EarthScope
Primary Instrument Center

Generating a PH5 Archive from SmartSolo and/or Fairfield Data

Created and last revised: September 9, 2025

Table of Contents

Scope of this document:

This is a guide to building a PH5 from Fairfield nodes, SmartSolo nodes, or both. Note that regarding input data formats compatible with PH5:

- The data must be in manufacturer-specified SEG-D format
- Building a PH5 from Texan (RT125) and/or RT130 raw data has not been deprecated from PH5, but is not covered in this document
- Other data formats (e.g., miniSEED, SAC, SEG-Y) *cannot* be ingested into PH5

If you do not already have it on your machine, install PH5 using the directions at https://github.com/PIC-IRIS/PH5/wiki/Installation-Instructions, and install with full GUIs (environment-gui.yml).

Steps in this document have visual indications for the node type to which they apply: 🟤 for Magseis Fairfield (most commonly the ZLand 3C) and 🟦 for DTCC SmartSolo (most commonly the IGU16-HR3C model). If you have other node models produced by these manufacturers, you may be able to use this document to build a PH5 archive if the output formats are the same as discussed here.

In this document, scripts and commands are **bolded**, commands are highlighted yellow, and GUI items are highlighted gray.

Essential terminology

- PH5: a version of the HDF5 hierarchical data format.
- master.ph5: The PH5 file with metadata. A PH5 can only have one master.ph5.
- miniPH5_xxxxx.ph5: The PH5 files with data, where xxxxx is a number (e.g., 00001).
- SmartSolo .segd: SEGD 2.1 files written out using SmartSolo software.
- Fairfield .fcnt and .rg16: Fairfield continuous SEGD 1.6 files written out using Fairfield software.
- Array table(s): Metadata table(s) with .kef extension describing the stations in the PH5.
- Event table(s): Metadata table(s) with .kef extension describing any events (sources) in the PH5.

Tips/Essential Notes Prior to Starting

- If your experiment used both SmartSolo and Fairfield nodes, and/or two or more sample rates of either SmartSolo or Fairfield nodes, and/or two or more gains of either SmartSolo or Fairfield nodes, offloading the files (.segd, .fcnt) with different line (array) numbers is strongly recommended. For example, 18 dB gain SmartSolos in array 1, 24 dB gain SmartSolos in array 2, and 18 dB gain Fairfields in array 3. This separation will be required for PH5 completion and doing so at the offload step will save time later in the PH5 building process. See Appendix A for more information about preparing to build a PH5 that involves multiple pforma runs.
- Do NOT build multiple PH5s for the same experiment. If you have a need to build a PH5 partway through, for example, a 12 month nodal deployment, contact data-help@earthscope.org to discuss your needs. In almost all cases, it is more

computationally efficient to rebuild a PH5 from scratch to accommodate additional data, as opposed to adding that data to an existing PH5.

- EPIC uses a window length of 30 minutes when writing out SmartSolo .segd and Fairfield .fcnt. Please be aware that almost all PH5 enhancements and bug fixes have been tested with data that have 30 minute windows.
- The datalogger serial number is determined by the line and station number, not the physical node serial number. Do NOT duplicate line and station number in the same experiment unless you have swapped in different nodes at the same station site at different times (e.g., node A at station 1001 for 30 days, then node B at station 1001 for 30 days, and so on).
- Verify your PH5 install is reasonably up to date. Run **ph5_validate -h** and if the version is less than 2022.066, you need to update your PH5 install. If you installed PH5 before May 2021, you will need to do a fresh installation of PH5 because the GUI package dependencies changed.
- Check the raw data size and verify that you have at least three times that space available on your machine.
- PH5 automatically retains all metadata tables loaded, so if you need to update a table, follow directions carefully.
- You need an FDSN network code and DMC assembled ID to complete the PH5. If you are building a PH5 for an EPIC experiment, log in to EPIC and fill out the mobilization form. If you are building a PH5 for a non-EPIC experiment, go to the FDSN temporary network request and specify IRISPH5 as the web service when you complete the form.

0. Preparatory steps (SmartSolo and/or Fairfield)

First, set up the directory structure, such as

mkdir exp_name
mkdir exp_name/Archive-PH5

The Archive-PH5 (or PH5v1, ph5v1, etc.) is the directory in which the PH5 will be built. Then, optionally make a metadata directory:

mkdir exp_name/metadata

If you know the window length used on writeout, go to step 1. If not, run

dumpsegd filename | grep samples_per_trace | head -1

Use this output and the sample rate to determine the duration of each trace window. If it is less than 30 minutes, it is strongly recommended that you make a note of how many trace windows to combine to make a 30 minute window.

1. Input lists

Make input file lists. If you are using SmartSolos, go to step 1.a; if you are using Fairfields, go to step 1.b. If you are using both, go to step 1.a and then step 1.b. Caution: If you swapped nodes and used the same station number and line number for multiple deployments, you must include all files from that station number in a single run.

1.a (SmartSolo only)

Make a file list from the .segd files under SOLODATA. For example:

ls -d $PWD/Oct5_SEGD/*segd > /path/to/experiment/processing/directory/segd.list

If you used more than one gain setting in the experiment, repeat for each gain setting to make separate input file lists. If your experiment used a large number of nodes and/or ran for a long time, you may exceed the ls limit on your system and need to make the file list in steps (e.g., by channel). See Appendix A for example cases requiring two or more input file lists and recommendations on pre-offload steps. You may have to make your segd.list in parts if the number of files exceeds the ls limit on your system.

1.a.i (optional)

If you have a lot of files, go back to your experiment processing directory and run **map_header**, for example

map_header -f segd.list

This may take several minutes to more than an hour to pre-organize the files for the loading step so loading the data in **pforma** runs faster. Note that each run of **map_header** overwrites the previous output file smartsolo_map. Go to step 2.

1.b (Fairfield only)

If you deployed the Fairfield nodes in more than one UTM zone, repeat the following for each zone to make separate input file lists. See Appendix A for example cases requiring two or more input file lists and recommendations on pre-offload steps.

ls -d $PWD/Other/*/*fcnt > /path/to/experiment/processing/directory/fcnt.txt

Next, go back to the experiment processing directory and make a list that can be used to load data.

unsimpleton -f fcnt.txt -d Data_nodes --hardlinks
ls -d $PWD/Data_nodes/*rg16 > rg16.txt

If your system supports soft links, you can exclude the hardlinks flag in unsimpleton (be aware EPIC has not tested this recently).
Go to step 2.

2. Loading data

Now you will load files into PH5 using **pforma**, a GUI that multithreads the process for faster building of the PH5 archive. Start **pforma** going to the experiment processing directory and running:

pforma

Please read the notes below. If you have only SmartSolo data, go to step 2.a; if you have only Fairfield data, go to step 2.b; if you have both Fairfield and SmartSolo data, go to step 2.c.

Notes on **pforma**:

-If a progress bar turns orange, this is a failure to ingest data. Do not proceed with any further steps to build the archive; instead, click Log to see where it failed and details on the problem; there is also a menu to save the log file. You will need to exit **pforma** and after the problem file(s) is/are resolved (may require writing out the affected file(s) again), make a new PH5 directory (e.g., mkdir PH5_v2) and start again from step 0. If you are low on disk space, remove the original directory or back it up on external storage before starting again.

-Progress bar scrolling: this is a known bug and tends to occur when only one file has been assigned to the processing directory. It is not cause for concern.

-Progress bar does not reach 100%: this is known to occur when a large number of files (few hundred or more) are assigned to a processing directory and is most likely with SmartSolo data. It is not cause for concern.

-SmartSolo data written directly to external hard drive with SmartSolo software is known to result in a limited number of files that will not load: building a test PH5 is recommended in this case. Affected files can be re-offloaded to internal storage using SmartSolo software. Fairfield files are not affected.

2.a. Load data (SmartSolo only)

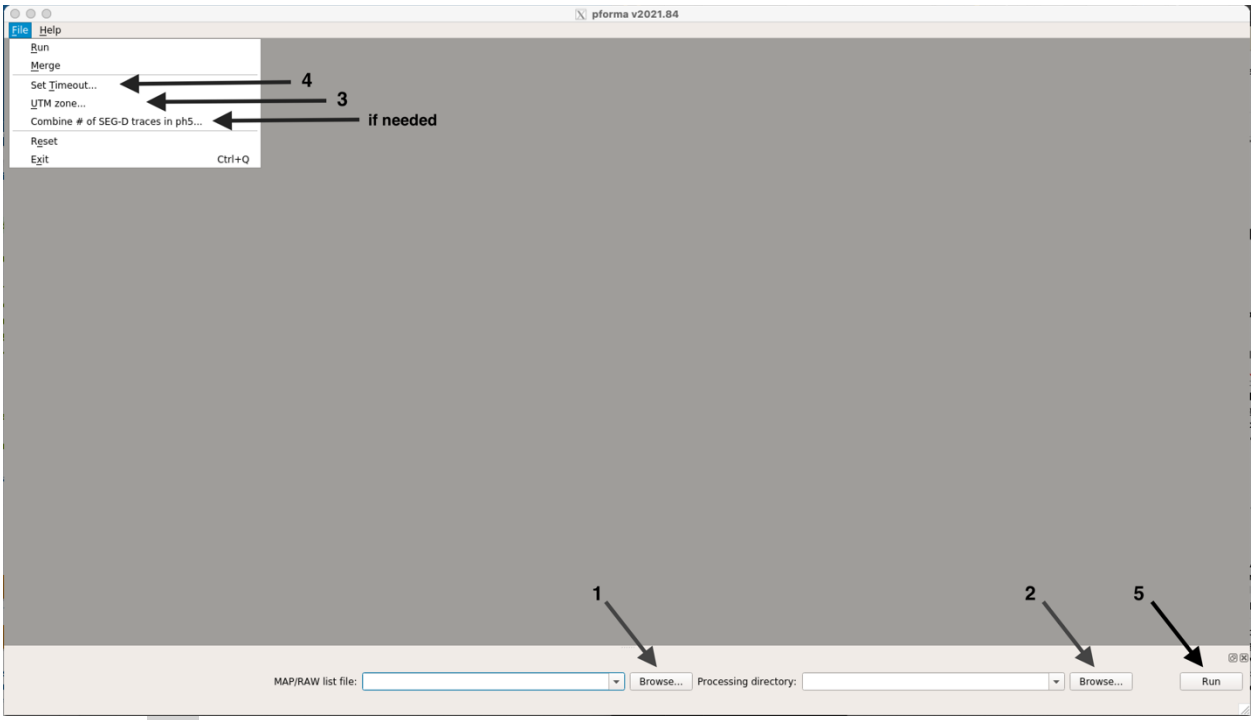
Click Browse to the right of MAP/RAW list file. If you did not use optional step 1.a.i, select the segd.list you made earlier. If you did use optional step 1.a.i, select the smartsolo_map file. Next, click Browse and select the PH5 directory (e.g., Archive-PH5) you made earlier. Then, click File -> Set Timeout, enter 200000, and click OK. If your window lengths are short, click File -> Combine # of SEG-D traces in PH5 and enter the appropriate value (see step 0).

Then, click Run. It may take a while for the processing directories appear in pforma. This does not mean pforma is frozen: let the program continue to run.

Wait until all processing directories in the GUI show the Log button and the lower left corner of **pforma** shows the message 'All processes finished. Wrote pforma.cfg'. After this, if you have more than one input list, do not select Merge and instead close **pforma**. Then, start it again and repeat the steps from Browse to Run. Repeat as needed without merging. When you are done reading in all the input_list files, click File -> Merge. After both the Merge Progress and Completed popups appear in **pforma**, you can close the popups and **pforma**. Go to step 3.

2.b. Load data (Fairfield only)

Click Browse to the right of MAP/RAW list file and select the rg16.txt file you made earlier. Next, click Browse to the right of Processing directory and select the PH5 directory you made earlier. After this, click File-> UTM Zone and enter the appropriate zone (e.g., 11N or 17S) and click OK. Then, click File -> Set Timeout, enter 200000, and click OK. If your window lengths are short, click File -> Combine # of SEG-D traces in PH5 and enter the appropriate value (see preparing files).



Then, click Run. The processing directories will appear within a few minutes.
Wait until all processing directories in the GUI show the Log button and the lower left corner of **pforma** shows the message 'All processes finished. Wrote pforma.cfg'. After this, if you have more than one input list, close **pforma**. Then, start it again and repeat the steps from Browse to Run. Repeat as needed. When you are done repeating these steps, or if you do not need to repeat steps because you have only one input list, click File -> Merge. After both the Merge Progress and Completed popups appear in **pforma**, you can close the popups and **pforma**. Go to step 3.

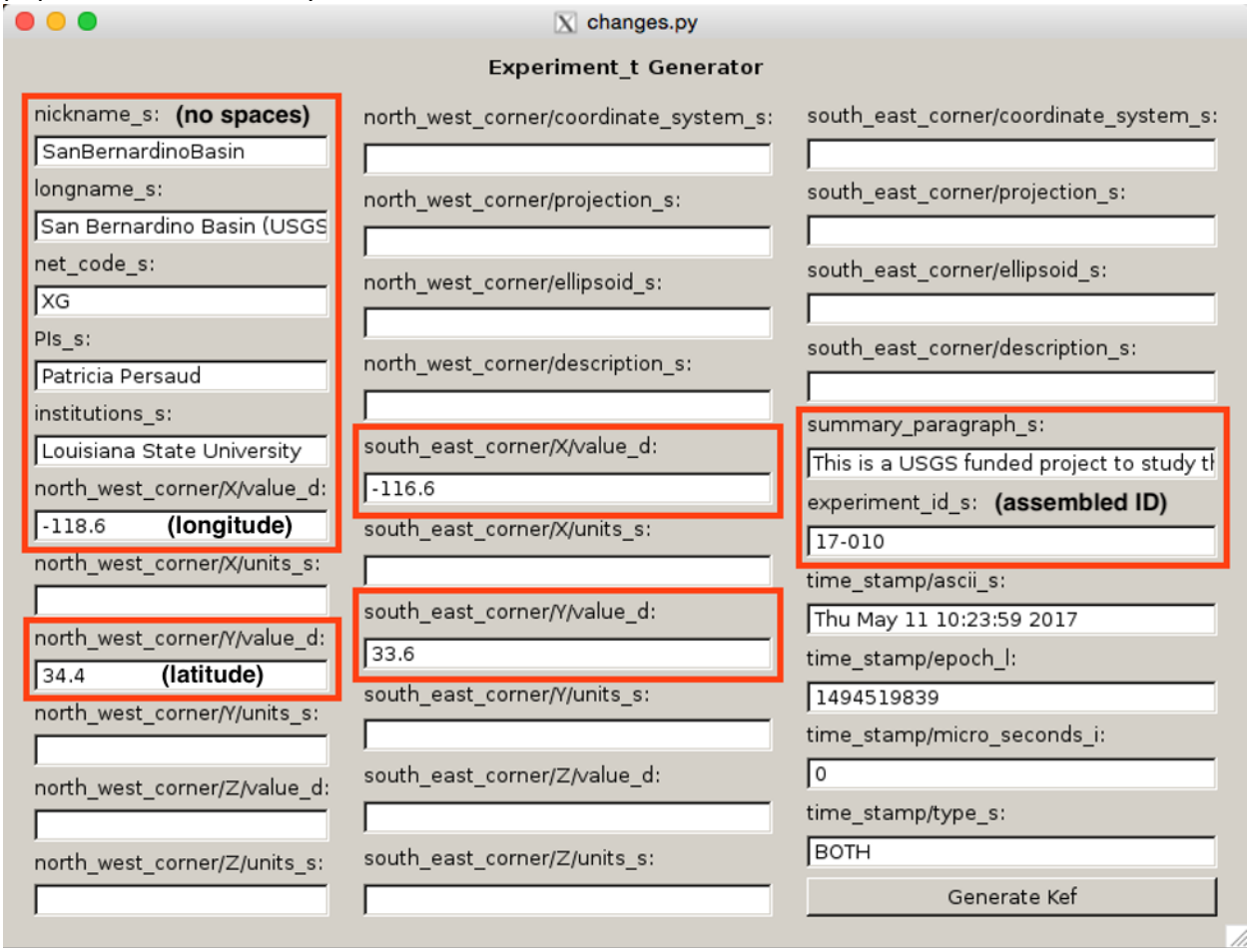
2.c. Load data (SmartSolo and Fairfield)

Click Browse to the right of MAP/RAW list file and select either a Solo input file (segd.list or smartsolo_map) or a Fairfield input file (rg16.txt). Next, click Browse to the right of Processing directory and select the PH5 directory you made earlier Then, click File -> Set Timeout, enter 200000, and click OK. If your window lengths are short, click File -> Combine # of SEG-D traces in PH5 and enter the appropriate integer (see preparing files). If you selected a Fairfield input file, also click File -> UTM Zone, enter the appropriate zone, and click OK.

Then, click Run. If you selected a Fairfield input file, the processing directories will appear within a few minutes. If you selected a Solo input file and did not use optional step 1.a.i, the processing directories may take several minutes to over an hour to appear.
Wait until all processing directories in the GUI show the Log button and the lower left corner of **pforma** shows the message 'All processes finished. Wrote pforma.cfg'. Close **pforma**, then start it again and repeat the steps from Browse to Run. Repeat as needed, including closing and starting **pforma** for each input list, until you have fed **pforma** all the input file lists. Then, click File -> Merge. After both the Merge Progress and Completed popups appear in **pforma**, you can close the popups and **pforma**. Go to step 3.

3. Experiment table (SmartSolo and/or Fairfield)

Generate and load the experiment table. Note: you need to have a network code and assembled ID before you can complete this step.
Move into the combined directory:
cd PH5_Archive/Sigma
Start the experiment table generation GUI:
experiment_t_gen
The required fields are noted in red boxes, below. The last four fields on the right column are populated automatically and these fields should not be modified.



After filling out the required fields, click Generate Kef and save the file as experiment.kef. After exiting, load the kef:
keftoph5 -n master.ph5 -k experiment.kef

There is a kef2ph5.log file that tracks the filenames loaded, the timestamp, and the last time the file was modified before loading.
If your experiment used active sources and you are loading events, go to step 4. If not, go to step 5.

4. Load event metadata (SmartSolo and/or Fairfield)

4.a. Preparing event metadata

Events in the PH5 can include explosive sources, hammer strikes, local earthquakes, or other sources of interest during the experiment. Prepare a .csv file with the following required fields: event array number, event ID, latitude, longitude, elevation, and UTC time. Optional fields include description, event size & units, and depth. You can have one .csv file per event array, or include all event arrays in the same .csv file. If you made a metadata directory earlier, place the .csv file(s) there. Tables describing the required and optional fields are below.

Required fields	**Description**
Array	Event line/array number. Numeric, $1 \leq n \leq 999$.
Event ID (id_s)	SEG-Y compatible event number, less than 2,147,483,647. Unique for experiment.
Latitude	Latitude in degrees
Longitude	Longitude in degrees
Elevation	Elevation in meters
Event time	Time formatted as *YYYY:JJJ:HH:MM:SS.ssssss* where JJJ is day of year, or *YYYY-MM-DDTHH:MM:SS.ssssss*

Optional fields	**Description**
Event size	Numeric source size if available.
Event size units	Units for source size if available: kg, g, lb, magnitude, etc.
Depth	Source depth below surface in meters. Below surface is positive, above is negative.
Description	ASCII text field for providing additional information about a source. Limit to 80 characters.

Notes: having more than 20 event arrays and more than 5 station arrays, or any equivalent combination, is considered an edge case and may not be supported by PH5. Station array numbers and event array numbers can be re-used, e.g., station array 2 and event array 2 can exist in the same PH5, and a station ID 2002 and event ID 2002 can also exist in the same PH5. However, a PH5 cannot have two event IDs 2002, nor can event ID 2002 be assigned to event array 2 and event array 4, only one event array.

4.b. Check and load event metadata

Once your .csv file is ready, start the **noven** GUI:

noven

Select File -> Open and select the event .csv file.
Select File -> Change Parameters. In the menu that comes up, click on the Input Type dropdown, select 'event', click Apply, and then click Close.
Next, use the dropdown above each column to specify which column is the event array, event ID, etc. A brief description of the column shows in the lower left corner when you hover over each option in the dropdown. When you are finished, select File -> Change Parameters, and for Skip Lines enter 1. Click Apply, and then click Close.
Next, select File -> Check input. **Noven** will verify all fields in the open .csv file. If the file passes verification, a popup with the message 'Nothing funky found!' will appear. Click OK, then select File -> Save As to save as a kef (e.g., events.kef) that can be loaded into PH5. If the file does not pass verification, a message window will appear with information about the affected line(s) and column(s).
If you have multiple event arrays and opted to make one .csv per event array, repeat these steps as needed (e.g., events_1.kef, events_2.kef, etc.). Close **noven** when you are done.
Next, load the event kef(s) into the PH5. Make sure you are in the Sigma directory.
keftoph5 -n master.ph5 -k /path/to/events.kef

Go to step 5.

5. Load responses (SmartSolo and/or Fairfield)

Request from the Nominal Response Library (NRL) version 1, compile your own RESP file in the same format as NRL version 1, or contact epic.data.group@earthscope.org to request the response file(s).
Place the RESP file(s) in the metadata directory if you made one, or in the Sigma directory.
Do NOT use response files from the new NRL web service because they are incompatible with the ph5tostationxml code.
Notes: Make sure that stations are separated into different arrays if they have different node types, sample rates, or gains. For Fairfields, do NOT use the NRL RESP files because their bit weight handling in PH5 will result in amplitudes that are incorrect by orders of magnitude. Use RESP provided by EPIC instead.

Make sure you are in the Sigma directory, then run
resp_load -n master.ph5 -a <comma separated list of arrays>

where the array list is non-zero-padded array numbers, such as 1,5,9.
If this command finishes and produces an input.csv without any warnings about no DAS table found, proceed. The input.csv contains the framework of essential response information for all nodes & settings (sample rate, gain) used in the PH5. You complete the Das RESP path and, if using SmartSolos, the Sensor RESP path. If the **resp_load** command finishes with warnings about no DAS table found, do not proceed further and contact epic.data.group@earthscope.org for assistance. If the input.csv only has the column description row after this step, do not proceed further and contact the EPIC data group at the above email for assistance. An example of an input.csv file generated by resp_load:

```
Das Model, Sensor Model, Sample Rate, sample rate multiplier,Gain, Das RESP path, Sensor RESP path
SmartSoloIGU16,GS-30CT,1000,1,12,,
ZLAND3C,,1000,1,12,,
```

Below is an example of an input.csv file after the RESP file names have been added; in this example, the RESP files exist in the Sigma directory. The RESP files may exist elsewhere on the machine and can be referenced by relative or absolute filepath. Note that SmartSolos have a separate Das and Sensor RESP file, and the Fairfields have one RESP file, as well as the required trailing comma at the end of the Fairfield row. Save the updated input.csv file.

```
Das Model, Sensor Model, Sample Rate, sample rate multiplier,Gain, Das RESP path, Sensor RESP path
SmartSoloIGU16,GS-30CT,1000,1,12,RESP.XX.DT100..GPZ.SSOLO_IGU16.4.1000.MP.OFF,RESP.XX.NS680..SPZ.DTSOLO.5.1850.43000.76_6
ZLAND3C,,1000,1,12,RESP.YE.N303.GPZ.1000SPS.12DB,
```

Then, run:
resp_load -n master.ph5 -a <comma separated list of arrays> -i input.csv

This will load the RESP files, update response references in the array table, and also update receiver table references in the array table using default orientations. Go to step 6 (optional but strongly recommended), or go to step 7.

6. Initial validation (SmartSolo and/or Fairfield)

This involves validating the PH5 and the stationXML. PH5 validation is included with PH5. The stationXML validator can be downloaded from the EarthScope github, and it is included with PASSOFT installed after 2021. It is not included with PH5.
To validate the PH5:
ph5_validate -n master.ph5

Progress information will be printed to screen, and the validation information will be saved in ph5_validate.log.
To extract and validate the stationXML:
ph5tostationxml -n master.ph5 -o exp-resp1.xml --level=response
stationxml-validator --input exp-resp1.xml --ignore-warnings

On some systems, you may have to run the validator by:
java -jar ${CONDA_PREFIX}/jar/stationxml-validator.jar --input exp-resp1.xml --ignore-warnings

Make note of any errors and go to step 7.

7. Updating metadata tables (SmartSolo and/or Fairfield)

Check that the station coordinates, start & stop times, array numbers, station IDs, and channel orientations are as intended. To view the array tables in csv format, run
ph5tokef -n master.ph5 --all_arrays > arrays_all.kef
keftocsv -f arrays_all.kef -o arrays_all.csv

Examine the array tables with preferred viewer. Recommended checks when viewing array tables:

- Station coordinates
- If you swapped in different nodes at different times for the same station ID, check that all time epochs are represented and that the correct node ID and DAS serial number are referenced in the right epoch in the array table
- Start and stop times are generally correct: exceptions are accidental inclusion of huddle test data or a node not turning off when intended

Based on your checks, field notes, and initial validation results, then:

- If you need to update station coordinates or channel orientation, and/or change array number, station ID, or start or stop times from what was in segd headers, go to step 7.a.
- If you only need to change channel orientation from the default (Z +90 dip, 1/N 0 azimuth, 2/E 90 azimuth) go to step 7.b.
- If you need to correct errors in the event table, go to step 7.c.
- If you need to correct errors in the experiment or response table, go to Appendix B.
- If all station coordinates and other metadata are as intended, go to step 8.

Note: if you prefer to edit csvs in Microsoft Excel or any other software that detects column format and assigns attributes, proofreading for un-commanded or undesirable changes in a plain text editor is strongly recommended.

7.a. Update array table

Open csv in preferred editor. Update the relevant metadata. Coordinates must be in decimal degrees (latitude & longitude) and meters (elevation). Note that location/X is longitude and location/Y is latitude; see Appendix C for a table fully describing station metadata requirements. Please note that if you update start and/or stop times, you need to modify both the ascii and epoch timestamp columns. If you did optional step 6, correct any errors (e.g., station ID too large error from ph5_validate, overlapping station epochs from stationxml validator). For each station requiring updates e.g. to coordinates, change all three channels. Do not modify columns unless needed or modify the format used for columns (such as the ASCII date), as doing so can invalidate the PH5. Save the csv under a new filename and convert to a kef:
csvtokef -f arrays_updatedv1.csv -o arrays_updatedv1.kef

Note: the channel orientation convention 1,2 (e.g., GP1, GP2) is recommended to follow GSN convention and is required if nodes were installed more than 5 degrees from geographic north. GP1 is nominally N-S and GP2 is nominally E-W.
If some nodes were installed not oriented to geographic north, go to step 7.b. If all nodes were installed oriented to geographic north but you followed step 6 and have one or more errors in the experiment or event table, go to step 7.c. Otherwise, go to step 7.d.

7.b. Update receiver and array table

The receiver table will need to be edited for this step, as well as the array tables.

ph5tokef -n master.ph5 -C > receiver_original.kef
keftocsv -f receiver_original.kef -o receiver_original.csv

Open this csv in preferred editor.
Use field notes to populate orientations in addition to the default 0,90. Important: do NOT change the first row. Below is an example of an updated receiver table csv with notes:

	A	B	C	D	E	F	G	H	I	J	K	L
1	table	orientation/azimu	orientation/az	orientation/dip	orientation/di	orientation/descri	orientation/channel_number_i		index (implicit)		Stations	
2	/Experiment_g/Receivers_g/Receiver_t	0	degrees	90	degrees	Z	0		0			
3	/Experiment_g/Receivers_g/Receiver_t	0	degrees	0	degrees	N	0		1		1030, 1045, 1062, 1071, 1079	
4	/Experiment_g/Receivers_g/Receiver_t	90	degrees	0	degrees	E	0		2			
5	/Experiment_g/Receivers_g/Receiver_t	9	degrees	0	degrees	N	0		3		1000, 1001, 1016, 1020, 1027	
6	/Experiment_g/Receivers_g/Receiver_t	99	degrees	0	degrees	E	0		4			
7	/Experiment_g/Receivers_g/Receiver_t	356	degrees	0	degrees	N	0		5		1002	
8	/Experiment_g/Receivers_g/Receiver_t	86	degrees	0	degrees	E	0		6			
9	/Experiment_g/Receivers_g/Receiver_t	23	degrees	0	degrees	N	0		7		1004, 1092	
10	/Experiment_g/Receivers_g/Receiver_t	113	degrees	0	degrees	E	0		8			

The index(implicit) and stations columns are for in-progress tracking only and if used, must be removed prior to saving the finalized version and converting to a kef, as shown below:

	A	B	C	D	E	F	G	H
1	table	orientation/azimut	orientation/az	orientation/dip/	orientation/di	orientation/descri	orientation/channel_number_i	
2	/Experiment_g/Receivers_g/Receiver_t	0	degrees	90	degrees	Z	0	
3	/Experiment_g/Receivers_g/Receiver_t	0	degrees	0	degrees	N	0	
4	/Experiment_g/Receivers_g/Receiver_t	90	degrees	0	degrees	E	0	
5	/Experiment_g/Receivers_g/Receiver_t	9	degrees	0	degrees	N	0	
6	/Experiment_g/Receivers_g/Receiver_t	99	degrees	0	degrees	E	0	
7	/Experiment_g/Receivers_g/Receiver_t	356	degrees	0	degrees	N	0	
8	/Experiment_g/Receivers_g/Receiver_t	86	degrees	0	degrees	E	0	
9	/Experiment_g/Receivers_g/Receiver_t	23	degrees	0	degrees	N	0	
10	/Experiment_g/Receivers_g/Receiver_t	113	degrees	0	degrees	E	0	

Open the array csv; if you needed step 7.a., use that csv. Go to the receiver_table_n_i column and enter the relevant receiver table indices for each station. Note the receiver table starts with an implied index of 0, which is the Z channel dip lookup.
Save the csvs under new filenames and then convert them to kefs:

csvtokef -f arrays_updatedv2.csv -o arrays_updatedv2.kef
csvtokef -f receiver_updated.csv -o receiver_updated.kef

If you followed step 6 and have one or more errors in the event table, go to step 7.c. Otherwise, go to step 7.d.

7.c. Update event table

If you need to correct any errors in the event table(s), run

ph5tokef -n master.ph5 --all_events > events_original.kef
keftocsv -f events_original.kef -o events_original.csv

Open this csv in a preferred editor, and make the needed changes. Save under a new filename and convert to a kef:
csvtokef -f events_updatedv1.csv -o events_updated.kef

Go to step 7.d.i.

7.d.i. Remove the tables being replaced

PH5 automatically retains what is loaded, so tables being replaced first need the old version deleted from the PH5.
If you followed step 7.a and/or step 7.b:
delete_table -n master.ph5 --all_arrays

If you followed step 7.b, also run:
delete_table -n master.ph5 -C

If you followed step 7.c, run:
delete_table -n master.ph5 -V n

Where n is the event line number for each event line in the PH5. If you installed or updated PH5 after mid-February 2025 and plan to replace all event lines, you can run:
delete_table -n master.ph5 --all_events

After you have finished deleting the table(s) you will replace, go to step 7.d.ii.

7.d.ii. Load new tables

If you followed step 7.a and/or step 7.b, run:
keftoph5 -n master.ph5 -k arrays_updatedv2.kef

If you followed step 7.b., also run:
keftoph5 -n master.ph5 -k receiver_updated.kef

If you followed step 7.c, run:
keftoph5 -n master.ph5 -k events_updated.kef

Go to step 8.

8. Make ancillary tables (SmartSolo and/or Fairfield)

Generate ancillary tables for PH5 to use. Run:
sort_kef_gen -n master.ph5 -a > sort_t.kef

This will generate a sort table to optimize searches.
If the experiment has any events (you used step 4), also create the event-station offset table by running:
geo_kef_gen -n master.ph5 > offset.kef

Go to step 9.

9. Final validation & loading ancillary tables (SmartSolo and/or Fairfield)

This involves validating the PH5 and the stationXML. PH5 validation is included with PH5. The stationXML validator can be downloaded from the EarthScope github, and it is included with PASSOFT installed after 2021. It is not included with PH5.
To validate the PH5:
ph5_validate -n master.ph5

Progress information will be printed to screen, and the validation information will be saved in ph5_validate.log. Note that if you followed optional step 6 that ph5_validate.log will be overwritten unless you change the filename.
To extract and validate the stationXML:
ph5tostationxml -n master.ph5 -o exp-resp1.xml --level=response
stationxml-validator --input exp-resp1.xml --ignore-warnings

On some systems, you may have to run the validator by:
java -jar ${CONDA_PREFIX}/jar/stationxml-validator.jar --input exp-resp1.xml --ignore-warnings

If there are any errors, go to step 7 and follow as needed to fix the errors.
If you did step 7.b., check that azimuth and dip values are as expected. If any are incorrect, go to step 7 and follow as needed to correct. If you need to go through step 7.a and/or step 7.c, repeat step 8.
Now, load the table(s) from step 8.
keftoph5 -n master.ph5 -k sort_t.kef

If applicable, also run
keftoph5 -n master.ph5 -k offset.kef

Go to step 10.

10. Test data extraction (SmartSolo and/or Fairfield)

Output miniseed, such as the following run from the Sigma directory:

ph5toms -n master.ph5 -a 009 -s 2024-04-10T00:00:00 -t 2024-04-11T23:59:59 -o ms_test -F MSEED

where -a 009 is the station array (leading zeroes required), -s and -t define start and end time respectively, -o defines the output directory, and -F defines the format. See ph5toms -h for details of the available flags.

If you followed step 4, also test SEGY output, such as the following run from the Sigma directory:

ph5toevt -n master.ph5 --use_deploy_pickup -o gather_test -l 10 -A 9 -e 12002 --shot_line 5

where -l defines length in seconds, -A is the station array, and -e defines the event number. See ph5toevt -h for details of the available flags.

Appendix A. Tips to prepare for multiple pforma runs

If you are building a PH5 and did not use uniform settings throughout the deployment, this appendix contains some examples and tips for data handling with respect to multiple runs of pforma. Note that EPIC has not tested all possible combinations; if you have a combination type not listed here, you can contact epic.data.group@earthscope.org for recommendations. If your only settings change was sample rate, you can use a single input list but each sample rate needs to be in its own array.

A.1 Multiple node models (Fairfield and SmartSolo)

Case brief description: Used both FF and Solo and used same settings for all nodes of same type. Follow the directions in the main document for steps 1 & 2.
Examples of this case:

- Fairfield nodes in UTM zone 13N at 18 dB gain and 500 sps in array 2 and SmartSolo nodes at 18 dB gain and 500 sps in array 1
- Fairfield nodes in UTM zone 11N at 6 dB gain and 500 sps in array 7 and SmartSolo nodes at 12 db gain and 1000 sps in array 5
- Fairfield nodes in UTM zone 17S at 36 dB gain in array and 250 sps in array 4 and SmartSolo nodes at 0 db gain and 250 sps in array 2

Why separate input lists and pforma runs are required: to ensure the correct segdtoph5 routines are used, including UTM conversion for the Fairfield node data.

Does this need to be planned at the offloading steps: Yes, strongly recommended. If both node types have been assigned the same array number and re-offload is not available, you will need to update the array numbering in the array tables so both node types are not in the same array. Not doing so risks incorrect response loading and will make troubleshooting the PH5 more difficult. However if you used, for example, line 1 station 101 in the SmartSolo offload and line 1 station 101 in the Fairfield offload and any of their respective deployments overlapped, you *must* re-offload.

A.2 Multiple UTM zones (Fairfield only)

Case brief description: Used Fairfield only across more than one UTM zone, and experiment settings were otherwise uniform.
Examples of this case:

- List 1 for UTM zone 17S in array 1, list 2 for UTM zone 18S in array 3, and list 3 for UTM zone 17N in array 1
- List 1 for UTM zone 18N in array 5 and list 2 for UTM zone 17N in array 2

Why separate input lists and pforma runs are required: If all UTM zones are included in one list, coordinates will not convert correctly and the affected array table entries will require updating.

Does this need to be planned at the offloading steps: Preferably. If not done, you can use your notes to separate the files into lists by UTM zone. Note that for this type case, stations in

different UTM zones may be in the same array if that is the only difference. If that is not the only difference, see A.3 instead.

A.3 Non-uniform settings (Fairfield and/or SmartSolo)

Case brief description: Used Fairfield and/or SmartSolo and did not use the same settings for all nodes of same type.
Examples of this case:

- Fairfield nodes in UTM zone 11N at 12 dB gain & 500 sps in array 1 and Fairfield nodes in UTM zone 11N at 18 dB gain & 500 sps in array 5
- SmartSolo nodes at 24 dB gain & 250 sps in array 2 and SmartSolo nodes at 6 dB gain & 1000 sps in array 3
- Fairfield nodes in UTM zone 11N at 12 dB gain & 500 sps in array 3 and Fairfield nodes in UTM zone 12N at 12 dB gain & 500 sps in array 3 and SmartSolo nodes at 12 dB gain & 500 sps in array 1
- Fairfield nodes in UTM zone 13N at 18 dB gain & 1000 sps in array 2 and Fairfield nodes in UTM zone 12N at 12 dB gain & 1000 sps in array 1

Why separate input lists and pforma runs are required: Different gains in a single pforma run result in a malformed response table and a potentially invalid PH5.
Does this need to be planned at the offload step: Yes.

Additional Notes:
If you used SmartSolos with different gains, you need to create a prospect in SoloLite for each gain used to populate the gain correctly in the segd headers. If you used Fairfields with different gains, having one gain per array may be sufficient to correctly populate the gain in the segd headers.
Special case: Multiple deployment (e.g. node A at line 1 station 1001 for 30 days, then replaced by node B at line 1 station 1001 for 30 days) type of experiment but changed the gain partway through. For example, node A ran at 12 dB gain and node B at 18 dB gain. The simplest way to handle this is to change the line and/or station ID, such as node A is line 1 station 1001 and node B is line 2 station 1001. Keep in mind the note above regarding gain in segd headers. If you have this case and would like advice on handling it, please contact epic.data.group@earthscope.org. Do *not* attempt to build a PH5 with line 1 station 1001 at 12 dB and line 1 station 1001 at 18 dB (for example) in two different input lists: doing so will result in runaway processes and eventually a non-responsive machine.

Appendix B. Updating the experiment and/or response table

If you need to correct any errors in the experiment table, run
ph5tokef -n master.ph5 -E > experiment_original.kef

The experiment table is short enough that you can open it as-is in a text editor with the .kef extension. Make the changes required to fix the errors, then save with a new filename (experiment_updated.kef).
Then delete the old table:
delete_table -n master.ph5 -E

Load the new version:
keftoph5 -n master.ph5 -k experiment_updated.kef

If you discovered you loaded the incorrect response file(s) and need to correct the response table, make a copy of the original input.csv if desired, then run (following command will overwrite existing input.csv):
resp_load -n master.ph5 -a <comma separated list of arrays>

Complete the Das RESP path and, if using SmartSolos, the Sensor RESP path, as you did in step 5. Then, run
resp_load -n master.ph5 -a <comma separated list of arrays> -i input.csv -r

Appendix C. Station metadata requirements table

The fields are listed in the order in which they appear in the array table(s).

Field name	Description	Directions and Limits
table	Array or line number for stations (e.g., /Experiment_g/Sorts_g/Array_t_001)	Numeric, equal to or less than 999; do not change non-numeric parts of the field
id_s	SEG-Y compatible station number	Numeric, less than 32,767
location/X/value_d	Station longitude in decimal degrees	Positive or negative number, must be the same for all channels of a station
location/X/unit_s	Specify longitude units	Default is degrees, do not change
location/Y/value_d	Station latitude in decimal degrees	Positive or negative number, must be the same for all channels of a station
location/Y/unit_s	Specify latitude units	Default is degrees, do not change
location/Z/value_d	Station elevation in meters	Numeric, must be the same for all channels of a station.
location/Z/unit_s	Specify elevation units	Default is unknown; can update to m
location/coordinate_system_s	Optionally provide a coordinate system	Default is geographic, recommend leaving as default
location/projection_s	Optionally provide a datum name	Default is WGS84. If Fairfield, leave as default since that is the reference used in converting from UTM
location/ellipsoid_s	Optionally provide a geoid name	Default is blank, can be left blank
location/description_s	List information about the station location	Default is 'Converted from UTM Zone' (Fairfield) or 'Read from SEG-D as-is'

		(SmartSolo). Can leave as default. If modified, this field is limited to ~80 ASCII characters and the contents will show in the Site column/field in EarthScope DMC metadata queries (MDA, station query, etc.)
deploy_time/ascii_s	Deploy time in ASCII format	Date and time in whole seconds, such as Wed Apr 24 16:26:25 2024; if day is less than 10, space-pad (e.g., Wed Apr 3 16:26:25 2024); if epoch deploy time is updated, this field must be updated to match and it must be the same for all channels of a station
deploy_time/epoch_l	Deploy time in epoch seconds	Must be whole seconds in reference to standard epoch time; if ASCII deploy time is updated, this field must be updated to match and it must be the same for all channels of a station
deploy_time/micro_seconds_i	If applicable, decimal seconds	Must be a number equal to or less than 999999; in most cases can be left as default and must be the same for all channels of a station
deploy_time/type_s	None	Default is BOTH, do not change
pickup_time/ascii_s	Pickup time in ASCII format	Date and time in whole seconds, such as Wed May 22 06:00:00 2024; if day is less than 10, space-pad (e.g., Wed May 1 06:00:00 2024); if epoch pickup time is updated, this field must be updated to match and it must

		be the same for all channels of a station
pickup_time/epoch_l	Pickup time in epoch seconds	Must be whole seconds in reference to standard epoch time; if ASCII deploy time is updated, this field must be updated to match and it must be the same for all channels of a station
pickup_time/micro_seconds_i	If applicable, decimal seconds	Must be a number equal to or less than 999999; in most cases can be left as default and must be the same for all channels of a station
pickup_time/type_s	None	Default is BOTH, do not change
das/serial_number_s	Full datalogger serial number	DAS serial number as marked in raw data headers, varies by manufacturer; do not change
das/model_s	Datalogger model	Text string; e.g., SmartSolo IGU16, can leave as default
das/manufacturer_s	Datalogger manufacturer	Text string; e.g., SmartSolo, can leave as default
das/note_s	None	Can leave as default
sensor/serial_number_s	Sensor serial number	Varies by manufacturer
sensor/model_s	Sensor model	Can leave as default, do not remove.
sensor/manufacturer_s	Sensor manufacturer	Can leave as default, do not remove.
sensor/note_s	None	Can leave as default
description_s	Information about serial number and node unit	Can leave as default; do not change either the DAS or node ID unless you have swapped in different node IDs at different times for the

		same DAS serial number and are adding lines to the array table to accurately reflect the swap times in which case the Node ID can be updated but the DAS cannot
seed_band_code_s	SEED-defined character for band (sample rate-instrument corner)	1 letter, default should be correctly assigned as per SEED convention
sample_rate_i	Sample rate for each recorded channels	Integer, must be greater than 1, do not change
sample_rate_multiplier_i	Multiplier for sample rate	1, do not change
seed_instrument_code_s	SEED-defined character for instrument	1 letter, default should be correctly assigned as per SEED convention
seed_orientation_code_s	SEED-defined character for channel orientation	1 letter or number; defaults to N,E,Z for SmartSolo or 1,2,Z for Fairfield; use of 1,2,Z (1 corresponds to N, 2 to E) is strongly recommended to align with GSN and EPIC SEED convention
seed_location_code_s	Optionally, 2-character SEED-defined location code	Default is blank, leaving as default is strongly recommended; cannot use different location codes within a station
seed_station_name_s	SEED compatible station name	Alphanumeric, 1-5 characters, usually the same as id_s; if alphabetical must be all capital letters
channel_number_i	Channel number of each component	Do not change
receiver_table_n_i	Index to look up in the receiver table for channel's numeric dip and azimuth	Automatically populated during resp_load; do not change unless following step 7.b

response_table_n_i	Index to look up in the response table for response information	Automatically populated during resp_load; if arrays are set up correctly for different parameters, leave as-is