

**IRIS PASSCAL INSTRUMENT CENTER**

**GENERATING A PH5 ARCHIVE FOR ACTIVE SOURCE AND MIXED MODE  
EXPERIMENTS**

For PASSCAL Platforms: Mac OSX & Linux  
Version 2020:119



This document was written and produced by the  
PASSCAL Data Group

Last revision 04/28/2020 IRIS PASSCAL Data Group

Please send your feedback, questions, and comments to: [data\\_group@passcal.nmt.edu](mailto:data_group@passcal.nmt.edu)

## Table of Contents

Introduction .....	3
Supported data formats .....	4
Pre-archiving steps and installing PH5 .....	4
I. Activating PH5 and organizing your data .....	5
1. Activating the PH5 environment .....	5
2. Creating an organized directory structure.....	5
II. Loading raw data into PH5 .....	5
1. Preparatory steps for node data.....	6
2. Loading data with pforma.....	7
3. Adding more data using the command line.....	12
III. Creating the experiment table .....	13
IV. Receiver/station and event metadata tables .....	14
1. Receiver/station metadata – the Array tables.....	15
2. Event metadata.....	16
3. Creating kef files from metadata tables with noven.....	17
4. Loading metadata kef files .....	20
V. Loading response information.....	20
1. Updating table indices for multiple datalogger types .....	20
2. Adding response information.....	20
VI. Creating and loading additional tables .....	22
VII. Validating your archive .....	22
1. Validate PH5 archive.....	22
2. Checking metadata by outputting stationXML.....	23
VIII. Testing PH5 archive by outputting data .....	23
1. Build and view SEG-Y gathers.....	23
2. Output SAC or miniSEED data .....	24
IX. Transferring PH5 to PASSCAL.....	24
X. Appendix .....	25
1. Updating the PH5 software .....	25
2. Loading all data using the command line .....	25
3. Deleting Tables.....	26
4. If Receivers do not Align True North or East .....	26
5. Loading miniSEED data into PH5 .....	27
6. Updating Magseis Fairfield node locations.....	28
7. Additional metadata guidelines.....	29
8. Lookup table for equipment types/SEED channel names.....	30
9. Glossary.....	32

## Introduction

PH5 is the PASSCAL recommended archival format for active and mixed mode data sets. This program uses a hierarchical data format, version 5 (HDF5) data model, which has several strengths including:

- Extensible and portable, allowing it to evolve and operate on a variety of platforms through a number of programming interfaces
- Self-describing and allows direct access to parts of the file without the need to parse the entire file
- Has been in use for two decades and is utilized in a variety of disciplines, from aerospace to the medical field

PH5 is a unified archiving format designed to make archival and retrieval of data as simple as possible. Because the metadata and data are kept separate, the architecture allows for quick and easy updating of the metadata without having to reprocess the entire data set.

PH5 is built using multiple HDF5 files. The complete set contains a master.ph5 file and a number of mini.ph5 files. The master contains much of the metadata as well as links to the minis, which contain the actual trace data.

The archiving of your data fulfills the principal investigator (PI) responsibilities defined in the PASSCAL Data Delivery Policy: <https://www.passcal.nmt.edu/content/general-information/policy/data-delivery-policy>

Additional documentation can be found on our website: <https://www.passcal.nmt.edu/content/data-archiving/active-source>

Please take a moment to thoroughly review this guide before you start. If you have any questions please contact: [data\\_group@passcal.nmt.edu](mailto:data_group@passcal.nmt.edu).

Within this document:

**Headers, general scripts and commands are in bold.**

Command-line usage is highlighted yellow

GUI options and menus are highlighted gray

*Standard output is italicized.*

URLs and email addresses are blue

## Supported data formats

PH5 currently supports the following data formats from PASSCAL instrument pool:

Instrument type (PASSCAL pool)	Data format	PH5 command line	Notes
Magseis Fairfield ZLAND Node	Fairfield Continuous (modified SEG-D)	segdtoph5	
Reftek RT130	Reftek format	130toph5	
Reftek RT125 (Texans)	TRD	125atoph5	
Geodes	SEG-2	seg2toph5	
Q330, Centaur	MiniSEED	mstoph5, metadatatoph5	See Section X-5

PASSCAL's development on the PH5 software suit has been focused on digesting and converting data formats from the PASSCAL instrument pool. However, the extensible nature of PH5 structure allows for digestion of other data formats. This can be achieved by writing a reader for that specific format.

## Pre-archiving steps and installing PH5

Prior to starting this archival process you should contact the Data Group and acquire, complete, and/or review:

1. Acquire network ID: On the PASSCAL website (<http://www.passcal.nmt.edu>) log in to your account. Once you have logged in, go to the PI home page (<http://www.passcal.nmt.edu/pihomepage>), select your current experiment and then click on the "Mobilize Project" button. Completion of this form sends a request to the DMC for an assembled ID and network code for your experiment.

**Note: This form alerts the IRIS Data Management Center (DMC) of your temporary network and sets up the infrastructure needed for the DMC to accept your data.**

2. Review the PASSCAL Data Delivery Policy: <http://www.passcal.nmt.edu/content/general-information/policy/data-delivery-policy>

3. Review the installation process and required files to set up PH5: <https://github.com/PIC-IRIS/PH5/wiki>

If you encounter problems with the PH5 software, please contact [data\\_group@passcal.nmt.edu](mailto:data_group@passcal.nmt.edu) for assistance.

# I. Activating PH5 and organizing your data

## 1. Activating the PH5 environment

In order to run PH5 commands, you need to activate the PH5 environment. PH5 is run from an Anaconda environment that you created when you installed PH5. You need to be in bash in order to activate the PH5 environment:

```
>> source activate ph5
```

*\*Note: If you are on a shared server, you might need to load Anaconda before activating the PH5 environment:*

```
>> module load anaconda
```

After activating the PH5 environment, you should have (ph5) before your bash prompt in your terminal window.

To deactivate the PH5 environment, run a similar command:

```
>> source deactivate ph5
```

For newer versions of conda, to deactivate the PH5 environment run:

```
>> conda deactivate
```

## 2. Creating an organized directory structure

PASSCAL suggests using the following directory structure. Start by creating a main directory for the project. From your main project directory, create a subdirectory for your raw data and copy the raw data files into it. If you have multiple types of instrumentation, create a separate raw directory for each instrument:

```
>> mkdir RAW-Texans – put all Texan TRD files here
```

```
>> mkdir RAW-RT130 – put all RT130 ZIP files here
```

Next, create a PH5 directory:

```
>> mkdir PH5 – this is where your PH5 master and mini files will be created
```

Also, create a metadata directory. All of the metadata files for your data will be placed in this directory.

```
>> mkdir METADATA
```

Finally, create a RESPONSE folder in your main project directory:

```
>> mkdir RESPONSE
```

# II. Loading raw data into PH5

Make a list of all the raw data files and their full path. This can be done from the command line from your main project directory by running the following command:

```
>> ls -d $PWD/RAW-Texans/*.TRD > raw_list
```

This command will create a text file called `raw_list` in your main project directory. Open this file and make sure that only the TRD files are listed. If you have multiple types of instrumentation (ie., Texans and RT130s), add the paths to the raw data for each additional instrumentation type to this list as needed.

If you have 1- or 3-channel Magseis Fairfield ZLAND nodes, additional steps are required to load your data. Please read section II-1 Preparatory steps for node data. If you do not have any node data, skip to section II-2 Loading data with pforma.

### 1. Preparatory steps for node data

Processing Magseis Fairfield ZLAND node data with **pforma** requires a few additional steps.

Create more descriptive file names:

The default file names created by the node server are functional but are not descriptive enough for effective troubleshooting. The original file name (see examples below) only include the station ID and do not contain any information about the array or the node serial number that was deployed at that station. **pforma** requires more descriptive file names that include the array, station, and serial number of the node in the file name.

Original node file names:

```
/storage/SanBernardinoBasin/SanBernardinoBasin201710/RAW-node/10.fcnt  
/storage/SanBernardinoBasin/SanBernardinoBasin201710/RAW-node/11.fcnt  
/storage/SanBernardinoBasin/SanBernardinoBasin201710/RAW-node/12.fcnt
```

Or:

```
/storage/SanBernardinoBasin/SanBernardinoBasin201710/RAW-node/101.0.0.fcnt  
/storage/SanBernardinoBasin/SanBernardinoBasin201710/RAW-node/102.0.0.fcnt  
/storage/SanBernardinoBasin/SanBernardinoBasin201710/RAW-node/103.0.0.fcnt
```

In the first example, the file name consists solely of the station ID. In the second example the file name consist of the station ID and then numbers indicating how many deployments there were at that station and how many files were written out for that node.

Create a list of the original file names in the main project directory with the absolute path to each file:

```
>> ls -d $PWD/RAW-NODE/*.fcnt > fcnt-list.txt
```

Run **unsimpleton** to create soft links with more descriptive names:

```
>> unsimpleton -f fcnt-list.txt -d NODE_DATA
```

Where:

*fcnt-list.txt* is the list of original node file names with their absolute path

*NODE\_DATA* is the directory where the soft links will be located

If your systems does not support soft links, then you can use **unsimpleton** to create hard links instead:

```
>> unsimpleton -f fcnt-list.txt -d NODE_DATA --hardlinks
```

After running **unsimpleton**, you should have a new directory in your main directory named *NODE\_DATA*:

```
>> ls NODE_DATA
```

```
PIC_1_101_1897.0.0.rg16
```

```
PIC_1_102_1896.0.0.rg16
```

```
PIC_1_103_1893.0.0.rg16
```

The new file names created by **unsimpleton** include, from left to right, array number, station number, node serial number, and then numbers indicating how many deployments there were at that station and how many files were written out for that node.

After running **unsimpleton**, make a new list of raw file names referencing the files in the NODE\_DATA folder:

```
>> ls -d $PWD/NODE_DATA/*.rg16 > node-list.txt
```

In **pforma**, select the new list with more descriptive names (node-list.txt) for the 'RAW list file'. If you have multiple instrument types (i.e. RT130s and nodes) you can append this list for the node data to the list of all other raw files.

### Combine SEG-D traces for faster processing

Nodes can write data to little tiny windows (30 sec) in the raw data, which makes processing in PH5 very slow. You will need to determine what the event windows are in the raw SEG-D data and set **pforma** to combine traces to get ~30 minute windows. **If you already know how long the SEG-D traces in your data files are, you can skip this step and use that information to determine how many traces you will combine to get 30 minute windows.**

If you do not know how long the SEG-D traces are, you can determine how many samples are in each SEG-D trace using **dumpfair**. The number of samples is in the Trace Header Extension 1 as `samples_per_trace`.

To see the number of samples:

```
>> dumpfair 1.fcnt | grep samples_per_trace | head -1
```

Once you have identified the number of samples per trace, you can use this combined with the sample rate to determine the length of each trace in minutes.

For example, if each trace has 75,000 samples at a sample rate of 250sps:

*75,000 samples/250 sample per sec=300 sec/60 sec per min=5 minutes*

Calculate # of windows need to combine to get ~30 minute window:

- with 5 minutes windows, need to combine 6 traces to get 30 minute windows

Proceed with the steps outlined in section II-2 Loading data with pforma.

## 2. Loading data with pforma

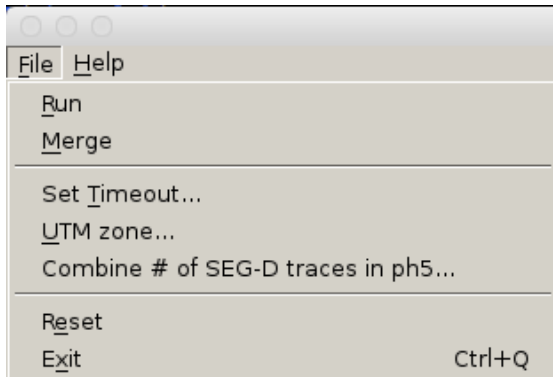
Data is loaded into PH5 using a GUI program called **pforma**. **pforma** takes all of the raw data files referenced in the list you created earlier and divides them up to multi-thread the loading process to make it faster. **pforma** will use all available cores on your system. Currently, this is not configurable.

**pforma** creates multiple subdirectories in the PH5 directory. After loading all of the data using **pforma**, you will merge the resulting subdirectories into a complete PH5.

To launch the **pforma** GUI, type into the command line:

```
>> pforma
```

The GUI that opens will allow you to load data into a PH5 archive using the raw file list that you created earlier. In the main **pforma** window you can select the list of raw data files that you created and the processing directory where the PH5 archive will be created. The main **pforma** menu (under **File**) has several useful options.



**Run** – Run the load process after selecting the raw file list and the processing directory

**Merge** – Combine the data from the sub-directories created by loading data with **pforma** into a final, complete PH5 archive in the Sigma directory

**Set Timeout** – Change the timeout from the default (default is 2000 seconds)

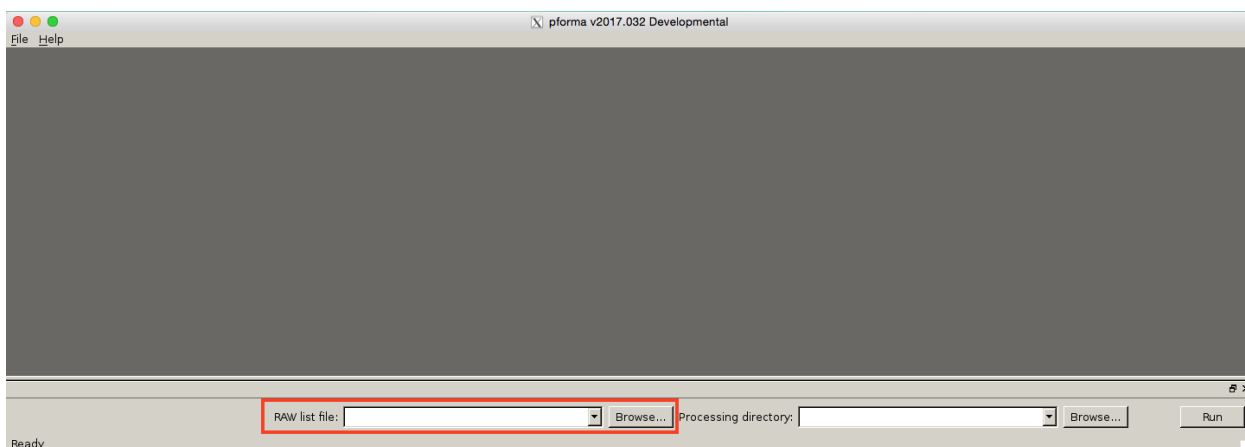
**UTM Zone** – Set the appropriate UTM zone for processing node data (ie 13N)

**Combine # of SEG-D traces in ph5** – Sets the number of SEG-D traces that will be combined into a single trace in PH5 for node data. Nodes can write out the data in very short traces which increases processing time when loading data and retrieving data from the PH5 archive

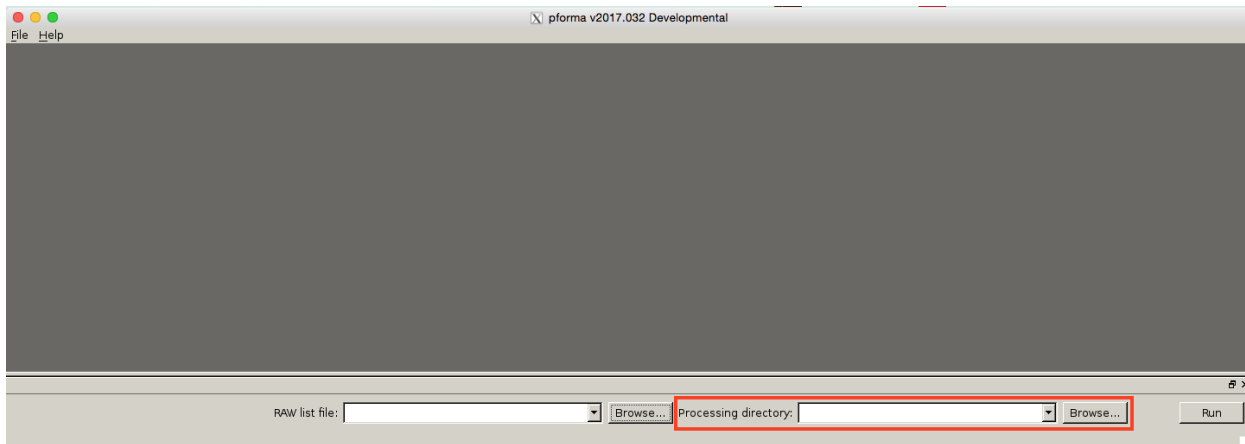
**Reset** – Reset settings to default

**Exit** – Close **pforma**

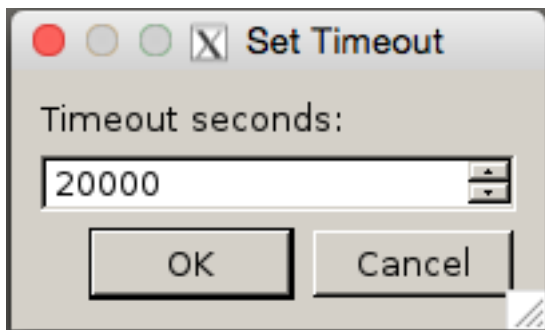
Select the **Browse** button to the right of the “RAW list file” field and browse for the file list you created in your main project directory. This list should include all raw data files that will be included in your PH5 archive.



Select the **Browse** button to the right of the “Processing directory” field and browse for the PH5 directory you created earlier.



Increase timeout: **File > Set Timeout**. For Texans, the default timeout (2000 seconds) should be sufficient but when processing data from RT130s or nodes, increase the timeout by a factor of 10 to 20000.



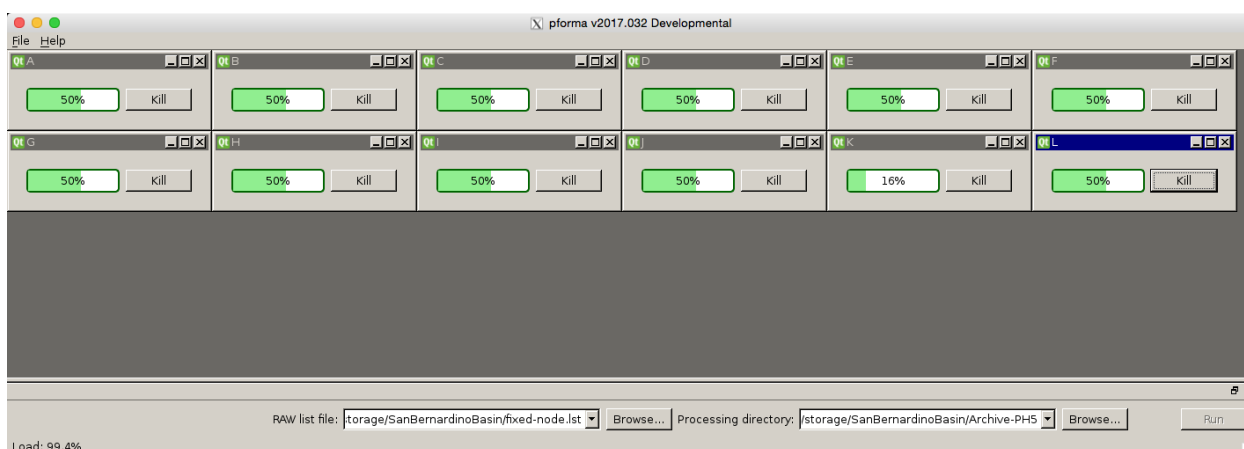
*\*Note: For nodes: In **pforma**, you will need to specify both how many SEG-D traces to combine in the PH5 as well as the UTM Zone. You can set these by selecting the appropriate option in the **File** dropdown menu in **pforma**. [See section above: Combine SEG-D traces for faster processing]*

*Number of traces to combine: **File > Combine # of SEG-D traces in ph5***

*Set UTM zone: **File > UTM Zone***

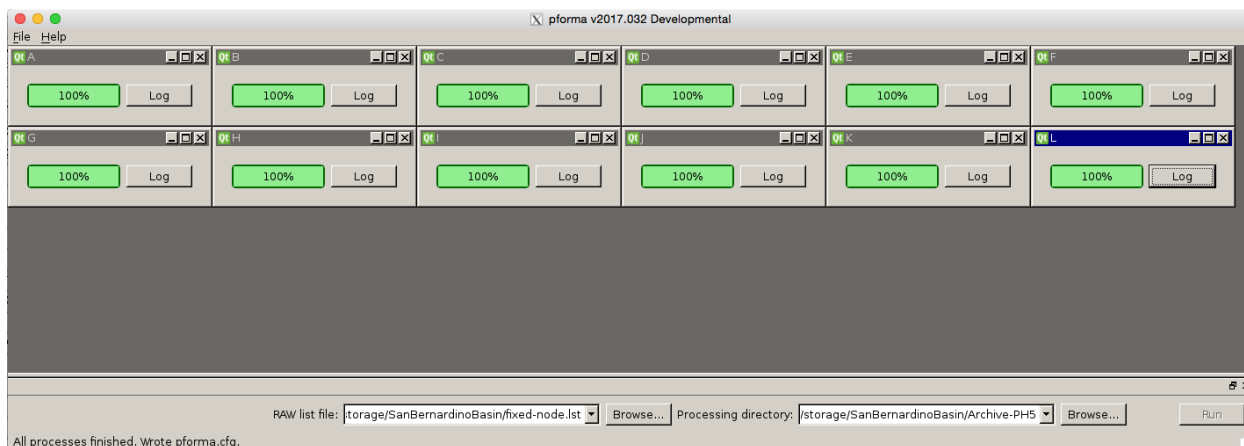
*The UTM zone consists of the zone number and either N or S depending on the hemisphere (ie 13N). If you have node data from multiple UTM zones, you will need to run **pforma** for the data for each UTM zone separately in order to correctly map the node locations into the PH5 archive.*

Click **Run** and wait for all the bars to show 100%.



When the processes are complete, the **Kill** button will change to **Log**. You can select the **Log** button to see a summary of what was processed. Occasionally a process will finish and the **Kill** button will change to **Log**, but the green bar will continue to scroll rather than show 100%. This is a known bug in the GUI. As long as all processes have the **Log** button and the bottom of the **pforma** window shows “All processes finished. Wrote pforma.cfg”, the process has finished.

Do not close **pforma** yet!



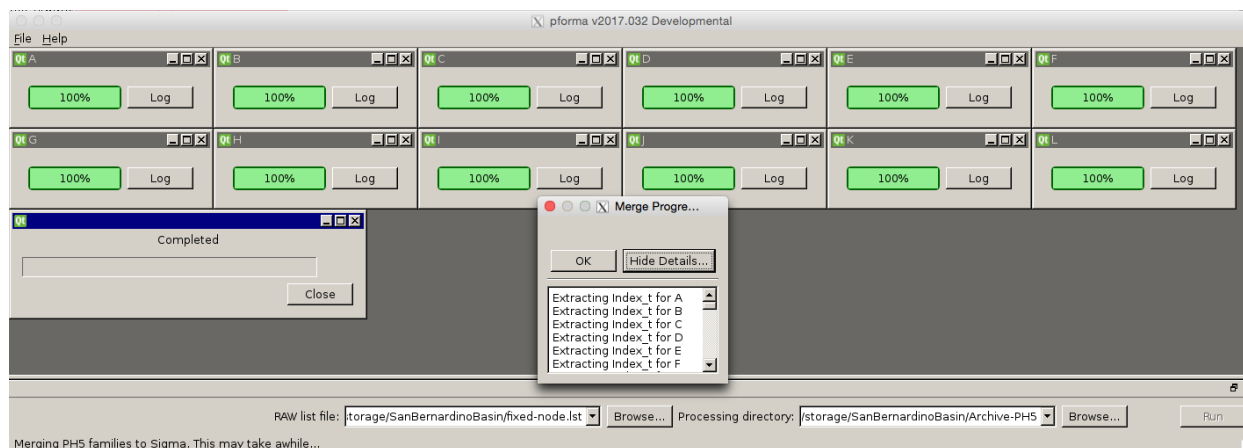
If **pforma** fails to process a data file and stops, the progress bar for that process will turn orange (indicating there was a problem, but **pforma** finished running) or red (**pforma** stopped running without finishing processing the data). Click on the **Log** button to the right of the progress bar for that process to determine what went wrong. Please send the information in this log file to [data\\_group@passcal.nmt.edu](mailto:data_group@passcal.nmt.edu) if you are not able to resolve the problem.

Common problems include:

**Timeout:** **pforma** times out when processing a specific data file. If this occurs, select **File > Set Timeout** and increase the timeout. Delete all of the subdirectories in the PH5 processing directory and run **pforma** again.

**Corrupted RT130 data:** **pforma** stops running on a specific ZIP file due to the data file being corrupted. If this occurs, check the log for that process to identify the problem data. Move that zip file to another folder and remove the entry from the raw file list. Delete all of the subdirectories in the PH5 processing directory and run **pforma** again. Email [data\\_group@passcal.nmt.edu](mailto:data_group@passcal.nmt.edu) for recommendations for dealing with the corrupted data.

Click **File** on the main GUI, and select **Merge**, and wait for this to complete as well. This creates combined PH5 master and mini files that can be found in a subdirectory called “Sigma” within the PH5 processing directory you specified. Once the merge is complete, a Merge Progress window will pop up to let you know it is finished. You can close **pforma** now.



If you close **pforma** before merging the PH5, open **pforma** GUI from the command line. You will need to re-identify the raw file list (even though the data are already processed) and processing directory. Select the **Browse** button to the right of the “RAW list file” field and browse for the file list you created in your main project directory. Select the **Browse** button to the right of the “Processing directory” field and browse for the PH5 directory that you selected for **pforma** to use earlier. Click **File** on the main GUI, and select **Merge**, and wait for the merge to complete.

**pforma** creates a directory for each subset of data that it loads. In the example above, **pforma** created 13 subdirectories in the processing directory named A through M. After merging the PH5 archive, a new directory named Sigma is created where the final PH5 archive is located.

```
[field@texserv2 SanBernardinoBasin]$ ls PH5/
A C          commands1494519102.json E G I K M      pforma.json
B commands1494443465.json D       F H J L pforma.cfg Sigma
[field@texserv2 SanBernardinoBasin]$ ls PH5/Sigma/
kef2ph5.log      miniPH5_00007.ph5 miniPH5_00015.ph5 miniPH5_00023.ph5 miniPH5_00031.ph5
master.ph5       miniPH5_00008.ph5 miniPH5_00016.ph5 miniPH5_00024.ph5 miniPH5_00032.ph5
miniPH5_00001.ph5 miniPH5_00009.ph5 miniPH5_00017.ph5 miniPH5_00025.ph5 miniPH5_00033.ph5
miniPH5_00002.ph5 miniPH5_00010.ph5 miniPH5_00018.ph5 miniPH5_00026.ph5 miniPH5_00034.ph5
miniPH5_00003.ph5 miniPH5_00011.ph5 miniPH5_00019.ph5 miniPH5_00027.ph5 miniPH5_00035.ph5
miniPH5_00004.ph5 miniPH5_00012.ph5 miniPH5_00020.ph5 miniPH5_00028.ph5 miniPH5_00036.ph5
miniPH5_00005.ph5 miniPH5_00013.ph5 miniPH5_00021.ph5 miniPH5_00029.ph5
miniPH5_00006.ph5 miniPH5_00014.ph5 miniPH5_00022.ph5 miniPH5_00030.ph5
```

If you need to add more data files you can add them by repeating the above steps, then delete or move the Sigma directory and re-merge the PH5.

### 3. Adding more data using the command line

**Skip this step if all of your data was loaded with `pforma`.** If you have already run `pforma` to load your data into PH5 and merged your PH5 archive, but need to add additional raw data, you can add data via the command line from within the Sigma folder. In most cases, all data should be loaded with `pforma`, but you can add additional data to the archive after running `pforma`. Since adding data via the command line programs takes longer than using `pforma`, it is recommended to only use the command line to add small amounts of data to your archive (ie a few stations).

Adding additional data using the command line programs is done from within the Sigma folder where your merged PH5 archive is located. The command used will vary depending on what type of data you are trying to add to your archive. Data in different formats must be added using the specific command for that format. For all supported data formats and their corresponding commands, see Table 1 in the "Supported Data Formats" section.

Each command has a help accessed by running the command with the `-h` flag.

Example for adding additional Texan data:

```
>> 125atoph5 -f <file_list> -n master.ph5 >& 125a2ph5.out
```

The file list should contain the full path for the additional raw data files. You can also add data from a single raw data file rather than a file list.

For `segdtoph5`, you will need to specify the UTM zone after the `-U` flag:

```
>> segdtoph5 -f <file_list> -n master.ph5 -U 5N -c 6
```

In addition to specifying the master PH5 file and the location of the raw data files that need to be added to the PH5, you can also specify to which mini-PH5 file you want to begin adding data. For example, in the screenshot below, there are already 36 mini-PH5 files in the Sigma directory. In this case, the additional data should be added to the PH5 starting with mini-PH5 file number 37.

```
[field@texserv2 SanBernardinoBasin]$ ls PH5/
A C          commands1494519102.json E G I K M          pforma.json
B commands1494443465.json D          F H J L pforma.cfg Sigma
[field@texserv2 SanBernardinoBasin]$ ls PH5/Sigma/
kef2ph5.log      miniPH5_00007.ph5 miniPH5_00015.ph5 miniPH5_00023.ph5 miniPH5_00031.ph5
master.ph5       miniPH5_00008.ph5 miniPH5_00016.ph5 miniPH5_00024.ph5 miniPH5_00032.ph5
miniPH5_00001.ph5 miniPH5_00009.ph5 miniPH5_00017.ph5 miniPH5_00025.ph5 miniPH5_00033.ph5
miniPH5_00002.ph5 miniPH5_00010.ph5 miniPH5_00018.ph5 miniPH5_00026.ph5 miniPH5_00034.ph5
miniPH5_00003.ph5 miniPH5_00011.ph5 miniPH5_00019.ph5 miniPH5_00027.ph5 miniPH5_00035.ph5
miniPH5_00004.ph5 miniPH5_00012.ph5 miniPH5_00020.ph5 miniPH5_00028.ph5 miniPH5_00036.ph5
miniPH5_00005.ph5 miniPH5_00013.ph5 miniPH5_00021.ph5 miniPH5_00029.ph5
miniPH5_00006.ph5 miniPH5_00014.ph5 miniPH5_00022.ph5 miniPH5_00030.ph5
```

### III. Creating the experiment table

Kitchen exchange format files, or kefs, are used to import metadata into the PH5 file. In the METADATA directory, build the experiment summary kef with the GUI:

```
>>> experiment_t_gen
```

Required fields include nickname, long name, net code, PIs, institutions, coordinates for the bounding box, summary paragraph, and experiment ID (assembled ID).

If you have not already submitted a mobilization form to the DMC and received an assembled ID and network code for your experiment, please visit the PASSCAL PI Home Page (<http://www.passcal.nmt.edu/pihomepage>) and select the “Mobilize Project” button for your experiment. An assembled ID and network code are required for the experiment summary kef.

The screenshot shows a window titled "changes.py" with a sub-header "Experiment\_t Generator". The window contains a grid of input fields for experiment metadata. Several fields are highlighted with red boxes:

- nickname\_s:** SanBernardinoBasin
- longname\_s:** San Bernardino Basin (USGS)
- net\_code\_s:** XG
- PIs\_s:** Patricia Persaud
- institutions\_s:** Louisiana State University
- north\_west\_corner/X/value\_d:** -118.6
- north\_west\_corner/X/units\_s:** (empty)
- north\_west\_corner/Y/value\_d:** 34.4
- north\_west\_corner/Y/units\_s:** (empty)
- north\_west\_corner/Z/value\_d:** (empty)
- north\_west\_corner/Z/units\_s:** (empty)
- south\_east\_corner/X/value\_d:** -116.6
- south\_east\_corner/X/units\_s:** (empty)
- south\_east\_corner/Y/value\_d:** 33.6
- south\_east\_corner/Y/units\_s:** (empty)
- south\_east\_corner/Z/value\_d:** (empty)
- south\_east\_corner/Z/units\_s:** (empty)
- summary\_paragraph\_s:** This is a USGS funded project to study t
- experiment\_id\_s:** 17-010
- time\_stamp/ascii\_s:** Thu May 11 10:23:59 2017
- time\_stamp/epoch\_l:** 1494519839
- time\_stamp/micro\_seconds\_i:** 0
- time\_stamp/type\_s:** BOTH
- Generate Kef** button

*nickname\_s*: Experiment nickname. This is used to name SEG-Y files generated by **ph5toevt**. We recommend using an experiment nickname without any spaces.

*longname\_s*: The long name of the experiment. It is recommended that this match the experiment long name in the PASSCAL experiment database and on the DMC mobilization form.

*net\_code\_s*: The 2-character FDSN network code for the experiment. If you do not have a network code for your experiment, please confirm that you have filled out the mobilization form for the experiment on the PASSCAL PI Home Page (<http://www.passcal.nmt.edu/pihomepage>). Contact PASSCAL Data

Group ([data\\_group@passcal.nmt.edu](mailto:data_group@passcal.nmt.edu)) if you did not receive a network code when you filled out the mobilization form.

*PIs\_s*: A list of the experiment Principal Investigators.

*institutions\_s*: The institutions with which the experiment PIs are affiliated. It is recommended that the institutions be listed in the same order as the PIs.

*north\_west\_corner/X/value\_d*: Longitude in decimal degrees of the northwest corner of a bounding box for the experiment.

*north\_west\_corner/Y/value\_d*: Latitude in decimal degrees of the northwest corner of a bounding box for the experiment.

*south\_east\_corner/X/value\_d*: Longitude in decimal degrees of the southeast corner of a bounding box for the experiment.

*south\_east\_corner/Y/value\_d*: Latitude in decimal degrees of the southeast corner of a bounding box for the experiment.

*summary\_paragraph\_s*: A brief paragraph summarizing the experiment.

*experiment\_id\_s*: The DMC assigned assembled ID for the experiment. If you do not have an assembled ID for your experiment, please confirm that you have filled out the mobilization form for the experiment on the PASSCAL PI Home Page (<http://www.passcal.nmt.edu/pihomepage>) or contact the PASSCAL Data Group ([data\\_group@passcal.nmt.edu](mailto:data_group@passcal.nmt.edu)) if you did not receive an assembled ID when you filled out the mobilization form.

After filling out all of the necessary fields, select **Generate Kef** and save the experiment kef in the METADATA folder.

Afterwards, navigate to the Sigma directory and load the resulting kef (in this example it was saved as experiment\_t.kef) into the PH5 file:

```
>> keftoph5 -n master.ph5 -k ../../METADATA/experiment_t.kef
```

*\*Note: Every time you load a kef using keftoph5, an entry is made in a kef2ph5.log file. This can be useful for checking the progress of your steps.*

## IV. Receiver/station and event metadata tables

Receiver/station and event metadata is loaded into PH5 by first creating a series of CSV (comma-separated variable) files for the receiver/station and event metadata and then converting those files to kef files using the GUI program **noven**. While the columns of the CSV file may be in different orders, the format used in each field has very specific requirements.

Example files for both array and event tables in Excel format can be found on the PASSCAL website: <http://www.passcal.nmt.edu/content/data-archiving/active-source>

*\*Note: For Magseis Fairfield nodes, if the receiver locations have been configured properly in the node server prior to writing out the data, pforma/segdtoph5 will automatically generate the appropriate array tables when the data are loaded into PH5 and no additional user input is required. If you need to update the receiver locations see Appendix 6.*

## 1. Receiver/station metadata – the Array tables

Receiver/station metadata is required for all data included in a PH5 archive. For most instrumentation, the user must prepare a .csv format file containing all required fields (see table below). The only exception is for Magseis Fairfield nodes (see note above).

For the purposes of PH5, the station metadata is referred to as the “array metadata” or “array tables”. The array tables in a PH5 archive are where the station metadata, including location, sample rate, instrumentation, and deploy and pickup times are stored. Each station is assigned to an array. A PH5 archive can contain as many arrays as are necessary, where an array is defined as a group of stations that have some logical connection (e.g. sample rate, geographic location on a line). In many cases, there will only be a single array of stations. It is recommended that a user create separate .csv files for each array. This allows for easier editing of problem metadata in that a user only needs to edit and replace a single array.

Each row in the array table corresponds to a single channel of data. Therefore, for a 3-channel instrument, there will be 3 rows for each station.

*\*Note: Each array should only contain stations with the same sample rate. If a station is recording multiple sample rates, that station will be listed under multiple arrays, with each array corresponding to a sample rate. For example, a station with an RT130 recording both 100sps and 1sps data would have entries under Array 1 for the 100sps HH? channels and a second entry (with the same station name) under Array 2 for the 1sps LH? channels.*

The required fields for the array table are listed below with brief descriptions. Example array tables can be found on the PASSCAL website (see above).

Required fields	Description	Limitations
Array	Array or line number for stations	Numeric
Station number (id_s)	SEG-Y compatible station number	Numeric, less than 32,767
SEED station name	SEED compatible station name	Alphanumeric, 3-5 characters, all capital letters
DAS serial number	Full datalogger serial number	DAS serial number as marked in raw data headers, varies by manufacturer
Sensor serial number	Sensor serial number	Varies by manufacturer
Channel number	Channel number	Numeric, normally 1 for single channel dataloggers, 1, 2, or 3 for 3-channel. Must match channel number in data headers.
Sample rate	Sample rate for each recorded channel	Integer, must be greater than 1. For sample rates less than 1, adjust the sample rate multiplier (See Note)
Sample rate multiplier	Multiplier for sample rate	Normally 1 unless sample rates less than 1sps are recorded (See Note)
SEED channel name	SEED compatible channel name	3 characters, see Section X-8
Latitude (location/Y)	Latitude in degrees	Positive or negative number; Must be the same for all channels of a station

Longitude (location/X)	Longitude in degrees	Positive or negative number; Must be the same for all channels of a station
Elevation (location/Z)	Elevation in meters	Numeric
Deploy time	Deploy time in appropriate date format	YYYY:JJJ:HH:MM:SS.sss or YYYY-MM-DDTHH:MM:SS.ssss
Pickup time	Pickup time in appropriate date format	YYYY:JJJ:HH:MM:SS.sss or YYYY-MM-DDTHH:MM:SS.ssss
DAS manufacturer	Datalogger manufacturer	See Section X-8
DAS model	Datalogger model	See Section X-8

Optional fields	Description	Limitations
Sensor manufacturer	Sensor manufacturer	Required in most cases, unless using an all-in-one instrument (sensor and datalogger combined). See Section X-8
Sensor model	Sensor model	Required in most cases, unless using an all-in-one instrument (sensor and datalogger combined). See Section X-8
Description	Station description/comments	Optional station comments

*\*Note: In most cases, the sample rate multiplier is set to 1. For all sample rates 1sps or greater, the sample rate multiplier will be set to 1. For sample rates less than 1sps (i.e. mass positions recorded at 0.1sps, or UH? or VH? seismic data (0.01sps and 0.1sps respectively)), the sample rate column should be set to 1 and the sample rate multiplier should be 10 (for 0.1sps) or 100 (for 0.01sps).*

## 2. Event metadata

Event/shot metadata is not required. The event table can store information about any sort of point source used in your experiment, including earthquakes of interest that occurred under your array. If you have event metadata, then you will also need to create an event table in addition to the array tables you have already created. The event table has fewer required fields than the array table.

Required fields	Description
Shot line/array	Shot line or array number; Numeric
Event ID (id_s)	SEG-Y compatible shot number, less than 32,767
Latitude	Latitude in degrees
Longitude	Longitude in degrees
Elevation	Elevation in meters
Event time	Time: YYYY:JJJ:HH:MM:SS.sss or YYYY-MM-DDTHH:MM:SS.ssss

Optional fields	Description
Event size	Event or shot size if available; kg, lbs, or magnitude

Event size units	Units for event or shot size if available: kg, lbs, or magnitude
Depth	Event or shot depth below surface in meters

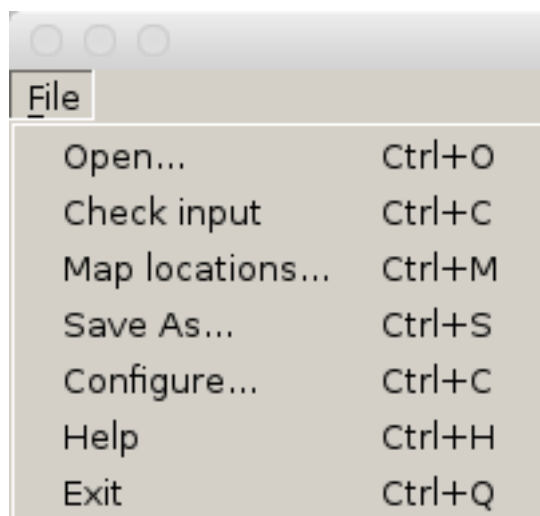
### 3. Creating kef files from metadata tables with *noven*

Array and event information is loaded into PH5 using a series of kef files. You can create these kef files by loading the CSV files that you just created into a GUI program called **noven**. This program loads CSV files for both arrays and events, checks the formatting of the required columns, and then outputs a kef file that can then be loaded into the master.ph5 file using **keftoph5**.

To launch the **noven** GUI, type into the command line:

```
>> noven
```

The GUI that opens will allow you to load the CSV format metadata files that you created for the receiver array and event tables. The main **noven** menu (under **File**) has several useful options.



**Open** – Open a receiver or event CSV file

**Check input** – verify that the input for each column is correctly formatted and that all required metadata are present

**Map locations** – create a Google Earth kml file from the currently loaded metadata table

**Save As** – save input out to a kef file after the input has been checked

**Configure** – change settings for **noven** including specifying if file is receiver or event metadata

**Help** – displays **noven**'s help for the specified file type (receiver or event)

**Exit** – close **noven**

*\*Note: On some operating systems, the **Configure** menu is located under **Python > Preferences** instead of under **File > Configure**.*

Select **File > Open** and navigate to the METADATA folder and select either the receiver array or event CSV file that you created earlier. When you initially open a file, **noven** defaults to displaying the top 3 rows of the file. You can modify this in the Configure menu.

Array	Station	SEED_station_name	DAS-SN	Sensor-SN	Component
4	4001	4001	965E	T3544	1
4	4001	4001	965E	T3544	2

The ‘receiver’ header on the GUI window indicates that **noven** is interpreting the currently open file as a receiver array table.

You can configure several of **noven**’s setting by opening the **File > Configure** window:

**Configure**

Input Type: receiver

Output Format: kef

Column Separator: comma

Skip Lines: 0

View Lines: 3

Close Apply

**Input Type** – Specify if the file you are currently working with is for a receiver or an event table.

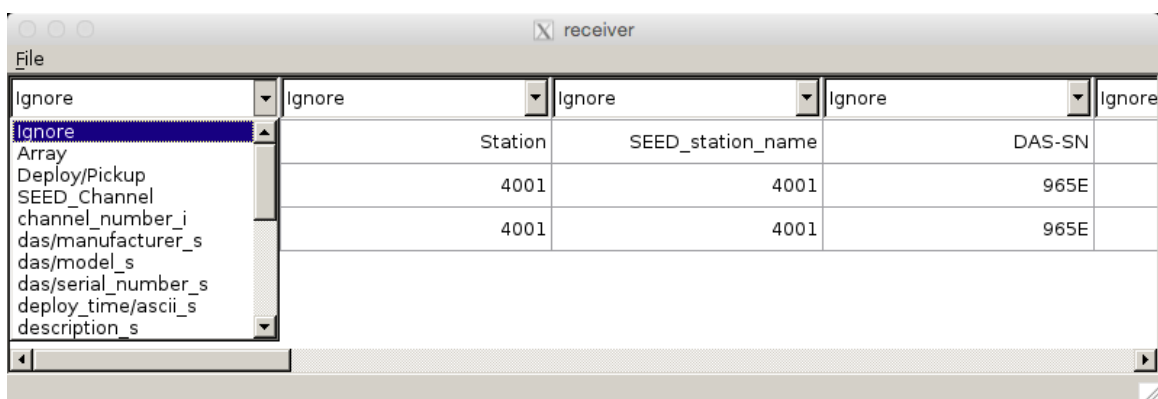
**Output Format** – output metadata in a kef file.

**Column Separator** – Select the appropriate delimiter for your input file. The default is comma-delimited.

**Skip Lines** – Number of lines at the top of the file to skip. This allows you to load a file with a header field for each column, and then skip that line when checking the input and saving out the kef file.

**View Lines** – Number of lines to show. **noven** defaults to showing the top 3 lines of a file, but the number of visible lines can be increased if desired.

After setting the appropriate input type in the Configure window (receiver or event) and selecting **Apply**, you can specify which columns in your file match the required input columns. While the format of each column is specified, the order of those columns in your file is flexible. When you initially open a file in **noven** the “Ignore” at the top of each column indicates that that column will not be included in the output kef file. You can select the dropdown menu at the top of each column and change the “Ignore” to the appropriate column identifier.

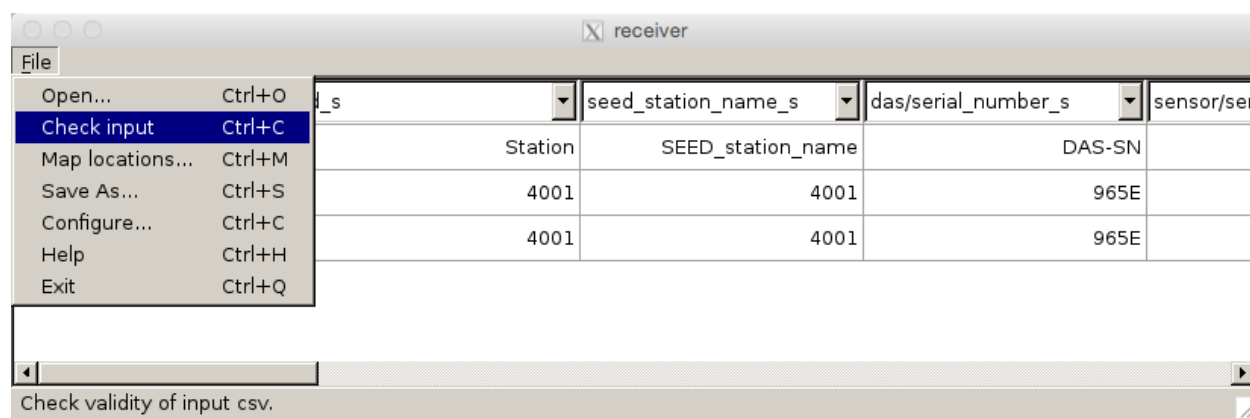


**noven**'s Help menu has a complete list of the possible columns for the specified file type (receiver or event) and the required format for each column. Select **File > Help** to view the help for that file type.

If there are additional columns in the file that are not required for PH5 (notes, contact information for landowners, etc), leave those columns specified as "Ignore" and they will not be included in the kef file.

Once you have identified all of the columns in your file using the dropdown menus, if your file has a header line, return to the **File > Configure** window and change the number of lines to skip so that all header lines are skipped and the only lines displayed are the desired metadata.

You can then check the file input.



If all required columns are defined, and the contents of all columns are formatted correctly, a go ahead message will be displayed.

Once you have received this confirmation that the file is formatted properly, click **File > Save As...** to save a .kef file of the information. This file should be saved into the METADATA directory. **noven** will not allow you to save the .kef file until you have checked the file using the **Check input** option.

*\*Note: noven will not be able to handle any empty lines in your file. If, after running **Check input** successfully, you go to save the file and receive an error message "Must open csv and check inputs first", check your CSV file to confirm there are no empty lines.*

You can also export the CSV into a .kml file to look at in Google Earth. With **noven** still open, select **File > Map Locations**, and save the file as example.kml. You can then open this file with Google Earth

to ensure that your receivers and events are in their proper locations. If necessary, correct any errors in your metadata and rebuild the .kef file.

#### 4. Loading metadata kef files

Once your .kef file is created for your array and/or event table, you may then load them into PH5. From the Sigma directory, use the command:

```
>>keftoph5 -n master.ph5 -k ../METADATA/array_1.kef
>>keftoph5 -n master.ph5 -k ../METADATA/shot_1.kef
```

In these two examples, the array metadata was named array\_1.kef, and the event metadata was named shot\_1.kef. You can load multiple array or event kef files if needed.

If you discover errors in the metadata after loading the array or event kef files, you can remove the receiver or event metadata from the PH5 by using the **delete\_table** command. See Section X-3 for instructions on removing tables from the PH5 archive.

## V. Loading response information

PH5 requires you to specify the response information for your instrumentation. There are two steps to this process: updating the response table references and merging the response information.

### 1. Updating table indices for multiple datalogger types

Skip this step if you only have one type of datalogger and go to the next section (Section V-2 Adding response information). In your PH5 directory containing the sub families and Sigma directory run:

```
>> set_n_i_response
```

This will create a new directory called RESPONSE\_T\_N\_I. Move into this directory.

From the RESPONSE\_T\_N\_I directory run:

```
>> load_das_t --path=../Sigma --onlysave
>> load_das_t --path=../Sigma --onlyload
```

Delete the old response table and load the new Response\_t\_cor.kef file found in the RESPONSE\_T\_N\_I directory.

```
>> delete_table -n ../Sigma/master.ph5 -R
>> keftoph5 -n ../Sigma/master.ph5 -k Response_t_cor.kef
```

### 2. Adding response information

Response files for individual dataloggers and sensors can be downloaded from the IRIS DMC Nominal Response Library (NRL): <http://ds.iris.edu/NRL/>

From the NRL, save the RESP files for all of your sensors and dataloggers in the response directory.

**resp\_load** is run in the Sigma directory where your combined PH5 archive is located.

Start by creating a template for your arrays:

```
resp_load -n master.ph5 -a <comma separated list of arrays>
```

For example, to generate a template csv file for array 1:

```
>> resp_load -n master.ph5 -a 1
```

This will generate a CSV file that contains every combination of das and sensor used in the specified arrays and save it as input.csv

Example of input.csv contents:

```
DAS Model, Sensor Model, Sample Rate, sample rate multiplier, Gain, DAS RESP path, Sensor RESP path  
rt130,cmg3t,100,1,1,,
```

Edit this template file with the path to the RESP files for the DAS and sensor.

Example:

```
DAS Model, Sensor Model, Sample Rate, sample rate multiplier, Gain, DAS RESP path, Sensor RESP path  
rt130,cmg3t,100,1,1,/path/to/RESP/RT130_100SPS.RESP,/path/to/RESP/CMG3T.RESP
```

If you have a combined response file (sensor and datalogger together) rather than separate response files for the sensor and datalogger, you will only put the path to the RESP files in once. The example below is for a Magseis Fairfield 3-channel node where sensor and datalogger are part of a single unit. In this case only the DAS Model field should be populated.

Original input.csv file for Magseis Fairfield node:

```
DAS Model, Sensor Model, Sample Rate, sample rate multiplier, Gain, DAS RESP path, Sensor RESP path  
ZLAND3C,,250,1,12,,
```

Modified file for Magseis Fairfield node:

```
DAS Model, Sensor Model, Sample Rate, sample rate multiplier, Gain, DAS RESP path, Sensor RESP path  
ZLAND3C,,250,1,12,/path/to/RESP/NODE_3C.RESP,
```

Do not forget to include the trailing comma.

*\*Note: The trailing comma is required in all cases where a single combined response file that describes both the sensor and datalogger in a single file is used. In the example above for the Magseis Fairfield nodes, which are an all-in-one unit, there is only one response file (NODE\_3C.RESP) that describes the entire response for the node. The response file is in the DAS RESP Path column of the input file while the Sensor RESP Path column is left blank. The trailing comma is left to mark the Sensor RESP Path column even though it is empty.*

Next you run:

```
>> resp_load -n master.ph5 -a <comma separated list of arrays> -i input.csv
```

This will generate a new response.kef and new array\_t\_XXX.kef files for each array you specified and replace the existing tables with the updated response and array tables.

## VI. Creating and loading additional tables

Before finalizing the PH5 archive, any additional tables that are calculated from information already existing in the PH5 (for example, source-to-receiver offsets) need to be generated and loaded into the PH5. The following commands are all run in your Sigma directory.

### *Source-to-receiver offset table*

If you do not have any events/shots, skip this step. **geo\_kef\_gen** calculates the source-to-receiver offset for each station and shot, and then loads the resulting tables back into PH5 with the following commands.

```
>> geo_kef_gen -n master.ph5 > offset_t.kef followed by  
>> keftoph5 -n master.ph5 -k offset_t.kef
```

### *Texan timing tables*

**You only need to perform this step if you have Texan (RT125) data. If you are using RT130s and/or nodes, skip this step and go to sort\_kef\_gen.**

Texan data may need timing-drift corrections, which can be easily done with the command

```
>> time_kef_gen -n master.ph5 > time_t.kef followed by  
>> keftoph5 -n master.ph5 -k time_t.kef
```

### *Sort table*

The **sort\_kef\_gen** command produces a .kef containing the information that will help optimize data searches, which is done with the following commands

```
>> sort_kef_gen -n master.ph5 -a >sort_t.kef followed by  
>> keftoph5 -n master.ph5 -k sort_t.kef
```

## VII. Validating your archive

PH5 has several tools to confirm the accuracy of your archive and view your metadata and data before extracting your data in the desired format.

### *1. Validate PH5 archive*

**ph5\_validate** will output a text file of any warnings about your archive and can automatically fix some errors. Run the following command in the Sigma directory

```
>> ph5_validate -n master.ph5
```

Check the ph5\_validate.log file for any errors. Correct those errors before running **ph5\_validate** again.

## 2. Checking metadata by outputting stationXML

To look at the stationXML of your PH5 archive, you can use the **ph5tostationxml** command. **ph5tostationxml** generates stationXML format metadata. To generate stationXML run:

```
>> ph5tostationxml -n master.ph5 -p /path/to/your/PH5 -o exp-sta.xml --level=station  
>> ph5tostationxml -n master.ph5 -p /path/to/your/PH5 -o exp-resp.xml --level=response
```

**ph5tostationxml** will also generate a Google Earth KML file for all arrays and shot lines:

```
>> ph5tostationxml -n master.ph5 -p /path/to/PH5 -o exp-geo.kml -f KML
```

## VIII. Testing PH5 archive by outputting data

Once your PH5 database has been completely built and validated, you can write the data out in various formats. In the following steps, you can either output to SEG-Y gathers with the **ph5toevt** command, or into miniSEED or SAC using the **ph5toms** command.

### 1. Build and view SEG-Y gathers

PH5 can create event or receiver gathers via command line. Here is an example of a shot gather:

```
>> ph5toevt -n master.ph5 --use_deploy_pickup -o Gathers -N -l 10 -A 1 -x U -e 5012 --shot_line  
1
```

where:

- n specifies the master.ph5 file
- use\_deploy\_pickup causes PH5 to use the deploy and pickup times in the array tables to determine if data exists for a station
- o specifies the output directory
- N does not time correct the data
- l specifies the trace length in seconds
- A specifies which array to pull data from
- x specified the extended header style (U is Menlo USGS style headers)
- e specifies which shot to pull data for
- shot\_line specifies which shot line the shot is part of

The SEG-Y gather in this example is written to the Gathers directory and contains 10 second long un-time-corrected traces from array 1 receivers (all components) starting at the time of event 5012. A log of **ph5toevt** activity is also saved in the specified output directory. If you are ready to produce gathers for all events, you can run **ph5toevt** with the '-E' flag to create gathers for all of the events. Run **ph5toevt -h** for more options.

PH5 can also output receiver gathers using the **ph5storec** command.

## 2. Output SAC or miniSEED data

PH5 also supports writing data out as SAC or miniSEED. You can write out all of the data from the PH5 archive or just specified time windows using the **ph5toms** command. **ph5toms** defaults to miniSEED for the output format unless the **-F** flag is specified. Try **ph5toms -h** for all options.

To output miniSEED for two days of data for array 1, run the following:

```
>> ph5toms -n master.ph5 -p . -a 001 -s 2017:052:00:00:00 -t 2017:053:23:59:59 -o mseed -F MSEED
```

where:

- n specifies the master.ph5 file
- p specifies the location of the master.ph5 file (in the example above, -p . indicates the master.ph5 file is in the current directory)
- a specifies which array to pull data from. The leading zeroes (-a 001 in the example above) are required.
- s specifies the start time of the desired data
- t specifies the end time of the desired data
- o specifies the output directory
- F specifies the output data format (SAC or MSEED)

To output SAC files for all data contained in the PH5 archive, run the following:

```
>> ph5toms -n master.ph5 -F SAC -o sac_out/
```

## IX. Transferring PH5 to PASSCAL

The completed PH5 volume that will be archived at the IRIS DMC consists only of the master.ph5 file and all of the mini???.ph5 files. All other files (kef files, csv files, etc) are not considered to be part of the final archive and do not need to be submitted to PASSCAL for archiving.

For PASSCAL experiments, after you finish building your PH5 archive and have verified it is complete, please contact the PASSCAL Data Group ([data\\_group@passcal.nmt.edu](mailto:data_group@passcal.nmt.edu)) via email. In your email, include the experiment name, assembled ID, and size of the completed PH5 volume in the email. The PASSCAL Data Group will send instructions for transferring the PH5 archive to PASSCAL for further verification before sending it to the IRIS DMC.

A report is not required for PH5 archives as the experiment table in the PH5 archive contains a brief description of the experiment. However, supplemental files and information can be submitted as a report if their association with the archive would make the data more useful for a future user. If submitting a report associated with a PH5 archive, please include any files with the email to the PASSCAL Data Group.

## X. Appendix

### 1. Updating the PH5 software

The PH5 software can be updated by pulling the most recent version from the PASSCAL PH5 GitHub.

To update the PH5 software, first locate the PH5 directory that was created when you initially pulled the PH5 package from GitHub.

From the PH5 directory, run:

```
>> git pull
```

This will go to the PH5 GitHub and pull the most recent version.

Now, activate PH5:

```
>> source activate ph5
```

Once PH5 is active, you can uninstall the current version and install the new version:

```
>> pip uninstall ph5
```

Select 'y' when prompted. You should see a message saying that you "Successfully uninstalled ph5-????".

Now install the new version:

```
>> python setup.py install
```

Your PH5 software installation should now be up-to-date.

### 2. Loading all data using the command line

Using the command line to load data allows for precise loading of data into PH5, and is recommended only in specific situations. For the majority of experiments, loading data using **pforma** is recommended.

In the PH5 directory, initialize the PH5 file. Name it "master":

```
>> initialize_ph5 -n master.ph5
```

Add Texan raw data to the PH5 file with **125atoph5**. The "trd\_list" should list the absolute or relative path and file names as shown to the right. Use the "M" option, specifying 1/6<sup>th</sup> the data volume to be loaded, in integer GB (you can use **du -sh ../RAW** to estimate data volume). The "M" option causes the data to be uploaded into a number of "data-only" ph5 files (called 'mini' files) linked to the master ph5 file, optimizing subsequent data processing and extraction.

Here we load ~100 GB of data:

```
>> 125atoph5 -f trd_list -n master.ph5 -M 17 >& 125a2ph5.out
```

Node data can also be loaded via command line using the **segdtoph5** command similar to the **125atoph5** example above.

Add RT130 data with **130toph5**, setting the “M” option to 1/2<sup>th</sup> the RT130 data volume to be loaded, in integer GB. A list of RT130 raw files may be referenced as input to **130toph5** similar to the **125atoph5** example above.

### 3. Deleting Tables

If you have loaded any metadata that is later found to be incorrect, you can delete the associated tables containing the metadata or tables with calculations based upon the metadata with **delete\_table**.

**delete\_table** will create a backup of the table you are deleting before removing it from the PH5. After rebuilding the .csv into an appropriate .kef file, you can use the following commands to remove any old metadata. For example, if a receiver location in array 1 is in error after loading the array\_1.kef into the PH5 file:

Delete affected receiver tables and any related tables:

```
>> delete_table -n master.ph5 -A 1
```

If you have already created and loaded the source-to-receiver offset table (created using **geo\_kef\_gen**), you will also need to remove that table when updating the array tables:

```
>> delete_table -n master.ph5 -O
```

This removes the source-receiver offset information. If all receivers/arrays are contained in one csv, you must remove all array tables.

Once you have removed all affected tables, you can replace them with the new updated tables using **ph5tokef**.

```
>> delete_table -h
```

 gives you all options for deleting other tables.

*\*Note: If you delete the DAS tables (-D flag) then the associated data will become inaccessible and cannot be retrieved without reloading it from the original format.*

### 4. If Receivers do not Align True North or East

This step should only be followed in situation where a sensor’s horizontal components are not oriented with N as True North and E as 90 East of True North, or where a sensor’s vertical component provides positive voltage output with upward ground motion. Skip this step for standard installations of nodes, GS-11, L-4C, L-22, and L-28 sensors.

For the aforementioned situations:

- Dump the receiver table to a .kef: **ph5tokef -n master.ph5 -C > receiver\_t.kef**
- Delete all lines from the .kef corresponding to rows in the table not requiring a change to the azimuth or dip, and only keep lines that need update otherwise these rows will be duplicated in the table upon upload of the modified .kef.
- Add the string **‘:Update:orientation/description\_s’** to the row description of the remaining rows (Remember the rows in need of revision should be the only ones left in the .kef).

In the example below the Z orientation needs an update. We removed the lines corresponding to the N and E rows of the table, added `:Update:orientation/description_s` and changed the dip value from '90' to '-90'

```
# Table row 1
/Experiment_g/Receivers_g/Receiver_t:Update:orientation/description_s
orientation/azimuth/value_f = 0
orientation/azimuth/units_s = degrees
orientation/dip/value_f = -90
orientation/dip/units_s = degrees
orientation/description_s = Z
```

- d. Load the receiver\_t.kef into the PH5 file with  
`>> keftoph5 -n master.ph5 -k receiver_t.kef -c >& receiver_keftoph5.out`  
This command uses the '-c' option as a check of .kef syntax (necessary for hand-edited files).
- e. Inspect the output file for errors. You can try `egrep -i "err|warn|prob|fail|fatal"` `receiver_keftoph5.out` to search for the strings in the output file. Correct any issues and re-check.
- f. Repeat the keftoph5 command without the '-c' option for the changes to be written into the PH5 file.

## 5. Loading miniSEED data into PH5

Unlike other supported data formats in PH5, loading miniSEED data into PH5 requires you to load the metadata first from a stationXML or dataless file before adding the data to the archive.

If you do not already have a stationXML or dataless file, please see documentation for using Nexus on the PASSCAL website: <https://www.passcal.nmt.edu/content/data-archiving/documentation/passive-source>

*\*Note: PH5 requires serial numbers for sensor and datalogger.*

Once you have a stationXML or dataless file prepared, use:

```
>> metadatatoph5 -n master.ph5 -f <stationXML>
```

Check the metadatatoph5 log to verify that all stations and channels had responses found and loaded.

If you have multiple sample rates, sensors, and/or dataloggers in your original stationXML file (e.g., mass position channels), you will need to split them out to one sample rate-sensor-datalogger combination per array. Use:

```
>> ph5tokef -n master.ph5 --all_arrays -o arrays.kef followed by
>> keftocsv -f arrays.kef -o arrays.csv
```

Then use Excel or another csv editor to assign different array numbers for each combination, move all the lines for each array into separate files (array1.csv, array2.csv, etc.) and convert the csv files back to kef files with `csvtokef`. Remove the original array table from the PH5 before loading the new array tables:

```
>> delete_table -n master.ph5 --all_arrays
>> keftoph5 -n master.ph5 -k array_1.kef
```

Run `keftoph5` for all new array tables.

Once you have loaded the metadata and reorganized the array tables, load the miniSEED data with:

```
>> mstoph5 -n master.ph5 -d <miniseed_dir>
```

Check the `datatoph5.log` to verify that all the miniseed files were successfully added to the PH5 archive.

Next, create and load an experiment table using the directions in section III of this document.

If there are events to load into your PH5 (e.g., earthquakes, controlled sources), see section IV-2 of this document for instructions.

To check that the responses were fully loaded and the bit weight appears correct, use:

```
>> ph5tokef -n master.ph5 -R -o response.kef
```

If any of the bit weights are 0, or otherwise appear incorrect, compute the correct information and update the `response.kef`, then delete the original response table using `delete_table` (see section X-3), and load the updated `response.kef`.

To complete and verify the archive, follow the steps in section VI, VII, and VIII of this document.

## **6. Updating Magseis Fairfield node locations**

PH5 takes locations for the Magseis Fairfield nodes from the headers of the `.fcnt` files and automatically generates the array tables. If the locations of the deployed nodes were not updated in the Fairfield node server before writing out the `.fcnt` files, you will need to replace the array tables created by **pforma/segdtoph5** with new array tables with the correct locations. You can accomplish this by either using `noven` to create new array tables or you can modify the existing array tables that were created by **segdtoph5**. Once you have `kef` files for the array tables with correct locations, there are instructions at the end of this section for removing the old array tables and replacing them with the new array tables.

To create new array tables, follow the instructions in Section IV to create a new array `kef` file using **noven**. For DAS model, use "ZLAND 3C" and for DAS manufacturer use "FairfieldNodal". For Sensor model, use "GS-30CT" and for Sensor manufacturer use "Geo Space".

To modify the existing array tables, start by writing out the array tables from the PH5 into a file. In the Sigma directory (or wherever your complete PH5 is located) run:

```
ph5tokef -n master.ph5 -A 1 > array1-fromph5.kef
```

Do this for each array that was defined in the Fairfield server for this project. You can also use `--all_arrays` instead of dumping each array individually.

After you have written out the array tables, convert the `kef` files to `.csv`:

```
keftocsv -f array1-fromPH5.kef -o array1-fromph5.csv
```

Take the `.csv` file and replace the locations with the correct locations. All locations in this file should be in decimal degrees. The 3 fields that need to be replaced with the correct locations are `location/X/value_d` (longitude), `location/Y/value_d` (latitude) and `location/Z/value_d` (elevation

in meters). Note that for 3-channel nodes, the .csv will have 3 lines for every instrument, one for each component. You will need to correct the locations for all 3 components.

After you have finished modifying the .csv, convert it back to kef format:

```
csvtokef -f array1-fromph5.csv -o array1-fixedlocs.kef
```

Once you have a kef file with the correct locations, either from creating a new one using the instructions in Section IV or by modifying the array tables written out from the PH5 as described above, then you need to remove the current array tables and replace them with the new version.

To remove an individual array table:

```
delete_table -n master.ph5 -A 1
```

or to remove all of the array tables:

```
delete_table -n master.ph5 --all_arrays
```

**delete\_table** will automatically create a backup of the table that was removed. Once you have deleted the original array tables, load the new version of the array table:

```
keftoph5 -n master.ph5 -k array1-fixedlocs.kef
```

The node locations should now be correct.

## **7. Additional metadata guidelines**

Examples for array/receiver metadata in both Excel and CSV formats can be found on the PASSCAL data archiving website: <https://www.passcal.nmt.edu/content/data-archiving/documentation/active-source>

### **General guidelinesIV**

- Each array can only include one sample rate. If you recorded more than one sample rate, you must separate the different sample rates into separate arrays.
- For MSEED channel names, use SEED convention for your instrumentation and sample rate according to Appendix A of the SEED Manual. There is a lookup table for most of the PASSCAL instrument pool in this document (See Section X-8).
- SEED station name is required; SEED station names are required to be 3 to 5 characters, alphanumeric, all capital letters. If your station IDs (id\_s) fits these requirements, you can just repeat those values in this column.
- Separate files for node metadata are not required unless something in the original experiment setup needs to be changed (ie, changing the station names/numbers, etc). See Section X-6.
- For 3-channel instruments, the array metadata must have 1 line for each channel at a station; for example, station 1001 would have separate lines in the array table for channels 1, 2, and 3.
- Once you have finished editing your receiver array table, save the file out as a comma-separated variable file (.csv).

### **Instrument specific guidelines**

#### **Magseis Fairfield ZLAND 1C nodes**

- In most cases, PH5 takes node metadata from the SEG-D headers. Only create and load new node metadata if the SEG-D headers are incorrect.
- For 1C nodes, channel number 1 is Z.

- DAS-SN is made up of a combination of the Array/Line and Station, ie Line -> 321, Station -> 654, DAS-SN -> 321X654.
- For Sensor-SN, use the node serial number (since sensor and digitizer are in one unit).

### Magseis Fairfield ZLAND 3C nodes

- In most cases, PH5 takes node metadata from the SEG-D headers. Only create and load new node metadata if the SEG-D headers are incorrect.
- For 3C nodes, channel number 1 is N/S, 2 is E/W, and 3 is Z.
- DAS-SN is made up of a combination of the Array/Line and Station, ie Line -> 321, Station -> 654, DAS-SN -> 321X654.
- For Sensor-SN, use the node serial number.

### Reftek RT125 (Texans)

- For Texans, channel number is 1 (vertical) for 1C geophones.
- For DAS/Texan serial numbers, add 10,000 to the serial number written on the Texan to get a 5-digit number.
- For sensor serial numbers, create unique dummy serial numbers for geophones if unique serial numbers do not exist.

### Reftek RT130

- For RT130s, channel number 1 is Z, 2 is N/S, and 3 is E/W.
- Each recorded data stream/unique sample rate must be in its own array, ie 100sps in Array 1, 1sps in Array 2, and 0.1sps (ex. mass positions) in Array 3.
- For sample rates less than 1 (ie 0.1sps) the sample rate multiplier field (SR-mult) should be set appropriately. For 0.1 sps, SR-mult should be 10; for 0.01 sps, SR-mult should be 100.

## 8. Lookup table for equipment types/SEED channel names

Instrument	Sample Rates (Hz)							
	$1000 \leq \text{SPS} < 5000$	$250 \leq \text{SPS} < 1000$	$80 \leq \text{SPS} < 250$	$10 \leq \text{SPS} < 80$	$1 < \text{SPS} < 10$	1	0.1	0.01
STS-2	FH?	CH?	HH?	BH?	MH?	LH?	VH?	UH?
CMG-3T	FH?	CH?	HH?	BH?	MH?	LH?	VH?	UH?
CMG3ESP	FH?	CH?	HH?	BH?	MH?	LH?	VH?	UH?
TR-240	FH?	CH?	HH?	BH?	MH?	LH?	VH?	UH?
TR-120	FH?	CH?	HH?	BH?	MH?	LH?	VH?	UH?
TR-40	FH?	CH?	HH?	BH?	MH?	LH?	VH?	UH?
CMG-40T 30s	FH?	CH?	HH?	BH?	MH?	LH?	VH?	UH?
CMG-40T 1s	GH?	DH?	EH?	SH?	MH?	LH?	VH?	UH?
S-13	GH?	DH?	EH?	SH?	MH?	LH?	VH?	UH?
HS-10	GH?	DH?	EH?	SH?	MH?	LH?	VH?	UH?
L-4C*	GL?	DL?	EL?	SL?	ML?	LL?	VL?	UL?
L-22*	GL?	DL?	EL?	SL?	ML?	LL?	VL?	UL?
L-28	GL?	DL?	EL?	SL?	ML?	LL?	VL?	UL?

L-28LB (4.5Hz geophone)*	GL?	DL?	EL?	SL?	ML?	LL?	VL?	UL?
L-40A (40 Hz geophone)	GP?	DP?	EP?	SP?	MP?	LP?	VP?	UP?
Fairfield Node	GP?	DP?	EP?	SP?	MP?	LP?	VP?	UP?
FBA ES-T	FN?	CN?	HN?	BN?	MN?	LN?	VN?	UN?
<b>Table 1. Recommended SEED channel names for many of the sensors available from PASSCAL</b>								
* The use of “H”, denoting high gain, assumes that the dataloggers are programmed using a gain of 32, which is the recommended gain setting for a typical PASSCAL experiment. In the event the gain is set to 1 at the datalogger, then the second character of the channel name should be set to “L”.								

## PH5 DAS and Sensor references

DAS	PH5-Manufacturer	PH5-Model
<b>ZLAND Node-1C (all-in-one)*</b>	FairfieldNodal	ZLAND 1C
<b>ZLAND Node-3C (all-in-one)*</b>	FairfieldNodal	ZLAND 3C
<b>RT130</b>	Reftek	rt130
<b>RT125 (Texan)</b>	Reftek	rt125
<i>*Note: While the ZLand nodes are all-in-one units and therefore do not technically have a separate sensor reference, when building PH5 metadata use "Geo Space" as the sensor Manufacturer and "GS-30CT" as the Sensor Model.</i>		

Sensor	PH5-Manufacturer	PH5-Model
<b>CMG-3T</b>	Guralp	cmg3t
<b>STS-2 Gen 1</b>	Streckeisen	sts2_g1
<b>STS-2 Gen 2</b>	Streckeisen	sts2_g2
<b>STS-2 Gen 3</b>	Streckeisen	sts2_g3
<b>L-22</b>	Sercel	l22
<b>L-28</b>	Sercel	l28
<b>PIC Geophone (4.5Hz 1C)</b>	Geospace	GS11V
<b>Trillium 120</b>	Nanometrics	trillium_120
<b>Trillium 240</b>	Nanometrics	trillium_240
<b>Trillium 120PH</b>	Nanometrics	trillium_120PH
<b>Trillium Compact 120s</b>	Nanometrics	trillium_comp
<b>CMG-40T</b>	Guralp	cmg40t
<b>Hyperion</b>	Hyperion	hyperion
<b>Trillium 40</b>	Nanometrics	trillium_40
<b>CMG-40T 1sec</b>	Guralp	cmg40t_1

## 9. Glossary

This glossary is intended to clarify PH5-specific terms and general terms used in building a PH5 archive. This is not intended to be a comprehensive glossary of seismology terms.

**Array** – group of stations/receivers. Every station must be assigned to at least 1 array. Each array must have a single sample rate. A station can be part of multiple arrays due to recording multiple sample rates (ie station 1001 recording both 100sps (array 1) and 1sps (array 2)).

**Assembled ID** – also referred to as “experiment ID” in `experiment_t_gen`. The assembled ID is a number assigned to a PH5 archive by the IRIS DMC consisting of the year assigned and a number indicating the order in which the assembled ID was assigned (for example, 19-018). This assembled ID uniquely identifies the data set.

**Event** – any point source temporally coincident with the stations/receiver deployments. While “shot” is the term used more commonly in active source seismology, an event in PH5 can include both manmade sources such as a weight drop or airgun as well as natural sources such as earthquakes.

**Event table** – table in PH5 with event information, typically imported via `kef` file created by `noven`. In `noven`, specifying “shot” in the Preferences/Configure menu will create a `kef` file for your event metadata. Each event must be assigned to a shot line/array.

**Master PH5 file** – file containing all metadata information as well as tables indexing which dataloggers are contained in which mini PH5 file.

**Mini PH5 file** – files containing data in PH5 format. Each mini file can contain data from multiple dataloggers. The number of mini files depends on how much data is in the archive. `pforma` automatically determines the number of mini files. Number of mini files can also be specified when adding data via command line.

**Network code** – two character code assigned to a network by the FDSN (for example, YE). This code combined with the year/years the network was operating uniquely identifies the data set.

**Receiver** – another term for station. In `noven`, specifying “receiver” in the Preferences/Configure menu will create a `kef` file for your station metadata.

**Receiver Table** – table in PH5 with channel orientation information. This table is created automatically when data are loaded via `pforma` or the command line. The receiver table can be dumped from the PH5 via the command `ph5tokef -n master.ph5 --C`

**Response Table** – table in PH5 with response information including bit weight and references to RESP files added to the PH5 with `resp_load`.

**Shot** - a point source, typically manmade, temporally coincident with the stations/receiver deployments. Also referred to as events, particularly in `noven`’s Preferences/Configure menu.

**Shot Line** – group of events/shots. Also referred to as an array in noven when “shot” is selected in Preferences/Configure. Each event/shot must be assigned to a shot line. The shot line numbers can repeat already existing array numbers used for station/receiver metadata.

**Station** – an individual site where equipment was deployed. A single station can contain multiple dataloggers or sensors (for example, collocated short period and strong motion instruments). A station should have a single unchanging location.