A Domain Availability Check (DCHK) Registry Type for the Internet Registry Information Service (IRIS)
draft-ietf-crisp-iris-dchk-09

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 17, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).
Abstract

This document describes a lightweight domain availability service using the Internet Registry Information Service (IRIS) framework and the data model of the IRIS Domain Registry (DREG) service.

Table of Contents

1. Introduction .................................................. 3
2. Document Terminology ....................................... 4
3. Domain Availability Check Registry ......................... 5
   3.1. Schema Description ..................................... 5
       3.1.1. The <domain> Result .............................. 5
       3.1.2. Support for <iris:lookupEntity> .................. 8
   3.2. DCHK Formal XML Syntax .................................. 8
   3.3. BEEP Transport Compliance ............................... 13
       3.3.1. Message Pattern .................................. 13
       3.3.2. Server Authentication ............................. 13
   3.4. URI Resolution ........................................... 13
       3.4.1. Application Service Label ......................... 14
       3.4.2. Bottom-Up Resolution ............................... 14
       3.4.3. Top-Down Resolution ............................... 14
4. Internationalization Considerations ......................... 15
5. IANA Considerations ........................................... 16
   5.1. XML Namespace Registration .............................. 16
   5.2. XML Schema Registration .................................. 16
   5.3. S-NAPTR Registration .................................... 16
   5.4. BEEP Registration ....................................... 17
6. Security Considerations ....................................... 18
7. References ..................................................... 19
   7.1. Normative References .................................... 19
   7.2. Informative References ................................... 19
Authors’ Addresses ............................................... 20
Intellectual Property and Copyright Statements ................ 21
1. Introduction

This document describes a lightweight service for checking the availability of domain names. This service is based on the IRIS framework and uses the data model defined by RFC3982 [7]. By doing this, the domain availability service has the advantages provided by IRIS and DREG, such as well-known methods for server navigation, structured queries and results, and layered extensibility.

The use of IRIS for this service also allows seamless integration between the domain availability service and the service provided by DREG. This allows a user to find the availability status of a domain and reference the full registration information in DREG.

The data model in this service (called a registry schema in IRIS terms) is a strict subset of the DREG data model. This enables implementors to directly reuse DREG code paths and allows operators to deploy the service in either the same server processes as a DREG service (same host and port) or in a different server process (different port) or machine (different host).

As an example, an operator may wish to deploy both types of service on the same set of machines. As time goes on, the operator may then decide to segregate the services, placing the domain availability service on one set of machines and the DREG service on a separate set of machines with a stricter set of controls. Either deployment scenario is transparent to the end user and always appear to be seamlessly complementary.

When coupled with [9], this domain availability service is lightweight and extremely efficient for high-volume, public-facing service.
2. Document Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [2].
3. Domain Availability Check Registry

The data model used for the domain availability check (DCHK) service is a strict subset of the DREG data model. This section describes the DCHK registry type.

3.1. Schema Description

References to XML elements with no namespace qualifier are from the schema defined in Section 3.2. References to elements and attributes with the "iris" XML namespace qualifier are from the schema defined in IRIS [6].

The schema present in this document is tied to the protocol version associated to the XML namespace URI defined in Section 5.2. Extensions to the present DCHK schema are allowed—though not recommended—but would require a new version. Please refer to RFC3981 [6] for more details about versioning the IRIS protocol.

The descriptions contained within this section refer to XML elements and attributes and their relation to the exchange of data within the protocol. These descriptions also contain specifications outside the scope of the formal XML syntax. Therefore, this section will use terms defined by RFC 2119 [2] to describe the specification outside the scope of the formal XML syntax. While reading this section, please reference Section 3.2 for needed details on the formal XML syntax.

3.1.1. The <domain> Result

An example of a <domain> result:

```xml
<domain
  authority="iana.org" registryType="dchk1"
  entityClass="domain-name" entityName="example.com">
  <domainName>example.com</domainName>
  <status><active/></status>
</domain>
```

The <domain> result represents an instance of a domain assignment. The children of the <domain> element are as follows:

- <domainName> - the full name of the domain as it is in DNS. The contents of this element MUST be a domain name as specified by RFC 1035 [1].
o <idn> - the name of the domain in nameprep form if applicable.
   See RFC 3491 [3].

o <status> - this element contains child elements representing
domain status information. It defines the following status types:

* <active> - available via DNS (either via delegation or direct
  publication)

* <inactive> - unavailable via DNS

* <dispute> - registrant assignment is in dispute

* <renew> - renewal of domain registration

* <addPeriod> - the domain is in the grace period after creation
  or activation (see RFC 3915 [5]).

* <renewPeriod> - the domain is in the grace period after renewal
  (see RFC 3915 [5]).

* <autoRenewPeriod> - the domain is in the grace period after
  automatic renewal (see RFC 3915 [5]).

* <transferPeriod> - the domain is in the grace period after
  transfer (see RFC 3915 [5]).

* <redemptionPeriod> - the domain is in the grace period after
  deletion (see RFC 3915 [5]).

* <restore> - change to previous status of this domain

* <policyCompliant> - the domain is considered compliant
  according to a given policy specified by the substatus
  identifier.

* <policyNoncompliant> - the domain is not considered compliant
  according to a given policy specified by the substatus
  identifier.

* <reserved> - the domain is reserved and is not available for
  registration under normal registration procedures.

* <create> - specifies the creation status of the domain in the
  registration system.

* <delete> - specifies the deletion status of the domain in the
  registration system.
* <transfer> - specifies the transfer status of the domain from one responsible or owning entity in the registration system to another.

* <update> - specifies the status of the domain as it relates to information in the domain being modified or having the ability to be modified.

* <other> - specifies a registration system specific status of the domain.

- <registrationReference> - an element containing an entity reference, the referent of which MUST be either a <domain> (Section 3.1.1) or a <domain> as defined by RFC3982 [7]. The intent of this element is to point to the downstream registration reference. Therefore, if this is a result given back by a domain registry, it should point to the domain in the domain registrar or registrant service.

- <createdDateTime> - an element containing the date and time of the creation of this domain

- <initialDelegationDateTime> - an element containing the date and time of the initial delegation of this domain.

- <expirationDateTime> - an element containing the date and time of the expiration of this domain.

- <lastDatabaseUpdateDateTime> - an element containing the date and time of the last actualization of the database that is source for this result

- <iris:seeAlso> - an element containing an entity reference specifying a referent that is indirectly associated with this domain.

3.1.1.1. Domain Status Type

Each element of type ‘domainStatusType’ has the following composition:

- <appliedDate> - an optional child element containing the date applicable to creation of the status.

- <ticket> - an optional child element containing a service ticket identifier relevant to the status.
<description> - zero or more child elements with text to describe the status in natural language. Each of these elements MUST have a 'language' attribute describing the language of the description element.

<subStatus> - a child element indicating further status information. Values for this element are not defined by this specification. This child element has a required 'authority' attribute to indicate the origin of the specification of the value of this element.

'actor' - an optional attribute indicating the acting entity for which this status is applied. The values may be "registry", "registrar" or "registrationServiceProvider".

'disposition' - an optional attribute indicating the nature of this status. The values may be "pending" or "prohibited".

'scope' - an optional attribute indicating the context or origin of the status value.

3.1.2. Support for <iris:lookupEntity>

The following types of entity classes are recognized by the <lookupEntity> query of IRIS for this registry:

- domain-name - the fully qualified name of a domain. This a domain name as specified by RFC 1035 [1]. Yields a <domain> (Section 3.1.1) in the response.

- idn - the fully qualified name of a domain in nameprep form (see RFC 3491 [3]). Yields a <domain> (Section 3.1.1) in the response.

3.2. DCHK Formal XML Syntax

This registry schema is specified in the XML Schema notation (see [10] and [11]). The formal syntax presented here is a complete schema representation suitable for automated validation of an XML instance when combined with the formal schema syntax of IRIS.

```xml
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:dchk="urn:ietf:params:xml:ns:dchk1"
  xmlns:iris="urn:ietf:params:xml:ns:iris1"
  targetNamespace="urn:ietf:params:xml:ns:dchk1"
  elementFormDefault="qualified" >
```
<import namespace="urn:ietf:params:xml:ns:iris1" />

<annotation>
<documentation>
Domain availability check schema
derived from IRIS schema
</documentation>
</annotation>

<!-- ========================================= -->
<!--                                           -->
<!-- Result Types                              -->
<!--                                           -->
<!-- ========================================= -->
<!--                                           -->
<!-- Domain                                    -->
<!-- -->

<complexType name="domainType">
<complexContent>
<extension base="iris:resultType">
<sequence>
<element name="domainName" type="token" />
<element name="idn" type="token" minOccurs="0" maxOccurs="1" />
<element name="status" minOccurs="0" maxOccurs="1">
<complexType>
<choice minOccurs="0" maxOccurs="unbounded">
<element name="active" type="dchk:domainStatusType" />
<element name="inactive" type="dchk:domainStatusType" />
<element name="dispute" type="dchk:domainStatusType" />
</choice>
</element>
</sequence>
</extension>
</complexType>
<element
    name="renew"
    type="dchk:domainStatusType" />
<element
    name="addPeriod"
    type="dchk:domainStatusType" />
<element
    name="renewPeriod"
    type="dchk:domainStatusType" />
<element
    name="autoRenewPeriod"
    type="dchk:domainStatusType" />
<element
    name="transferPeriod"
    type="dchk:domainStatusType" />
<element
    name="redemptionPeriod"
    type="dchk:domainStatusType" />
<element
    name="restore"
    type="dchk:domainStatusType" />
<element
    name="policyCompliant"
    type="dchk:domainStatusType" />
<element
    name="policyNoncompliant"
    type="dchk:domainStatusType" />
<element
    name="reserved"
    type="dchk:domainStatusType" />
<element
    name="create"
    type="dchk:domainStatusType" />
<element
    name="delete"
    type="dchk:domainStatusType" />
<element
    name="transfer"
    type="dchk:domainStatusType" />
<element
    name="update"
    type="dchk:domainStatusType" />
<element
    name="other"
    type="dchk:domainStatusType" />
</choice>
</complexType>
</element>
<element
    name="registrationReference"
    type="iris:entityType"
    minOccurs="0"
    maxOccurs="1" />
<element
    name="createdDateTime"
    type="dateTime"
    minOccurs="0"
    maxOccurs="1" />
<element
    name="initialDelegationDateTime"
    type="dateTime"
    minOccurs="0"
    maxOccurs="1" />
<element
    name="expirationDateTime"
    type="dateTime"
    minOccurs="0"
    maxOccurs="1" />
<element
    name="lastDatabaseUpdateDateTime"
    type="dateTime"
    minOccurs="0"
    maxOccurs="1" />
<element
    ref="iris:seeAlso"
    minOccurs="0"
    maxOccurs="unbounded" />
</sequence>
</extension>
</complexContent>
</complexType>

<element
    name="domain"
    type="dchk:domainType"
    substitutionGroup="iris:result" />

<complexType
    name="domainStatusType">
    <sequence>
        <element
            name="appliedDate"
            type="dateTime"
            minOccurs="0"
            maxOccurs="1" />
        <element
            name="ticket"
<element
    name="description"
    minOccurs="0"
    maxOccurs="unbounded">
    <complexType>
        <simpleContent>
            <extension
                base="string">
                <attribute
                    name="language"
                    type="language"
                    use="required" />
            </extension>
        </simpleContent>
    </complexType>
</element>

<element
    name="subStatus"
    minOccurs="0"
    maxOccurs="1">
    <complexType>
        <simpleContent>
            <extension
                base="token">
                <attribute
                    type="token"
                    use="required"
                    name="authority" />
            </extension>
        </simpleContent>
    </complexType>
</element>
</sequence>

<attribute
    name="actor">
    <simpleType>
        <restriction
            base="string">
            <enumeration
                value="registry" />
            <enumeration
                value="registrar" />
            <enumeration
                value="registrationServiceProvider" />
        </restriction>
    </simpleType>
</attribute>
3.3. BEEP Transport Compliance

All DCHK client and servers MUST implement the Lightweight UDP Transport Protocol (IRIS-LWZ) [9]. The use of other transports like the XML Pipelining with Chunks (IRIS-XPC) transport [12] or the BEEP transport [8] is optional and completely at the discretion of the server operator. The XPC transport is in any case preferable to the BEEP transport.

IRIS allows several extensions of the core capabilities. This section outlines those extensions allowable by IRIS-BEEP [8].

3.3.1. Message Pattern

This registry type uses the default message pattern as described in IRIS-BEEP [8].

3.3.2. Server Authentication

This registry type uses the default server authentication method as described in IRIS-BEEP [8].

3.4. URI Resolution
3.4.1. Application Service Label

The application service label associated with this registry type MUST be "DCHK1". This is the abbreviated form of the URN for this registry type, urn:ietf:params:xml:ns:dchk1.

3.4.2. Bottom-Up Resolution

The bottom-up alternative resolution method MUST be identified as ‘bottom’ in IRIS URI’s. Its process is identical to the ‘bottom’ process described by RFC3982 [7].

3.4.3. Top-Down Resolution

The top-down alternative resolution method MUST be identified as ‘top’ in IRIS URI’s. Its process is identical to the ‘top’ process described by RFC3982 [7].
4. Internationalization Considerations

Implementers should be aware of considerations for internationalization in IRIS [6].

Clients needing to localize the data tags in this protocol should take note that localization is only needed on the names of XML elements and attributes with the exception of elements containing date and time information. The schema for this registry has been designed so that clients need not interpret the content of elements or attributes for localization, other than those elements containing date and time information.

Clients should also make use of the <language> elements provided in many of the results. Results containing internationalized data can be accompanied by these elements in order to aid better localization of the data by the user.

All date and time elements make use of the XML Schema [10] data type "dateTime". If their contents are Coordinated Universal Time (UTC) timestamps, they MUST be specified by using the capitalized 'Z' indicator (instead of 'z').
5. IANA Considerations

5.1. XML Namespace Registration

This document makes use of the XML registry specified in RFC 3688 [4]. Accordingly, the following registration information is provided for the IANA:

- XML Namespace URN/URI:
  - urn:ietf:params:xml:ns:dchk1

- Contact:
  - Andrew Newton <andy@hxr.us>
  - Marcos Sanz <sanz@denic.de>

- XML:
  - None.

5.2. XML Schema Registration

This document makes use of the XML registry specified in RFC 3688 [4]. Accordingly, the following registration information is provided for the IANA:

- XML Schema URN/URI:
  - urn:ietf:params:xml:ns:dchk1

- Contact:
  - Andrew Newton <andy@hxr.us>
  - Marcos Sanz <sanz@denic.de>

- XML:
  - The XML Schema specified in Section 3.2

5.3. S-NAPTR Registration

The following S-NAPTR application service label will need to be registered with IANA according to the IANA considerations defined in IRIS [6]:

Newton & Sanz                 Expires May 17, 2008
5.4.  BEEP Registration

The following BEEP Profile URI is to be registered with IANA, in addition to the registration provided in IRIS-BEEP [8].

http://iana.org/beep/iris1/dchk1
6. Security Considerations

Being a proper subset of RFC3982 [7], the registry described in this document introduces no security considerations beyond those documented in RFC3981 [6].
7. References

7.1. Normative References


7.2. Informative References

Authors’ Addresses

Andrew L. Newton
VeriSign, Inc.
21345 Ridgetop Circle
Sterling, VA  20166
USA
Phone: +1 703 948 3382
Email: andy@hxr.us
URI:  http://www.verisignlabs.com/

Marcos Sanz
DENIC eG
Wiesenhuettenplatz 26
D-60329 Frankfurt
Germany
Email: sanz@denic.de
URI:  http://www.denic.de/