CMIP5 Dataset Version Directory Structure

Stephen Pascoe, Version 0.4, 2010-04-19

This document recommends a directory layout for supporting multiple versions of datasets within the CMIP5 archive. In drawing up this proposal the following factors have been considered:

1. The structure should conform as closely as possible to the CMIP5 Data Reference Syntax document [DRS].
2. The structure should be workable with the current ESG publisher software.
3. The structure should present a consistent user interface for downloading data files.
4. The structure should be manageable by operators of ESGF data nodes.

To give some background each of considerations are discussed below then the proposed directory structure is described in detail.

1 Discussion of Considerations

The current plans for version control in the CMIP5 archive derive from 2 independent sources: The DRS document and the ESG publisher software. **[MORE]**

1.1 DRS Document

The DRS document divides the CMIP5 archive into a set of *atomic datasets* defined terms of several components including a version number, therefore it implies datasets are immutable with a fixed version number. The assumption is that when data within an atomic dataset is superseded a new dataset is generated with an incremented version number; the original dataset remains unchanged and available to users.

The DRS document recommends version should be part of the URL syntax used to allow users to download data and also part of the directory structure used to store the data at the data-node.

The document is not precise on many details of how the DRS version should be used. For instance how data managers update datasets from one version to another and how data and metadata from previous versions are stored to maximise efficiency and manageability are not addressed

1.2 ESG Publisher

ESG publisher uses the terms *dataset* and *dataset version* to represent a set of files published in one operation. A dataset version is an immutable object which can represent a 'DRS dataset including version number'. The published 'dataset version' itself has an identifier which typically consists of dataset_id+version number; this appears in the THREDDS catalog of the dataset version**.**

Within ESG Publisher each dataset version is assigned a single dataset id and version number. Republishing a dataset with a pre-existing dataset id usually triggers a new version of that dataset. For each dataset version a new THREDDS catalogue is produced. ESG Publisher also detects when files change such that an updated file will be recorded as a new version of that file. In this case the ESG publisher database keeps track of the metadata of previous file versions but does not manage previous versions of the actual files. The paths to files within a dataset can be changed with the *rename* operation, however this operation does not increment the version number.

1.3 User Interface to Data Files

It is expected that users will visit an ESG Gateway to generate wget or DML scripts containing lists of URLs to data files. These URLs should follow the URL syntax suggested in the DRS document. Importantly the path portion of the URL identifies the atomic dataset and therefore, from the user's perspective, these URLs will be the primary identifiers of both atomic datasets and files in the archive. The version directory structure has been designed such that URLs have the following user-centric features, assuming URLs follow reflect the directory structure:

1. The user has a record of which version of the data they downloaded from the version numbers embedded in the URLs
2. Running the script still retrieves the original data even after a new version is published (except for "latest", see below)
3. URLs can be edited to retrieve other versions of an atomic dataset if the user knows that version exists.
4. When a new dataset version is published it should be possible for a user to download only those files that have changed in the new version. This may require tool support.
5. A special DRS version "latest" can be used to retrieve the latest version of a dataset. Downloading "latest" has the disadvantage that it is not immediately clear to the user which version you are downloading. Additional tools should be made available to deduce the exact DRS version of a set of files based on their *tracking_id* or checksum
6. A possible desirable feature is for wget scripts to support recursive download below the atomic dataset level so that if the user doesn't know the exact filenames in a dataset version they can still retrieve it.

1.4 Data Management

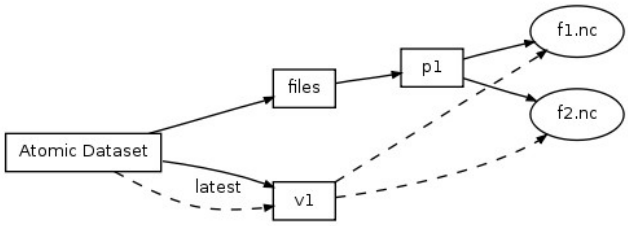
Maintaining previous versions of *atomic datasets* must not put too high a burden on the data management practices of data-node administrators.

1. If a file is unchanged between versions there should be no need to store the file twice. Using symbolic links is recommended for solving this problem.
2. Filenames should remain unchanged when a version is superseded.
3. Mixing data files and symbolic links within one directory should be avoided as the asymmetry complicates data management.
4. It should be possible to remove the data of previous versions without breaking symbolic links. This will probably require tool support.

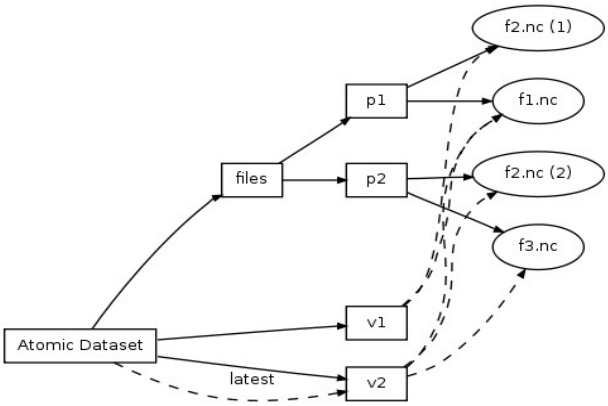
2 Proposed Version Directory Structure

The following directory structure directly reflects the DRS path syntax with each atomic dataset in a single directory whilst maintaining changes between each dataset versions in a separate directory tree. The directory structure below each atomic dataset directory is divided into a **files** directory containing all data files, the directories **v<n>** containing links to files at a particular version and a link **latest** to the most recent **v<n>** directory.

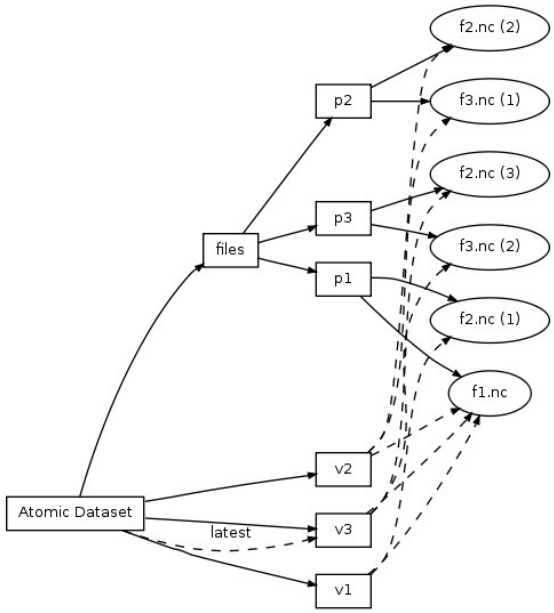
Within **files** a series of subdirectories **files/p<n>** contain files changed or added at each version. This is illustrated with the following scenario of an atomic dataset at 3 successive versions.



*Version 1: Add **f1.nc** and **f2.nc***



*Version 2: Replace **f2.nc**, add **f3.nc***



*Version 3: Replace **f2.nc** and **f3.nc***

2.1 Advantages to this structure

1. The filesystem holds a record of *versions* and *version changes* independently. Directories **v<n>** hold *versions* and directories **files/p<n>** hold version changes.
2. No files need moving between version changes.
3. Links and files are completely separated.
4. Users could update a dataset simply by downloading files in a **files/p<n>** directory.

2.2 Disadvantages to this structure

1. The link **latest** is fairly complex, pointing to a **v<n>** directory containing links to multiple **files/p<n>** directories.
2. Removing previous versions is complex. If **v1** and **files/p1** were removed **f1.nc** would need to be moved to **files/p2** and all symbolic links to it updated.

2.3 Publishing atomic datasets

It is expected that ESG publisher will publish multiple atomic datasets as *realm datasets*. There is therefore a potential mismatch between the dataset version as recorded in ESG publisher and the version of each individual atomic dataset. This could be resolved by:

1. When a realm dataset changes all atomic datasets are republished as new versions. This would lead to new atom datasets that are unchanged from their previous versions.
2. When a realm dataset changes only those atomic datasets that have changed are republished as new versions. This would make the realm dataset version independent of any atomic dataset version.

Both these options are likely to be very confusing to users without clear guidance and software support. Ideally the user should only be aware of a single concept of version. This would be achieved by no. 1 above however, if a version change affected only a small part of a realm dataset, no. 1 could lead to a large number of duplicate symbolic links.

2.4 User download experience

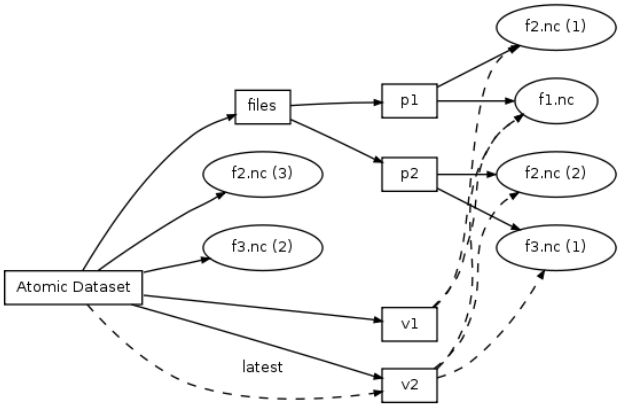
When a new dataset version is published users will wish to download only those parts of the dataset that have changed. This information is contained in the **files/p<n>** subdirectories that are not expected to be visible to the user. We therefore need a service to provide users with wget scripts to download these changes.

2.5 Version Transition Procedure

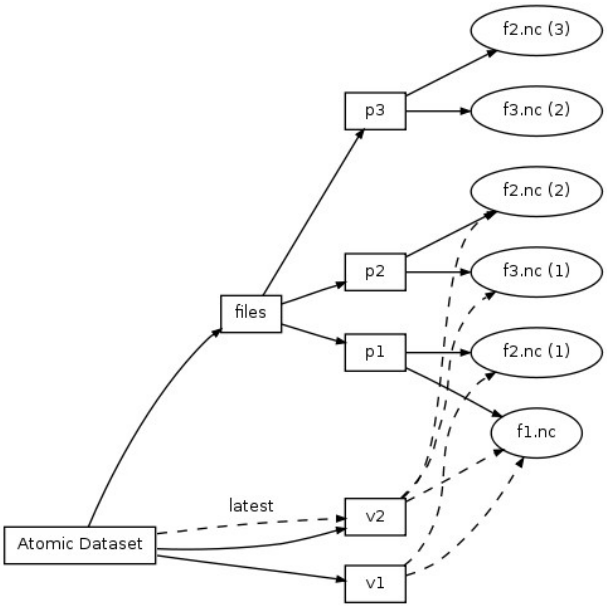
An important consideration is how the version directory structure is manipulated during version changes. In general the procedure of moving an atomic dataset from one version to another is as follows.

1. Files are deposited in the atomic dataset directory by CMOR or manually.
2. The new files are moved into **latest**. If a new file replaces an existing one the old file is moved to **old/v<n>**.
3. The symbolic link directories **v<n>** are updated to reflect any files that have moved.

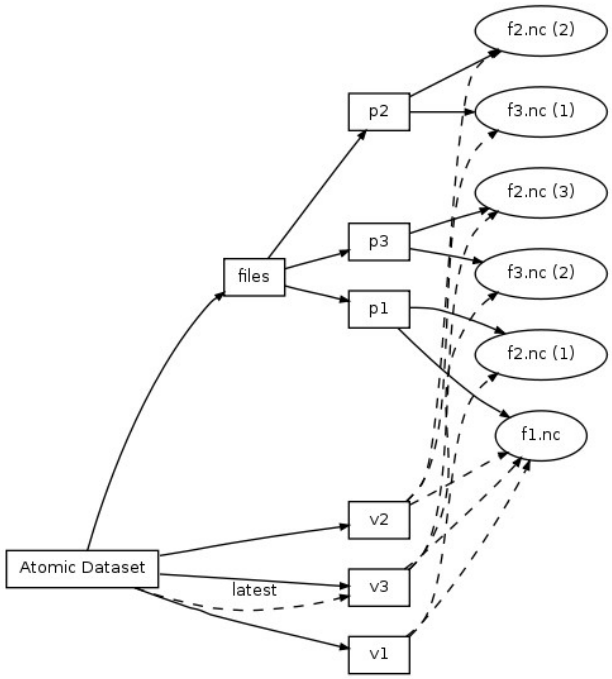
A more detailed algorithm is outlined below with reference to the transition between version 2 and version 3.



*Step 1: The new files **f2.nc** and **f3.nc** are deposited in the atomic dataset directory.*



*Step 2: Create directory **p2** and move all new files into it.*



*Step 3: Create directory **v3**. For each file **f** in **files/p3** create links **v3/f → files/p3/f**. For each link **f** in **v2** where **f** is not in **v3** duplicate the link in **v3**. Change latest to point to **v3***

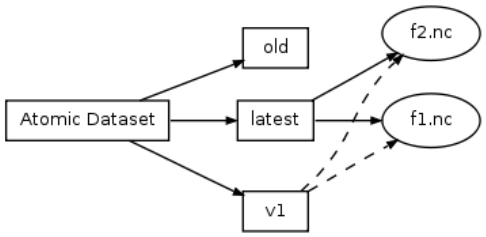
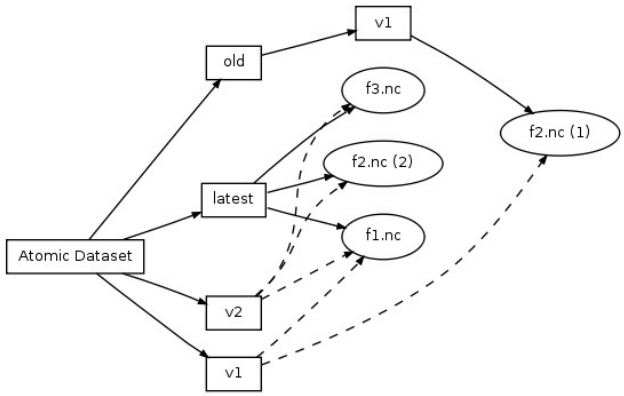
3 Old Version Directory Structure Recommendation

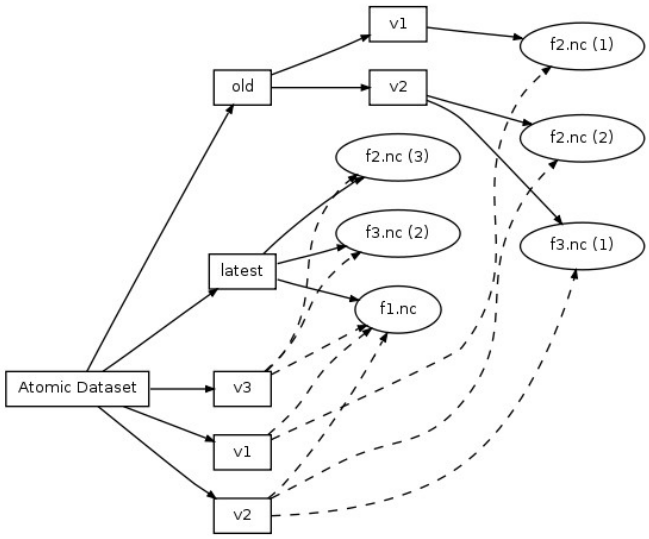
The Old directory structure recommendation is included here for reference.

The directory structure below each atomic dataset directory is divided into 2 groups of subdirectories: files and links. The **latest** and **old** directories contain the data files. The **latest** directory always contains the files for the latest version and **old/v<n>** contains those files that existed in previous versions but not in the latest.

The directories **v<n>** (not to be confused with **old/v<n>** above) contain symbolic links to the files available for each version **n**.

This is illustrated with the following scenario of an atomic dataset at 3 successive versions.

Version	Contents / Change	Directory Structure
1	f1.nc, f2.nc	
2	f2.nc replaced, f3.nc added	

3	f2.nc and f3.nc replaced	

3.1 Advantages to this structure

5. The two groups of subdirectories mirror the two approaches taken by the DRS document and ESG Publisher. The subdirectories **latest** and **old** represent a mutable view of the atomic dataset whereas the subdirectories **v<n>** represent an immutable view of atomic dataset versions.
6. Filesystem complexity is confined to previous versions. **latest** always contains the files of the most recent version and the most recent **v<n>** directory always contains links to **latest**.
7. Links and files are completely separated.
8. Data files for previous versions can be removed with manageable implications for fixing broken links. For instance if **old/v1** and **v1** were deleted a relatively simple algorithm could detect and move the link **v2/f2.nc** to point to **old/v2/f2.nc**.

3.2 Disadvantages to this structure

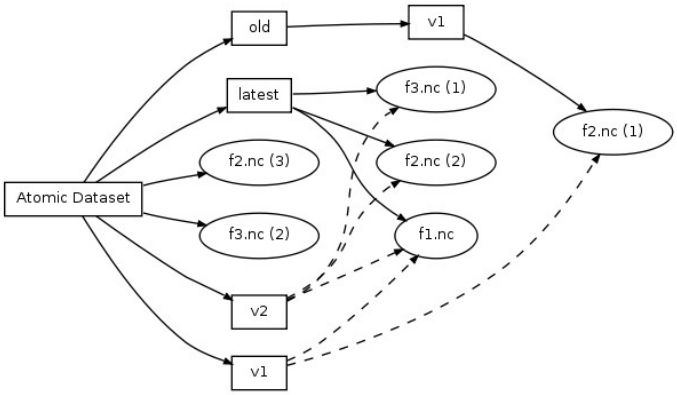
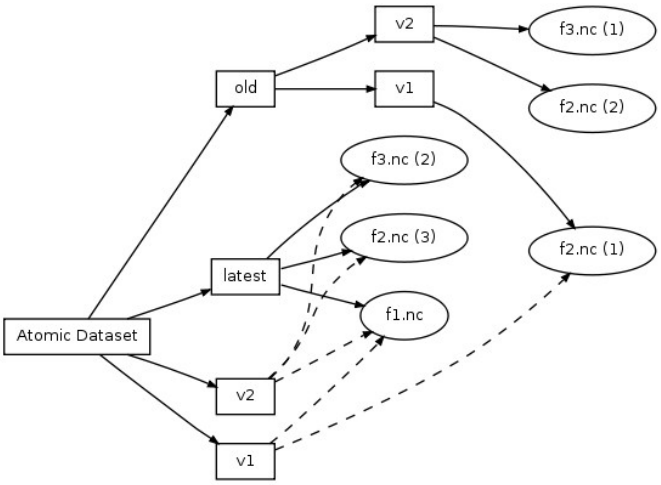
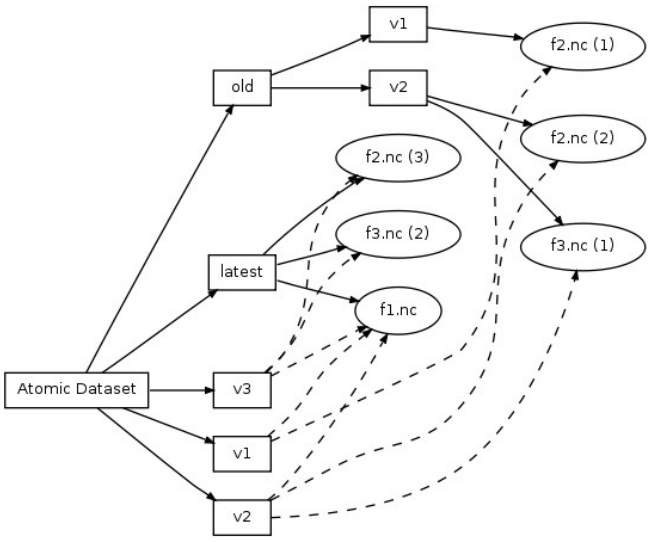
1. Creating backups may become more complex because files are moved during version changes.

3.3 Version Transition Procedure

An important consideration is how the version directory structure is manipulated during version changes. In general the procedure of moving an atomic dataset from one version to another is as follows.

1. Files are deposited in the atomic dataset directory by CMOR or manually.
2. The new files are moved into **latest**. If a new file replaces an existing one the old file is moved to **old/v<n>**.
3. The symbolic link directories **v<n>** are updated to reflect any files that have moved.

A more detailed algorithm is outlined below with reference to the transition between version 2 and version 3.

The new files f2.nc and f3.nc are deposited in the atomic dataset directory.	
1. Create directory **old/v2** 2. For each new file **f** in this version 1. If **latest/f** exists and is different from **f** move **latest/f** to **old/v2/f** 2. Move **f** to **latest/f** **NOTE:** At this point symbolic links **v2/f2.nc** and **v2/f3.nc** are wrong.	
1. Create directory **v3** 2. For each file f in **latest** create links **v3/f → latest/f.** 3. For each file **f** in **old/v2** find any links **v1/f → latest/f** and change them to **v1/f → old/v2/f**	

4 Citations

[DRS] CMIP5 Data Reference Syntax (DRS) and Controlled Vocabularies. *http://cmip-*

pcmdi.llnl.gov/cmip5/docs/cmip5_data_reference_syntax.pdf