HDF Product Designer Documentation

Release 1.3.0

The HDF Group

December 14, 2015

Contents:

ONE

INTRODUCTION

HDF Product Designer (HPD) helps data producers design conventional HDF5 products easily and generate consistently interoperable data products by utilizing best practices and standards if they exist in their data user communities. Conventions are defined using the powerful CLIPS expert system and designs can be re-used across product suites.

Key goals this application strives to fulfill:

- Facilitate creation of interoperable and standards-compliant data products in HDF5 as early as possible in the project development process.

- Support multiple computing platforms without requiring the full software stack of development tools and libraries installed.

- Easy and intuitive editing (create, update, move, copy, delete) of HDF5 objects.

- Collaborative approach to product design (project/team/organization-wide).

- Incorporation of best practices and standards from targeted data user communities.

- Integration of compliance and interoperability tests into the design workflow.

- Content import from existing files.

- Export of designs as HDF5 files, HDF5/JSON, or as source code in several programming languages.

The Hierarchical Data Format (HDF5) provides a flexible container that supports groups and datasets, each of which can have attributes. In many ways, HDF5 is similar to a directory structure in a file and, like directory structures, the same data can be structured and annotated in many ways. This flexibility empowers HDF5 users to arrange data in ways that make sense to them. However, it can make it difficult to share data as users, and tools, must understand the structure and the properties of data in order to use and understand it.

Many communities have successfully addressed this problem by creating conventional structures and annotations for data in HDF5. This approach depends on data files (e.g., products) that carefully follow these conventions. In some cases, designing and writing those files can be challenging or the user creating the product may be driven by local needs that lead to deviations from the conventions. Unfortunately, even small deviations can cause problems for downstream tools and future users.

What is a *HDF5 product* in the context of this software? HDF5 product is the content that should exist in a single HDF5 file. This content is defined by the HDF5 objects (groups, attributes, datasets), their names, the hierarchies they create (links and references), and attribute values. Dataset values are typically not stored in such files (unless they qualify as *metadata*) thus this software cannot be used as a data server. Once completed, a HDF5 product is replicated in many files (commonly on the order of tens of thousands or more) and filled with real data.

The HDF Product Designer code repository is hosted in the NASA Earthdata Code Collaborative.

TWO

WHAT'S NEW

The current version of HDF Product Designer is 1.3.0. It was released on 2016-01-01. Mentioned below are a few notable highlights:

- NetCDF CDL design importer fixed

- Design import from *OPeNDAP DMR* output

- Support for the variable-length string datatype

- Product *documentation* in the Microsoft Word `docx` format

- *Notification* about a newer version is displayed in login window on startup

The Known Issues section lists known issues and limitations.

Release notes for previous versions are available below:

2.1 Version 1.2.0

Version 1.2.0 of the HDF Product Designer was released on 2015-09-30. Here are a few highlights:

- Design import from netCDF CDL and HDF5 files.

- Design export as Python, MATLAB, or IDL source code is greatly improved.

- Support for unlimited dataset dimension size.

- Improved handling of dimension scales in template files.

- For null-terminated string values, the string length in template files is now always one character longer to accommodate the terminating null character.

2.2 Version 1.1.0

Version 1.1.0 of the HDF Product Designer was released on 2015-07-01. Here are a few highlights:

- If a dataset has _FillValue attribute and CF or NUG convention is active, the value of _FillValue attribute will be used as fillvalue of the dataset's creation property when a design is imported.

- Compound dataset can be edited. Compound field can be dragged and dropped.

- Tree control uses white background and color icons.

- stderr/stdout messages are redicted to a separate window.

- IDL code generation is improved.

2.3 Version 1.0.0

Version 1.0.0 of the HDF Product Designer was released on 2015-04-01. This is the first official release of HDF Product Designer Desktop.

THREE

KNOWN ISSUES

HDF Product Designer is still being actively developed. Below is a list of currently known issues or limitations:

- Nested compound datatype is not supported.

- Exporting designs with an array datatype compound field is not yet supported by all source code generators.

- `vlstring` (variable-length string) datatype cannot be used as a compound field.

- Unsupported menu items are disabled. They are placeholders for future features.

- Choosing both CF and NUG for conventions when creating a new design is not supported. Select either one only.

- Convention support works for new designs only. Changing design's convention later is not supported.

- Support for the HDF-EOS convention is still in very early stages of development. It will not yield valid HDF-EOS5 template files because it is not possible to assign values to datasets. Specifically, it is not possible to store ODL string into the `StructMetadata` dataset which is essential for HDF-EOS5 library to access the file.

- **Export as** works only for the current working version (label:`HEAD`).

- Design/dataset/attribute/group names cannot be longer than 255 characters.

- The / character is not allowed in object names and will be replaced with the _ character.

- The size of attribute's value cannot exceed one gigabyte.

- The system utilizes user sessions that expire one day after the login time. The best practice is to log out (quit) the application daily.

- There is a 30-second timeout limit for all operations. Typically long executing operations, like template file validation, may fail due to this timeout.

- Generating HDF5 template files for designs with large (> 64 kB) attributes may fail with an error: `Unable to create attribute (object header message is too large)`.

To report any new issue please visit the issue tracking website. (NASA Earthdata login required.)

FOUR

GETTING STARTED

Preparing to use HDF Product Designer is very simple. The following sections explain what and how.

4.1 Download

We provide the binaries for the following platforms:

Platform	Binary	SHA256 Checksum
Windows 7/8/10 Setup	HPD_setup.exe	255f7296f7fa3c2e780366d2dcbe23ea1966a828a9c88d2cf198ac6f4c715e1f
Windows 7/8/10	HPD.exe	9d13c47642c1407d0e98596cf9c0d237e4bce91a4626c3c1c468b5c2c59f0f1e
Mac OS X (Mavericks or newer)	HPD.dmg	8f1410c79139222331f186a1f0d07c594af3f9c1d679c8bd3fdaad6a9a265ce3
Linux 2.6 (x86_64)	HPD	035a6f241bc19930340e2e1e22c4c9e5a73c1aa6fce0f62006db0e84ef50183b

Note: The Windows Setup binaries require administrator privileges to install. Use the other build for your Windows platform if not having sufficient user privileges.

The Windows binaries require Microsoft Visual C++ 2008 Redistributable Package (x86).

4.2 Installation

The binaries in the *Download* section will work as is, no additional software tools or libraries are needed. Double click on the icon to launch Product Designer.

For the Mac OS X binary, drag & drop the *HPD.app* from the HPD drive into the *Application folder*. Then double click the application. Typically a security exception is required to open applications from unknown developers. Please follow these instructions in that case.

When the application is installed correctly and starts successfully, it will display a login window requesting user authentication information for the NASA Earthdata Login service.

If a new version is available the login window will also display a hyperlink next to the Login button with the text: *New version x.y.z of HPD is available!* (Fig. 4.1). Clicking the hyperlink will open a browser window with the web page from which the latest version of the HPD binaries can be downloaded.

If you want to build this application from the source, please visit its code repository.

Fig. 4.1: New version notification hyperlink next to the Login button

4.3 User Authentication

HDF Product Designer does not collect or store any user information that can be used for authentication. It relies instead on each user's organization to provide such service for them.

Upon first-time successful authentication, user's name (first and last) and email address are collected from the authentication service and kept. User's email represents their identity and that email is compared after each subsequent authentication.

Note: Since HDF Product Designer obtains user personal information from their authentication service they have full control of what information about them is shared between the application and the service.

Currently supported authentication services and how to configure them are explained below.

4.3.1 NASA Earthdata Login

If you don't have NASA Earthdata Login account, please register first. The process requires NASA's approval so it may take some time.

Sign in to NASA Earthdata Login website, go to **My Applications** tab, press "Approve More Applications" (if necessary), find and approve HDF Product Designer application to access your profile as shown in Fig. 4.2.

My Applications

🖥 Approved Applications
Applications that use your URS profile for authentication.

Earthdata Code Collaborative
Earthdata Wiki
Earthdata Feedback Module
Earthdata Website
ECHO Reverb
HDF Product Designer

APPROVE MORE APPLICATIONS

Fig. 4.2: Web page for approving HDF Product Designer application

USAGE

Major capabilities of the application are showcased and explained in this part of the guide. Please see the Getting Started section on how to install this software if you would like to try out the features described here.

5.1 Project

Projects are the main organizational and collaborative unit of this application. One project can have one or more users. Every user must belong to at least one project. Collaboration among users is determined by project membership.

By default, every user belongs to two projects: **Sandbox** (shared among all users), and the personal project with the user's name. For any issue regarding projects and users, please contact the HPD Server administrator (ajelenak@hdfgroup.org). If requesting more users to join a project provide their emails as registered in the URS.

To open any of the user's projects use the menu **Project > Open**.

5.2 Design

Designs represent content stored in HDF5 files. A single project can have many designs and all users of the same project can access its designs.

A design can have multiple versions and the version that can be edited is called *current working* version. The design versioning system is a simple timeline of saved versions (checkpoints). At any point while working on a design, user can opt to save its version as a checkpoint. Any of these checkpoints can be later promoted into the current working version. Checkpoints must have unique labels, and the current working version's label is HEAD and cannot be changed.

Currently supported design actions:

- Create a new design: **Design > New**.

- Open an existing design: **Design > Open**.

- Import design: **Design > Import**.

- Checkpoint design's current version: **Design > Save**.

- Close design in order to work on another one: **Design > Close**.

- Delete a design: **Design > Delete**. This action cannot be undone so please be cautious.

5.2.1 Import

Importing designs allows reuse of existing HDF5/HDF4/netCDF products as a starting point for creating new products. Currently HPD Desktop supports import from the formats below:

- HDF5 File
- HDF5/JSON
- HDF4 File Content Map
- NcML
- CDL
- OPeNDAP DMR

All import formats except the HDF5 File are text-based so can be viewed or edited easily with any text editor before import.

HDF5

To import a design from an HDF5 file, please follow the steps below:

1. Select a project from the tree view (Fig. 5.4).

2. Choose **Design > Import from > HDF5** from the menu (Fig. 5.5).

3. A file selection dialog will appear. Select an HDF5 file and then **Open**. (Fig. 5.6).

4. HPD Desktop will start loading the design from the file. It will update tree view dynamically during import (Fig. 5.7) by scrolling down the view automatically as the tree grows. The importing process is completed when the scrolling stops. Inspecting the import is recommended at this point.

5. If a parsing error occurs during the import, the design will be saved up to the point where the parsing error occurred. The error message(s) will be shown in a separate wxPython stderr/stdout window.

Note: HDF5 file content is converted to HDF5/JSON in memory prior to importing. Any HDF5 feature not supported by the hdf5-json tools will cause import operation to fail.

HDF5/JSON

HDF5/JSON is a JSON representation of HDF5 file content. JSON (JavaScript Object Notation) is a lightweight, open format for data exchange, very popular in web applications. The h5tojson.py command line program can be used to generate the JSON from any HDF5 file.

To illustrate this functionality we have added to our HDF-EOS THREDDS server a capability to generate HDF5/JSON for the sample NASA HDF-EOS5/HDF5 products hosted there, as shown in Fig. 5.1 and Fig. 5.2.

If you click one of sample NASA products, you'll get **H5JSON** link under **Access** (Fig. 5.2).

Clicking the link will generate HDF5/JSON file *without* actual data (i.e., metadata information only) as shown in Fig. 5.3.

If the HDF5/JSON representation of an existing HDF5 file using the above method is saved to a file, it can be imported by following the steps below.

1. Select project node from the tree view (Fig. 5.4).

2. Press **Design > Import from > HDF5 JSON** from menu (Fig. 5.5).

3. A file selection dialog will appear. Select an HDF5/JSON file (Fig. 5.6).

4. Press the **Open** button. HPD Desktop will start loading the design from the file. It will update tree view dynamically during importing (Fig. 5.7) and the tree view control will scroll down automatically as the tree

Fig. 5.1: THREDDS web service for generating HDF5/JSON
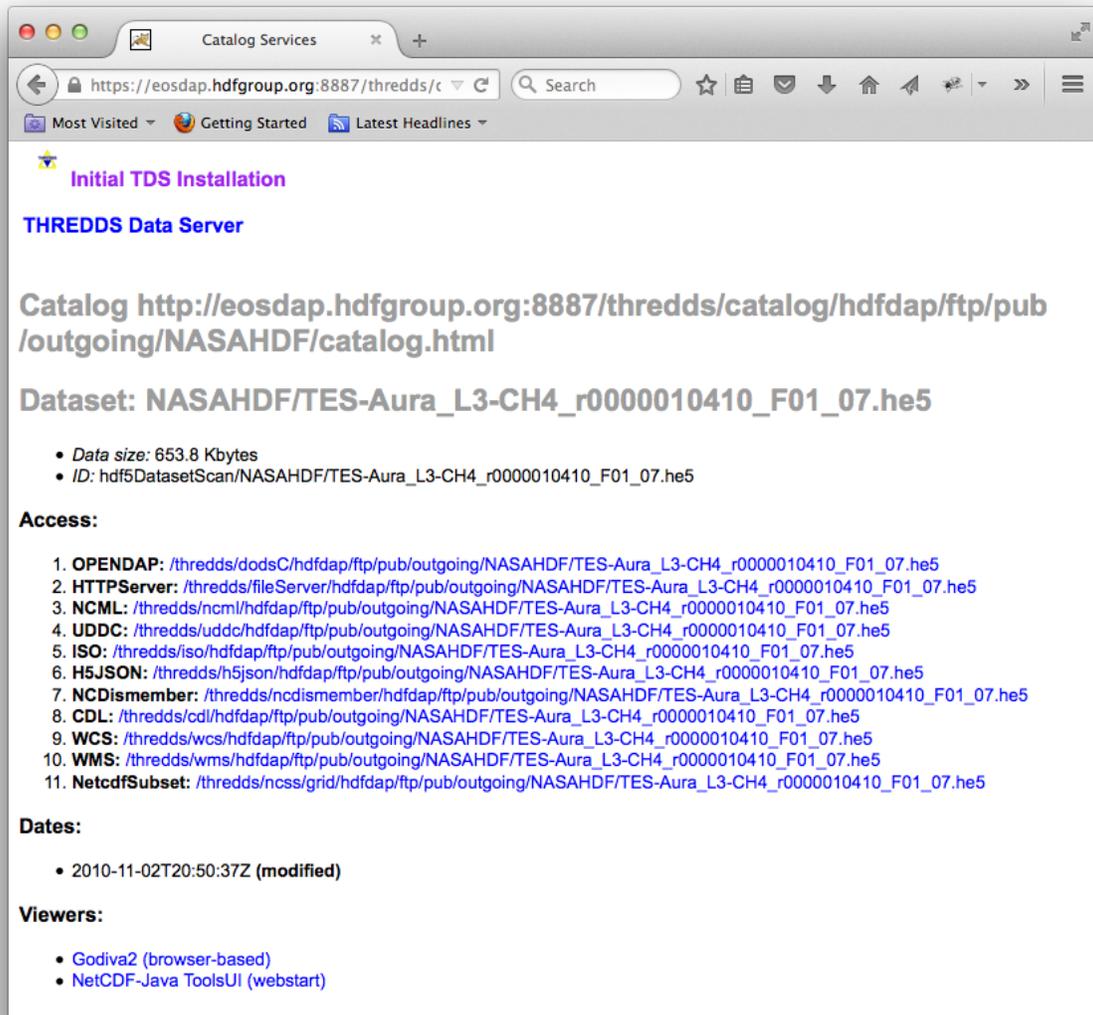
Fig. 5.2: THREDDS catalog page displaying various access methods for one HDF5 file

```
https://eosd..._F01_07.he5    +

https://eosdap.hdfgroup.org:8887/thred    Q Search

Most Visited    Getting Started    Latest Headlines

{
    "apiVersion": "1.0.0",
    "datasets": {
        "4f393406-307b-11e5-b51a-005056008d34": {
            "alias": [
                "/HDFEOS/GRIDS/NadirGrid/Data Fields/CH4"
            ],
            "attributes": [
                {
                    "name": "_FillValue",
                    "shape": {
                        "class": "H5S_SIMPLE",
                        "dims": [
                            1
                        ]
                    },
                    "type": {
                        "base": "H5T_IEEE_F32LE",
                        "class": "H5T_FLOAT"
                    },
                    "value": [
                        -999.0
                    ]
                },
                {
                    "name": "MissingValue",
                    "shape": {
                        "class": "H5S_SIMPLE",
                        "dims": [
                            1
                        ]
                    },
                    "type": {
                        "base": "H5T_IEEE_F32LE",
                        "class": "H5T_FLOAT"
                    },
                    "value": [
                        -999.0
                    ]
                },
                {
                    "name": "Title",
                    "shape": {
                        "class": "H5S_SCALAR"
                    },
                    "type": {
                        "charSet": "H5T_CSET_ASCII",
```

Fig. 5.3: HDF5/JSON output from THREDDS H5JSON access
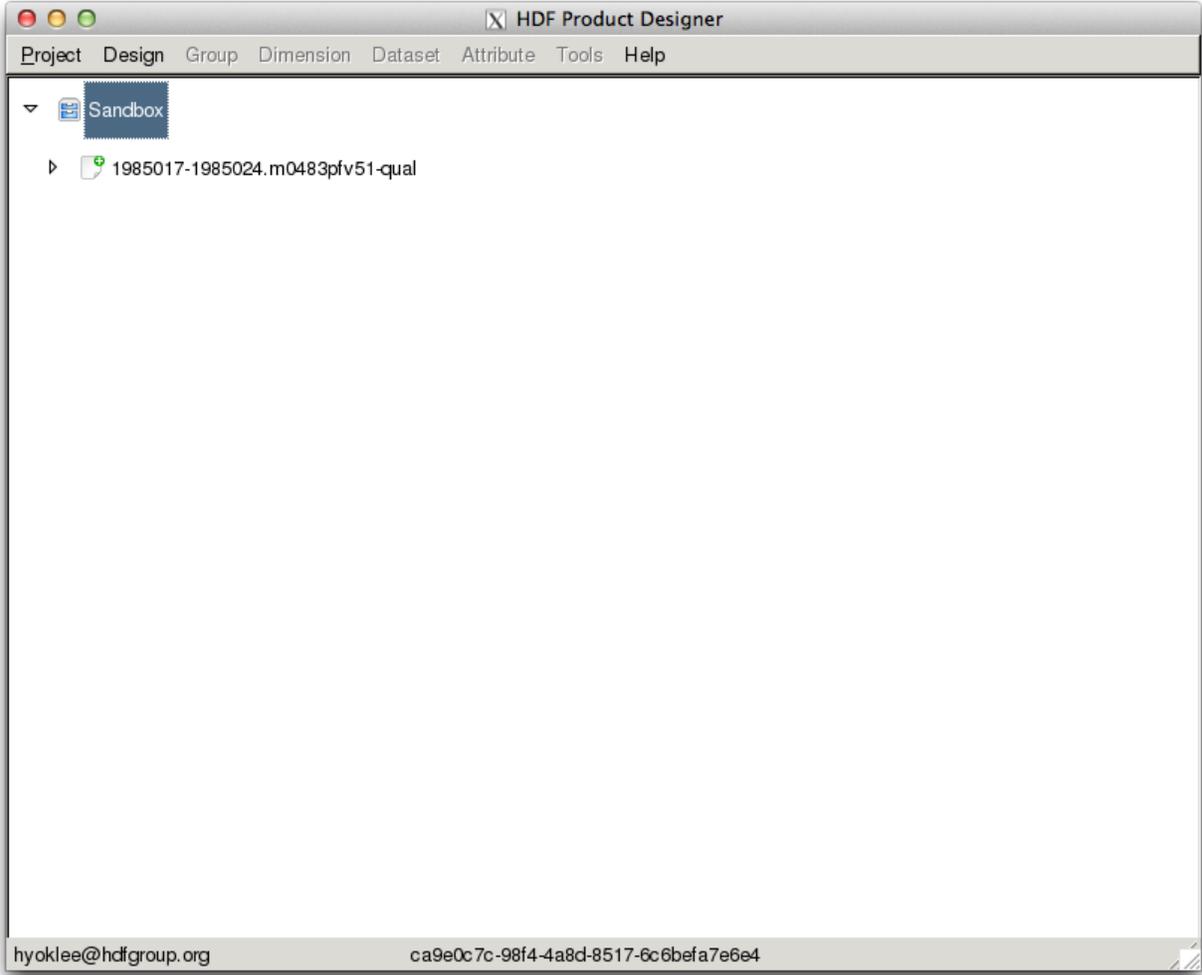
Fig. 5.4: HPD Desktop tree control view with a selected project
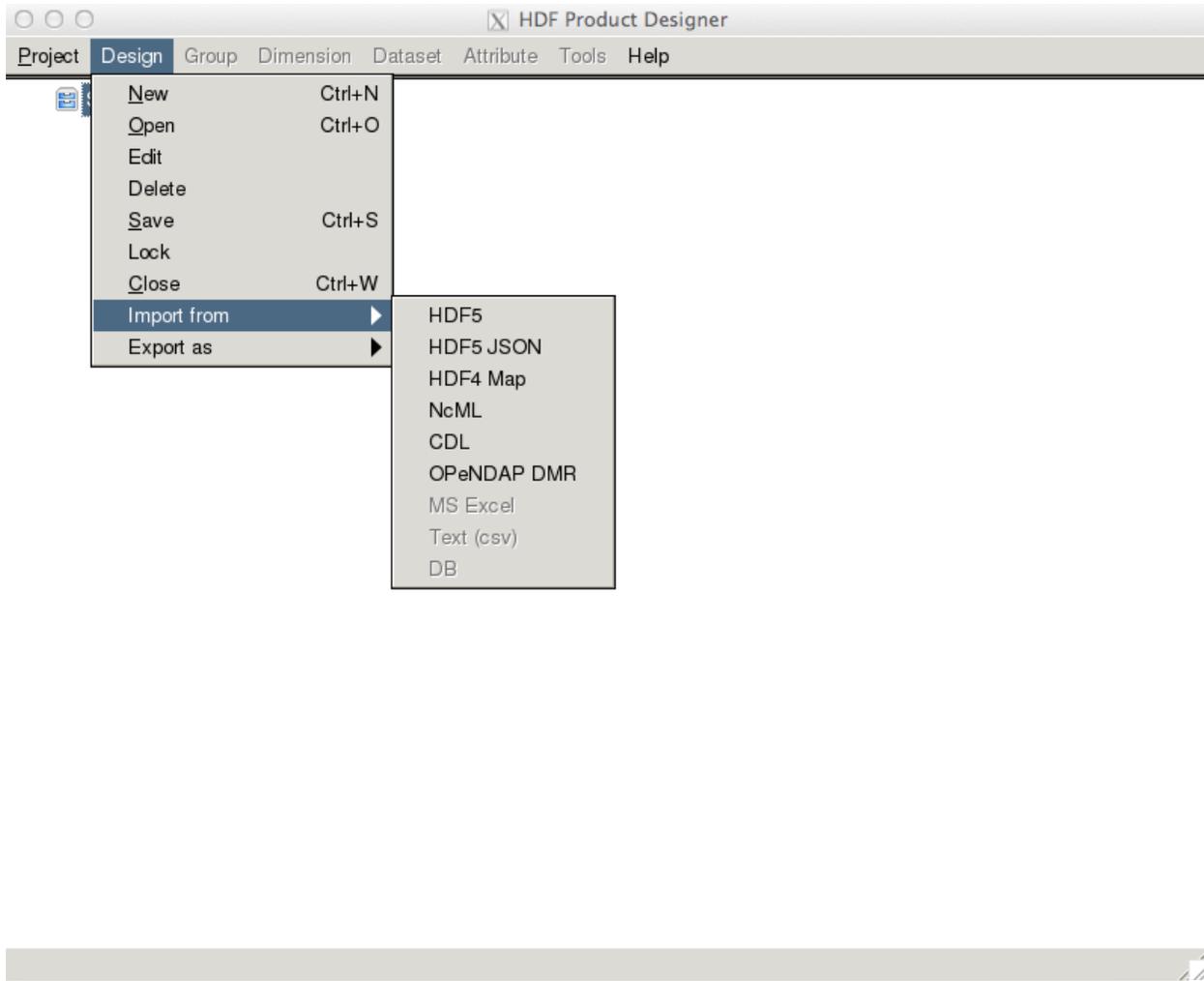
Fig. 5.5: HPD Desktop menu selection for importing design from various sources
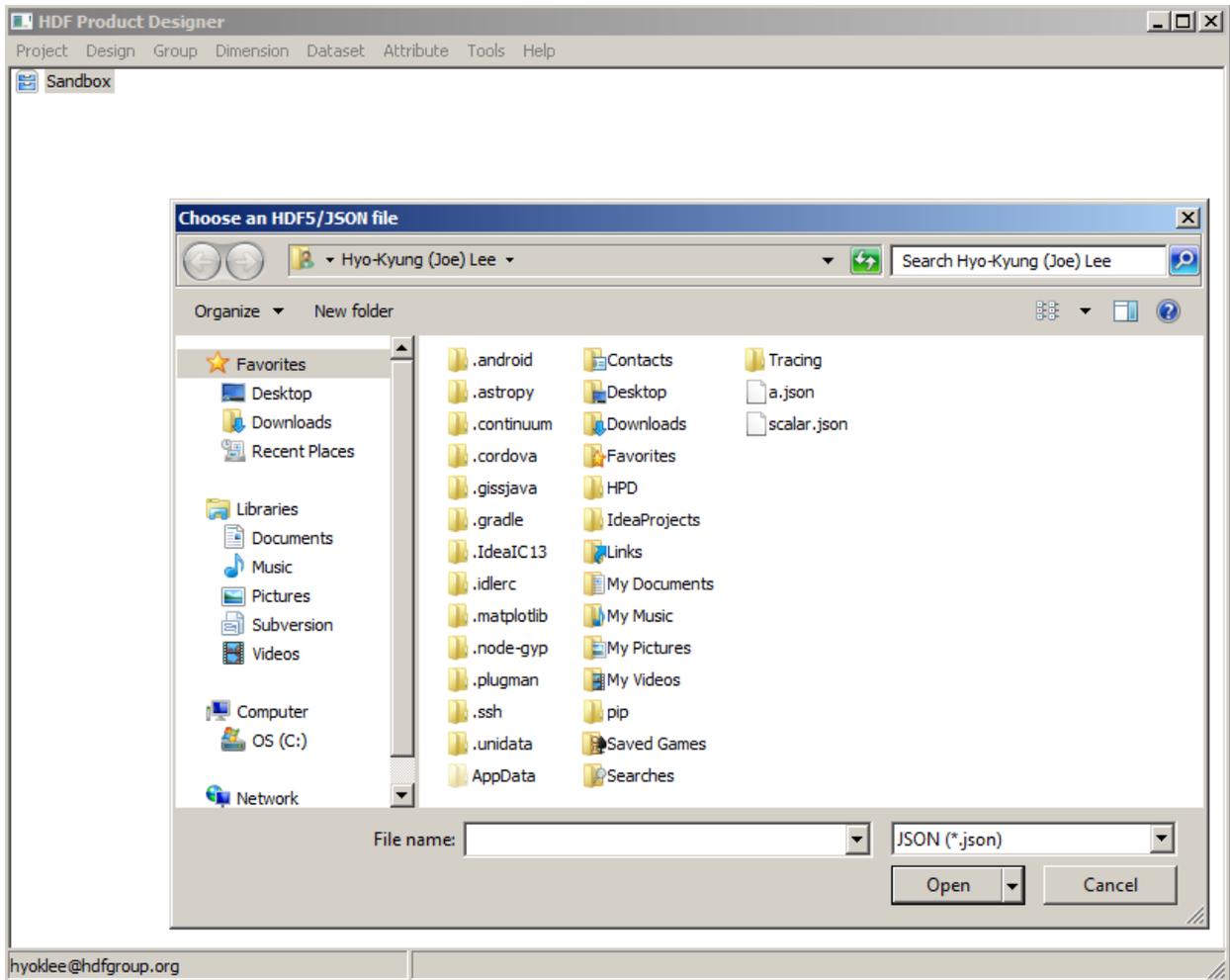
Fig. 5.6: HPD Desktop dialog box for opening an HDF5/JSON file

grows. If you don't see any change in the tree view, it means it has completed importing. You may want to scroll up to check your imported design.
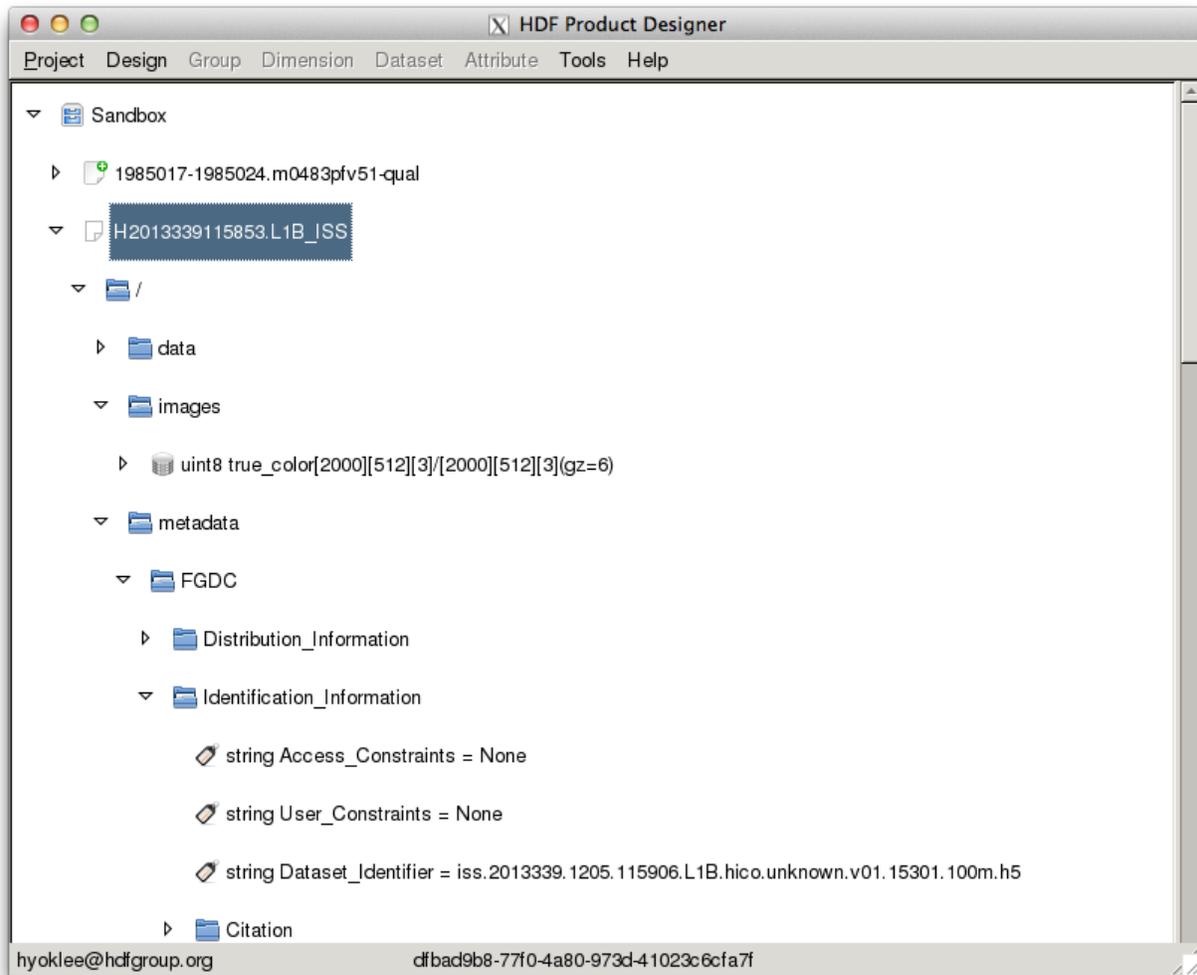
Fig. 5.7: Updated tree view in HPD Desktop during HDF5/JSON file import

5. If a parsing error occurs during the import, the design will be saved up to the point where the parsing error occurred. You can examine an error message from the wxPython stderr/stdout message window.

Note: The HDF5/JSON importer in HPD Desktop has some limitations. For example, it cannot handle references and custom datatypes. They will be ignored without any warning messages because they can make products inoperable with netCDF.

Importing datasets without layout and filter creation properties will make them chunked with the size equal to their size (the entire dataset is one chunk), and enable gzip (deflate) compression filter with level 6.

HDF4 XML

Importing designs from HDF4 files, for example existing NASA HDF-EOS2/HDF4 products, is possible using the `h4mapwriter` tool which generates HDF4 file content map. HDF4 file content maps are XML documents that describe file structure, metadata, and offsets and lengths of raw data. `h4mapwriter` is available from The HDF

Group project website.

To demonstrate the tool's capability we exposed it as a THREDDS server's data access service for sample NASA HDF4 products, as shown in Fig. 5.8.

Fig. 5.8: THREDDS web service for generating HDF4 file content map

If you click one of sample NASA products, you'll get **H4MAP** link under **Access** (Fig. 5.9).

Clicking the link will generate HDF4 map file in XML (Fig. 5.10).

Once the HDF4 XML representation of an existing HDF4 file using the above method is saved to a file, it can be imported by following the steps below.

1. Select project node from the tree view (Fig. 5.4).

2. Press **Design > Import from > HDF4 Map** from menu (Fig. 5.5).

3. A file selection dialog will appear. Select an XML file (Fig. 5.6).
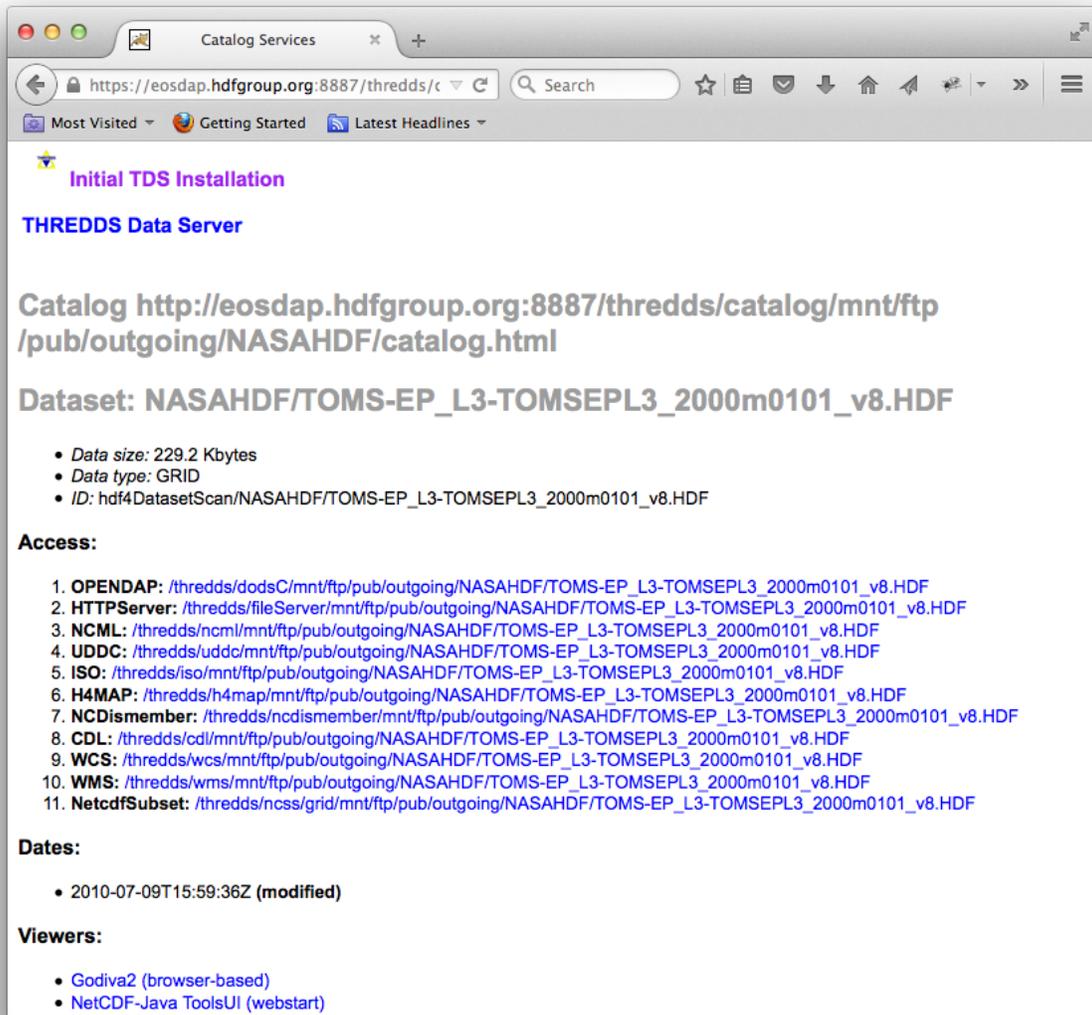
Fig. 5.9: THREDDS catalog page displaying various access methods for HDF4 file

Fig. 5.10: HDF4 XML output from THREDDS H4MAP access

4. Press the **Open** button. HPD Desktop will start loading the design from the file. It will update tree view dynamically during importing (:numref :*tree-import-hdf5*) and the tree view will scroll down automatically as the tree grows. If you don't see any change in the tree view, it means it has completed importing. You may want to scroll up to check your imported design.

5. If a parsing error occurs during the import, the design will be saved up to the point where the parsing error occurred. You can examine an error message from the wxPython stderr/stdout message window.

A useful feature of the HDF4 map writer is that it can merge a set of split file attributes (due to the attribute size limitation in HDF4) into a single big string value in XML. To support such merged ODL string attributes, HPD Desktop's HDF4 map importer can parse the entire merged ODL string and re-represent it as a hierarchy consisting of groups and attributes. Those re-constructed groups start with name `Metadata_` in the imported design. For example, Fig. 5.11 illustrates that the content of coremetadata is re-represented under the `Metadata_core` group.
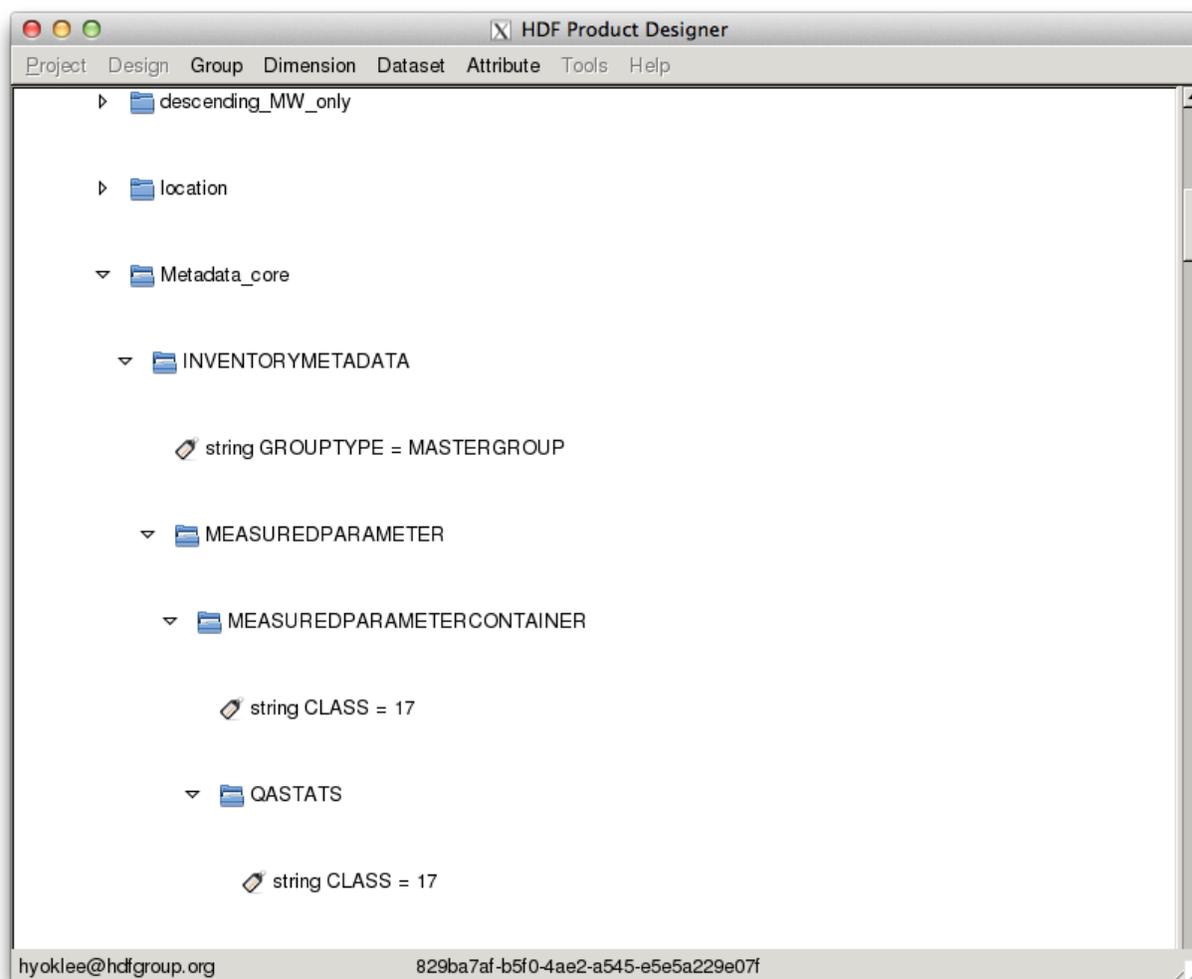
Fig. 5.11: An imported HDF-EOS2 file design with merged coremetadata

The current HDF4 map importer has several limitations. It cannot import Table (a.k.a Vdata in HDF4) data. You will see the following message in the console window:

```
INFO: Skipping unsupported Table Cloud Data
```

If importing Vdata is important, please try the *NcML* method.

Please note that HDF4 allows duplicate names under the same group but HDF5 does not (Fig. 5.12).
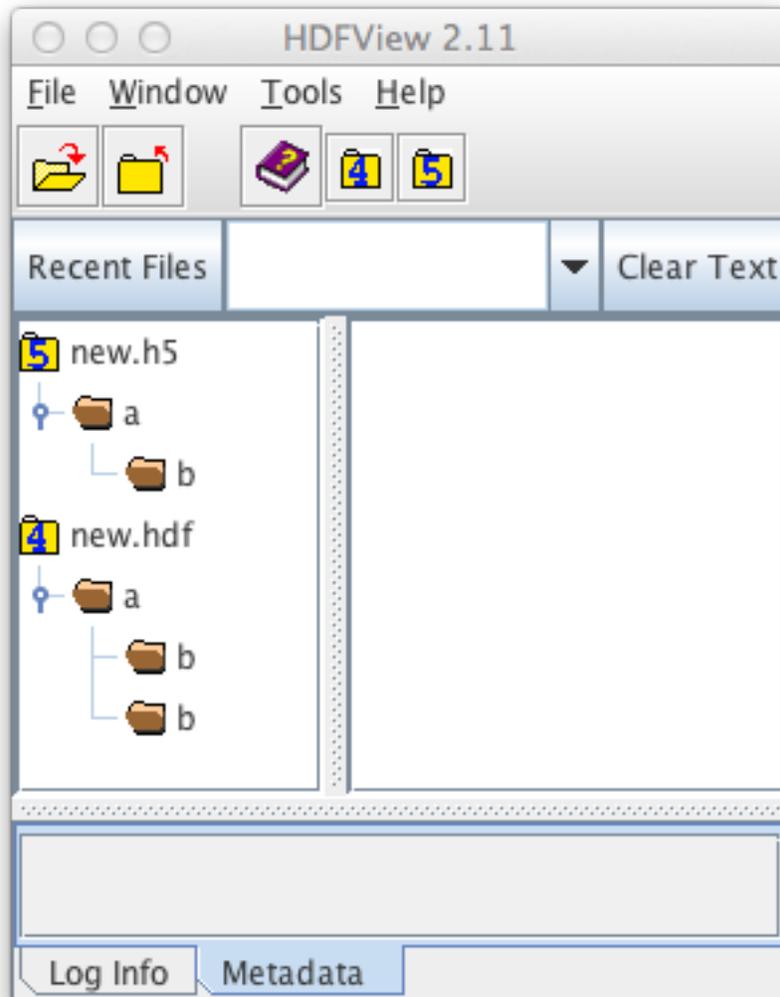
Fig. 5.12: HDFView displaying an HDF4 file that has two groups with the same name

In such case you'll see an error message in the console window:

```
hpdws.create_group():put:409
...
{"message": "Strip: Object with this name already exists.", ...}
```

HPD Desktop doesn't give any pop-up warning dialog messages for the above cases because they may happen quite often. However, the HDF4 map importer will give a pop-up notification dialog message when it replaces / character with _ for group/dataset/attribute names.

NcML

NcML is an XML representation of the metadata in a netCDF file (the information similar to the output of the ncdump -h). The netCDF-Java library from Unidata can read most NASA HDF4 and HDF5 files. Thus, the toolsUI, which is

based on netCDF-Java, can produce NcML easily from local HDF files. By clicking on the NcML tab, an HDF file can be opened and then saved as NcML (Fig. 5.13).
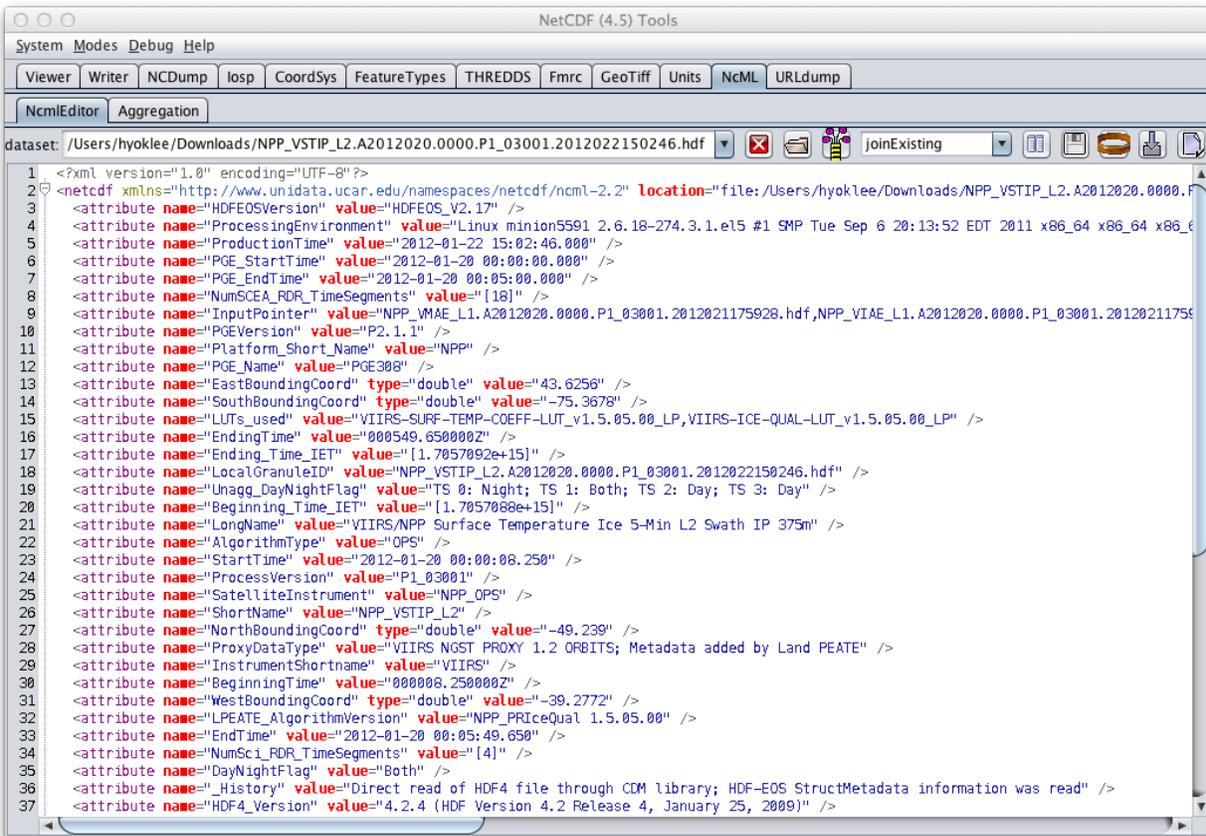
Fig. 5.13: ToolsUI displaying an HDF4 file content in NcML

The THREDDS data server, which is also based on NetCDF-Java, can generate NcML easily using a web browser. We provide a demo THREDDS server for many NASA products from different NASA Earth data centers (Fig. 5.14) for which NcML representation can be obtained. Please note that this generic THREDDS server is different from the custom THREDDS server that provides HDF5/JSON and HDF4 Map access in the previous sections (e.g., port number is different).

Clicking folders will lead you to a specific HDF product catalog that provides various access methods (Fig. 5.15).

Simply click the link next to **NCML** to get the NcML representation of NASA HDF products (Fig. 5.16).

Once obtained NcML representation of an existing HDF or netCDF file is saved to a file, importing it is done by following the steps below:

1. Select project node from the tree view (Fig. 5.4).

2. Press **Design > Import from > NcML** from menu (Fig. 5.5).

3. A file selection dialog will appear. Select an NcML file (Fig. 5.6).

4. Press the **Open** button. HPD Desktop will start loading the design from the file and save the design on HPD Server. It will update the tree view dynamically (Fig. 5.7) by scrolling down as the tree grows. If you don't see any change in the tree view, it means it has completed importing. You may want to scroll up to check your imported design.
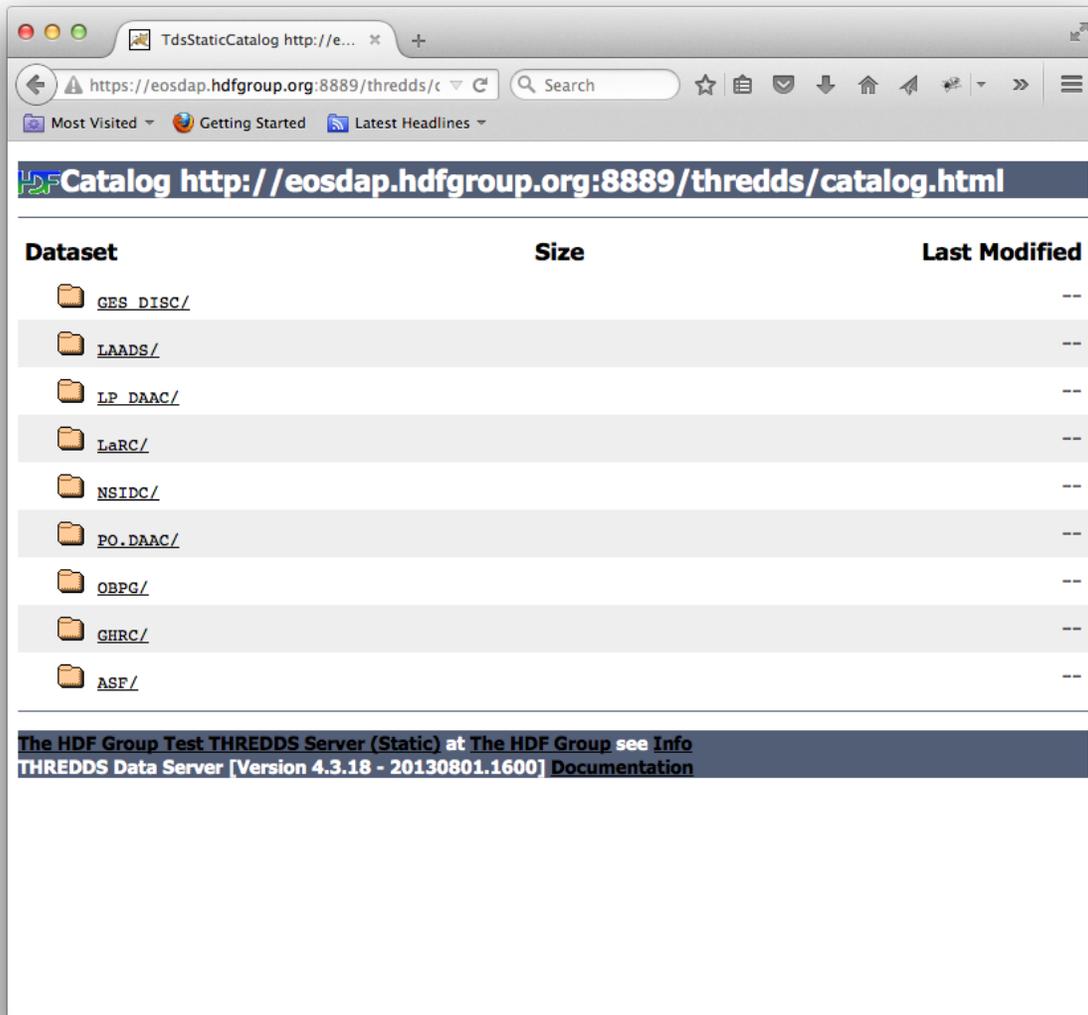
Fig. 5.14: THREDDS web service for generating NcML from HDF files
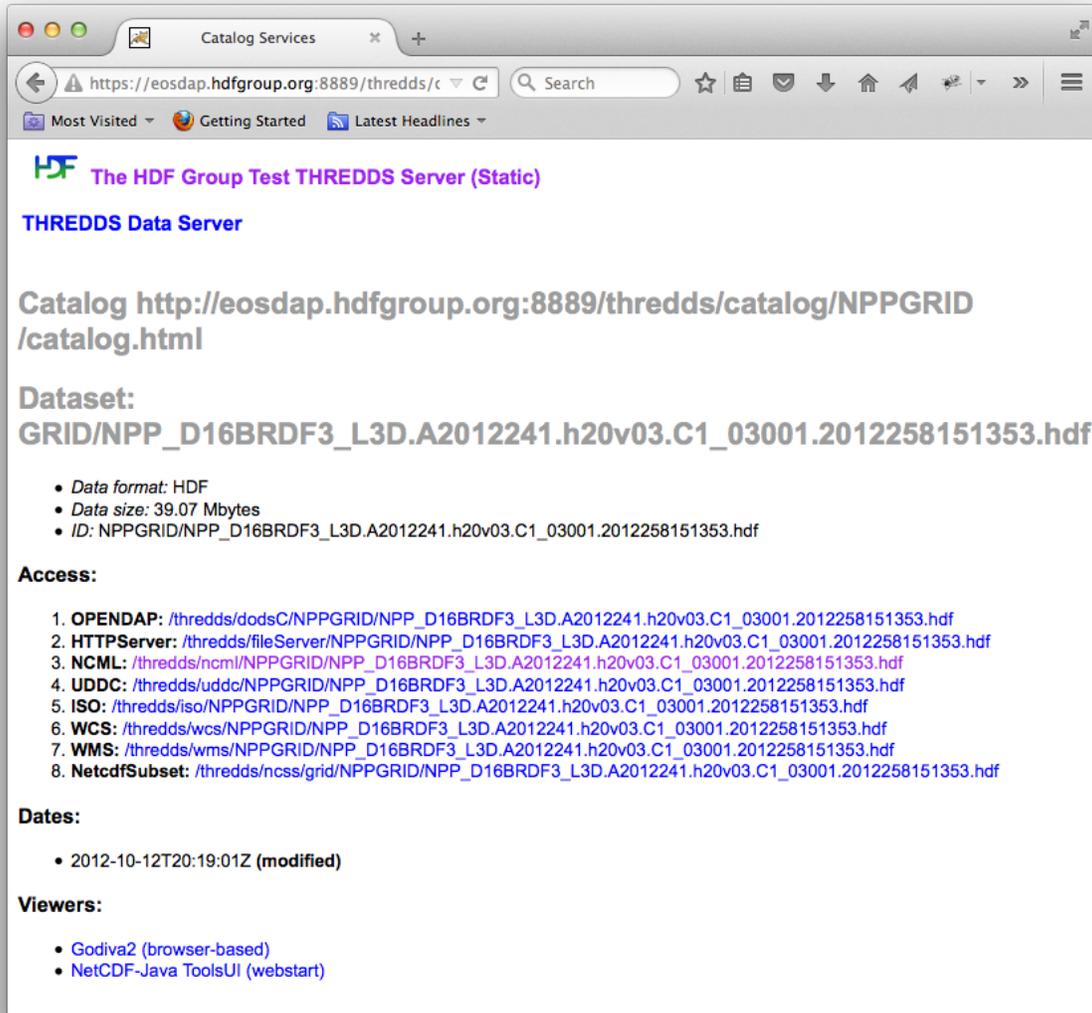
Fig. 5.15: THREDDS catalog page displaying various access methods including NcML

Fig. 5.16: NcML output from the THREDDS NCML service

5. If a parsing error occurs during the import, the design will be saved up to the point where the parsing error occurred. The error message(s) will be displayed from the wxPython stderr/stdout message window.

NetCDF CDL

The network Common data form Description Language (CDL) is a text representation of a netCDF file that is used as input and output by the netCDF utilities.

To import a CDL file, please follow the steps below:

1. Select a project from the tree view (Fig. 5.4).

2. Choose **Design > Import from > CDL** from the menu (Fig. 5.5).

3. From a file selection dialog pick a CDL file (Fig. 5.6).

4. HPD Desktop will start loading the design from the file. It will update the tree view dynamically (Fig. 5.7) by scrolling down as the tree grows. Importing is finished when the scrolling stops.

5. If a parsing error occurs during the import, the design will be saved up to the point where the parsing error occurred. The error message(s) will be displayed in the wxPython stderr/stdout message window.

OPeNDAP DMR

The OPeNDAP Dataset Metada Response (DMR) is an XML representation of the metadata from a remote data source hosted by an OPeNDAP server. DMR is based on the DAP 4.0 protocol. DMR is replacing the DAP 2.0-based DDX and keeps evolving. DMR support may break at any time if its specification changes in between this application's releases.

To import an OPeNDAP DMR, please follow the steps below:

1. Select a project from the tree view (Fig. 5.4).

2. Choose **Design > Import from > OPeNDAP DMR** from the menu (Fig. 5.5).

3. A text dialog box will appear. Enter an OPeNDAP server DMR URL (Fig. 5.2.1).
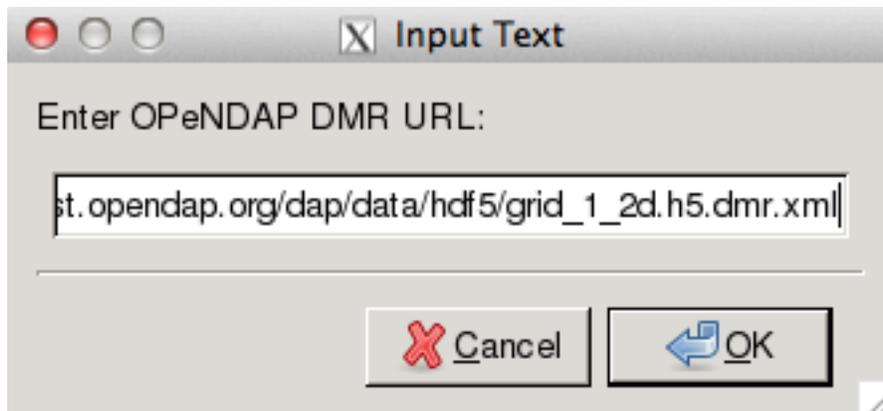
4. HPD Desktop will start loading the design from the server. It will update the tree view dynamically (Fig. 5.7) by scrolling down as the tree grows. Importing is finished when the scrolling stops.

5. If a network error or parsing error occurs during the import, the design will be saved up to the point where the parsing error occurred. The error message(s) will be displayed in the wxPython stderr/stdout message window.

5.2.2 Export

A design can be exported in several formats: HDF5 template file, HDF5/JSON, and as Python, MATLAB, IDL, or FORTRAN source code. The generated source code can produce the template file and, if modified to write actual data, final HDF5 products.

Note: Due to varying level of support for the HDF5 features in different programming languages not all source code exports may be available for some designs.

HDF5 Template File

Template files represent a design as an HDF5 file. They do not contain any real data but their structure, the names of groups/datasets/attributes, and metadata (attribute values) accurately represent the design. Therefore these files are ideal for quickly verifying the design by testing its template file with various display or visualization software, for example HDFView.

To generate an HDF5 template file from a design, please follow the steps below:

1. Select the design item in the tree first (Fig. 5.17).

2. Press **Design > Export as > HDF5** from menu (Fig. 5.18).

3. You will be asked to provide a name for HDF5 file. By default, it uses the same design name with extension `.h5`.

4. Press the **Save** button. HPD Desktop will send a request to HPD server to a new HDF5 file from design and start downloading the file as soon as it is ready.

5. If exporting fails, the pop-up window will display the error message (Fig. 5.20).

6. A pop-up window will confirm successful export and show the full file path to the template file.

HDF5/JSON

HPD Desktop can export a design as HDF5/JSON.

1. Select a design item in the tree (Fig. 5.17).

2. Press **Design > Export as > JSON** from menu (Fig. 5.18).

3. If exporting fails, the pop-up window will display the error message (Fig. 5.20).

4. A pop-up window will confirm successful export and show the full file path to the exported file.

HDF5/JSON files can be edited with any text editor. This makes it possible to perform certain editing operations that are currently not yet supported in the HPD Desktop, such as bulk search and replace. The modified file can then be imported back or turned into an HDF5 file with the jsontoh5.py.

Python Source Code

HPD Desktop can export a design as Python source code. The code requires the h5py package.

1. Select a design item in the tree (Fig. 5.17).

2. Press **Design > Export as > Python** from menu (Fig. 5.18).

3. If exporting fails, the pop-up window will display the error message (Fig. 5.20).

4. A pop-up window will confirm successful export and show the full file path to the exported file.
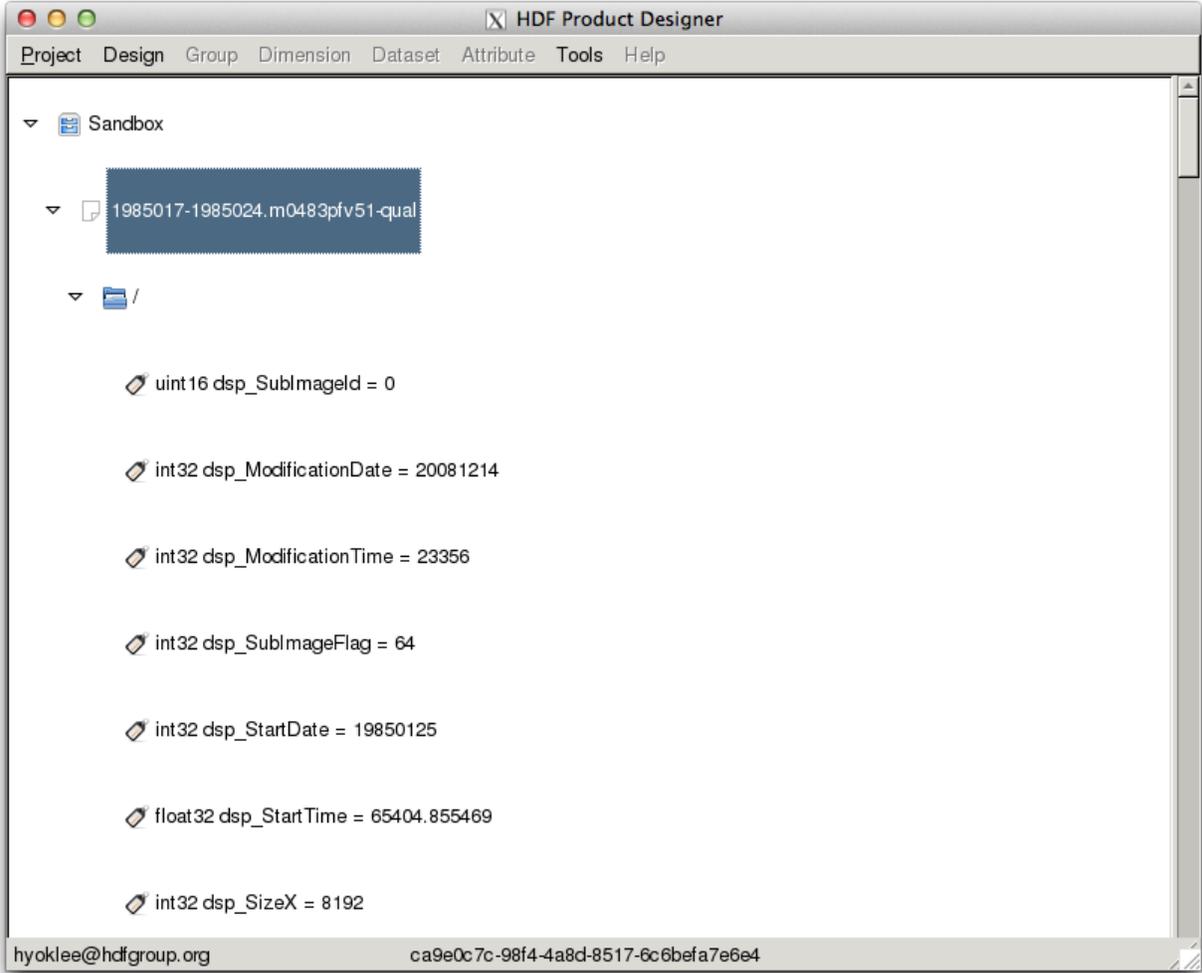
Fig. 5.17: HPD Desktop tree control view with a selected design
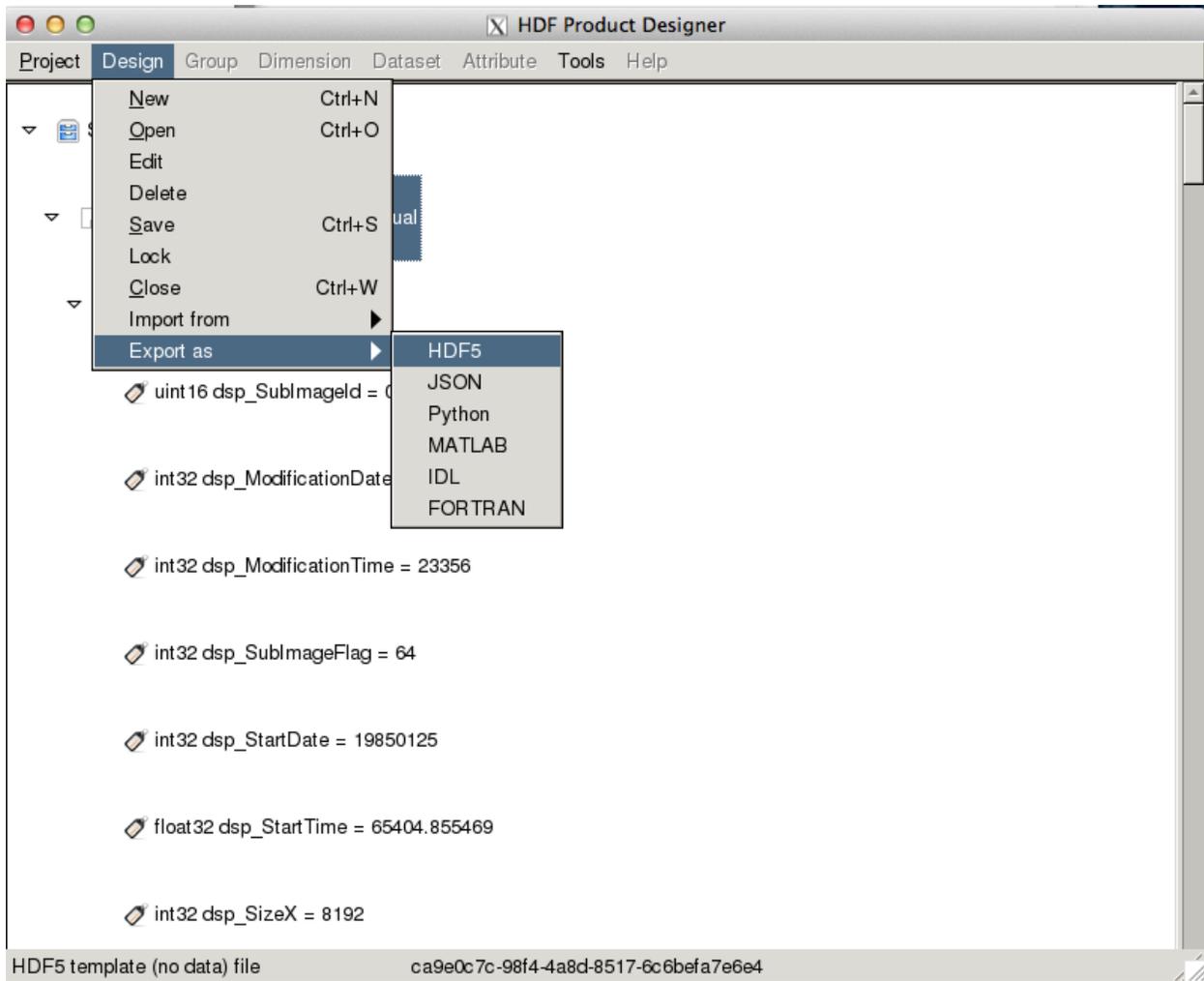
Fig. 5.18: HPD Desktop menu selection for exporting design as HDF5 template file
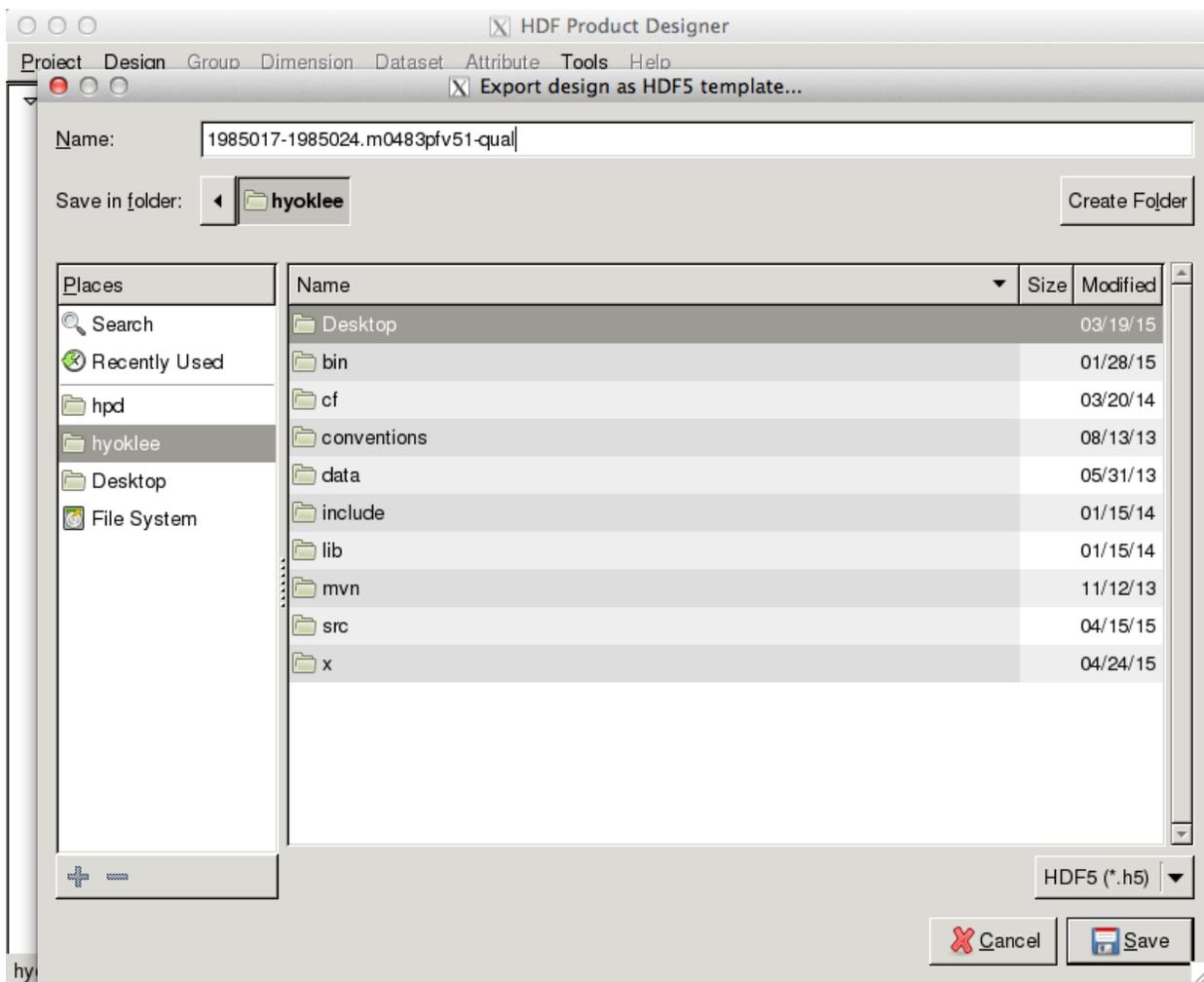
Fig. 5.19: HPD Desktop dialog box for saving design as HDF5 template file
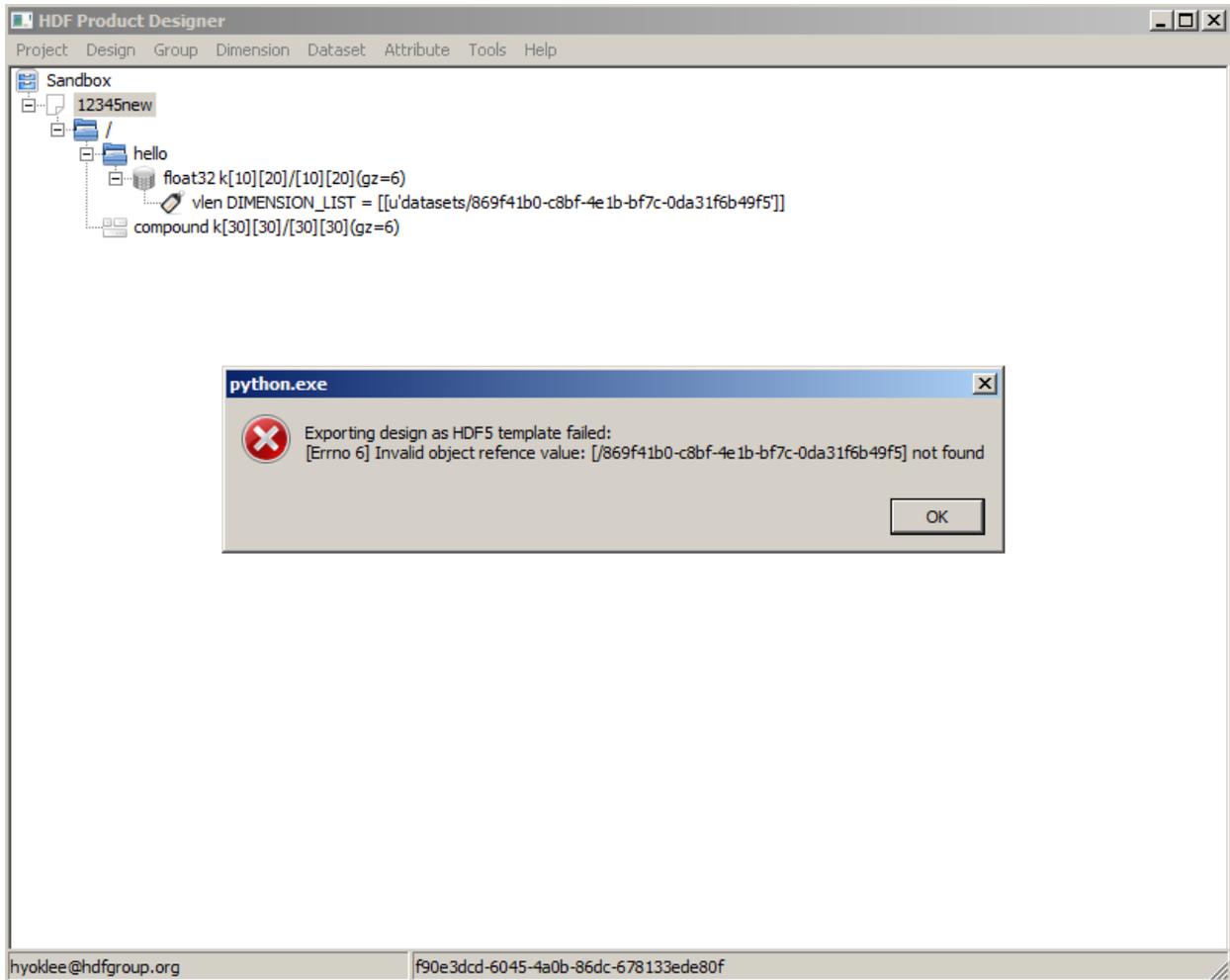
Fig. 5.20: HPD Desktop dialog message box for download failure

The Python code was tested with Python 2.7 and h5py versions 2.4 and 2.5.

IDL Source Code

HPD Server can generate IDL code using IDL's HDF5 routines.

1. Select a design item in the tree (Fig. 5.17).

2. Press **Design > Export as > IDL** from menu (Fig. 5.18).

3. If exporting fails, the pop-up window will display the error message (Fig. 5.20).

4. A pop-up window will confirm successful export and show the full file path to the exported file.

Note: IDL source code export will fail if:

- 8-bit signed integer (`int8`) or variable string (`vlstring`) datatypes are used.

MATLAB Source Code

HPD Server can generate MATLAB code using MATLAB's HDF5 routines.

1. Select a design item in the tree first (Fig. 5.17).

2. Press **Design > Export as > MATLAB** from menu (Fig. 5.18).

3. If exporting fails, the pop-up window will display the error message (Fig. 5.20).

4. A pop-up window will confirm successful export and show the full file path to the exported file.

Note: MATLAB source code export will fail if:

- Any dataset in a design has the total number of elements greater than 2^{48} - 1.

FORTRAN Source Code

Warning: The functionality and support for HDF5 features of this source code generator currently lags significanly behind the others so it is not recommended for any serious use.

HPD Server can generate FORTRAN 90 code using HDF5 FORTRAN APIs.

1. Select a design item in the tree first (Fig. 5.17).

2. Press **Design > Export as > FORTRAN** from menu (Fig. 5.18).

3. If exporting fails, the pop-up window will display the error message (Fig. 5.20).

4. A pop-up window will confirm successful export and show the full file path to the exported file.

To compile the exported FORTRAN code, the HDF5 library is required and the HDF5 library must be compiled with these flags: `--enable-fortran` and `--enable-fortran2003`.

5.3 Group

You can add new group using menu items under **Group**. You can edit or delete an existing group using menu items. You can also edit group directly by clicking and editing from the tree view. As soon as you change group, it will be saved on HPD Server.

5.4 Dimension

A dimension is attached by dragging & dropping it onto a dataset. The `DIMENSION_LIST` and `REFERENCE_LIST` attributes of the dataset and its dimension scale will be updated accordingly.

To detach a dimension, drag & drop dataset onto that dimension. Deleting a dimension will detach that dimension from the dataset automatically.

5.5 Dataset

You can add new dataset using menu items under **Dataset**. You can edit or delete an existing dataset using menu items. As soon as you change dataset, it will be saved on HPD Server.

The shape of dataset must be a set of integers separated by comma. For example, to create a dataset with 2x4 shape like **dset[2][4]**, please type **2,4** in the shape text box of New Dataset dialogue box. If you leave the shape empty, a scalar dataset will be created. To specify unlimited dimension size, use `*`. For example, to create a dataset with unlimited x 4 shape, please type ***,4**.

To create a compound dataset, create a dataset with compound type first. Then, select the dataset and create another dataset under it. The child dataset becomes the field of compound dataset.

When you create a dataset, you can specify compression filters such as gzip, storage layout (chunk shape), and fill value. By default, HPD users gzip compression with level 6 and chunked storage. If you leave chunk shape blank, chunk shape will be same as dataset shape.

In tree view, storage layout is indicated with '/' after dataset's shape. If '/' doesn't appear, it means contiguous storage is used. If an array style shape (e.g., [2][4]) is specified after '/', it means chunking storage layout is used. Single '/' means compact storage layout is used. The following table summarizes the storage layout notation in tree view.

Example	Meaning
int8 d[10][20]	Contiguous storage layout
int8 d[10][20]/[2][4]	Chunking storage layout with shape 2x4
int8 d[10][20]/	Compact storage layout

Filters and the fill value are indicated within parenthesis and they are separated by commas. For example, `int8 dset[10](fv=-1,gz=6)` means: gzip filter with compression level 6, with -1 as the fill value. The following table explains the two letter notation of filters and fill value in dataset creation property.

Notation	Meaning	Value Required?
fv	fill value	Yes
gz	gzip	Yes - [1,9]
sh	shuffle	No
fl	fletcher32	No

Since interoperability is most important, fill value in dataset creation property will be overwritten by `_FillValue` attribute's value if CF or NUG conventions are active and a `_FillValue` attribute is defined for a dataset. The same applies when importing designs from NcML and HDF4 map if either CF or NUG conventions are in effect.

5.6 Attribute

All operations on HDF5 attributes are available from the **Attribute** menu. Please note that creating, deleting, or modifying any attribute is immediately recorded.

If you want to specify multiple values for a single attribute, you can surround the values in square braket and separte them with comma (Fig. 5.21).
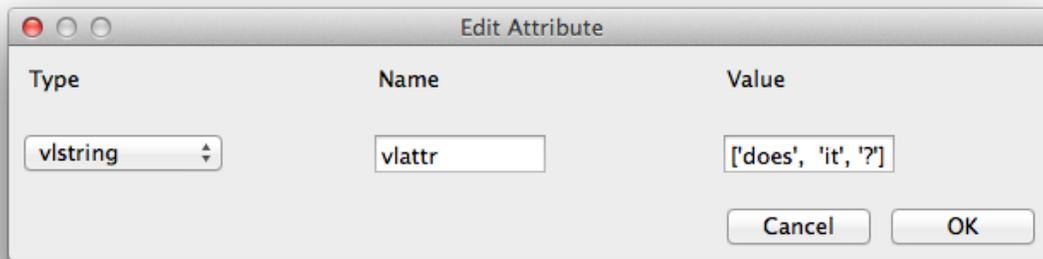
Fig. 5.21: How to add multiple values for a single attribute

5.7 Convention Support

- When you create a new design, you can pick a set of conventions that you want to use. However, you cannot choose both CF and NUG at the same time.
- If you select convention, a toolbar will appear automatically.
- Using conventions will activate CLIPS expert engine except for HDF-EOS convention.
- To apply convention, you can select individual root group or dataset and press a button in toolbar.
- For HDF-EOS convention, you need to select the root group before you press any Point/ZA/Profile/Swath/Grid button.
- Choosing custom convention allow you to load your own CLIPS code. You may want to download and edit a sample CLIPS code that modifies the default values of ACDD convention and to test custom convention.

5.7.1 CLIPS

Unless you're comfortable with CLIPS programming, we don't recommend you to modify existing CLIPS rules. If you dare to customize CLIPS rules, please try them first on generic CLIPS engine, not PyCLIPS engine used in HPD Desktop because a subtle mistake in CLIPS programming is hard to debug with HPD Desktop. HPD Desktop doesn't provide enough debugging message on system console. Although PyCLIPS generates **my_trace.log**, it may not be useful for debugging either. The way PyCLIPS works in a packaged environment is a still mystery and we are still investigating its limitation by trial-and-error.

The current user interface is only focused on adding attributes to groups and datasets suggested by CLIPS rules. However, the current rule set is more sophisticated than what HPD Desktop GUI utilizes. For example, CLIPS rules can handle dimensions but HPD Desktop doesn't utilize them yet. We will address them in future versions.

Therefore, the only thing that HPD Desktop tested is to customize global name values using *defglobal* when ACDD/NUG/CF conventions are used. We don't know what will happen to HPD Desktop by writing a custom *defrule*, so it's entirely up to your hacking ability. If you have trouble with using PyCLIPS in HPD, please figure it out yourself from source code or please support us by providing more funding if a serious effort needs to be made for custom conventions.

5.8 Tools

HDF Product Designer provides tools for validating design and generating documentation.

5.8.1 Validation

HDF Product Designer provides several tools for testing design's interoperability. Currently these tools rely on the web services provided by the THREDDS and Hyrax data servers. We collectively refer to these interoperability tools and services as *HPD Online* to distinguish them from those of the HPD Server.

HPD Online services can be invoked from the **Tools** menu at any time while working on a design (Fig. 5.22).

The selected design will be exported as an HDF5 template file and this file will be made available to the HPD Online tools. An error message will be displayed if this process fails for any reason (Fig. 5.23).

Finally, the chosen HPD Online tool will run against the design's HDF5 file and report its result back via a web browser window opened by the HPD Desktop.

This workflow can be repeated as many times as needed to either correct any reported error or maximize design's interoperability as defined by the tool.

For requesting addition of new tools to HPD Online please contact eoshelp@hdfgroup.org.

CDL

The network Common data form Description Language (CDL) is a text representation of a netCDF file that is used as input and output by the netCDF utilities, which are based on the netCDF C library. These netCDF utilities are very useful for interoperability testing so HPD Online supports obtaining a CDL output of a design.

The CDL output from HPD Online is directly generated by the ncdump tool. Correct display of the design's content assures it will be interoperable with any netCDF-C based tools.

- To validate a design with the `ncdump -h` command use the **Tools > Validate > CDL** menu.

- If no errors, the CDL output will be displayed in a web browser window (Fig. 5.24).

Get as netCDF-3

netCDF-3 is a widely supported file format especially by the legacy tools. NASA data centers allow users to download HDF files in the netCDF-3 format on demand.

HPD Online connects the Hyrax service which enables download of a design as a netCDF-3 file. This can benefit users to ensure the files generated from the design will be interoperable with software which relies on the netCDF-3 format.

- To validate a design with netCDF-3 tools use the **Tools > Validate > Get as netCDF-3** menu.
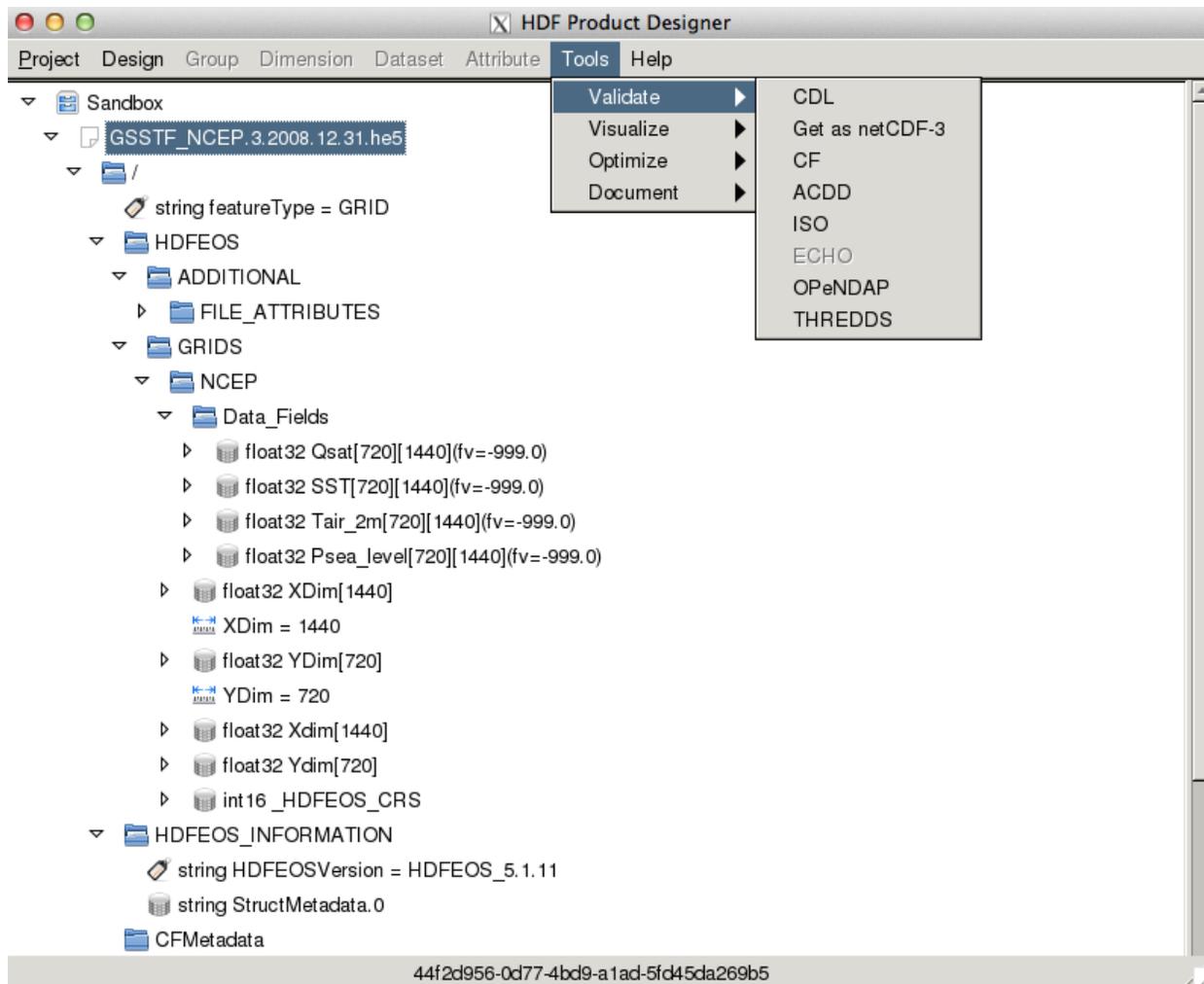
Fig. 5.22: HPD Desktop Tools menu for HPD Online tools
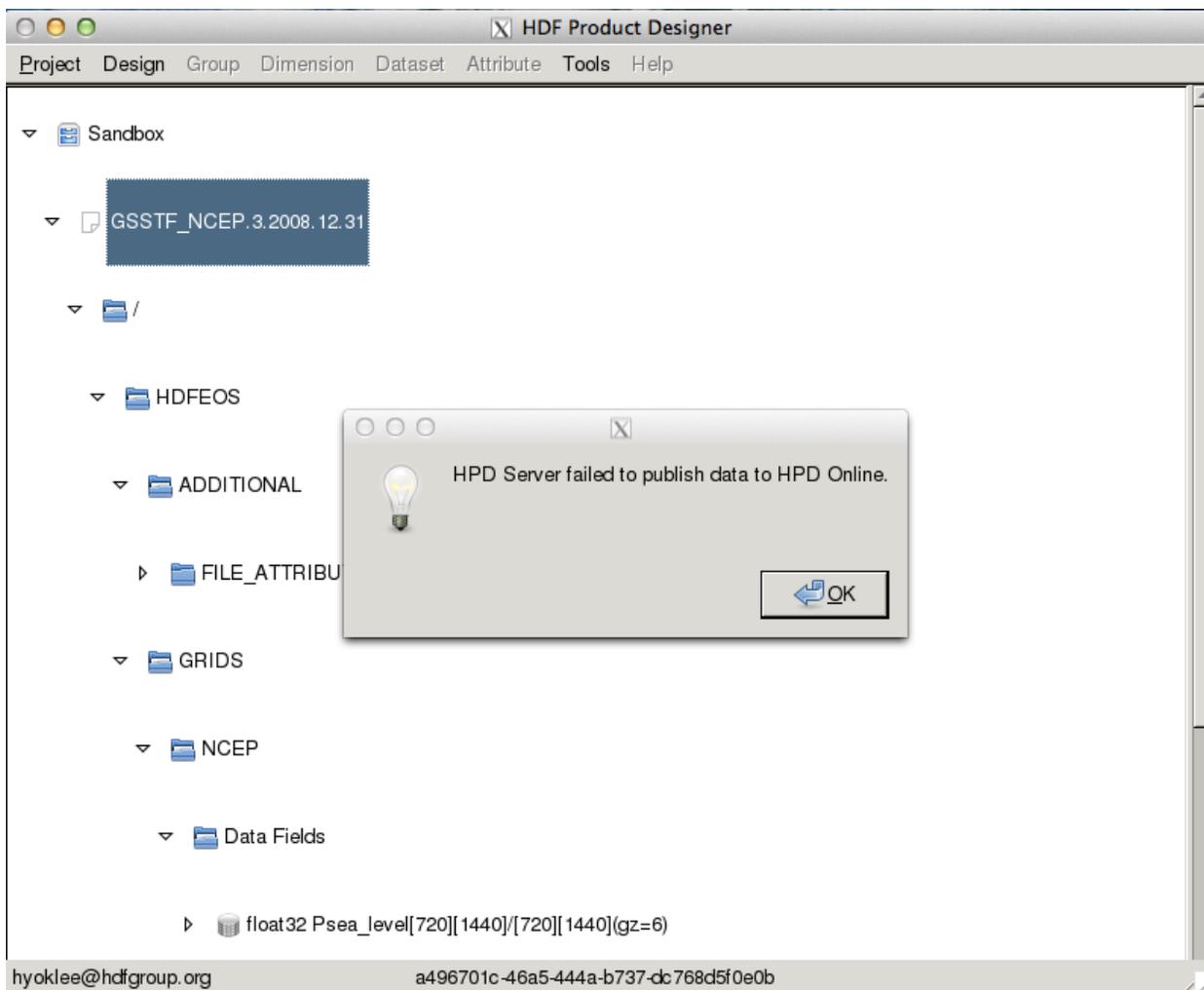
Fig. 5.23: Error message for publishing HDF5 file on HPD Online
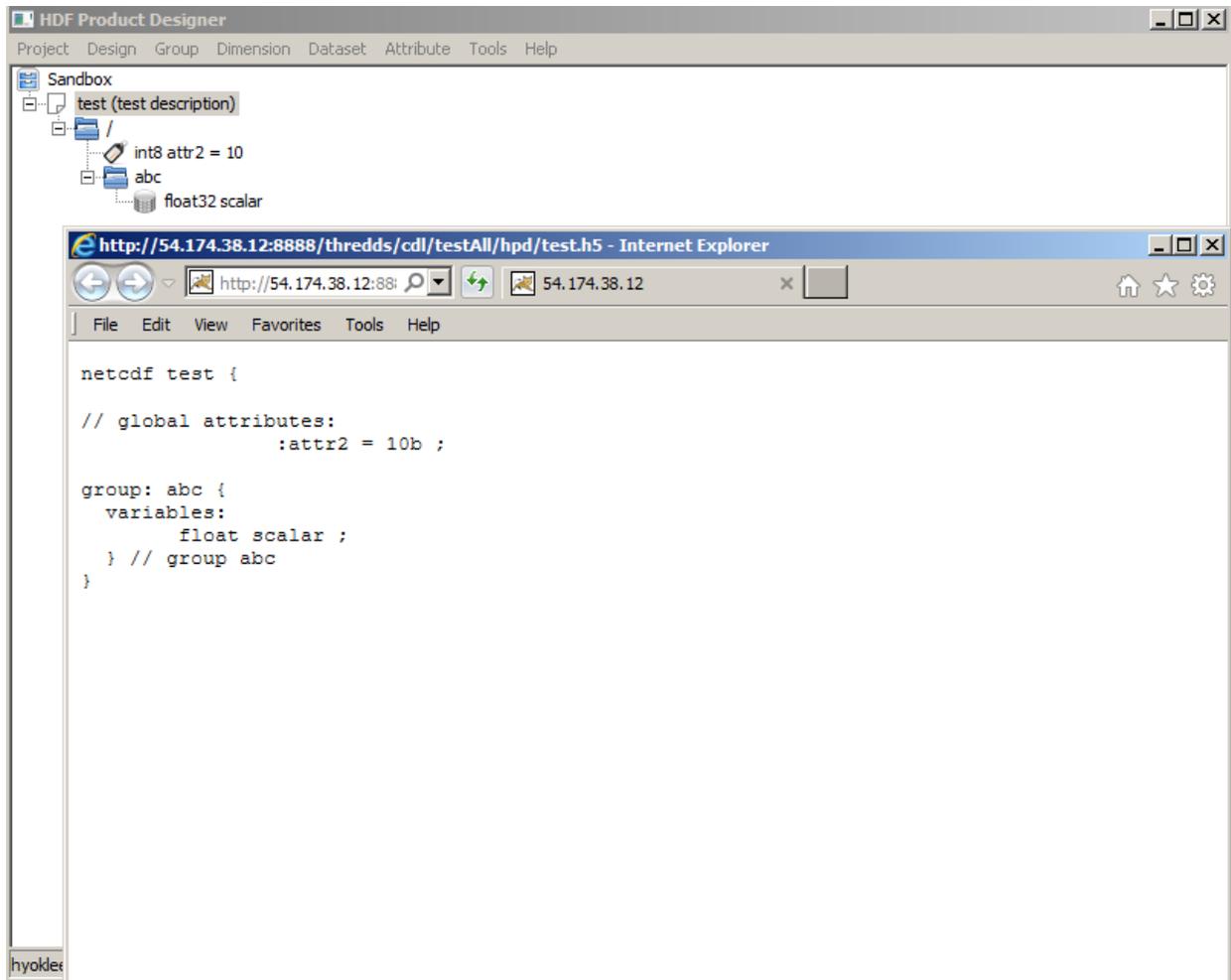
Fig. 5.24: CDL output from HPD Online

- If no errors, HPD Desktop will prompt for a file name and save the netCDF-3 content (Fig. 5.25).
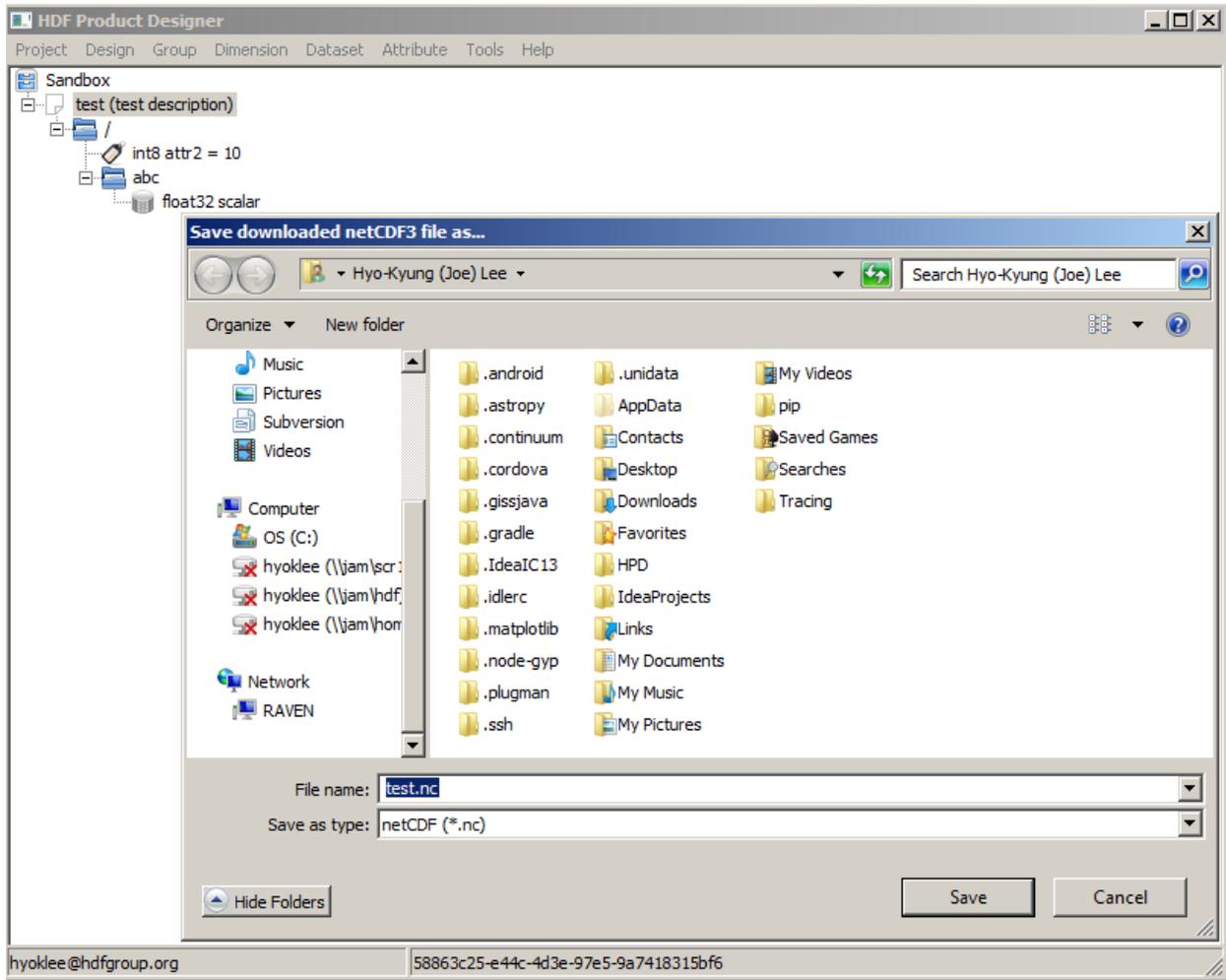

Fig. 5.25: HPD Desktop dialog box for saving design as netCDF-3 file

CF

The Climate and Forecast convention (CF) is one of the NASA ESDIS Standards Office (ESO) approved metadata standards. The convention is quite important because there are many data access software that utilize it to geolocate, plot, or subset data correctly. A subtle mistake in following the CF convention can make CF tools fail to load data.

HPD Online can check the CF convention compliance by using the ncdismember service powered by the CFChecker via the THREDDS server. The `ncdismember` tool will split the contents of a file along its HDF5 groups into multiple files. The CFChecker will then check each *dismembered* file and report on its CF compliance.

- To validate a design with CFChecker use the **Tools > Validate > CF** menu.
- If no errors, a CF compliance report will be displayed in a web browser window (Fig. 5.26).
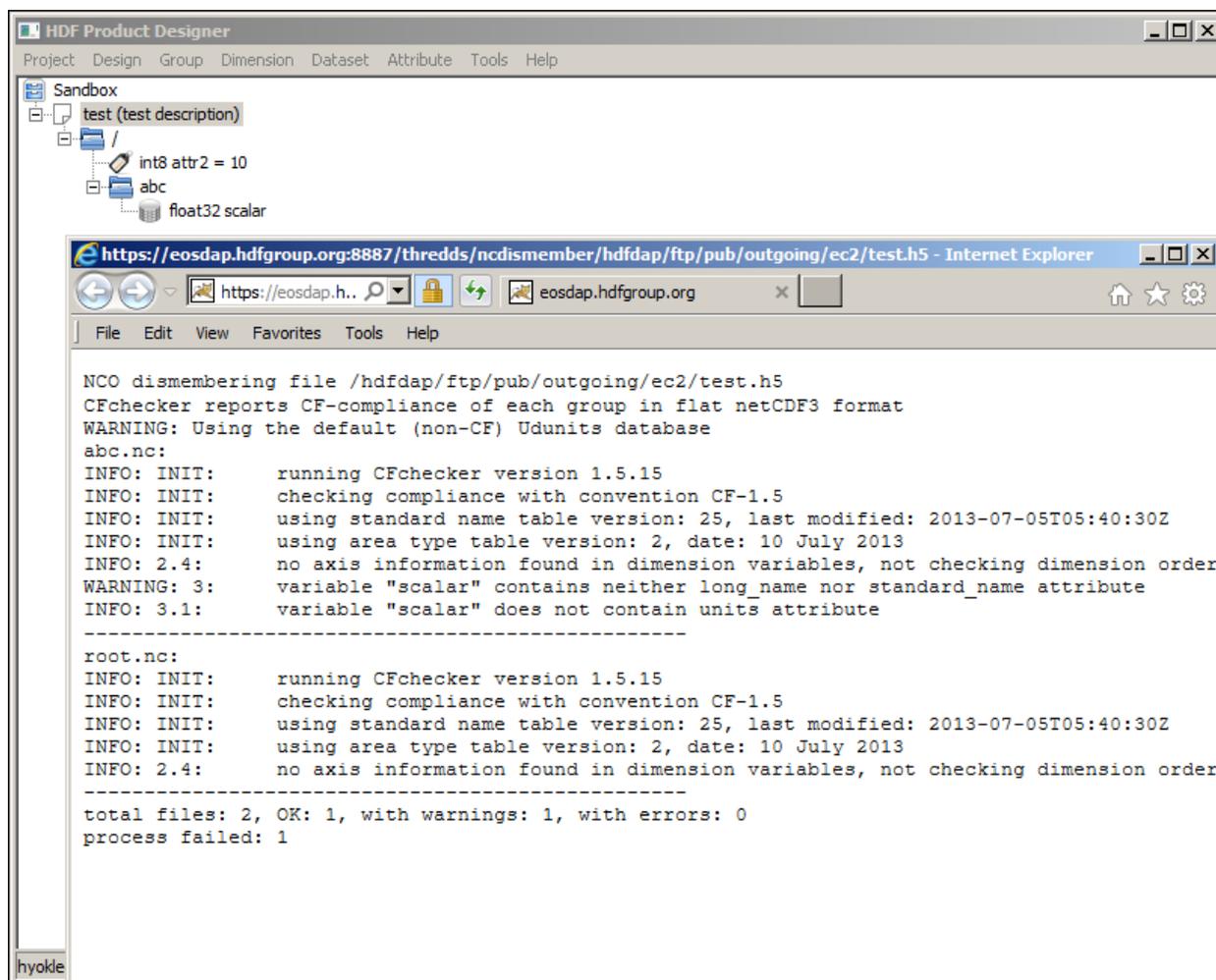
Fig. 5.26: CF compliance report from HPD Online. Note how the `test.h5` file is split into the `abc.nc` and `root.nc` files in the report.

ACDD

The Attribute Convention for Data Discovery ACDD is a set of global (root group) attributes that provide metadata for improving the data's discoverability. It works very well with other Earth Science conventions because there is no overlap in the metadata content between them. Metadata cataloging tools use ACDD attributes to extract metadata from files to support search or translation into other metadata formats such as DIF, FGDC, or ISO 19115. We highly recommend using the ACDD attributes in your designs.

HPD Online utilizes THREDDS UDDC service. It provides an evaluation of how well the metadata contained in a design conforms to the ACDD.

- To validate a design with the ACDD checker use **Tools > Validate > ACDD** menu.

- If no errors, a report will be displayed in a web browser window (Fig. 5.27).
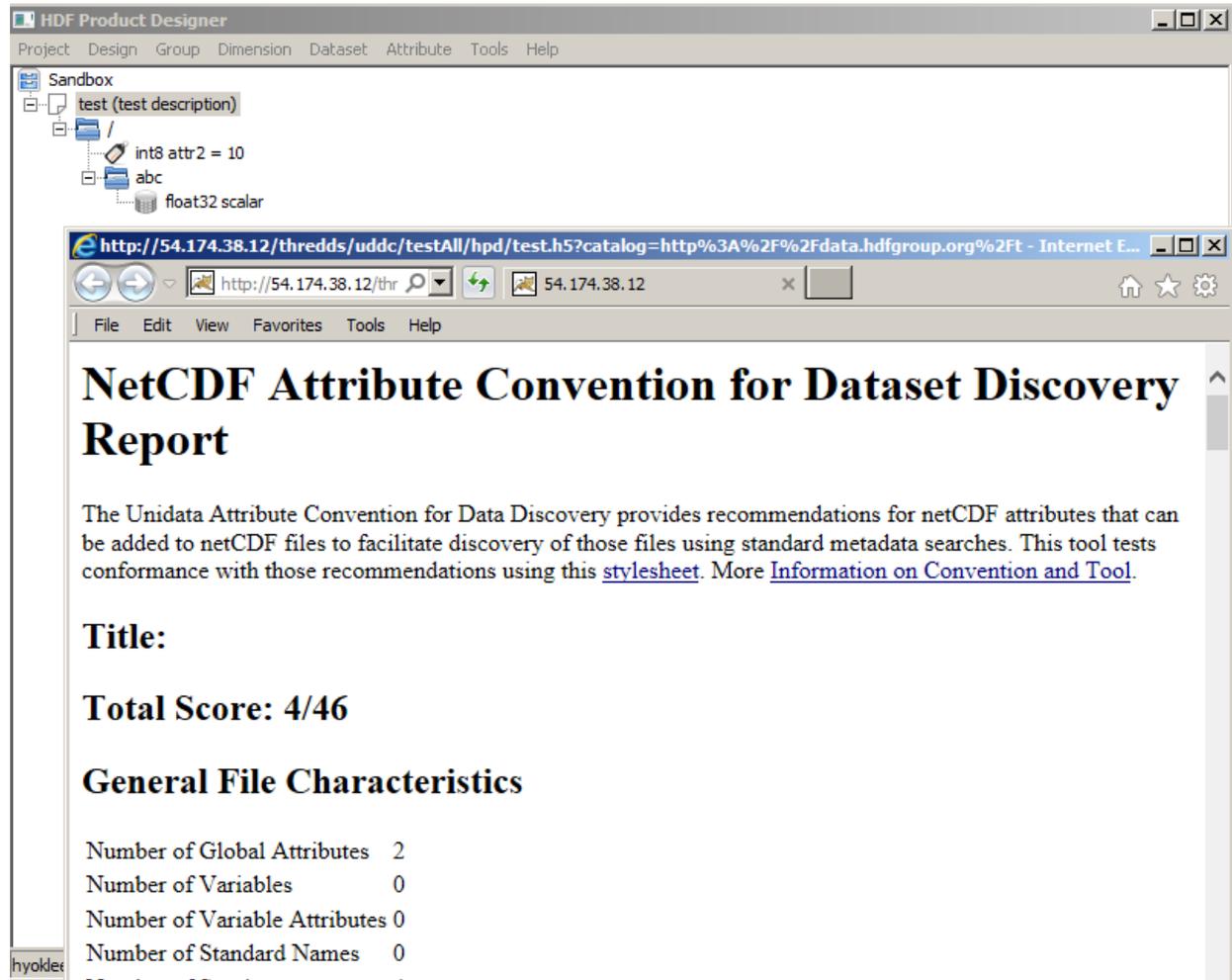
Fig. 5.27: ACDD report from HPD Online

ISO

International Organization for Standardization (ISO) has published Geographic Information Metadata standard 19115. NASA conventions and best practices for ISO 19115 are currently under development. HPD Online simply utilizes THREDDS's ISO service to provide ISO 19115 metadata description of a design.

- To obtain ISO metadata record of a design use **Tools > Validate > ISO** menu.

- If no errors, the design's ISO metadata will appear in a web browser window (Fig. 5.28).
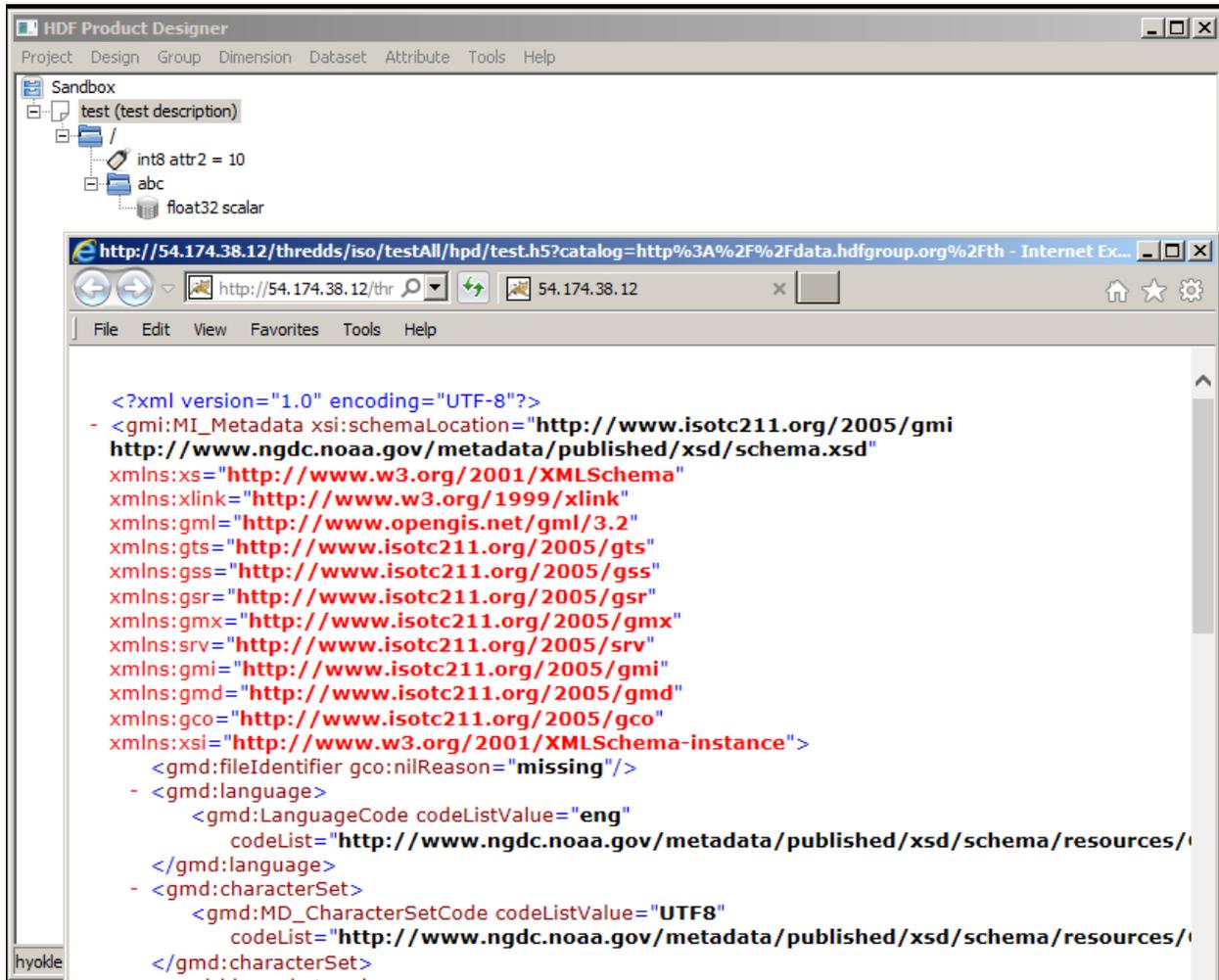
Fig. 5.28: ISO metadata output from HPD Online

OPeNDAP

OPeNDAP stands for "Open-source Project for a Network Data Access Protocol". OPeNDAP is both the name of a non-profit organization and the commonly-used name of a protocol which the OPeNDAP organization has developed. Hyrax is the official OPeNDAP server by OPeNDAP organization. Hyrax can serve both HDF4 and HDF5 data using the HDF4 OPeNDAP handler and the HDF5 OPeNDAP handler.

The HDF5 OPeNDAP handler is a software that can be used to access HDF5 data via OPeNDAP's Data Access Protocol. The HDF Group has designed and implemented this software. The handler can support OPeNDAP's visualization client tools that access NASA HDF-EOS5 (OMI, HIRDLS, MLS, TES and MOPITT) and some HDF5 (GPM, OBPG and some MeASUREs) products.

The HDF5 handler provides two options: CF and generic. The CF option tries to make HDF5 products follow the CF convention. The generic option serves HDF5 data as-is, which can break many CF tools. HPD Online uses HDF5 handler with CF option enabled to test the interoperability of a design.

- To validate a design with OPeNDAP server (Hyrax) use the **Tools > Validate > OPeNDAP** menu.

- If no errors, an OPeNDAP HTML form will be displayed in a web browser window (Fig. 5.29).
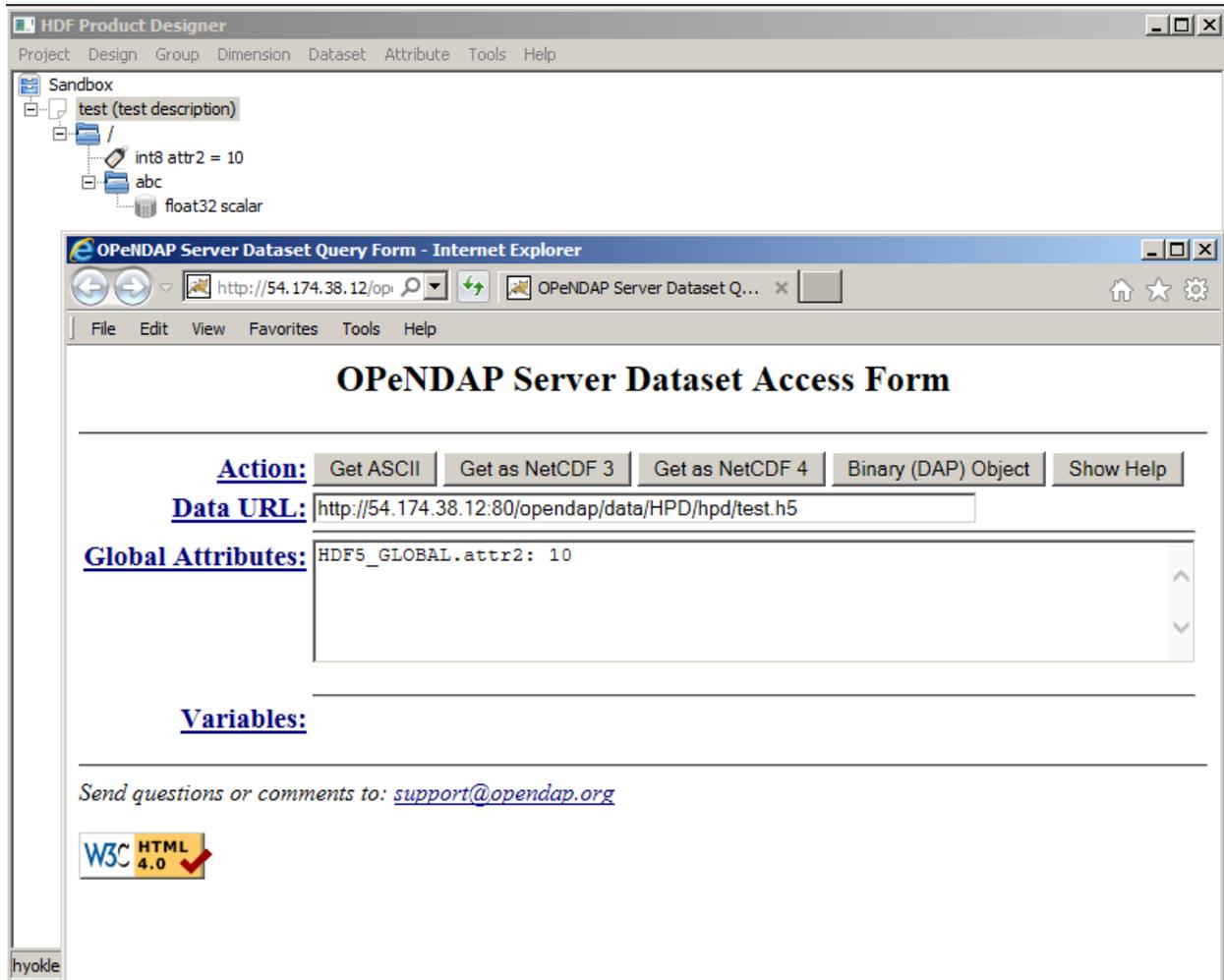
Fig. 5.29: OPeNDAP server dataset access form from HPD Online. Note that the group hierarchy *abc* is flattened as part of the variable name `_abc_` by CF- enabled HDF5 handler because the CF convention do not have the notion of group hierarchy.

THREDDS

The Unidata's THREDDS Data Server (TDS) is a web server that provides metadata and data access for scientific files, using OPeNDAP, OGC Web Mapping Service (WMS) and Web Coverage Service (WCS), HTTP, and other remote data access protocols. The TDS is implemented as a Java servlet on top of the netCDF-Java library. Therefore, TDS is a good testbed for all netCDF-Java applications such as Panoply and IDV. If THREDDS can access and interpret a design's HDF5 file correctly and other netCDF-Java tool can, too.

- To validate a design with THREDDS use the **Tools > Validate > THREDDS** menu.

- If no errors, a THREDDS dataset access page will be displayed in a web browser window (Fig. 5.30).
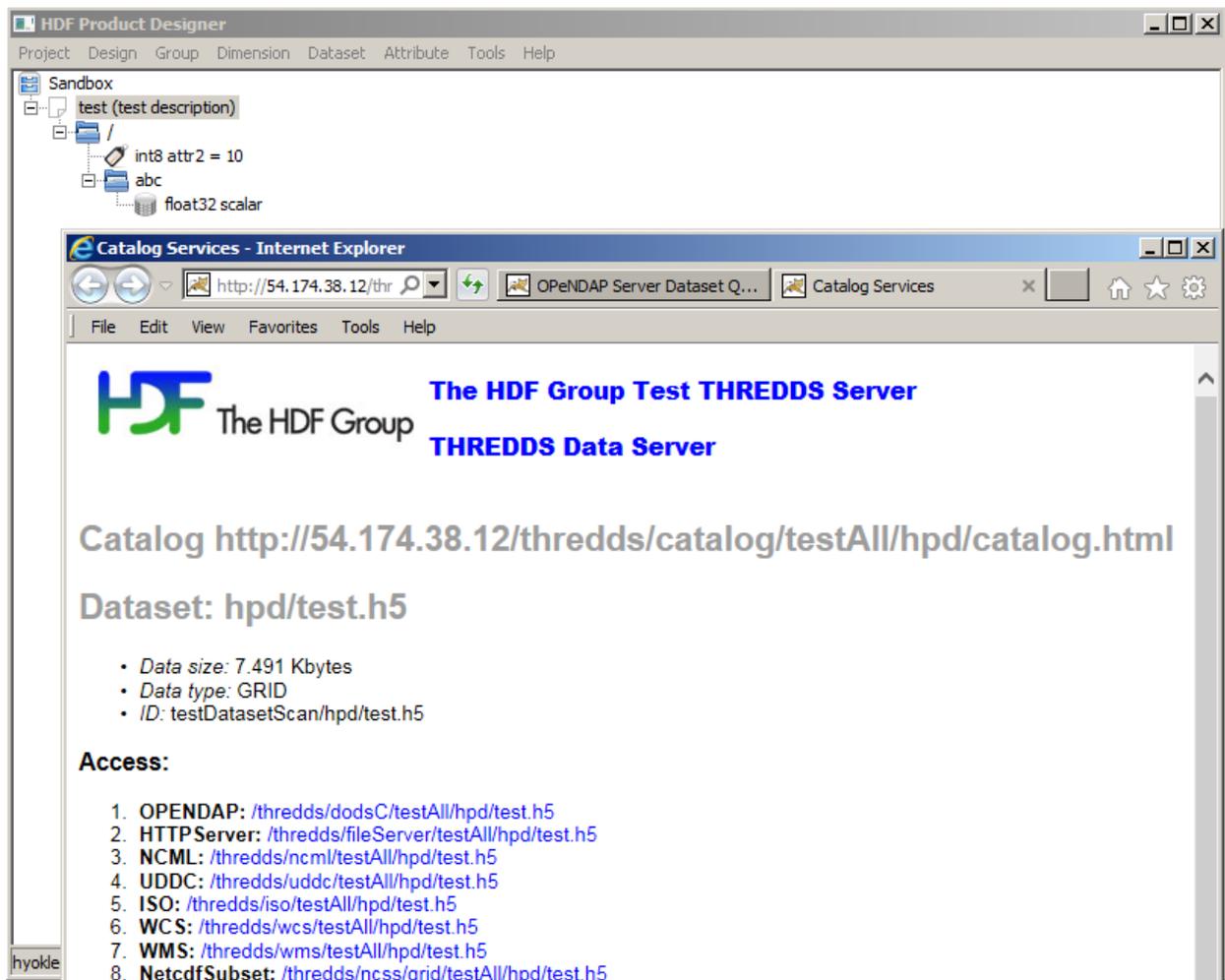
Fig. 5.30: THREDDS dataset access from HPD Online

5.8.2 Documentation

Preparing product documentation is often a repetitive task, especially for designs where large number of HDF5 groups, datasets, and attributes have to be described. To reduce the mundane workload and speed up the process, design documentation as Microsoft Word `docx` file can be stored locally as a starting point for further editing. The formatting of such documents is deliberately minimal to allow easier adjustment to the final style. Documentation content consists of:

- Design's name and version as the document title.

- A sentence below the title with the timestamp of document's creation.

- Group names as section titles.

- Group's attribute information (name, datatype, shape, value) in a table with single line borders.

- Dataset names as titles of subsections under their parent groups.

- Dataset information (parent group, datatype, shape, and max. shape) in a table with single line borders.

- Dataset's attribute information (name, datatype, shape, value) in a table with single line borders.

- HDF5 dimension scales are distinguished from ordinary HDF5 datasets by having *Dimension Scale* in their subsection title.

Note: This is an experimental feature and the formatting and content of generated Microsoft Word files may change. We solicit user comments and suggestions on how to improve the utility of the produced material.

To obtain design documentation:

- Select the **Tools > Document > MS Word** menu option.

- If no errors, a dialog window to choose a file name for the MS Word document will appear (Fig. 5.31).

5.9 Shortcut Keys

The following shortcut keys are supported.

Key	Function
Ctrl+Shift+O	Open Project
Ctrl+Shift+W	Close Project
Ctrl+Q	Quit
Ctrl+O	Open Design
Ctrl+W	Close Design
Ctrl+C	Copy
Ctrl+X	Cut
Ctrl+V	Paste
DEL	Delete
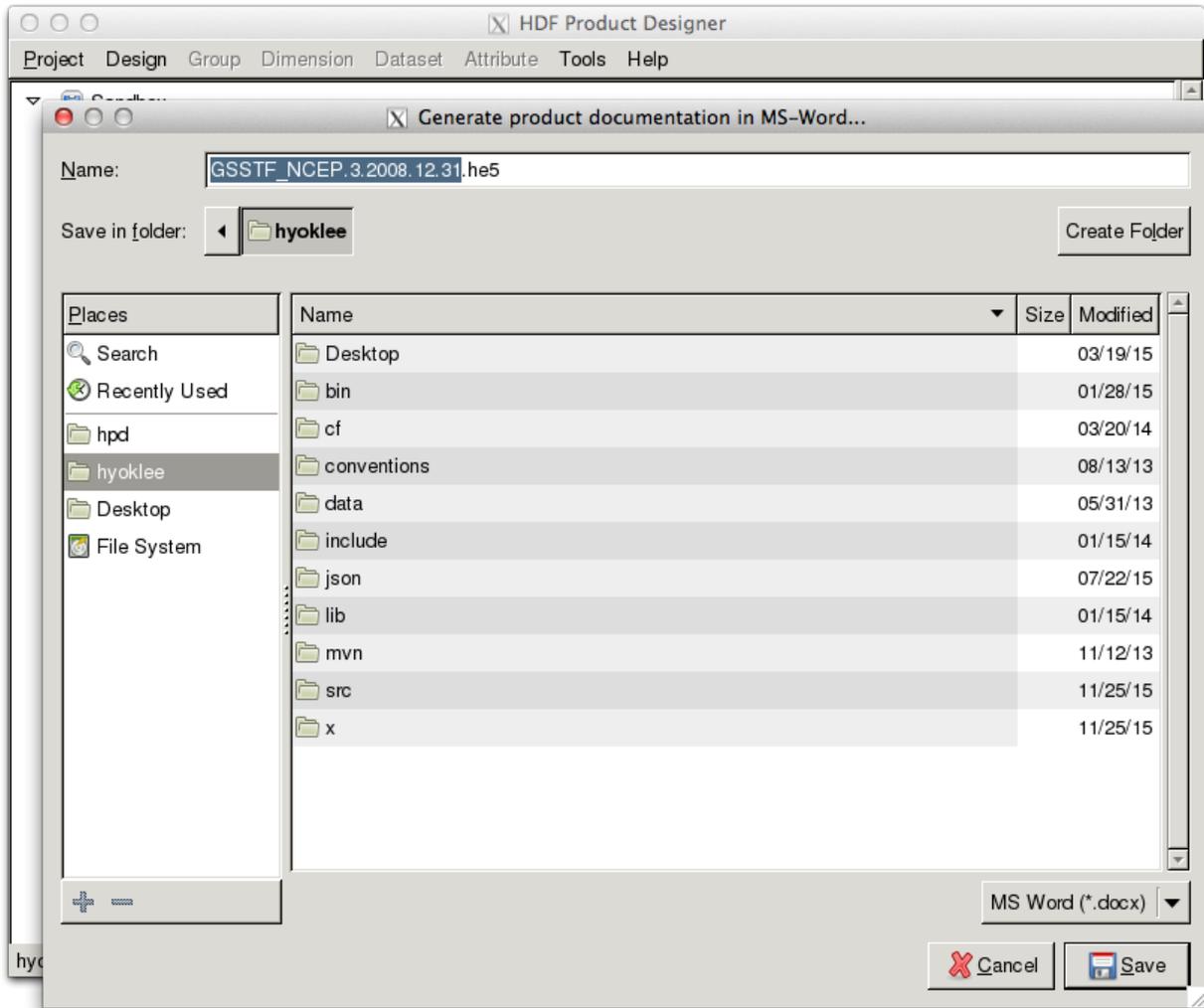
Fig. 5.31: Saving design's product documentation as a Microsoft Word file

SIX

APPENDIX

This part of the guide is not essential for using the application. It provides additional information about the overall system design and specific software components.

6.1 System Design
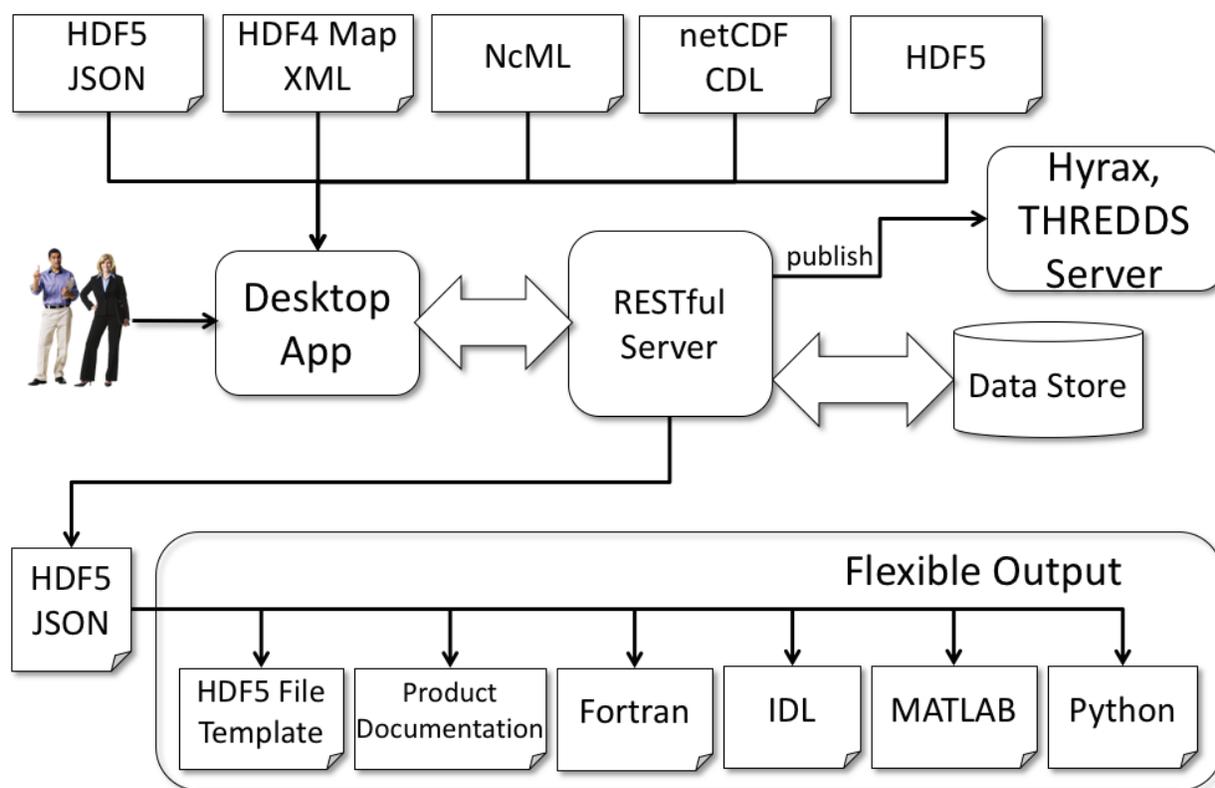
Fig. 6.1: HDF Product Designer system diagram

HDF Product Designer has three main components: Desktop Client, RESTful Server, and Data Store. The fourth component, HDF5 Server, at present provides the online validation services, such as those afforded by the Hyrax and THREDDS data servers. The best approach to integrating it with the other components is being investigated.

The main components are described in more detail below.

6.1.1 Desktop Client

The client is the only part of the system directly accessible to users. It serves as the GUI front end to the capabilities provided by the RESTful server. It is written in Python using the wxPython GUI toolkit. This makes it possible to support the three major computing platforms (Windows, Mac OS X, and Linux) from the same source code while maintaining the native GUI look and feel.

Besides providing GUI functionality to the standard HDF5 content editing, the capabilities unique to the client are design import and convention-based support during design development phase. For design import, the client reads in file content description in several supported formats and translates it into the series of web requests to the RESTful server which is responsible for storing that information.

Convention support during design development phase is achieved by running internally a CLIPS rule engine and executing a set of rules applicable for specific convention and editing action, for example, creating an HDF5 dataset and the CF convention. PyCLIPS module is used as the interface between the client and the CLIPS engine.

6.1.2 RESTful Server

This server provides the back end support for all the capabilities of the HDF Product Designer. These capabilities are exposed as RESTful application programming interface with the intention to allow more than one client application to be developed. The client-server communication is secure because only requests that use the HTTPS communication protocol are accepted.

Capabilities unique to the server are:

- User authentication
- Create, read, update, delete project/design actions
- Design export (template file, source code)
- Generate design template files for validation services

The latest server's documentation is available here.

6.1.3 Data Store

The store is the data persistence layer of the HDF Product Designer. It keeps all the information about users, projects, designs, HDF5 objects, and their relationships. The RESTful Server communicates exclusively with the store via the SSL-encrypted connection thus all the exchanged information is secure.

The Data Store is using the PostgreSQL relational database. Because of the variety of possible HDF5 content the store utilizes JSON in the database extensively. This permits greater flexibility while not jeopardizing the proven data consistency safeguards of relational databases. For example, all design modifications are executed as database transactions.

INDICES AND TABLES

- genindex
- modindex
- search