

ESIP Atom-based OpenSearch and Collection Casting Conventions

Christopher Lynnes, Hook Hua, James Gallagher
ESIP Discovery Cluster

1 Status of this Memo

This memo provides information to the NASA Earth Science Data Systems (ESDS) community. This memo specifies an ESDS standard for Earth Science metadata. Distribution of this memo is unlimited.

2 Change Explanation

N/A

3 Abstract

The Discovery Cluster within the Earth Science Information Partners (ESIP) Federation has developed a set of conventions around the Atom syndication specification and OpenSearch query/response specification to enable lightweight mechanisms of data advertising, discovery and access. This framework supports both a query/response pattern as well as a publish/subscribe pattern called collection casting. The OpenSearch specifications from www.opensearch.org have been adopted along with the draft specifications for Time and Geo(spatial) extensions, the Atom specification, and Dublin Core's specification for date/time range in the response and syndication cases. The result is a data advertising and discovery framework with low buy-in for search providers and low learning curves for client developers.

4 Copyright Notice

The majority of the document was authored by a U.S. Government employee and is therefore noncopyright. Sections that incorporated content from non-federal employees are provided under copyright by the Federation of Earth Science Information Partners, with all rights reserved.

5 Table of Contents

TBD

6 Introduction

Earth science data abound in cyberspace, yet many of them are so-called “dark data”, i.e., difficult or impossible to find in a meaningful way. Even data that are offered by well-funded science data archives may be accessible only through proprietary search interfaces. This complicates the life of a scientist trying to collect datasets from a variety of sources. Often, the job devolves to:

- (a) Discover that Dataset A is available at website X
- (b) Learn the user interface for website X
- (c) Conduct the search at website X
- (d) Order and then acquire data for website X through interface
- (e) Repeat (a)-(d) for Dataset B at website Y

To be sure, several protocols have developed over time to support “one-stop-shopping”. For example, the Z39.50 became popular in the 1990’s for searching documents and still has some currency in the digital library community, though this is lacking in some of the spatio-temporal concepts needed for effective Earth science queries. Also in the 1990’s was the EOSDIS Version 0 protocol, a standard developed at NASA to support dataset- and granule-level searching over sockets, and later over HyperText Transfer Protocol (HTTP). More recently, the Catalog Services for the Web (CSW) protocol has been developed within the Open Geospatial Consortium (OGC), with many adherents particularly in the area of structured searches. However, both the EOSDIS V0 and CSW protocols can be complicated to implement (and even understand) for a science researcher with little or no information technology staff.

The Earth Science Information Partners is a federation of data providers (and consumers) in the area of Earth science. The ESIP federation includes large, federal partners, such as NASA, NOAA, USGS and some of their subsidiary data centers, as well as academic research groups, state/local government, educational organizations, non-profit consortia and private companies. While all share an interest in Earth science data, the diversity in both size and perspective within ESIP defines the challenge of finding Earth science data in an increasingly heterogeneous world of providers. Thus, ESIP members and their colleagues in the community could benefit from a simplified federated search mechanism to support basic data searches across a diverse community. It should be simple enough for a computer-literate scientist to write his or her own scripts as search clients. It should also be simple enough for small organizations or scientists with minimal or even no informatics support to make their data available and discoverable to the wider community. With this set of general goals in mind, the ESIP Discovery group was constituted to work on the problem of highly distributed, diverse data advertising and discovery.

7 ESIP Data Discovery

7.1 Introduction to ESIP OpenSearch Query and Collection Casting

7.1.1 Introduction to OpenSearch

In searching for a discovery framework that could serve a highly distributed set of data and be implemented by the widest diversity of providers, ESIP members began exploring the OpenSearch convention for distributed search. OpenSearch was originally developed by the A9 search technology company, a subsidiary of Amazon. The specification for OpenSearch query and response was released to the community in 2005, and adopted in a variety of settings. The OpenSearch convention is currently maintained on www.opensearch.org, a site provided by A9 to the community.

At its core, OpenSearch is simply a description of simple document formats:

- an OpenSearch Description Document, which describes a particular search provider's capabilities and search syntax, and
- common elements of query response documents

In most cases, OpenSearch queries are simply specified within the URL, though there is a proposed extension for "POST" submissions as well. This simplicity lends itself to widespread adoption, not just among the search service providers, but also among client developers. (The importance of the latter group cannot be overstated in the success of an interoperability-related technology.)

OpenSearch supports several different formats for returning results, including the Atom syndication mechanism (Nottingham and Sayre, 2005), which was chosen for machine readability.

7.1.2 Introduction to Collection Casting

Collection casting is a publish-subscribe method, which can be used to advertise the availability of data sets. Once advertised the casts are discoverable, can be aggregated using standard web crawling technologies and the resulting aggregation queried using query and response specifications described in the next section. This builds off of a similar pre-existing publish-subscribe method called Datacasting, whereby a data provider publishes the availability of new data granules (data files) via an RSS feed and a data consumer subscribes to the feed to learn about recently available data (Bingham et al., 2009). For more information on the DatacastingRSS framework, see the website <http://datacasting.jpl.nasa.gov> and the NASA RFC for DatacastingRSS, under consideration as of 28 Jan 2013, in NASA's Standards and Processes group at <https://earthdata.nasa.gov/wiki/main/images/a/a6/Datacasting-Standard-RFC-20121108.pdf>.

The Atom specification for Collection casting enables unification of the response format for

Collection casting and the OpenSearch query response in this specification. Thus, a data provider can provide access to its data simply by publishing an OpenSearch Response as an Atom feed file, that is, either by putting it out on the Web to be crawled or registering it in a data registry. In the Collection casting case, there is no user query to constrain the results in the OpenSearch Response file; constraints (if any) are instead left up to the data provider. Rather, the provider can cast all the data, only updating the feed when new records are added or old records updated or removed (corresponding to an update or retirement of the data set)

7.2 Query and Response Specifications

The ESIP Discovery conventions for query are based on the OpenSearch Description Document, which describes how queries are made to a given search service. Response and casting conventions follow the Atom specification, with customizations to adapt the representation of data items in Atom. Some of these customizations are designed to help machine-level clients parse the search results.

7.2.1 OpenSearch Description Documents (Required)

OpenSearch Description Documents (OSDD) are XML files that are the key to specifying to clients how queries are to be done with a given search provider. The OpenSearch specification of OSDDs is available in detail on the OpenSearch site at

http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch_description_document

. The query specifications are contained within XML elements of type `<Url>`, which includes a template for the URL query and a mime-type for the expected response, e.g.:

```
<Url type="application/atom+xml"
```

```
template="http://example.com/?q={searchTerms}&pw={startPage?}&format=atom"/>
```

The template gives the URL to execute the query, with placeholders indicated by curly braces ‘{’ and ‘}’.

7.2.2 Search Terms (Optional)

The searchTerms parameter is expected to be a simple set of keywords to be used in a freetext search. This contrasts somewhat with the traditional method of searching (e.g., within EOSDIS Version 0), which emphasized structured searches based on specific attributes like satellite and instrument. However, freetext search is simpler to implement at the client end and more universal, not relying on a common schema, and with judicious use of acronyms by the client or user (e.g., “MODIS”) can be nearly as precise. Note that searchTerms are an important discriminator for data collections; however, they are often not useful for files within a data collection, where the discriminator is more likely to be space or time coordinates (see Sections 7.2.3, 7.2.4) of the files. Thus searchTerms is not a required parameter; rather it is the server that specifies in a template whether they are required for that particular search type.

When multiple searchTerms are included, they should be separated by ‘+’, e.g., “MODIS+fires”.

Servers are **required** to treat this combination as a boolean “AND” operation by default.

Multi-word phrases can be specified by setting double quotes (") around the phrase. In this case, a server must treat this as a phrase search to be compliant with the ESIP specification.

Boolean keywords “OR” and “NOT” are not yet codified in the ESIP OpenSearch specification, but are likely candidates for the future. Some servers may support this, but there is as yet no specific way for them to advertise that fact.

7.2.3 Time Extension (Optional)

A draft of a time extension has been proposed for the OpenSearch specification at http://www.opensearch.org/Specifications/OpenSearch/Extensions/Time/1.0/Draft_1.

Despite its draft nature, the ESIP Discovery Cluster has accepted it as an ESIP federated search convention. If the Time extension is used, then:

(a) the namespace is defined as

xmlns:time=<http://a9.com/~opensearch/extensions/time/1.0/>, i.e., a namespace created for this extension. In theory, any abbreviation for the namespace aside from “time” can be used. In practice the ESIP Discovery cluster recommends the use of “time”.

(b) **start** and **end** in the above namespace are defined as the beginning and end (respectively) of the search period. Both are defined as conforming to the [IETF RFC 3339](http://tools.ietf.org/html/rfc3339), “Date and Time on the Internet: Timestamps”, which is itself a profile of the ISO 8601 standard.

The draft extension does not address date and time in the returned Atom response document. Based on a Discovery Change Proposal, the date/time range of a dataset or data file is given using the Dublin Core date element, <http://purl.org/dc/elements/1.1/date>. The dates are given in ISO 8601 format. Although this specification admits a diversity of formats, the ESIP convention requires one of the following to simplify date parsing on behalf of clients. The format can be either:

- date only: YYYY-MM-DD
- date-time: YYYY-MM-DDTHH:MI:SS.SSSZ

where YYYY is a four-digit year,

MM is a two digit month, with leading zero if month < 10,

DD is a two digit day of the month, with leading zero if day < 10.

HH is hour in the day,

MI is minutes from the start of the hour

SS is seconds from the start of the minute. It may contain fractional seconds after a decimal point.

The letter ‘T’ is used to separate date and time when time is specified. The letter ‘Z’ indicates Zulu or Greenwich Mean Time.

Some example dates are:

- 2009-01-09
- 2009-01-09T00:00:00Z
- 2009-01-09T00:00:00.000Z

If a date *range* is specified, then the two dates are separated by a slash, e.g.:

<dc:date>2009-01-01T05:23:24Z/2009-01-01T05:29:24Z</dc:date>

7.2.4 Spatial Extension (Optional)

The ESIP Discovery conventions have adopted the proposed [Draft 2 of the GEO extension](#) in OpenSearch, which is in turn based on the GeoRSS Simple specification specifications, <http://www.georss.org/simple> GeoRSS-Simple.

By convention, ESIP Discovery **requires** the OpenSearch GEO extension namespace:

xmlns geo="<http://a9.com/-/opensearch/extensions/geo/1.0/>".

in the OpenSearch Description Document to define the query format. The draft specification includes the following query parameters:

- "name", e.g.,: `http://example.com/?q={searchTerms}&loc={geo:name?}&format=atom`
- "lat" and "lon", with optional "radius"
- "box", in the form west, south, east, north, e.g., template and query are:
`http://example.com/?q={searchTerms}&bbox={geo:box?}&format=atom`
`http://example.com/?q=pizza&bbox=-111.0,42.9,-119.8,43&format=atom`

In addition, the OpenSearch GEO extension draft includes several Well Known Text (WKT) geometries, including polygon. However, ESIP search services may support only a subset of these specifications. The OpenSearch Description Document will inform the clients, by way of including a {geo:geometry?} placeholder in the Url templates, which ones are supported by that search provider. An example, modified from the OpenSearch Draft Geo Extension (Clinton et al., 2007) is:

```
http://example.com/?q={searchTerms}&pw={startPage?}&g={geo:geometry?}&format=atom
```

However, the WKT is optional and sparsely implemented within ESIP; as a result, some details (such as identification of the spatial reference system) have not been fully elaborated in the ESIP Discovery cluster activities.

In the Atom response, the "georss" namespace and abbreviation are used for defining box, circle, and polygon regions for each entry. Although support of the Spatial Extension overall is optional, when spatial information is available for a response object, the <georss:box> element must always be provided. Additional spatial elements may be provided along with the <georss:box> element. When this occurs, it is to be assumed that the coordinates of the <georss:box> element represent a [Minimum Bounding Rectangle](#) encompassing the object's spatial coverage. If an object's spatial area requires a more complicated representation than that which is available through the usage of the GeoRSS-Simple specification, the GeoRSS GML (<http://www.georss.org/gml>) specification may be supplemented.

Thus, a client will always receive a Minimum Bounding Rectangle <georss:box> element, but may look to see if there are additional GML elements to obtain a more precise measure of the location or areal bounds.

box (required support)

A bounding box with coordinates (west, south, east, north).

Example: <georss:box>-180.0 -90.0 180.0 90.0</georss:box>

circle (optional)

A shape containing three coordinates (centerpoint latitude, centerpoint longitude, circle radius), where the lat/lon are in WGS84 format, and radius is in meter units.

Example of radius around a lat/lon point:

```
<georss:circle>34.0 -118.0 10000</georss:circle>
```

polygon (optional)

A list of lat-lon pairs in WGS84 format. space-delimited.

Example of radius around a lat/lon point:

```
<georss:polygon>45.3 -110.4 46.5 -109.5 43.8 -109.9 45.3  
-110.4</georss:polygon>
```

7.2.5 Recursive OpenSearch Queries (Optional)

One serious hurdle to overcome in searching for data is the enormous number of data items to account for in responses, as well as the expected number of successful “hits” for a query. In ordinary web searches, the searcher is usually looking for a small number of web pages or documents. Relevance ranking typically does a good job of presenting these successful hits near the top of the returned list, followed by single point-and-click retrievals. However, when searching for Earth science data covering large time periods or spatial areas, a user will often specify a set of constraints to find an appropriate data collection together with space-time criteria for files within that data collection. Often, the precision of the data collections returned for the search is low, with many spurious hits. However, the space-time precision of the files is often quite high: that is, the user truly wants to use all the data files of a desirable data collection set that fall within the space-time region of interest. Thus, searching for all data satisfying both dataset content and space-time region at the same time can produce a great many spurious hits, i.e., all the files for data collections that are *not* desired.

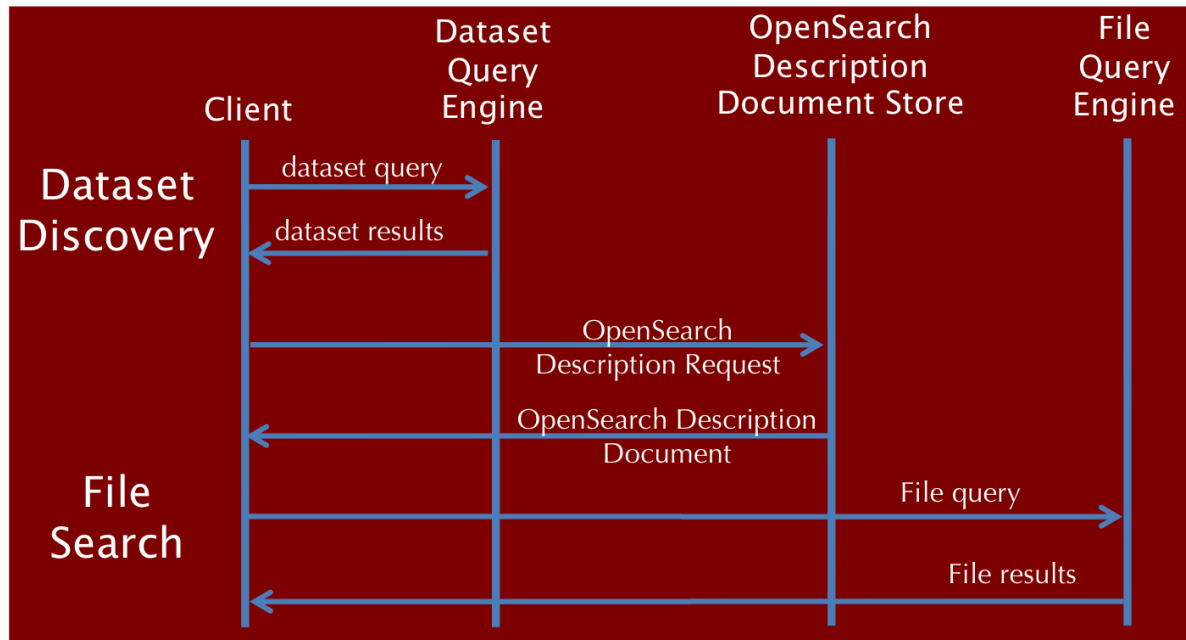


Fig.1. Event trace of a recursive ESIP search.

To get around this precision problem, the ESIP Discovery cluster defines a recursive search process (Fig. 1). In the simplest version of this, the data collection search is performed first. The results list, rather than being specific data items, comprises data collections; each data collection includes a link to the OpenSearch Description document that describes how to search for files within that data collection. OpenSearch Description Documents can be recognized in the Atom response by the type attribute: **type="application/opensearchdescription+xml"**.

Typically, the data collection being searched is constrained in the search template URL through an extra parameter, e.g., "collection_id=AIRX2RET". However, each search provider typically uses different terminology for identifying the data collection in a file-level search; thus the template URL has the effect of providing to the client the exact syntax for identify the data collection in the subsequent file-level query to the search provider. Note that the Dataset Query Engine, File Query Engine and OpenSearch Description Document may reside on different hosts. Thus this scheme not only supports a highly distributed system of data and search resources, but also admits for the possibility of cooperative or third-party search services.

Although two steps, collection-level and granule-level, are illustrated here, note that the recursive hierarchy could be more than two levels deep. That is, one could have an OpenSearch server that searches top-level (search engine) documents, returning the URLs to OpenSearch Description Documents for each search engine, identified by the mime-type `x-application/opensearchdescription+xml` (Fig. 2).

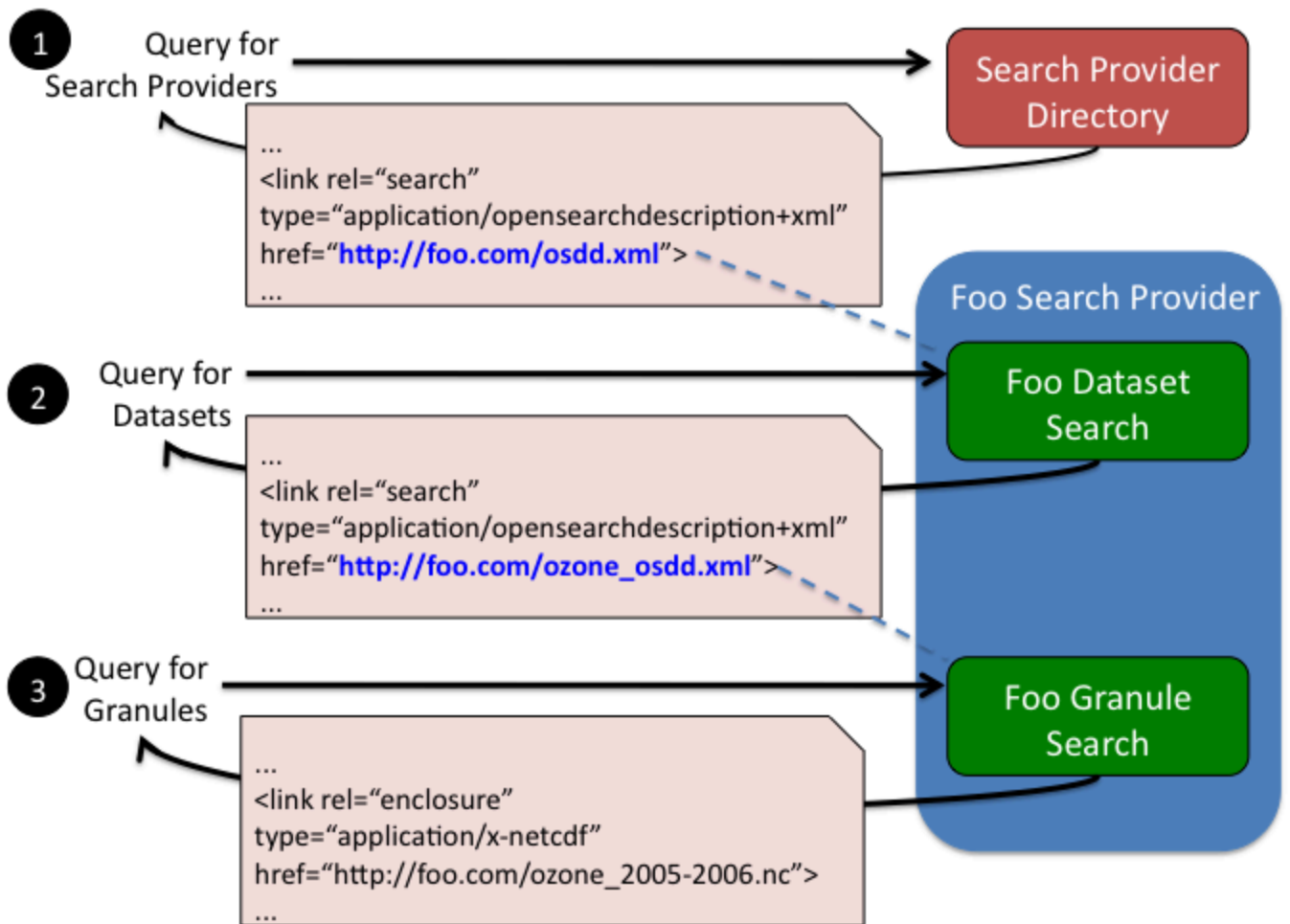


Fig. 2. Though the focus so far has been on two level recursion (Dataset search to Granule Search), this need not be the case. This figure shows a three-level recursion, beginning with a search for search providers (or engines) satisfying user criteria (e.g., searchTerms="ozone"). Each step in the recursion returns links to OpenSearch Description documents, which in turn inform the client on how to execute the next level search, until the lowest level, where the return consists of links to actual data of interest.

The Recursive Query is an **optional** aspect of the convention. There are use cases where only a dataset-level query is needed, as well as some where only the granule-level query is needed. Also, in most cases, the Recursive Query goes from a broader category (e.g., datasets) to a narrower one (files within a dataset).

7.2.6 Characterizing Atom Link Elements

The ESIIP Discovery conventions were designed to enable the easy incorporation of search into client tools, which may include complex clients conducting further operations on the result, such

as science processing. Many of the entries returned may contain multiple `<link>` elements to such resources as data, browse, metadata, etc. As a result, the `<link>` elements need to be unambiguously described. For most types of items, the type of link is recognized by a combination of the “rel” attribute’s value and the mime-type, represented in a “type” attribute. Both the `rel` and `type` attributes are **required** to specify the character of the link.

Type of link	Definition	rel value	mime-type (type value)
data	link representing a data file or other science data resource; may be large in size	enclosure	application/x-netcdf, application/x-hdf, text/csv
browse	image of the data typically used for making data request decisions	icon	image/jpeg image/png image/pdf image/gif
metadata	file with (usually) structured information about corresponding data files	describedBy	text/xml
OSDD	link to an OpenSearch Description Document; useful for recursive searching	search	application/ opensearchdescription+xml

7.2.7 Characterizing OPeNDAP Link Elements (Optional)

(This section falls under ESIP Federation Copyright, 2012).

In addition to these basic link types, some data providers may offer a variety of data access points to the same data. An example of this is data providers that offer data via the OPeNDAP standard, which itself has several different return types: specific forms of metadata or data in varying formats. In order to distinguish OPeNDAP data access points from simple FTP or HTTP URLs, as well as from each other, an xlink-based scheme has been adapted to identify the type of OPeNDAP access point. An OPeNDAP service link is identified by a `<link>` attribute of the type “xlink:role”. In this case, specifically, the xlink role and type attributes are given as:

`xlink:role="http://xml.opendap.org/dap/dap2.xsd" xlink:type="simple"/`

In addition, the combination of the `rel` attribute, mime-type (i.e., `type`) attribute and `xlink:arcrole` attribute are used to distinguish the different types of OPeNDAP response:

Server Response Type	<i>rel</i>	<i>type</i>	<i>xlink:arcrole</i>
HTML	describedby	text/html	#html
info	describedby	text/html	#info

DDX	describedby	application/xml	#ddx
RDF	describedby	application/rdf+xml	#rdf
ASCII	enclosure	text/plain	#ascii
NetCDF	enclosure	application/x-netcdf	#netcdf

Note that a search engine is not required to return all of these responses when identifying OPeNDAP links, just the ones that it (a) supports and (b) wishes to advertise to clients.

It is understood that inclusion of these xlink elements can cause a Discovery response to fail validation by some Atom feed validators, but this has been traded off for the extra information this construct provides to client programs about the options for obtaining data and metadata from the provider.

7.2.8 Versioning (Required)

In many cases, a client must know what version of the ESIP conventions a discovery server or cast is conforming to. (The ESIP convention version is distinct from dataset or file versions, which vary from dataset to dataset and are not treated in this specification.) Features may be added (e.g., new elements in the response) or in some cases even rolled back in subsequent versions (e.g. DCP-2). Thus, clients need an *easy* way to identify which version a given document conforms to, be it:

- Atom response from an OpenSearch query
- Data cast
- OpenSearch Description Document

The ESIP Discovery Version with which a provider complies would be identified by declaring an ESIP namespace within the root element of the document, i.e.:

a) the <feed> element for Atom responses

b) the <OpenSearchDescription> element for OpenSearch Description documents

The convention for the namespace prefix is "esipdiscovery". The namespace URI to which it points is <http://commons.esipfed.org/ns/discovery/<version>/>. The version must also be included as esipdiscovery:version="<version>" to make it visible to programs using off-the-shelf XML parsers. Thus, the opening element for an ESIP-compliant atom feed may look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:esipdiscovery="http://commons.esipfed.org/ns/discovery/1.2/"
  esipdiscovery:version="1.2">
```

Likewise, the beginning of an OpenSearch Description Document would look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:esipdiscovery="http://commons.esipfed.org/ns/discovery/1.2/"
  esipdiscovery:version="1.2">
```

Note that the rather long "esipdiscovery" namespace is used in preference to the simpler "esip" to allow for other ESIP efforts (e.g. semantic web) to carve out their own namespaces within ESIP.

The version that this RFC pertains to is Version 1.2 of the ESIP Discovery specification.

7.2.9 Optional Nature of Search-specific Url Template Fields (Required)

Placeholders for optional search parameters in URL templates must include a question mark character '?' at the end of the search term, e.g., `http://example.com/search/{instrument?}`.

However, the search-specific attributes themselves are not considered themselves to be part of the specification. Also, such attributes **MUST** be optional; required search-specific terms that are not spelled out in the ESIP OpenSearch specification render the associated search service non-compliant with that convention.

7.3 Error Handling (Required)

(This section falls under ESIP Federation Copyright, 2012).

In general, error handling conventions for ESIP Discovery follow the standards laid out in the W3C for HTTP/1.1 status codes. However, a key convention is the handling of zero results, which must not be treated as an error. Instead, the HTTP status code shall be success, and the response document shall contain a valid response with no item entries. The `<opensearch:totalResults>` tag must always indicate the number of results, where for this case be equal to 0. Also, the `<subtitle>` tag should provide a user-friendly message indicating that no results were found.

7.4 Implementations

Several implementations that loosely conform to some version of the ESIP Discovery conventions have been developed. (It is expected that this RFC will provide a mechanism to tighten conformance for both servers and clients.) OpenSearch servers have been developed by several search providers in NASA, including

- Global Hydrology Resource Center
- Goddard Earth Sciences Data and Information Services Center (GES DISC)
- Moderate Resolution Imaging Spectro-radiometer Adaptive Processing System (MODAPS).
- EOS Clearinghouse
- Web Services - NEWS

Clients that have been developed range from small scripts to automate data search to full-scale interactive systems such as Libre at the National Snow & Ice Data Center (NSIDC), the EOSDIS Simple Subset Wizard and the GES DISC Giovanni systems.

There are as yet no reference implementations or compliance test suites for the ESIP Discovery

conventions, but plans to implement same will follow approval of this RFC. The ESIP Discovery Cluster also maintains a mailing list that provides a forum for discussions of the conventions and related implementations. Implementors outside the ESIP cluster can submit questions or comments to the list as well (albeit moderated), at esip-discovery@lists.esipfed.org.

8 References

Bingham A.W, S. McCleese, T. Stough, R. G. Deen, K. Hussey and N. Toole, 2009, Earth Science Datacasting: Informed Pull and Information Integration, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 47, NO. 10, OCTOBER 2009.

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5191107>.

Clinton, D., Turner, A., Walsh, J., Fonts, O., and P. Goncalves, 2007. OpenSearch Geo Extension, Draft 2.,

http://www.opensearch.org/Specifications/OpenSearch/Extensions/Geo/1.0/Draft_2.

Gallagher, J., 2012, Discovery Change Proposal - 4, OPeNDAP Links in the Atom <link/> element, ratified, <http://wiki.esipfed.org/index.php/DCP-4>.

Hua, H., 2012. Discovery Change Proposal - 7, Error Handling Best Practices for Discovery Response, under review, http://wiki.esipfed.org/index.php/Discovery_Change_Proposal-7.

Klyne G, Newman C. 2001. Date and Time on the Internet: Timestamps. Internet draft, The Internet Society 2001, updated 2002 as Request for Comments 3339.

www.ietf.org/rfc/rfc3339.txt.

Lynnes, C., 2012. Discovery Change Proposal - 9, Identifying Version of ESIP Discovery Conformance, currently polling, http://wiki.esipfed.org/index.php/Discovery_Change_Proposal-9.

Nottingham, M. and R. Seyre (eds.), 2005. The Atom Syndication Format, IETF RFC 4287, <http://www.ietf.org/rfc/rfc4287.txt>.

9 Authors' Addresses

Christopher Lynnes, NASA/GSFC, christopher.s.lynnes@nasa.gov.

Hook Hua, NASA/JPL, hook.hua-1@nasa.gov.

James Gallagher, OPeNDAP.org, jgallagher@opendap.org.

10 Appendix A - Summary Table

Element or Attribute	Description	Namespace	Example
version attribute	Version of the ESIP Discovery Atom-based Specification supported by a search or collection casting provider. Required in OSDD and Atom response.	esipdiscovery	<pre><?xml version="1.0" encoding="UTF-8"?> <OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/" xmlns:esipdiscovery="http://commons.esipfed.org/ns/discovery/1.2/" esipdiscovery:version="1.2"></pre>
title element	Human-readable text describing about an entry, usually relatively short	Atom	
id element	Unique identifier for a returned resource in Atom response	Atom	
link element	URL for a resource referenced in Atom response	Atom	
rel attribute	Relationship represented by a given <link> element.	Atom	rel="enclosure" (used to describe <link> elements pointing to data resources)
type attribute	Type of resource, typically given as a mime-type	Atom	type="application/x-netcdf" (describes link elements that are netCDF files).
Url	External resource location	OpenSearch	<pre><Url type="text/html" template="http://example.com/search? q={searchTerms}&pw={startPage?}" /></pre>
template attribute	template attribute of <Url> element Required in OSDD	OpenSearch	
date	Temporal coverage of data in an Atom response.	Dublin Core	
geo:box	Minimum bounding rectangle, required in	OpenSearch Geo	

	Atom response if spatial information is included		
georss:geometry	Optional Well-Known-Text based geometry	GeoRSS	

11 APPENDIX B - Examples

11.1 OpenSearch Description Document (Top Level)

This is used by clients to form a search for data collections:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/"

xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/"

xmlns:time="http://a9.com/-/opensearch/extensions/time/1.0/"

xmlns:esipdiscovery="http://commons.esipfed.org/ns/discovery/1.2/"
    esipdiscovery:version="1.2">
    <ShortName>Example Dataset Search</ShortName>
    <Description>Use this example Dataset Search to obtain a list of Earth
Science Data Sets</Description>
    <Tags>Example Dataset Search</Tags>
    <Contact>help@example.com</Contact>
    <Url type="application/atom+xml"

template="http://example.com/cgi-bin/collectionlist.pl?keyword={searchTerms}&
amp;page=1&count={count}&osLocation={geo:box?}&osLocationPlaceName={geo:name?}&
startTime={time:start?}&endTime={time:end?}&format=atom"/>
    <Query role="example" searchTerms="ozone"
        title="Sample Bounding Box Search"
        time:start="2010-01-01"
        time:end="2010-01-10"
        geo:box="-130.0,25.0,-65.0,50.0"/>
<!--href="http://example.com/cgi-bin/collectionlist.pl?keyword=ozone&page=
1&count=10&osLocation=-130.0,25.0,-65.0,50.0&startTime=2010-01-0
1&endTime=2010-01-10&format=atom"/-->
    <Query role="example" searchTerms="Surface Air Temperature"
        title="Sample PlaceName Search"
        time:start="2010-01-01" time:end="2010-01-10"
        geo:name="New York"/>
```

```

<!--href="http://example.com/cgi-bin/collectionlist.pl?keyword=Surface%20Air%
20Temperature&page=1&count=100&osLocationPlaceName=greenbelt&
startTime=2010-01-01&endTime=2010-01-10&format=atom"/-->
</OpenSearchDescription>

```

11.2 Dataset Response

Following is an example of a dataset-level OpenSearch response (validated in the Atom validator at <http://www.validome.org/rss-atom/validate>).

```

<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:georss="http://www.georss.org/georss"
  xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:esipdiscovery="http://commons.esipfed.org/ns/discovery/1.2/">
  <title>Collection results for Example Data Collection</title>
  <subtitle type="html">Example Data Collection</subtitle>
  <updated>2013-02-06T23:12:01Z</updated>
  <author>
    <name>Example Data Provider</name>
    <email>help@example.com</email>
  </author>
  <id>http://example.com/cgi-bin/collectionlist.pl</id>
  <opensearch:totalResults>1</opensearch:totalResults>
  <opensearch:itemsPerPage>1</opensearch:itemsPerPage>
  <link rel="self" href="http://example.com/cgi-bin/collectionlist.pl"/>
  <entry>
    <title>Level 2 Standard physical retrieval</title>
    <id>http://dx.doi.org/10.1000/182</id>
    <updated>2013-02-02T23:00:01Z</updated>
    <author>
      <name>Example Search Provider</name>
      <email>help@example.com</email>
    </author>
    <dc:date>2002-08-30/2013-02-05</dc:date>
    <summary type="text">DataSet: ExampleDataCollection.005</summary>
    <content type="text">Example Data Collection, 2002-08-30 to
2013-02-05</content>
    <link rel="search" type="application/opensearchdescription+xml"
title="Example Data Collection Version 5"
href="http://example.com/granule_search_ExampleDataCollection_005.xml"/>
    <link rel="describedBy" type="text/html"
title="ExampleDataCollection_005.xml info"
href="http://example.com/collections/ExampleDataCollection_005.shtml"/>
  </entry>
</feed>

```


Upon receiving data collection results, the client can retrieve the document indicated in the `<link rel="search" type="application/opensearchdescription+xml...">` element in the response above. The resulting OSDD will contain the template for searching for files of a given dataset. Note that this particular template requires a parameter called `dataSet`, which has been pre-filled to search within the `ExampleDataCollection` data collection.

```
<?xml version="1.0" encoding="UTF-8"?>
<os:OpenSearchDescription xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
    xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/"
    xmlns:time="http://a9.com/-/opensearch/extensions/time/1.0/"
    xmlns:esipdiscovery="http://commons.esipfed.org/ns/discovery/1.2/"
    esipdiscovery:version="1.2">
    <os:ShortName>ExampleDataCollection.005</os:ShortName>
    <os:Description>Obtain a list of URLs for data collection
ExampleDataCollection.005</os:Description>
    <os:Tags>ExampleDataCollection.005</os:Tags>
    <os:Contact>help@example.com</os:Contact>
    <os:Url type="application/atom+xml"
template="http://example.com/cgi-bin/filelist.pl?dataSet=ExampleDataCollection.
005&page=1&maxgranules={os:count?}&osLocation={geo:box?}&order=
a&endTime={time:end}&startTime={time:start}&format=atom"/>
    <os:Query role="example" title="Sample Search"
        time:start="1980-01-01" time:end="2010-01-10"
        os:count="100"
        geo:box="-130.0,25.0,-65.0,50.0"/>
<!--href="http://example.com/cgi-bin/filelist.pl?page=1&dataSet=ExampleData
Collection.005&order=a&maxgranules=100&startTime=1980-01-01&end
Time=2010-01-10 23:59:59&format=atom"/--></os:OpenSearchDescription>
```

10.4 File-level Search Response

Upon substituting user search parameters into the Url template above, the following is a sample response.

```
<?xml version="1.0"?>
<feed xmlns="http://www.w3.org/2005/Atom"
    xmlns:esipdiscovery="http://commons.esipfed.org/ns/discovery/1.2/"
    xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
    xmlns:georss="http://www.georss.org/georss"
    xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns:time="http://a9.com/-/opensearch/extensions/time/1.0/"
```

```

xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xlink="http://www.w3.org/1999/xlink"
esipdiscovery:version="1.2">
<title>ExampleDataCollection Files</title>
<subtitle type="html">
    Total Results: 558
    Max Items Per Page:1</subtitle>
<updated>2009-07-16T18:30:02Z</updated>
<author>
    <name>Example Search Provider</name>
    <email>help@example.com</email>
</author>
<id>http://example.com/cgi-bin/granlist.pl</id>
<opensearch:totalResults>558</opensearch:totalResults>
<opensearch:itemsPerPage>1</opensearch:itemsPerPage>
<entry>
    <title>ExampleDataCollection.2009.01.01.052324Z.005.hdf</title>
    <updated>2009-01-01T05:23:24Z</updated>

<id>http://example.com/ExampleDataCollection.005/2009/001/Example.2009.01.01.
054.hdf</id>
    <link rel="enclosure" length="2510007"
href="http://example.com/ExampleDataCollection.005/2009/001/Example.2009.01.0
1.054.hdf"/>
    <link rel="enclosure" length="2510007" type="application/x-opensdap"
xlink:role="http://xml.opendap.org/dap/dap2.xsd" xlink:type="simple"
href="http://example.com/opensdap/ExampleDataCollection.005/2009/001/Example.2
009.01.01.054.hdf"/>
    <link rel="enclosure" length="4539076" type="application/x-netcdf"
xlink:role="http://xml.opendap.org/dap/dap2.xsd" xlink:type="simple"
href="http://example.com/opensdap/ExampleDataCollection.005/2009/001/Example.2
009.01.01.054.hdf.nc"/>
    <link rel="describedby" type="text/xml"
href="http://example.com/Example.005/2009/001/Example.2009.01.01.054.hdf.xml"
/>
    <link rel="icon"
href="http://example.com/ExampleDataCollection/Example.2009.01.01.054.hdf"
type="image/jpeg"/>
    <georss:box>-42.017, 33.9236, -63.6063, 10.6306</georss:box>
    <time:start>2009-01-01T05:23:24Z</time:start>
    <time:end>2009-01-01T05:29:24Z</time:end>
    <dc:date>2009-01-01T05:23:24Z/2009-01-01T05:29:24Z</dc:date>
    <summary type="html">start:2009-01-01T05:23:24Z,
end:2009-01-01T05:29:24Z, size:2510007</summary>
    <content type="text">Example data file for 2009-01-01T05:23:24Z to
2009-01-01T05:29:24Z</content>
</entry>
</feed>

```

N.B.: the inclusion of xlink will cause many Atom validators to fail. Implementations that rely on such a validator should strip out the xlink attributes before handing it to the validator.

12 APPENDIX B - Acronyms

CSW - Catalog Services for the Web

DCP - Discovery Change Proposal

EOS - Earth Observing System

EOSDIS - Earth Observing System Data and Information System

ESIP - Earth Science Information Partners

HTTP - Hypertext Transfer Protocol

IETF - Internet Engineering Task Force

MODAPS - Moderate Resolution Imaging Spectro-radiometer Adaptive Processing System

NSIDC - National Snow and Ice Data Center

OGC - Open Geospatial Consortium

OPeNDAP - Open-Source Network project for a Data Access Protocol

RFC - Request for Comments