PIC KITCHEN Cookbook

Draft: May 30, 2008

Processing steps to build ph5

Overview: This document outlines the steps to convert data from a seismic reflection or refraction experiment using the RefTek texan instruments to PASSCAL HDF5 (ph5) format. This example uses data from the Spears Ranch experiment in Oklahoma. Commands that are executed are shown in blue.
The steps required are to a) bring together the required files and information, in this case these include the raw data files, the TSP input files (geometry, das, and shot lists), and information about the experiment from the PIs, b) read this information into the ph5 file, c) verify the ph5 file, d) generate two text files, data_description.txt, and data_request_key.txt (used to generate the DMC's web form), and e) transmit the ph5 file to the DMC.

The method described here should be considered preliminary as it is likely to be greatly simplified as more experience with its use is gained.

1) Bring together the needed input files, raw TRD files, das file, shot file, geometry file (TSP input files). In addition, get information about the experiment, such as PI's names and institutions, the purpose of the experiment, the size and depth of the shots, and the type of geophones used.

2) Run tsp2dep to convert the TSP input files into a dep file as produced by rawmeet. This step is not necessary on PASSCAL run experiments using the rawmeet program.

```
usage: tsp2dep -u tsp_utm_file -d tsp_das_file -s tsp_shot_file

Convert TSP input files to a dep file. Writes to stdout.

options:
  -h, --help            show this help message and exit
  -u tsp_utm_file, --utm=tsp_utm_file
                        A TSP geometry file. "station, northing, easting,
                        elevation"
  -d tsp_das_file, --das=tsp_das_file
                        A TSP DAS file. "station, das_number"
  -s tsp_shot_file, --shot=tsp_shot_file
                        A TSP shot file. "year, doy, hr, mn, sc, station,
                        length"
  -D
```

```
flow@localhost% tsp2dep -s spears.shot -u spears.utm -d spears.das > SOAD.dep
```

3) Create a kef (Kitchen Exchange Format) file to describe the experiment. Each tabbed in section of the kef file describes, in key value pairs, a line in the preceding table line. In the following example a single line of the /Experiment_g/Experiment_t is modified. See example below.

```
#   Start of file SOAD-experiment.kef
/Experiment_g/Experiment_t
        PIs_s = Dr. G. Randy Keller, Dr. Steven H. Harder, Dr. Kate C. Miller,\
        Dr. Catherine M. Snelson
        institutions_s = University of Oklahoma, University of Texas, El Paso,\
        University of Nevada at Las Vegas
        longname_s = Spears Ranch Seismic Experiment Survey
        nickname_s = SOAD
        north_west_corner/X/units_s = degrees
        north_west_corner/X/value_d = -96.654163
        north_west_corner/Y/units_s = degrees
        north_west_corner/Y/value_d = 34.452259
        north_west_corner/Z/units_s = meters
        north_west_corner/Z/value_d = 315
        north_west_corner/coordinate_system_s = mercator
        north_west_corner/description_s = Estimated elevation
        north_west_corner/ellipsoid_s = WGS84
        north_west_corner/projection_s = None
        south_east_corner/X/units_s = degrees
        south_east_corner/X/value_d = -96.645916
        south_east_corner/Y/units_s = degrees
        south_east_corner/Y/value_d = 34.435873
        south_east_corner/Z/units_s = meters
        south_east_corner/Z/value_d = 315
        south_east_corner/coordinate_system_s = mercator
        south_east_corner/description_s = Estimated elevation
        south_east_corner/ellipsoid_s = WGS84
        south_east_corner/projection_s = None
        summary_paragraph_s = A 2-D seismic reflection experiment on the Spears Ranch\
        in Oklahoma. The purpose is to image the Arbuckle-Simpson aquifer.
        time_stamp/ascii_s = Thu Jun 28 14:58:00 2007
        time_stamp/epoch_l = 1183064280
        time_stamp/micro_seconds_i = 0
        time_stamp/type_s = "BOTH"
#   End of file
```

To update the line keyed on time_stamp/epoch_l substitute the line:
/Experiment_g/Experiment_t:Update:time_stamp/epoch_l
To delete the first line containing time_stamp/epoch_l with the value in the kef file substitute the line:
/Experiment_g/Experiment_t:Delete:time_stamp/epoch_l

Warning: You can not delete a line from the table if it is the only line.

4) Run initialize-ph5 to create an empty ph5 file for the experiment processing. See kef file above.

```
usage: initialize-ph5 [--help]--kef=kef_file --nickname=output_file

Program to initialize PH5 file at start of experiment. The kef file should
contain information for experiment table /Experiment_g/Experiment_t.

options:
  -h, --help            show this help message and exit
  -n output_file, --nickname=output_file
                        Experiment nickname.
  -k kef_file, --kef=kef_file
                        Kitchen Exchange Format file containing experiment
                        info.
```

```
flow@localhost% initialize-ph5 -n SOAD -k SOAD-experiment.kef
```

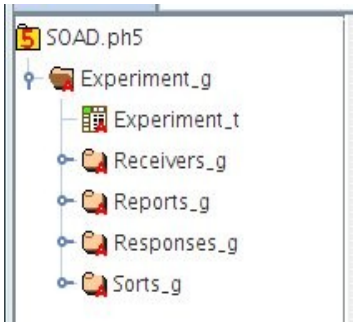

Illustration 1: ph5 file tree as viewed in hdfview

5) Process the raw TRD files and the dep file into the ph5 file using 125a2ph5. The file "SOAD-file-list.txt" contains a list of the raw files, the file SOAD.dep contains much of the meta-data for the experiment in dep (rawmeet) format. For an experiment using the rawmeet program the dep file would contain all of the meta-data.

```
usage: 125a2ph5 [--help][--dep dep_file][--kef kef_file][--raw raw_file | --file
file_list_file] --nickname output_file_prefix

Read the raw texan files and optionally a kef file into ph5 format.

options:
  -h, --help            show this help message and exit
  -r raw_file, --raw=raw_file
                        RT-125(a) texan raw file
  -f file_list_file, --file=file_list_file
                        File containing list of RT-125(a) raw file names.
  -n output_file_prefix, --nickname=output_file_prefix
                        The ph5 file prefix (experiment nick name).
  -k kef_file, --kef=kef_file
                        Kitchen Exchange Format file.
  -d dep_file, --dep=dep_file
                        Rawmeet dep file.

This program takes as input a file containing a list of raw TRD files and the dep
file created above.

flow@localhost% head SOAD-file-list.txt
RAW/I2631RAW.TRD
RAW/I2810RAW.TRD
RAW/I1554RAW.TRD
...
RAW/I2496RAW.TRD

flow@localhost% head SOAD.dep
# tsp2dep Version: 2007.177 dep Version: 0007.107 TSP Version: 3.1 Run: Tue Jun 26 09:16:46
2007
SHOT;1000;1000;1;N34.450543;W96.651557;306.116000;2007:151:15:01:0;;;;;;;;
SHOT;1002;1002;1;N34.450183;W96.651565;307.655000;2007:151:15:07:0;;;;;;;;
...
RECV;12839;2036;2;texan;1;;;N34.448596;W96.645643;308.465000;;;;;
```

```
flow@localhost% 125a2ph5 -n SOAD -d SOAD.dep -f SOAD-file-list.txt
```

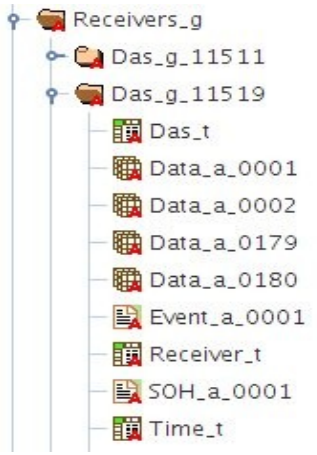

Illustration 2: Receivers group tree

6) Calculate offsets and insert them into the ph5 file using geod2kef then kef2ph5. This can only be done if geometry info was included in the dep file.

```
usage: geod2kef --nickname output_file_prefix [--path][-h][--listellipsoids][--
listunits][-U units][-E ellipsoid]

Read locations and calculate offsets from events to receivers. Produce kef
file to populate ph5 file.

options:
  -h, --help            show this help message and exit
  -n output_file_prefix, --nickname=output_file_prefix
                        The ph5 file prefix (experiment nick name).
  -p output_file_path, --path=output_file_path
                        Path to directory containing ph5 files. Defaults to
                        current directory
  -U output_units       Units to output offsets in. (Use -u to get list of
                        acceptable units.) Default == 'm' (meters)
  -E calculation_ellipsoid
                        Ellipsoid to use. (Use -e to get a list of acceptable
                        ellipsoids.) Default == 'WGS84'
  -e, --listellipsoids  List available ellipsoids.
  -u, --listunits       List all available output units.
```

Geometry is in geodetic units and meters, which are the defaults for geod2kef, rawmeet, as well as accommodated in ph5.

```
flow@localhost% geod2kef -n SOAD > SOAD-Offset_t.kef
```

Note: geod2kef writes to stdout.

Next insert the offsets into the ph5 file using kef2ph5.

```
usage: kef2ph5 --kef kef_file --nickname ph5_file_prefix [--path path]

Update a ph5 file from a kef file.

options:
  -h, --help            show this help message and exit
  -n OUTFILE, --nickname=OUTFILE
                        The ph5 file prefix (experiment nickname).
  -k KEFFILE, --kef=KEFFILE
                        Kitchen Exchange Format file.
  -p PATH, --path=PATH  Path to directory where ph5 files are stored.
  -c, --check           Show what will be done but don't do it!
```

```
flow@localhost% kef2ph5 -n SOAD -k SOAD-Offset_t.kef -c
```

If no errors, then run:

```
flow@localhost% kef2ph5 -n SOAD -k SOAD-Offset_t.kef
```

azimuth.value_f	azimuth.units_s	event_id_s	offset.value_d	offset.units_s	receiver_id_s
0.0	degrees	1000	0.0	m	1000
-179.73778	degrees	1000	20.079	m	1001
-178.94537	degrees	1000	39.942	m	1002
-178.8575	degrees	1000	59.914	m	1003
-178.68362	degrees	1000	80.002	m	1004
-178.7359	degrees	1000	99.973	m	1005
-178.72571	degrees	1000	119.835	m	1006
-178.49594	degrees	1000	140.043	m	1007
-178.5181	degrees	1000	159.904	m	1008
-178.38997	degrees	1000	179.889	m	1009

Illustration 3: Offset table

7) Run time-kef-gen to generate time corrections and then kef2ph5 to insert them into the ph5 file.

```
usage: time-kef-gen --nickname ph5-file-prefix [-p path]

Generates a kef file to populate Time_t from SOH_A_.

options:
  -h, --help            show this help message and exit
  -n ph5_file_prefix, --nickname=ph5_file_prefix
                        The ph5 file prefix (experiment nickname).
  -p ph5_path, --path=ph5_path
                        Path to ph5 files. Defaults to current directory.
```

8) Since the input files were in TSP format and not dep format some of the information is not in the ph5 file yet. This is easy to add. The example below adds information about the sensors used. First dump a table to kef format, then edit the kef file and use it to update the ph5 file.

```
usage: tabletokef --nickname ph5-file-prefix options

Dump a table to a kef file.

options:
  -h, --help            show this help message and exit
  -n ph5_file_prefix, --nickname=ph5_file_prefix
                        The ph5 file prefix (experiment nickname).
  -p ph5_path, --path=ph5_path
                        Path to ph5 files. Defaults to current directory.
  -d
  -E, --Experiment_t    Dump /Experiment_g/Experiment_t to a kef file.
  -S, --Sort_t          Dump /Experiment_g/Sorts_g/Sort_t to a kef file.
  -O, --Offset_t        Dump /Experiment_g/Sort_g/Offset_t to a kef file.
  -V, --Event_t         Dump /Experiment_g/Sorts_g/Event_t to a kef file.
  -A n, --Array_t_=n    Dump /Experiment_g/Sorts_g/Array_t_[n] to a kef file.
  -R, --Response_t      Dump /Experiment_g/Responses_g/Response_t to a kef
                        file.
  -P, --Report_t        Dump /Experiment_g/Reports_g/Report_t to a kef file.
  -C das, --Receiver_t_=das
                        Dump /Experiment_g/Receivers_g/Das_g_[das]/Receiver_t
                        to a kef file.
  -D das, --Das_t=das   Dump /Experiment_g/Receivers_g/Das_g_[das]/Das_t to a
                        kef file.
```

(Note: The -C and -D options do not work with current file organization. A known bug.)

Get a copy of /Experiment_g/Sorts_g/Array_t_001.

```
flow@localhost% tabletokef -n SOAD -A 1 > SOAD-Array_t_001.kef
```

Now edit SOAD-Array_t_001.kef, using the editor of your choice, and replace each instance of
/Experiment_g/Sorts_g/Array_t_001
with
/Experiment_g/Sorts_g/Array_t_001:Update:id_s
and
sensor/model_s =
with
sensor/model_s = 4.5Hz Vertical

```
#   Start of file
#
#       Wed Jun 27 13:49:26 2007        ph5 version: 2007.121 Bleeding
#
#   Table row 1
/Experiment_g/Sorts_g/Array_t_001:Update:id_s
        id_s = 1000
        location/Y/value_d = 34.450543
        location/Y/units_s = degrees
        location/X/value_d = -96.651557
        location/X/units_s = degrees
        location/Z/value_d = 306.116
        location/Z/units_s = meters
        location/coordinate_system_s =
        location/projection_s =
        location/ellipsoid_s =
        location/description_s =
        deploy_time/ascii_s =
        deploy_time/epoch_l = -1
        deploy_time/micro_seconds_i = 0
        deploy_time/type_s =
        pickup_time/ascii_s =
        pickup_time/epoch_l = -1
        pickup_time/micro_seconds_i = 0
        pickup_time/type_s =
        das/serial_number_s = 11642
        das/model_s = texan rt125a
        das/manufacturer_s = RefTek
        das/notes_s =
        sensor/serial_number_s =
        sensor/model_s = 4.5Hz Vertical
        sensor/manufacturer_s = GeoSpace Corp.
        sensor/notes_s =
        description_s =
        channel_number_i = 1
#   Table row 2
/Experiment_g/Sorts_g/Array_t_001:Update:id_s
        id_s = 1001
        location/Y/value_d = 34.450362
        location/Y/units_s = degrees
```

```
...

#   Table row 77
/Experiment_g/Sorts_g/Array_t_001:Update:id_s
        id_s = 1076
        location/Y/value_d = 34.43691
        location/Y/units_s = degrees

...

#   End of file
```

Next run kef2ph5. This will update /Experiment_g/Sorts_g/Array_t_001, keying the update on the value of id_s. This is done by the line `/Experiment_g/Sorts_g/Array_t_001:Update:id_s` shown above.

```
flow@localhost% kef2ph5 -n SOAD -k SOAD-Array_t_001.kef -c
```
if no errors:
```
flow@localhost% kef2ph5 -n SOAD -k SOAD-Array_t_001.kef
```

Note: /Experiment_g/Sorts_g/Event_t should also be updated to contain information about the size and depth of the shots.

9) Populate /Experiment_g/Sorts_g/Sort_t to describe all of the shot windows by running sort-kef-gen.

```
usage: sort-kef-gen --nickname ph5-file-prefix --serial-number DAS-SN [--path path-
to-ph5-files]

Generate a kef file to populate Sort_t.

options:
  -h, --help            show this help message and exit
  -n ph5_file_prefix, --nickname=ph5_file_prefix
                        The ph5 file prefix (experiment nickname).
  -p ph5_path, --path=ph5_path
                        Path to ph5 files. Defaults to current directory.
  -s sn, --serial-number=sn
                        DAS to use to get windows.
  -d
```

Use hdfview to find an instrument that recorded all of the shot windows. In this case DAS 11519.

```
flow@localhost% sort-kef-gen -n SOAD -s 11519 > SOAD-Sort_t.kef
```

```
flow@localhost% kef2ph5 -n SOAD -k SOAD-Sort_t.kef -c
```
if no error
```
flow@localhost% kef2ph5 -n SOAD -k SOAD-Sort_t.kef
```

Now generate a small kef file to delete the first two lines of the Sort_t table that were inserted there by 125a2ph5 as place holders.

10) Create data_request_key.txt and data_description.txt. This is done by running report-gen. You will need to edit the resulting data_description.txt for formatting. These two files are used by the DMC to generate the web form for this data set.

```
usage: report-gen --nickname=ph5-file-prefix options

Generate data_description.txt and/or data_request_key.txt.

options:
  -h, --help            show this help message and exit
  -n ph5_file_prefix, --nickname=ph5_file_prefix
                        The ph5 file prefix (experiment nickname).
  -p ph5_path, --path=ph5_path
                        Path to ph5 files. Defaults to current directory.
  -k, --key             Write data_request_key.txt.
  -d, --description     Write data_description.txt.
  --bug
```

```
flow@localhost% report-gen -n SOAD -k -d
```

11) Insert data_description.txt, and data_request_key.txt into SOAD.ph5 using report2ph5.

```
usage: report2ph5 --file report-file --kef kef-file --nickname experiment-nickname
[--path path-to-kef-file]

Load a report (pdf) into a ph5 file.

options:
  -h, --help            show this help message and exit
  -f REPORT_FILE, --file=REPORT_FILE
                        The file containing the report, (pdf, doc, ps, etc.).
  -k KEF_FILE, --kef=KEF_FILE
                        Kef file describing row in Report_t for the report.
  -n NICKNAME, --nickname=NICKNAME
                        Experiment nickname.
  -p PATH, --path=PATH  Path to where ph5 files are stored

flow@localhost% cat SOAD-desc.kef
/Experiment_g/Reports_g/Report_t
       array_name_a = data_description
       description_s = Text file describing the experiment layout.
       format_s = TXT
       title_s = data_description
flow@localhost% cat SOAD-key.kef
/Experiment_g/Reports_g/Report_t
       array_name_a = data_request_key
       description_s = Text file of sort request keys.
       format_s = TXT
       title_s = data_request_key

flow@localhost% report2ph5 -k SOAD-desc.kef -f  data_description.txt
flow@localhost% report2ph5 -k SOAD-key.kef -f data_request_key.txt
```

12) At this point run hdfview to verify the values in SOAD.ph5, and run ph5toseg to generate a test SEG-Y gather. Be sure to review the log generated by ph5tpseg.

```
usage: ph5toseg --nickname ph5-file-prefix [options]

Convert ph5 file to SEG format. (Presently only SEG-Y and PASSCAL SEGY trace
files)

options:
  -h, --help            show this help message and exit
  -n ph5_file_prefix, --nickname=ph5_file_prefix
                        The ph5 file prefix (experiment nickname).
  -p ph5_path, --path=ph5_path
                        Path to ph5 files. Defaults to current directory.
  -s sort_table_line, --sorttableline=sort_table_line
                        Sort table line number.
  -o gatherpath, --outpath=gatherpath
                        Directory to write gathers and reports to.
  -d
  -P, --passcal-segy    Generate PASSCAL SEGY trace files.

flow@localhost% mkdir GATHERS; ph5toseg -n SOAD -o ./GATHERS -s 2
```

13) Transmit SOAD.ph5 to bob.iris.washington.edu:/hdf5-data/incoming/SOAD (should use report number) along with data_description.txt and data_request_key.txt. The two text files are best extracted from the ph5 file at the DMC using dumpreports.

```
usage: dumpreports --nickname ph5-file-prefix [--path path-to-ph5-files]

Dump reports from a ph5 file

options:
  -h, --help            show this help message and exit
  -n ph5_file_prefix, --nickname=ph5_file_prefix
                        The ph5 file prefix (experiment nickname).
  -p ph5_path, --path=ph5_path
                        Path to ph5 files. Defaults to current directory.

[5] <bob:PIC>(SOAD) \pwd
/hdf5-data/incoming/SOAD
[6] <bob:PIC>(SOAD) dumpreports -n SOAD
File: data_description.TXT
Description: Text file describing the experiment layout.
Write: data_description.TXT? (y/n) y
File: data_request_key.TXT
Description: Text file of sort request keys.
Write: data_request_key.TXT? (y/n) y
File: Spears_Report.PDF
Description: Report generated for the PI supplied gather.
Write: Spears_Report.PDF? (y/n) n
```

Quick Check List

Initialize ph5 file and fill Experiment_t and Experiment_g/Receivers_g/Receiver_t.

```
initialize-ph5 -n XX-XXX -k Experiment_t.kef
```

Process raw-data and meta-data into ph5 file.

```
125a2ph5 -n XX-XXX -d XX-XXX.dep -f XX-XXX.files.txt
```

Populate Offset_t with offsets.

```
geod2kef -n XX-XXX > Offset_t.kef
kef2ph5 -n XX-XXX -k Offset_t.kef
```

Populate Time_t with timing information.

```
time-kef-gen -n XX-XXX > Time_t.kef
kef2ph5 -n XX-XX -k Time_t.kef
```

Populate Sort_t with recording window information.

```
sort-kef-gen -n XX-XXX -s nnnnn > Sort_t.kef
kef2ph5 -n XX-XXX -k Sort_t.kef
(Delete any NULL lines at start of table, one for each array)
```

Generate data_request_key.txt and data_description.txt and insert them into the ph5 file.

```
report-gen -n XX-XXX -k
report-gen -n XX-XXX -d
(Hand edit both resulting files)
report2ph5 -n XX-XXX -f data_request_key.txt
report2ph5 -n XX-XXX -f data_description.txt
```

Load default Receiver_t information.

```
kef2ph5 -n XX-XXX -k Receiver_t.kef -c
(if no errors)
kef2ph5 -n XX-XXX -k Receiver_t.kef
```

Verify ph5 file using hdfview and generate test gathers.

```
ph5toseg -n XX-XXX -s nn
(The value for the -s option comes from data_request_key.txt)
```

```
/Experiment_g/Experiment_t
     nickname_s
     longname_s
     PIs_s
     institutions_s
     north_west_corner/X/value_d
     north_west_corner/X/units_s
     north_west_corner/Y/value_d
     north_west_corner/Y/units_s
     north_west_corner/Z/value_d
     north_west_corner/Z/units_s
     north_west_corner/coordinate_system_s
     north_west_corner/projection_s
     north_west_corner/ellipsoid_s
     north_west_corner/description_s
     south_east_corner/X/value_d
     south_east_corner/X/units_s
     south_east_corner/Y/value_d
     south_east_corner/Y/units_s
     south_east_corner/Z/value_d
     south_east_corner/Z/units_s
     south_east_corner/coordinate_system_s
     south_east_corner/projection_s
     south_east_corner/ellipsoid_s
     south_east_corner/description_s
     summary_paragraph_s
     time_stamp/ascii_s
     time_stamp/epoch_l
     time_stamp/micro_seconds_i
     time_stamp/type_e
/Experiment_g/Receivers_g/Das_g_[sn]/Das_t
     array_name_data_a
     channel_number_i
     event_number_i
     raw_file_name_s
     receiver_table_n_i
     sample_rate_i
     stream_number_i
     time/ascii_s
     time/epoch_l
     time/micro_seconds_i
     time/type_e
```

```
/Experiment_g/Receivers_g/Das_g_[sn]/Receiver_t
     orientation/azimuth/value_f
     orientation/azimuth/units_s
     orientation/dip/value_f
     orientation/dip/units_s
     orientation/description_s
/Experiment_g/Sorts_g/Sort_t
     array_t_name_s
     array_name_s
     start_time/ascii_s
     start_time/epoch_l
     start_time/micro_seconds_i
     start_time/type_e
     end_time/ascii_s
     end_time/epoch_l
     end_time/micro_seconds_i
     end_time/type_e
     description_s
     time_stamp/ascii_s
     time_stamp/epoch_l
     time_stamp/micro_seconds_i
     time_stamp/type_e
```

```
/Experiment_g/Sorts_g/Array_t_[nnn]
     id_s
     location/X/value_d
     location/X/units_s
     location/Y/value_d
     location/Y/units_s
     location/Z/value_d
     location/Z/units_s
     location/coordinate_system_s
     location/projection_s
     location/ellipsoid_s
     location/description_s
     deploy_time/ascii_s
     deploy_time/epoch_l
     deploy_time/micro_seconds_i
     deploy_time/type_e
     pickup_time/ascii_s
     pickup_time/epoch_l
     pickup_time/micro_seconds_i
     pickup_time/type_e
     das/serial_number_s
     das/model_s
     das/manufacturer_s
     das/notes_s
     sensor/serial_number_s
     sensor/model_s
     sensor/manufacturer_s
     sensor/notes_s
     description_s
/Experiment_g/Sorts_g/Offset_t
     azimuth/value_f
     azimuth/units_s
     event_id_s
     offset/value_d
     offset/units_s
     receiver_id_s
```

```
/Experiment_g/Sorts_g/Event_t
     id_s
     location/X/value_d
     location/X/units_s
     location/Y/value_d
     location/Y/units_s
     location/Z/value_d
     location/Z/units_s
     location/coordinate_system_s
     location/projection_s
     location/ellipsoid_s
     location/description_s
     time/ascii_s
     time/epoch_l
     time/micro_seconds_i
     time/type_e
     size/value_d
     size/units_s
     depth/value_d
     depth/units_s
     description_s
/Experiment_g/Reports_g/Report_t
     array_name_a
     description_s
     format_s
     title_s
```

Meta-Data, Data Required

Shot/Source information:
- time of shot/source
- id of shot/source
- latitude of shot/source
- longitude of shot/source
- elevation of shot/source
- size of shot in kg or magnitude of source
- depth below surface of shot/source
- vibrator sweep information (if applicable)
- comments (if applicable)

Receiver/Station information:
- id of station
- serial number of receiver
- manufacturer and model of receiver
- type of sensor attached (including response info if applicable)
- latitude of station
- longitude of station
- elevation of station
- comments (if applicable)

General information about experiment:
- principal investigators
- institutions
- experiment name
- experiment nickname
- paragraph describing purpose of experiment
- paragraph describing layout of instruments

Raw data files:
- raw data files