  <identifier>ivo://ivoa.net/std/hips</identifier>
  <curation>
```

<sup>3</sup><https://ivoa.net/xml/index.html>

<sup>4</sup><https://www.ivoa.net/documents/StandardsRegExt/20230911/StandardsRegExt-v1.1.xsd>

```

<publisher>IVOA Apps Working Group</publisher>
<creator>
<name>Fernique, P.</name>
</creator>
...
<date role="created">2017-06-01T09:33:00</date>
<contact>
  <name>IVOA Apps Working Group</name>
  <email>apps@ivoa.net</email>
</contact>
</curation>
<content>
<subject>Virtual observatory</subject>
<subject>Standards</subject>
<subject>HiPS</subject>
  <description>
    HiPS: a hierarchical scheme for the description , storage and access of
    sky survey data. The system is based on hierarchical tiling of sky regions
    at finer and finer spatial resolution which facilitates a progressive
    view of a survey, and supports multi-resolution zooming and panning.
    HiPS uses the HEALPix tessellation of the sky as the basis for the scheme
    and is implemented as a simple file structure with a direct indexing
    scheme that leads to practical implementations.
  </description>
<referenceURL>http://ivoa.net/documents/HiPS</referenceURL>
</content>
<endorsedVersion status="rec" use="preferred">1.0</endorsedVersion>
<key>
  <name>hipslist-1.0</name>
  <description>
    A service returning a list of HiPS identifiers
    and metadata for HiPS. This term is used to form a standardID,
    for instance for use in vr:Capability.
  </description>
</key>
<key>
  <name>hips-1.0</name>
  <description>
    A single HiPS. This term is used to form a standardID,
    for instance for use in vr:Capability.
  </description>
</key>
</ri:Resource>

```

## 2.1 The Schema Namespace and Location

The namespace associated with StandardsRegExt extensions is

<http://www.ivoa.net/xml/StandardsRegExt/v1.0>.

Just like the namespace URI for the VOResource schema, the Standards-RegExt namespace URI can be interpreted as a URL. Resolving it will return

the XML Schema document that defines the StandardsRegExt schema. This namespace is constant for all versions of StandardsRegExt version one, in particular the current version 1.1. See [Harrison and Demleitner et al. \(2018\)](#) (section 2.2.3) for the background of this slightly confusing convention.

Authors of VOResource instance documents may choose to provide a location for the VOResource XML Schema document and its extensions using the *xsi:schemaLocation* attribute. While the choice of the location value is the choice of the author, this specification recommends using the StandardsRegExt namespace URI as its location URL (as illustrated in the example above), as in,

```
xsi:schemaLocation="http://www.ivoa.net/xml/StandardsRegExt/v1.0
                    http://www.ivoa.net/xml/StandardsRegExt/v1.0"
```

The prefix, *vstd*, is used by convention as the prefix defined for the StandardsRegExt schema; however, instance documents may use any prefix of the author's choosing. In applications where common use of prefixes is recommended (such as with the Registry Interface specification), use of the *vstd* prefix is recommended. Note also that in this document, the *vr* prefix is used to label, as shorthand, a type or element name that is defined in the VOResource schema, as in *vr:Resource*.

As recommend by the VOResource standard, the StandardsRegExt schema sets `elementFormDefault="unqualified"`. This means that it is not necessary to qualify element names defined in this schema with a namespace prefix (as there are no global elements defined). The only place it is usually needed is as a qualifier to a StandardsRegExt type name given as the value of an *xsi:type* attribute.

## 2.2 Summary of Metadata Concepts

The StandardsRegExt extension defines two new types of resources which are specifically for independently documented standards:

*vstd:Standard* This resource describes a general standard (e.g. data model, schema, protocol, etc.). The most important piece of metadata associated with this resource is the *referenceURL* (from the core VOResource schema) which should point to the human-readable specification document that defines the standard. It also allows one to provide the recommended version of the standard to use.

*vstd:ServiceStandard* This resource type, which extends from *vstd:Standard*, is specifically for describing a standard service protocol (e.g. Simple Cone Search). It differs from *vstd:Standard* in that it also allows one to describe specific constraints on the service interface via its *interface* element.



---

**Note**

As mentioned above, this standard allows controlled names to be defined either as part of a record of any of the above two types. When such names are being defined as part of an IVOA standard, it is recommended that the *vstd:Standard* or *vstd:ServiceStandard* record corresponding to the IVOA standard document be used to define the names. The deprecated *vstd:StandardKeyEnumeration* was originally envisioned to as a container for names. With the adoption of Vocabularies in the VO 2 (Demleitner and Gray et al., 2021), this type has become obsolete.

---

### 2.3 Defining Enumerations of Identifiers

A common practice when defining metadata is to restrict certain string values to a controlled set of defined names, each with a well-defined meaning. With XML Schema, the controlled set can be enforced by a validating parser (using the *xsd:enumeration* construct). One disadvantage of locking in the vocabulary in an XML Schema document is that in order to grow the list of allowed names, a revision of the XML Schema document is required, which can be a disruptive change. To avoid this, it is the practice within VOResource and its extensions to avoid “hard-coded” enumerations in the XML Schema document for sets of defined values that will likely change over time.

The StandardsRegExt schema provides an alternative to XML Schema-based definitions of controlled names. Instead, a controlled list of names, called *standard keys*, can be defined as part of any of the two StandardsRegExt resource types. Updating a resource description is much less disruptive than a Schema document, and as a resource is available via an IVOA-compliant registry, it is still possible for a (non-Schema-based) application to validate the use of the vocabulary.

The StandardsRegExt specification also defines a mapping from a key name to a URI. This allows these keys—and their underlying meaning – to be referenced in a globally unique way in a variety of contexts, not restricted to XML.

A key is defined using the *vstd:StandardKey* type which consists simply of a name and a description. The key is mapped to a URI by attaching the name as the fragment – i.e., appending after a pound sign, # – to the IVOA identifier for the resource description that defines the key:

$$ivoa-identifier\#key-name,$$

where *ivoa-identifier* is the resource’s IVOA identifier and *key-name* is the name of a key defined in that resource. Consistent with the URI standard, the *key-name* must not contain a pound (#) sign.

---

**Note**

When these enumerations are presented to a user in a GUI it is expected that only the fragment part that distinguishes the various members of the enumeration will be used as a choice value, as the full IVO ID is not usually particularly “user-friendly”.

---

For example, we consider a resource description with an IVOA identifier given by `ivo://ivoa.net/std/QueryProtocol` that defines a key named `case-insensitive`; the URI identifying this key would be:

`ivo://ivoa.net/std/QueryProtocol#case-insensitive`.

Note that registry references in the sense of *IVOA Identifiers 2* (i.e., the part in front of the hash character) must be compared case-insensitively. To facilitate comparisons of full IVOIDs, since version 1.1 this specification requires that standard key names do not contain uppercase letters. In this way, clients can safely lowercase the full standard key references before comparing them.

This is also true for all IVOA-approved standard keys before version 1.1 except for the keys for authentication methods from [Taffoni and Schaaf et al. \(2017\)](#). Clients implementing other IVOA standards can hence safely compare IVOIDs for standard keys by checking for identity after lowercasing.

This form of defining multiple keys, each with its own mapping to a URI, all in one resource has several advantages:

- The enumerations are naturally grouped under a single registry resource, and so can be retrieved with one registry query and need no further metadata to assert the association.
- The “Dublin core” metadata that is associated with a resource need only be entered once for the whole enumeration, rather than for each member of the enumeration – this saves both curation effort and space in the registry.
- If it is necessary to expand the list of controlled names (or shrink it), it is simple and fairly undistruptive to update the VOResource record.

Some applications may wish to publish additional metadata associated with a key definition through further extension of VOResource metadata. This can be done by deriving a new key metadatum type derived by extension from the *vstd:StandardKey*.

## 3 The StandardsRegExt Metadata

### 3.1 Resource Type Extensions

This specification defined two new resource types. As is spelled out in the VOResource specification, a resource description indicates that it refers to one of these types of resources by setting the *xsi:type* attribute to the namespace-qualified type name. Doing so implies that the semantic meaning of that type applies to the resource.

#### 3.1.1 Standard

The *vstd:Standard* resource type describes any general standard specification. This typically refers to an IVOA standard but is not limited to such. Generally, the *vstd:Standard* type is intended for standards *other than* standard protocols (which should use the *vstd:ServiceStandard* type instead). It extends the generic *vr:Resource* type as follows.

#### *vstd:Standard* Type Schema Documentation

a description of a standard specification.

This typically refers to an IVOA standard but is not limited to such.

#### *vstd:Standard* Type Schema Definition

```
<xs:complexType name="Standard" >
  <xs:complexContent >
    <xs:extension base="vr:Resource" >
      <xs:sequence >
        <xs:element name="endorsedVersion"
          type="vstd:EndorsedVersion"
          maxOccurs="unbounded" />
        <xs:element name="schema" type="vstd:Schema" minOccurs="0"
          maxOccurs="unbounded" />
        <xs:element name="deprecated" type="xs:token" minOccurs="0" />
        <xs:element name="key" type="vstd:StandardKey" minOccurs="0"
          maxOccurs="unbounded" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

#### *vstd:Standard* Extension Metadata Elements

##### Element *endorsedVersion*

*Type* a string with optional attributes

*Meaning* the version of the standard that is recommended for use.

*Occurrence* required; multiple occurrences allowed.

*Comment* More than one version can be listed, indicating that any of these versions are recognized as acceptable for use.

Element *schema*

*Type* composite: *vstd:Schema*

*Meaning* a description and pointer to a schema document defined by this standard.

*Occurrence* optional; multiple occurrences allowed.

*Comment* This is most typically an XML Schema, but it need not be strictly.

Element *deprecated*

*Type* string: *xs:token*

*Meaning* when present, this element indicates that all versions of the standard are considered deprecated by the publisher. The value is a human-readable explanation for the deprecation.

*Occurrence* optional

*Comment* The explanation should indicate if another standard should be preferred.

Element *key*

*Type* composite: *vstd:StandardKey*

*Meaning* a defined key associated with this standard.

*Occurrence* optional; multiple occurrences allowed.

As one of the purposes of this resource type is to enable users to discover the documentation that defines the standard that the resource describes, the *referenceURL* should point either to the standard’s specification document or to summary information about the standard that can lead one to the specification document.

The child *key* elements define terms with special meaning to the standard; see Sect. 3.2.

The purpose of the required *endorsedVersion* element is to point potential users of the standard to the version that is most preferred by the standard’s publisher. If multiple versions are relevant or in use, multiple elements may be given; in this case, the *use* attribute can further help steer the users to the preferred version.

### *vstd:EndorsedVersion* Type Schema Definition

```
<xs:complexType name="EndorsedVersion" >
  <xs:simpleContent >
    <xs:extension base="xs:string" >
      <xs:attribute name="status" default="n/a" >
        <xs:simpleType >
          <xs:restriction base="xs:string" >
            <xs:enumeration value="rec" />
            <xs:enumeration value="pr" />
            <xs:enumeration value="wd" />
            <xs:enumeration value="iwd" />
          </xs:restriction >
        </xs:simpleType >
      </xs:attribute >
    </xs:extension >
  </xs:simpleContent >
</xs:complexType >
```

```

        <xs:enumeration value="note" />
        <xs:enumeration value="pen" />
        <xs:enumeration value="en" />
        <xs:enumeration value="n/a" />
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="use" >
    <xs:simpleType >
        <xs:restriction base="xs:string" >
            <xs:enumeration value="preferred" />
            <xs:enumeration value="deprecated" />
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>

```

#### *vstd:EndorsedVersion* Attributes

##### status

<i>Type</i>	string with controlled vocabulary
<i>Meaning</i>	the IVOA status level of this version of the standard.
<i>Occurrence</i>	optional
<i>Allowed Values</i>	
rec	an IVOA Recommendation
pr	an IVOA Proposed Recommendation
wd	an IVOA Working Draft
iwd	an internal Working Draft of an IVOA Working Group
note	a published IVOA Note
pen	a Proposed Endorsed Note
en	an Endorsed Note
n/a	not an IVOA standard or protostandard at this time.
<i>Default</i>	n/a
<i>Comment</i>	For values of “rec”, “pr”, “wd”, “note”, “pen”, and “en” the record’s referenceURL element should point to the official specification document in the IVOA Document Repository; if the document does not appear there, these values should not be used.

##### use

<i>Type</i>	string with controlled vocabulary
<i>Meaning</i>	A designation of preference for the version compared to other versions in use.

---

**Note**

An example where the *deprecated* element might be used in the VO is in the case of the SkyNode standard. When StandardsRegExt was originally written, there were many instances of SkyNode services available in the VO, and where they were used, version 1.01 was endorsed; however, the IVOA has deprecated this standard in favor of the Table Access Protocol. Thus, a *vstd:ServiceStandards* record for SkyNode should include a *deprecated* element whose content refers viewers to the TAP standard.

---

*Occurrence* optional

*Allowed Values*

preferred

the most preferred version.

deprecated

a version whose use is now discouraged because a newer version is preferred.

When all versions of the standard are considered deprecated by the resource publisher, the *deprecated* child element should appear. The explanation given as a value should mention the standard that the current standard is deprecated by when relevant.

The *<schema>* element allows one to list the locations of any schemas defined by the standard thereby making them discoverable as well (just as the specification document is discoverable via the *referenceURL* element). It also can provide pointers to example uses of the schemas. Typically (especially for IVOA standards), the schema is an XML schema, and the location points to an XML Schema document; however, this is not required. Other schema types and definition formats are allowed.

***vstd:Schema* Type Schema Documentation**

a description of a schema definition

***vstd:Schema* Type Schema Definition**

```
<xs:complexType name="Schema" >
  <xs:sequence >
    <xs:element name="location" type="xs:anyURI" minOccurs="1"
      maxOccurs="1" />
    <xs:element name="description" type="xs:token" minOccurs="0"
      maxOccurs="1" />
    <xs:element name="example" type="xs:anyURI" minOccurs="0"
      maxOccurs="unbounded" />
  </xs:sequence>
```

```
<xs:attribute name="namespace" type="xs:token" use="required" />
</xs:complexType>
```

### *vst*:Schema Attributes

#### namespace

*Type* string: *xs:token*

*Meaning* an identifier for the schema being described. Each instance of this attribute must be unique within the resource description.

*Occurrence* required

*Comment* For XML schemas, this should be the schema's namespace URI. Otherwise, it should be a unique label to distinguish it from other schemas described in the same resource description.

### *vst*:Schema Metadata Elements

#### Element *location*

*Type* a URI: *xs:anyURI*

*Meaning* A URL pointing to a document that formally defines the schema.

*Occurrence* required

*Comment* The document should be in a machine-parsable format when applicable. For example, when referring to an XML schema, the document should be an XML Schema or similar document that can be used to validate an instance document.

#### Element *description*

*Type* string: *xs:token*

*Meaning* A human-readable description of what the schema defines or is used for.

*Occurrence* optional

*Comment* A brief description—e.g. one statement—is recommended for display purposes.

#### Element *example*

*Type* a URI: *xs:anyURI*

*Meaning* A URL pointing to a sample document that illustrates the use of the schema.

*Occurrence* optional; multiple occurrences allowed.

*Comment* When applicable (e.g. XML), the document should be in the format defined by the schema document.

As multiple schemas can be listed in the resource description, the **namespace** attribute provides an identifying label for each **<schema>** element.

The main component of the *schema* content is the URL pointing to the schema's definition document, but it can also provide additional information useful for display.

### 3.1.2 ServiceStandard

The *vstd:ServiceStandard* resource type extends *vstd:Standard* to describe more specifically a standard protocol. It adds an *interface* element to allow the interface defined by the standard to be described in a machine-readable way. Its type is defined to be *vr:Interface*, which is defined in the VOResource schema.

#### *vstd:ServiceStandard* Type Schema Documentation

a description of a standard service protocol.

This typically refers to an IVOA standard but is not limited to such.

#### *vstd:ServiceStandard* Type Schema Definition

```
<xs:complexType name="ServiceStandard" >
  <xs:complexContent >
    <xs:extension base="vstd:Standard" >
      <xs:sequence >
        <xs:element name="interface" type="vr:Interface" minOccurs="0"
          maxOccurs="unbounded" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

#### *vstd:ServiceStandard* Extension Metadata Elements

##### Element *interface*

*Type* composite: *vr:Interface*

*Meaning* an abstract description of one of the interfaces defined by this service standard.

*Occurrence* optional; multiple occurrences allowed.

*Comment* This element can provide details about the interface that apply to all implementations. Each interface element should specify a role with a value starting with "std:" or, if there is only one standard interface, is equal to "std".

*Comment* Applications that, for example, wish to build a GUI to the service on-the-fly would first access this generic description. Site-specific variations, such as supported optional arguments or site specific arguments, would be given in the actual implemented service description and overrides any common information found in this generic description. This generic interface description can be matched with the site-specific one using the role attribute.

*Comment* Even though the Interface type requires an accessURL child element, this element is intended to describe a service in the abstract—i.e. without reference to a particular installation of the service. Consequently, the accessURL may contain a bogus URL; applications should not expect it to be resolvable.



Even though the `vr:Interface` type requires an `<accessURL>` child element, the `<interface>` element in a `vstd:ServiceStandard` is intended to describe a service in the abstract—i.e. without reference to a particular installation of the service. Consequently, the `accessURL` should contain a bogus URL; applications should not expect it to be resolvable.

An application can, in principle, get a complete machine-readable description of a particular instance of a standard service (to, say, create a GUI for that service on-the-fly) by combining the general description in the `vstd:ServiceStandard` record with the service resource description for the specific instance. The intended process for building that description is as follows:

1. The application obtains a `VOResource` resource record for the service instance (e.g. from a registry).
2. The application extracts the `standardID` attribute for the desired service capability.
3. The `standardID` is resolved (via a registry) to a `vstd:ServiceStandard` record for the service. This description would capture the required and optional (but standard) components of the service interface.
4. The specific instance’s interface description is merged into the standard one. The service’s support of optional components as well as its allowed customizations would override the generic description from the `vstd:ServiceStandard` record.

The so-called “simple” data access layer (DAL) services, such as the Simple Image Access services (Dowler and Bonnarel et al., 2015), are registered using the `vs:ParamHTTP` interface type (Demleitner and Plante et al., 2021) to describe its interface. This interface type allows one to list input parameters accepted by the service. Each parameter can be marked as *required*, *optional*, or *ignored*. Typically with DAL services, parameters defined as optional by the standard may be legally ignored by an implementation. Consequently, this specification recommends special instruction for listing and interpreting input parameters in a `vstd:ServiceStandard` record when the interface is of type `vs:ParamHTTP`: parameters that can be optionally provided in a client’s query but are ignorable by the implementation should be marked as *ignored*. Applications that consume such `vstd:ServiceStandard` records should thus interpret the parameters marked *ignored* as *optional* for use by clients and *ignorable* by implementations. This minimizes the list of parameters that the service provider must list in the registration of a particular service instance to the ones that are actually supported (i.e., not ignored): when the list service instance description is merged into the list

from the *vstd:ServiceStandard* record (step 4 above), the result is an accurate list of the optional but supported and the ignored parameters for that service instance.

An example of an instance of a *vstd:ServiceStandard* record is shown in Appendix A. It describes the Simple Image Access Specification and in particular illustrates the recommended way to list input parameters defined by the standard.

### 3.2 Defining Keys: StandardKey and StandardKeyURI

The *vstd:StandardKey* type provides the means to define keys (as defined in Sect. 2.3) within a VOResource record.

#### *vstd:StandardKey* Type Schema Documentation

The name and definition of a key—a named concept, feature, or property.

This key can be identified via an IVOA identifier of the form `ivo://authority/resource#name` where name is the value of the child name element.

This type can be extended if the key has other metadata associated with it.

#### *vstd:StandardKey* Type Schema Definition

```
<xs:complexType name="StandardKey" >
  <xs:sequence >
    <xs:element name="name" type="vstd:fragment" />
    <xs:element name="description" type="xs:token" />
  </xs:sequence>
</xs:complexType>
```

#### *vstd:StandardKey* Metadata Elements

##### Element *name*

*Type* string of the form:  $([A-Za-z0-9;\backslash?:@\&=\backslash+\$,\_ \backslash.\! \backslash*\backslash\(\)])\%[A-Za-z0-9]\{2\})+$

*Meaning* The property identifier which would appear as the fragment (string after the pound sign, #) in an IVOA identifier.

*Occurrence* required

*Comment* To facilitate comparisons of such keys given that registry identifiers are case insensitive, all newly created standard keys must not contain uppercase letters.

##### Element *description*

*Type* string: *xs:token*

*Meaning* A human-readable definition of this property.

*Occurrence* required

Defining a key via a *key* element within a VOResource record implies the definition of a unique URI formed according to the syntax described in

---

**Note**

It is worth noting that just using or otherwise referencing a standard key URI in an application does not require importing the StandardsRegExt nor need there be any reference to the StandardsRegExt namespace. The role of the StandardsRegExt schema is simply to provide a way of documenting the definitions in a VOResource record. Thus, an application may dereference the URI for display or user help purposes; however, dereferencing is not necessary to use the URI.

---

Sect. 2.3 that represents the semantics given by the value of the *description* child element. Because the URI must be globally unique, the key name (given by the *name* child element) must be unique within the VOResource record.

Though it is not needed by StandardsRegExt resource records, the StandardsRegExt schema further defines a convenience type, *vstd:StandardKeyURI*, which defines the legal pattern for a full standard key identifier (as defined in Sect. 2.3). Applications that wish to use XML Schema to validate the form of a key URI may import the StandardsRegExt schema and use this type.

## 4 Operational Aspects

### 4.1 Management of Standards Records

Only IVOA standards that actually make use of identifiers to or into StandardsRegExt records MUST be accompanied with active records published by the *ivoa.net* registry. This, in particular, concerns all standards defining capabilities to be discerned using their *standardID* attribute. Other standards MAY be registered in this way once they have entered the IVOA document repository.

There are no regulations on StandardsRegExt records published under authorities other than *ivoa.net*; these can be published (and withdrawn) at any time.

Editors of standards requiring an *ivoa.net* StandardsRegExt record SHOULD upload one when they submit the first Working Draft in order to make IVOIDs valid even in prototyping phases. They MUST upload it at the beginning of the standard's RFC phase to ensure validators can work properly, and they MUST update the record when the standard reaches REC status.

Submission to the *ivoa.net* registry generally happens through e-mail. See <http://rofr.ivoa.net/> for contact information.

The *ivoatex* package (Demleitner and Taylor et al., 2022) offers templates for StandardsRegExt records and recommends keeping these in the respective

standards’ source code repositories.

## 4.2 Management of Standard Keys

Standards using standard keys must specify rules defining when and how standard keys are being added. In the simplest case, this would be “Keys in this standard’s StandardsRegExt record are defined in this text.” This will make sure that additions of keys receive the same review as the standard text itself.

However, other, more flexible regulations are conceivable, as for instance “Keys in this standard’s StandardsRegExt record can be added by the chair of the managing Working Group after consultation on the WG’s mailing list”. However, complicated processes should be avoided here; if regular key additions requiring consensus building are expected, the standard should probably rather define a proper IVOA vocabulary (Demleitner and Gray et al., 2021); this also has the advantage over StandardsRegExt keys that non-VO RDF tools can evaluate the resources produced, which is not the case for StandardsRegExt records.

## A A Sample Record

This example shows how one can describe an IVOA standard, the Simple Image Access Protocol. It includes a description of the input parameters defined in the specification. Note that this no longer corresponds to the actual SimpleDALRegExt record for the Simple Image Access Protocol as distributed by the Registry of Registries.

```
<resource xsi:type="vstd:ServiceStandard" status="active"
  created="2000-01-01T09:00:00" updated="2000-01-01T09:00:00"
  xmlns:vstd="http://www.ivoa.net/xml/StandardsRegExt/v1.0"
  xmlns:vs="http://www.ivoa.net/xml/VODataService/v1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ivoa.net/xml/VODataService/v1.1
    http://www.ivoa.net/xml/VODataService/v1.1">

  <title> Simple Image Access Protocol </title>
  <shortName> SIA </shortName>
  <identifier> ivo://ivoa.net/std/SIA </identifier>

  <curation>
    <publisher ivo-id="ivo://ivoa.net/IVOA">
      International Virtual Observatory Alliance
    </publisher>
    <creator>
      <name> Doug Tody </name>
    </creator>
    <creator>
      <name> Ray Plante </name>
```

```

</creator>
<date> 2004-05-24 </date>
<contact>
  <name> Data Access Layer Working Group </name>
<email> dal@ivoa.net </email>
</contact>
</curation>

<content>
  <subject> software standard </subject>
  <subject> virtual observatory </subject>
  <description>
    The Simple Image Access Protocol is a protocol for retrieving
    image data from a variety of astronomical image repositories
    through a uniform interface. The interface is meant to be
    reasonably simple to implement by service providers. A query
    defining a rectangular region on the sky is used to query for
    candidate images. The service returns a list of candidate
    images formatted as a VOTable. For each candidate image an
    access reference URL may be used to retrieve the image. Images
    may be returned in a variety of formats including FITS and
    various graphics formats. Referenced images are often computed
    on the fly, e.g., as cutouts from larger images.
  </description>
  <referenceURL>
    http://www.ivoa.net/Documents/latest/SIA.html
  </referenceURL>
  <type> Other </type>
  <contentLevel> Research </contentLevel>
</content>

<endorsedVersion status="rec"> 1.0 </endorsedVersion>

<interface xsi:type="vs:ParamHTTP" role="std" version="1.0">
  <accessURL use="base"> http://sample.org/cgi-bin/sia </accessURL>
  <queryType>GET</queryType>
  <resultType>text/xml+votable</resultType>

  <!--
  - These are the standard input parameters defined in the
  - SIA spec
  -->
  <param use="required">
    <name>POS</name>
    <description>
      Search Position in the form "ra,dec" where ra and dec are given
      in decimal degrees in the ICRS coordinate system.
    </description>
    <unit>degrees</unit>
    <dataType arraysize="2">real</dataType>
  </param>
  <param use="required">
    <name>SIZE</name>
    <description>

```

```

        Size of search region in the RA and Dec. directions in decimal
        degrees.
    </description>
    <unit>degrees</unit>
    <dataType arraysize="2">real</dataType>
</param>
<param use="optional">
    <name>FORMAT</name>
    <description>Requested format of images.</description>
    <dataType>string</dataType>
</param>
<param use="optional">
    <name>INTERSECT</name>
    <description>
        Choice of intersection of matched images with the region of
        interest .
    </description>
    <dataType>string</dataType>
</param>
<param use="ignored">
    <name>NAXIS</name>
    <description>
        The number of pixels desired along each axis
    </description>
    <dataType arraysize="2">integer</dataType>
</param>
<param use="ignored">
    <name>CFRAME</name>
    <description>
        the coordinate frame to impose on the image.
    </description>
    <dataType>string</dataType>
</param>
<param use="ignored">
    <name>EQUINOX</name>
    <description>
        the epoch of the mean equator and equinox for the specified
        coordinate system reference frame (CFRAME). coordinate frame to
        impose on the image.
    </description>
    <dataType>string</dataType>
</param>
<param use="ignored">
    <name>CRPIX</name>
    <description>
        the pixel position to locate the reference position in the output
        image.
    </description>
    <dataType arraysize="2">real</dataType>
</param>
<param use="ignored">
    <name>CRVAL</name>
    <description>
        the world coordinates of the reference position in the output

```

```

        image in decimal degrees
      </description>
      <dataType arraysize="2">real</dataType>
    </param>
    <param use="ignored">
      <name>CDELTA</name>
      <description>
        the scale of the output image in decimal degrees per pixel
      </description>
      <dataType arraysize="2">real</dataType>
    </param>
    <param use="ignored">
      <name>ROTANG</name>
      <description>
        the rotation angle to put into the coordinate system of the output
        image
      </description>
      <dataType arraysize="2">real</dataType>
    </param>
    <param use="ignored">
      <name>PROJ</name>
      <description>
        the projection to impose in the construction of the output image
      </description>
      <dataType>string</dataType>
    </param>
    <param use="ignored">
      <name>VERB</name>
      <description>
        the level of verbosity in the output.
      </description>
      <dataType>string</dataType>
    </param>
  </interface>
</resource>

```

## B Changes from Previous Versions

### B.1 Changes since Rec-1.0

- Corrected and clarified some wordings.
- Removed references to *vstd:StandardKeyEnumeration* (which is now deprecated) from this document and associated schema. Replaced computer language example by HiPS standard definition example.
- Added en and pen to the list of document types.
- Adding regulations on record upload and key additions.

- Requiring new keys to be all-lowercase in order to simplify comparisons.
- Ported to ivoatex. This entailed some re-shuffling the the previous content.
- Dropped the “example of a Standard resource that summarizes this specification” (we have an example in the appendix already, and it’s not clear why there would be another one inline).
- Dropped the in-document full schema (we now have auxiliaryurl, and it’s unclear whether it’s a good idea to have a second copy of the schema anyway).

## B.2 Changes since PR-v1.0 20120217:

- none other than date and status.

## B.3 Changes since PR-v1.0 20120213:

- added the *schema* element to *vstd:Standard*
- updated example for *vstd:Standard*

## B.4 Changes since PR-v1.0 20111017:

- added Note box that recommends against using *vstd:StandardKeyEnumeration* to describe keys when they are defined by an IVOA standard.
- added statement highlighting that # signs are not allowed in key names.
- added *iwid* and *note* as allowed values for *vstd:EndorsedVersion*’s *status* attribute.
- converted a Note box in to a normative paragraph that recommends listing optional *ParamHTTP* parameters as ignored. Note that there is a related error in the VODataService standard: while the *ignored* value is defined in the schema, it is not included in the document text.
- added sample *vstd:ServiceStandard* instance record back in as Appendix B
- added references to TAP and SIA
- fixed various grammatical typos.



### B.5 Changes since PR-v1.0 20110921:

- In *endorsedVersion*, changed “prop” to “pr”.
- various typo corrections

### B.6 Changes since PR-v1.0 20110316:

- Corrected ampersand representation in schema listing
- Various typo corrections and clarifications

### B.7 Changes since WD-v1.0 20100519:

- Prepped for PR
- improved discussion of example in section on the data model
- Added standard architecture sub-section
- updated in-lined schema (App. 1)

### B.8 Changes since WD-v1.0 20100514:

- short name changed from VOStandard to StandardsRegExt

### B.9 Changes since WD-v0.4:

- removed App. B. (Sample instance) as examples appear throughout the document.

## References

Bradner, S. (1997), ‘Key words for use in RFCs to indicate requirement levels’, RFC 2119.

<http://www.ietf.org/rfc/rfc2119.txt>

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E. and Yergeau, F. (2008), ‘Extensible markup language (XML) 1.0 (fifth edition)’, W3C Recommendation.

<http://www.w3.org/TR/REC-xml/>

Demleitner, M., Gray, N. and Taylor, M. (2021), ‘Vocabularies in the VO Version 2.0’, IVOA Recommendation 25 May 2021.

<http://doi.org/10.5479/ADS/bib/2021ivoa.spec.0525D>

- Demleitner, M., Plante, R., Linde, T., Williams, R. and Noddle, K. (2016), ‘IVOA Identifiers Version 2.0’, IVOA Recommendation 23 May 2016, arXiv:1605.07501.  
<http://doi.org/10.5479/ADS/bib/2016ivoa.spec.0523D>
- Demleitner, M., Plante, R., Stébé, A., Benson, K., Dowler, P., Graham, M., Greene, G., Harrison, P., Lemson, G., Linde, T. and Rixon, G. (2021), ‘VODataService: A VOResource Schema Extension for Describing Collections, Services Version 1.2’, IVOA Recommendation 02 November 2021.  
<http://doi.org/10.5479/ADS/bib/2021ivoa.spec.1102D>
- Demleitner, M., Taylor, M., Dempsey, J., Dowler, P., Harrison, P. and Molinaro, M. (2022), ‘The IvoaTeX document preparation system version 1.3’, IVOA Note 25 May 2022.  
<https://ivoa.net/documents/Notes/IVOATexDoc/>
- Dower, T., Demleitner, M., Benson, K., Plante, R., Auden, E., Graham, M., Greene, G., Hill, M., Linde, T., Morris, D., O’Mullane, W., Rixon, G., Stébé, A. and Andrews, K. (2018), ‘Registry Interfaces Version 1.1’, IVOA Recommendation 23 July 2018.  
<http://doi.org/10.5479/ADS/bib/2018ivoa.spec.0723D>
- Dowler, P., Bonnarel, F. and Tody, D. (2015), ‘IVOA Simple Image Access Version 2.0’, IVOA Recommendation 23 December 2015.  
<http://doi.org/10.5479/ADS/bib/2015ivoa.spec.1223D>
- Dowler, P., Evans, J., Arviset, C., Gaudet, S. and Technical Coordination Group (2021), ‘IVOA Architecture Version 2.0’, IVOA Endorsed Note 01 November 2021.  
<http://doi.org/10.5479/ADS/bib/2021ivoa.spec.1101D>
- Hanisch, R., IVOA Resource Registry Working Group and NVO Metadata Working Group (2007), ‘Resource Metadata for the Virtual Observatory Version 1.12’, IVOA Recommendation 02 March 2007, arXiv:1110.0514.  
<http://doi.org/10.5479/ADS/bib/2007ivoa.spec.0302H>
- Harrison, P., Demleitner, M., Major, B. and Dowler, P. (2018), ‘XML Schema Versioning Policies Version 1.0’, IVOA Endorsed Note 29 May 2018.  
<http://doi.org/10.5479/ADS/bib/2018ivoa.spec.0529H>
- Plante, R. (2007), ‘The Registry of Registries Version 1.00’, IVOA Note 28 June 2007.  
<http://doi.org/10.5479/ADS/bib/2007ivoa.rept.0628P>
- Plante, R., Demleitner, M., Benson, K., Graham, M., Greene, G., Harrison, P., Lemson, G., Linde, T. and Rixon, G. (2018), ‘VOResource: an XML

Encoding Schema for Resource Metadata Version 1.1', IVOA Recommendation 25 June 2018.

<http://doi.org/10.5479/ADS/bib/2018ivoa.spec.0625P>

Taffoni, G., Schaaf, A., Rixon, G. and Major, B. (2017), 'SSO - Single-Sign-On Profile: Authentication Mechanisms Version 2.0', IVOA Recommendation 24 May 2017, arXiv:1709.00171.

<http://doi.org/10.5479/ADS/bib/2017ivoa.spec.0524T>

Thompson, H. S., Beech, D., Maloney, M. and Mendelsohn, N. (2004), 'XML schema part 1: Structures second edition', W3C Recommendation.

<http://www.w3.org/TR/xmlschema-1/>