

# VOSpace Standards and Implementation Notes

**Brian Major**  
**Canadian Astronomy Data Centre**  
**IVOA - October 2016**



# VOSpace Standard and Implementation Notes

1. VOSpace Public Share Protocol
2. Code To Data using VOSpace Views
3. OpenCADC VOSpace project

# Simple Sharable VOspace download URL

The problem:

- VOspace is a complex protocol that requires a smart client to do transfer negotiation
- Users want to share simple download URLs to give to collaborator
- Must be compatible with browsers, curl, wget
- Must be referenceable in publications, DOIs

# Public Share Protocol

Dave Morris

Protocol URI (and documentation!)

<http://wiki.ivoa.net/twiki/bin/view/IVOA/VOSpacePublicShare>

# VOSpacePublicShare

## VOSpace public share protocol

This wiki page serves as the definition for a VOSpace transfer protocol for sharing node content.

The target use case is to enable users to get a simple URL that provides access to the node content using a simple HTTP GET request.

If a client includes the URI for this protocol in a PullFromVOSpace transfer request, then the endpoint URL in the VOSpace server response will contain a simple URL that provides HTTP GET access to the node content.

### Basic requirements

- The endpoint URL **MUST** work with a simple HTTP GET request.
- The endpoint URL **SHOULD** be suitable for listing in a published paper.
- The endpoint URL **SHOULD** be suitable for sending to someone in an email.
- The endpoint URL **MUST** be able to be used multiple times by multiple clients.

# VOSpacePublicShare (2)

## Technical details

- The URL of [this wiki page](#) serves as the URI for this transfer protocol.
- This protocol SHOULD only valid for a PullFromVOSpace transfer request.
- The UWS transfer job will complete immediately.
- The endpoint URL MUST continue to be valid after the transfer job has completed.
- The endpoint URL path MUST end with a valid file name based on the node name.
- The header fields of a HTTP response to a HTTP GET requests MUST contain a valid file name based on the node name.
- The header fields of a HTTP response to a HTTP GET requests MUST an appropriate mime-type based on the node content.

# VOSpacePublicShare (3)

## Technical issues

Things that need to be discussed..

- What if the transfer request contains more than one protocol - do the others remain valid even if this one has completed ?
- What happens if the node content changes ?
- Do we want to define a **ZippedArchive** view that can be used in combination with this protocol to download the contents of a ContainerNode as a zipfile ?

Things that probably go in a separate **ProtectedShare** protocol ..

- Do we want to add a property to control the URL lifetime ?
- Do we want to add a property to control the number of downloads allowed ?
- Do we want to add a property to control access ?

# Using OpenCADC to implement VOSpace 2.1

<http://www.github.com/opencadc>

- Giuliano Taffoni, Sara Bertocco and INAF / OATS team have created a successful implementation of the OpenCADC VOSpace

OpenCADC VOSpace:

- Provides all metadata support in a relational database
- Provides all transfer negotiation support over UWS



# Three Java Repositories

cadc-vos: data model classes and client

cadc-vos-server: RESTful vospace interface for:

- /nodes - node metadata
- /transfer /synctrans - transfer negotiations
- /capabilities /availability /views /properties /protocols

cadc-test-vos: set of server integration and conformance tests

# Newly ported to gradle

No more ant!

> gradle build

- Auto retrieval of dependencies
- Compile, compile test, run unit tests
- Integration testing manual

# Github and Travis Continuous Integration

gradle build - runs automatically on pull requests

Pull request will fail if test fails

To support integration tests in github, we'll need auto deployment to a container running the VOSpace service

# Supported Transfers

## File Downloads (pullFromVoSpace):

- Public read access (anonymous)
- Owner
- Group protected read via GMS

## File Uploads (pushToVoSpace):

- Owner
- Group protected writes via GMS

# Transfer Generator Interface

When transfer requests are made, the `cadcd-vos` code will call your transfer generator implementation to get the list of URLs for the `pullFromVoSpace` or `pushToVoSpace` request.

```
List<URL> getURLs(VOSURI target, Protocol protocol,  
    View view, Job job, List<Parameter> additionalParams)  
    throws FileNotFoundException, TransientException;
```

## TODO: Change post transfer UWS job update

When the download or upload is complete, a call needs to be made to change the transfer job phase from EXECUTING to COMPLETE (or ERROR).

We will likely be changing the transfer generation behaviour so that the job is considered complete when the transfer details (URLs) have been generated.

# TODO: Remove RESTlet dependency

We would like to switch to a plain Java servlet implementation

We found that RESTlet is difficult to configure and hides too many aspects of the HTTP request

# TODO: Simplify Configuration

There are too many ways to configure the VOSpace service implementation.



## Next: Server-to-Server transfers

`pullToVoSpace`, `pushFromVoSpace`

Uploads and downloads directly from one VOspace to another

# Users and Contributions Welcome!

We would be happy to hold your hand in getting a VOSpace implementation running.

# Code to Data

Some thoughts on how VOSpace can provide a nice interface to this problem...

# Code To At Data - Node Views for FITS files

- `view=cutout cutout=[2][1:100,1:100]` [1]
- `view=fhead` [1]
- `view=wcs (STC-S -> WCS)` [2] [3]

[1] CFITSIO - <http://heasarc.gsfc.nasa.gov/fitsio/fitsio.html>

[2] fits2wcs - <https://github.com/opencadc/wcs/tree/master/cadc-wcs>

[3] WCS - <http://www.atnf.csiro.au/people/mcalabre/WCS/>

# Code To At Data - Using capabilities

One may also use VOSpace capabilities to perform an operation on a file.

This differs from views in that the operation is performed immediately on file retrieval and only the result is saved on the Node.

Use cases: data reduction, metadata extraction

# Dynamic code to data using views

```
http://server.example.com/vospace/nodes/myvospace/myData?  
view=codeToData&  
container=vos://example.com~vospace/myvospace/myDocker
```



Other code-to-data ideas?



**Thank you**

**Brian Major  
Canadian Astronomy Data Centre  
IVOA - October 2016**

