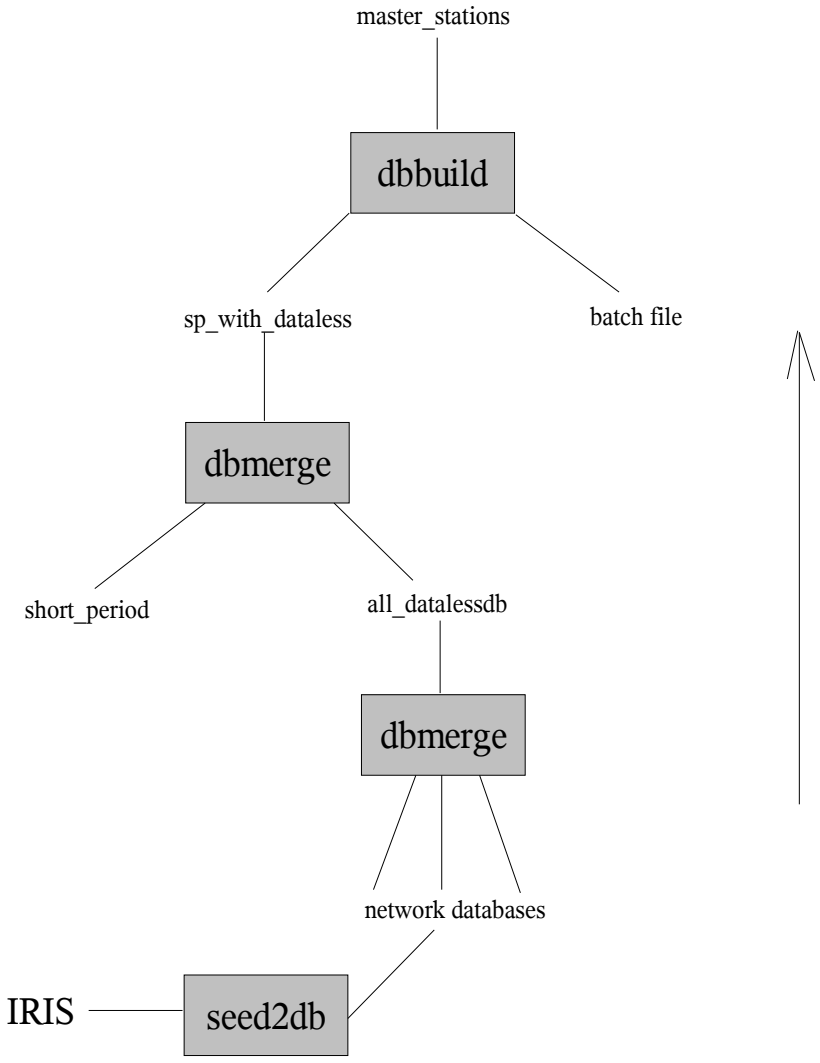
Maintaining the AEIC master stations database

Hierarchy:

This diagram gives a logical overview of how the master_stations database is constructed from dataless SEED volumes, a short-period database and a dbbuild batch file (for digital broadband stations).



Basic directory structure:

The master stations database resides in a repository at /home/martin/work/stationdb. The crucial directory hierarchy is represented below (Datascope databases are shown in brackets) :

```
stationdb/
----------all_stations/
-----------------------master_stations_db/
-----------------------------------------(master_stations)
-----------------------------------------Response/
-----------------------short_period_db/
-----------------------------------------(short_period)
-----------------------------------------Response/
-----------------------sp_with_dataless_db/
-----------------------------------------(sp_with_dataless)
-----------------------------------------Response/
-----------------------all_dataless_db/
-----------------------------------------(all_datalessdb)
-----------------------------------------Response/
/Seis/databases/dbbuild/data/
-----------------------------instruments
```

Henceforth, we use the environment variable $REPOSITORY to refer to the path / home/martin/work/stationdb/.

Description of the databases:

There are three databases that may be used in the update procedure.
1. (short_period). This database contains all short period stations and broadband stations with response not handled by dbbuild.
2. (all_datalessdb). This database contains all db's built from dataless seed volumes merged into one.
3. (sp_with_dataless). This database contains all of the short_period database plus data from imported stations obtained from dataless seed volumes downloaded from various sources.
4. (master_stations). This database is the final product used in data acquisition. It is built from merging short_period with all_dataless db, then running dbbuild in batch mode.

Useful man pages:

dbbuild
dbbuild_batch
dbbuild_examples

Useful programs:

dbbuild
dbverify
db2dbbuild (implemented soon)

Overview of steps for database update:

1. Make local copies of appropriate database(s).
2. Maintain (short_period) database.
3. Maintain (dataless_seed) database for imported stations.
4. Maintain dbbuild batch file.
5. Merge (dataless_seed) with (short_period).
6. Build complete database on merged (short_period) using dbbuild with batch file.
7. Hand edit tables, network, schanloc, snetsta, affiliation.
8. Deliver database with Response/ and stage/ directories to Mitch. Mitch will install new (master_stations) database, Response/ and stage/ directories to the appropriate systems.
9. Copy necessary files back to the database repository.

Note: BACKUP YOUR WORK AT EVERY STEP OF THE PROCESS!

Types of updates:

There are three types of updates:
1. An addition or change to the Alaska short period network (short_period).
2. An addition or change to imported stations via dataless SEED volumes (sp_with_dataless).
3. An addition or change to the Alaska broadband network (master_stations).

Depending on your update needs, you may be performing any number of these at the same time.

The procedures below are written in the chronological order in which they should be followed. Depending on the updates required, each procedure will tell you when its OK to move onto the next procedure.

Procedure 1: Creating a workspace:

1) Before starting any database work, create a workspace where you can carry out your edits For example, create a directory like /home/*$user*/work, and change to it:

```
>> cd
>> mkdir work
>> cd work
```

Procedure 2: Updating the Alaska short period network:

1. Copy the short_period directory (including its Response/ directory) to your workspace directory:

```
>> cpdir $REPOSITORY/all_stations/short_period_db .
```

2. If you do not need to make changes to the (short_period) database, omit steps 3 and 5 (but DO follow steps 4,6 and 7).

3. Assuming you need to edit the (short_period) database, run dbe in edit mode on your local copy of the (short-period) database to make the required changes:
   ```
   >> dbe -e short_period_db/short_period
   ```
4. Run dbverify to check your modifications, redirecting the output to a file to capture errors and warnings. This output file will be needed later, so don't delete it:
   ```
   >> dbverify short_period_db/short_period > verify_sp.txt
   ```

5. Sift through the output file with the "more" (or "less") command. Don't be surprised if there are of the order of 400-500 errors or warnings, but they are mostly of just two or three types. Consult Appendix 1 or talk to Martin, and move forward when only when you're convinced there are no unexpected errors:
   ```
   >> more verify_sp.txt
   ```

6. This output file will be required later, so don't delete it. Even if you did not edit the (short_period) database, you should have followed step 4. The contents of your workspace directory should now be:

   ```
   short_period_db/
   -------------(short_period)
   -------------Response/
   verify_sp.txt
   ```

7. Congratulations! You now have an updated copy of the short_period database. However, you still need to merge this with the (all_datalessdb) database (updating this first if there are any changes to the imported stations), to create a new (sp_with_dataless) database. And then you will need to run dbbuild to build a local copy of the (master_stations) database. And finally you will need to upload your work to the repository. Go to the next procedure.

Procedure 3: Updating the imported stations via dataless SEED volumes

These updates will be rather rare. And it's hard to even know when they are needed. The best you can hope for is that a network manager has let you know that there has been a response change or addition for an imported station. Or quarterly checks by downloading dataless SEED volumes from IRIS and comparing with installed databases. Always compare on and offdates of the database you create from the dataless seed volume with existing dates. Some networks do not keep historically accurate databases. If you are sure you the imported stations have not changed, jump to Procedure 4.

You can get dataless SEED volumes from IRIS at http://www.iris.edu/data/webRequest.htm. IRIS doesn't have historic information going back to the 1960s/70s – but our databases do. Make sure not to lose this important historic data! For Canadian stations, dataless SEED volumes can be got through a Canadian website.

1) Make sure you are in your workspace directory, create a dataless_seed/ directory, and change to it:

```
>> cd /home/$user/work
>> mkdir dataless_seed
>> cd dataless_seed
```

2) Download all dataless SEED volumes for your network from the repository, e.g.:
```
>> cpdir $REPOSITORY/atwc_dataless* .
>> cpdir $REPOSITORY/cnsn_dataless* .
>> cpdir $REPOSITORY/ii_dataless* .
>> cpdir $REPOSITORY/iu_dataless* .
>> cpdir $REPOSITORY/usarray_dataless* .
```

3) Convert each dataless seed volume to a database. It is also possible to download SEED seismic data from IRIS and extract the station metadata from the seed header file using seed2db with other options. Check the man page for seed2db if you want to use this method. Inspect each database well, not every network follows seed convention in their channel naming schemes. You are likely to have to make hand edits.
```
>> seed2db -respdir ./Response -stagedir ./Response/stage atwc_dataless atwc_db
```

4) Merge all the databases you created in step 3 to one big target dataless SEED database (all_datalessdb). The Response and stage dirs are handled by dbmerge behind the scenes, as long as they are in your working directories. Duplicates are removed and the instrument table links properly.
```
>> cp atwc_dataless*/atwc_2* .
>> /home/glenn/bin/rename atwc_2  all_datalessdb
>> dbmerge ii_dataless*/iidb all_datalessdb
>> dbmerge usarray/egak_wrakdb all_datalessdb
```
(NOTE: NOT SURE ABOUT cnsn & iu SEED volumes – not databases!)**yes they are!

5) Run dbverify on the (all_datalessdb) database, directing the output to a suitable output file:
```
>> dbverify all_datalessdb > verify_dataless.txt
```

6) Make a copy of the merged database and Response/ directory to be later copied to the repository.

7) Also create dated dirs in the dataless_networks dir and copy your dataless seed volumes and db's to the dir. (NOTE: instructions needed!!!!)

8) Procedure 4: Merging the (all_datalessdb) and (short_period) databases:

1. If you already have a copy of the (all_datalessdb) database from the last procedure (because you updated the dataless SEED volumes), jump to step 4. Otherwise, create a dataless_seed directory in your workspace directory:
```
>> cd /home/$user/work
>> mkdir dataless_seed
```

2. Copy the (all_datalessdb) database from the repository to your local dataless_seed directory. Don't forget the Response and stage dirs:
```
>> cp $REPOSITORY/dataless_seed/all_dataless_db/all_datalessdb*
dataless_seed
>> cp -r $REPOSITORY/dataless_seed/all_dataless_db/Response dataless_seed
```

3. Run dbverify on the (all_datalessdb) database, directing the output to a suitable output file:
```
>> dbverify all_datalessdb > verify_dataless.txt
```

4. Sift through the output file with the "more" (or "less") command. Consult Appendix 1 or talk to Martin, and move forward when only when you're convinced there are no unexpected errors:
```
>> more verify_dataless.txt
```

5. Make a new directory called sp_with_dataless in your local directory:
```
>> mkdir sp_with_dataless
```

6. Copy your (updated) short_period database into this sp_with_dataless directory:
```
>> cp short_period_db/short_period* sp_with_dataless/
>> cp -r short_period_db/Response sp_with_dataless/
```

7. The name "short_period" is no longer meaningful for this copy of the database, since you are about to merge rows from the (all_datalessdb) into it. So use the "rename" program to call it "sp_with_dataless" instead:
```
>> cd sp_with_dataless
>> /home/glenn/bin/rename short_period sp_with_dataless
```

8. Your workspace should now have the following structure (with additional directories under dataless_seed/ if you updated the dataless SEED volumes in Procedure 3):

```
dataless_seed_db/
-------------(all_datalessdb)
--------------Response/
short_period_db/
-------------(short_period)
-------------Response/
sp_with_dataless_db/
-------------(sp_with_dataless)
-------------Response/
verify_dataless.txt
verify_sp.txt
```

9. You should still be in the sp_with_dataless directory. Now you are ready to merge your local copy of the (all_datalessdb) database with the (sp_with_dataless) database you updated, with the dbmerge command:
```
>> dbmerge ../dataless_seed_db/all_datalessdb sp_with_dataless
>> cd ..
```

10. You should now be back in your workspace directory. The target database (sp_with_dataless) should now have the (all_datalessdb) database folded into it. Lets check with dbverify, again sending the output to a file:
```
>> dbverify sp_with_dataless_db/sp_with_dataless > verify_spwd.txt
```

11. Again, there will be lots of warnings and errors since this contains all the database rows from the (short_period) database you verified earlier. But are there any new errors or warnings? To find out, use the "diff" command:
```
>> diff verify_sp.txt verify_spwd.txt
```

12. If any differences occur, consult Appendix 1 or talk to Martin.

13. Congratulations! You now have an updated copy of the (sp_with_dataless) database. However, you still need to run dbbuild to build a local copy of the (master_stations) database. And finally you will need to upload your work to the repository. Go to the next procedure.

Procedure 5: Updating the broadband seismic network and/or master_stations database

Your workspace directory should now look like:

```
dataless_seed_db/
-------------(all_datalessdb)
--------------Response/
short_period_db/
-------------(short_period)
-------------Response/
sp_with_dataless_db/
-------------(sp_with_dataless)
--------------Response/
verify_dataless.txt
verify_sp.txt
verify_spwd.txt
```

There will be additional directories under dataless_seed/ if you updated the dataless SEED volumes in Procedure 3.

Now you're almost ready to run dbbuild and create the final (master_stations) database!

1. First copy the batch file "master_stations-dbbuild" from the repository to your workspace directory:
```
>> cd /home/$user/work
>> cp $REPOSITORY/all_stations/dbbuild_batch/master_stations-dbbuild .
```

2. Copy the database (sp_with_dataless) to the current (workspace) directory & rename it to (master_stations).
```
>> cp sp_with_dataless_db/sp_with_dataless* .
>> cp -r sp_with_dataless_db/Response .
>> /home/glenn/bin/rename sp_with_dataless master_stations
```

3. Rename snetsta, network, affiliation, and schanloc tables of the (master_stations) database to a backup name. Otherwise they may cause problems with dbbuild. See Appendix 2: Non-Standard tables for further details.
```
>> mv ../master_stations.snetsta backup.snetsta
>> mv ../master_stations.network backup.network
>> mv ../master_stations.affiliation backup.affiliation
>> mv ../master_stations.schanloc backup.schanloc
>> cd ..
```

4. Edit the batch file "master_stations-dbbuild" to suit your needs, if any changes have occurred to the broadband seismic network. Be sure that your choice of editor does not put in any control characters or end-of-line characters (I like simple old vi):
```
>> gvim master_stations-dbbuild
```
All sensor changes are tracked in the batch file. There is an example of every sensor somewhere in the database – best approach is to find something similar, and copy the syntax.

5. Run dbbuild in batch mode on the (sp_with_dataless) database, redirecting the screen output to a file you can inspect:
```
>> dbbuild -b -v master_stations master-stations-dbbuild >& dbbuild.out
```

6. Hand-edit the tables mentioned in the **Non-Standard tables** section below. Copy the tables to match your master_stations database name structure.

7. Run dbverify on your database.

Procedure 6: Uploading your changes back to the repository

Your workspace directory should now contain the following files:

```
dataless_seed_db/
-------------(all_datalessdb)
-------------/Response
short_period_db/
-------------(short_period)
-------------Response/
sp_with_dataless_db/
-------------(sp_with_dataless)
-------------Response/
master_stations_db/
-------------(master_stations)
```

```
-------------(backup)
-------------/Response
verify_dataless.txt
verify_ms.txt
verify_sp.txt
verify_spwd.txt
```

1. Send Mitch an email telling him the location of your workspace, and asking him to upload changes into the database repository.
2. Any new or modified databases you have created need to be checked back into the repository. Create directories with dates for each one to preserve chronological order.

3. Its probably good practice to keep your edits for now. Create a new directory, with todays date in the name, in your workspace and move everything to that directory:
   ```
   >> mkdir stationdb_20070314
   >> mv * stationdb_20070314
   ```

That's it!

Further work

- Create a CVS repository for the master_stations database? User would then 'checkout' latest database(s)/batch file, modify them, and commit changes back. Live install directory could then be a sandbox created by Mitch.

Appendix 1: Miscellaneous details

Dbmerge:
dbmerge merges two databases together. There is no independent output database, the second argument is the output. dbmerge also merges the Response and stage dirs as it does it's job. Kindly, dbmerge is nice about not doing anything if there is a problem with your database, it will point out the error and tell you to fix it before continuing. But still, backup your work before using!

Dbbuild
Dbbuild has two modes. The gui interface is quite useful as a practice tool to create a batch file. However, the gui does not accomplish all of our needs.
The batch mode is the most effective method of using dbbuild. It involves maintaining an input file. Proper syntax in the batch file is extremely important. Nearly all combinations of instruments are represented in the batch file. When making edits, search the file for the station you are interested in, then edit that section of the file to suit your needs.
The batch file command line will look something like:

>> dbbuild -b sp_with_dataless master_stations-dbbuild >& dbbuild_out

where dbbuild_out will save the output to a file for you to proofread. This output is different than the dbverify output and should be inspected for errors.

Dbverify:
Dbverify should be used before you consider any level update complete. Dbverify gives a lot of error messages that are not significant. You will recognize them as being repetitive. Dbverify tolerances for these errors can be customized to reduce the error output, consult the man page if you want to do this. You can redirect the dbverify output to a file with a command line like this:

```
>> dbverify short_period >& output_file
```

Closing Stations:
dbbuild gui will not close stations. It will close old channels when new channels are opened as long as the channel names stay the same (BHZ->BHZ). When channel names change or a station is closed, a close statement must be added manually to the dbbuild-batch file.

Dbbuild comments. Necessary dbbuild comments in the batch file include:
person editing the file,
date edits take place,
a reason such as "sensor change" or "site closed", "station installation"
Other comments are welcome when adding to the file, but not necessary.

Dataless SEED files additions (*these change all the time*)
Periodic updates of response for imported stations should take place when response is known to have change or for checking purposes.
web address: http://www.iris.washington.edu/data/DatalessRequest.htm

station list:
US net, EGAK, WRAK
AT net, AKUT, CRAG, PMR, SDPT, SIT, SKAG, SMY, OHAK
Convert the dataless seed files to a database with seed2db, setting the proper path and names for stage and response directories. After converting to a database, remove network and affiliation tables, these are maintined by AEIC needs, not the originating seismic lab. Merge with short_period database. The merge will not properly handle existing entries in the site table with historic value, the added entries may need to be removed by hand.
1/22/2007 sta removed from all_datalessdb.site: AKUT, SMY, PMR, SDPT, SIT
Be aware that US stations EGAK and WRAK have future offdates in the dataless seed files. For now, these dates are intact.
Delaney Park array stations in Anchorage are all under the official name 8040. DPA1, DPA2, DPA3, and DPA4 are technically channel names since the stations all have the same gps location. Response for these stations is not available as dataless seed, the parameters came from an E-mail and exist in the master_stations-dbbuild batch file.
ATWC stations may lag behind in the dataless SEED files from IRIS. Compare your database from the dataless seed file to the most current data in the database you are about to update.
**Some stations may have more current information coming from the dbbuild-batch file than exist in dataless seed! The source of the more current information is from the header file on a seed volume.
1/21/2007 PMR and all CNSN stations.

Non-standard tables:
schanloc, network, affliliation and snetsta tables are not standard tables and must be edited by hand. They are dynamically updated by Antelope is another network starts exporting new stations.

schanloc is a combination of hand and dbbuild. If a station has a loc code, then there must be an entry in this table. Seed2db handles this well. But Antelope will dynamically update this table when a new channel is imported. This often makes trouble when the chan row has the loc code appended to the sta code. These rows must be hand removed. There are many sta entries that have no loc code, these cause no problem are not needed in the table. This is a dynamically updated table by Antelope any time a new channel comes into data acquisition, but often needs hand edits when a channel contains a loc code that cannot be managed by dbloc for earthquake processing.

network is used by all stations. There are many different net entries based on stations that share particular processing needs. Some net entries are very broad, such as the AK network, some are quite narrow such as the Fourpkd network on Fourpeaked volcano.

affiliation is used by all stations. Groups of stations sharing characteristics are mapped to the network table via affiliation. Subset a particular station by name to determine what networks it is affiliated with.

snetsta is used to map stations AEIC imports from other networks. This table is dynamically updated by Antelope any time a new station comes into data acquisition. snetsta will need the proper snet hand applied when a new station is imported from another network, such as CN or AV. If this is not done, Antelope will try to incorrectly map an imported station to the AK net.

Instrument table:
There are stations that do not have known response. But the sensor table must have entries for an analyst to make picks. So dummy entries exist in the sensor table for short period stations and broadbands. Reference these dummy rows when adding or editing stations without known response.

Guralp issues:
the installed cmg5t response file is wrong in release 4.8. I used a 4.7 response file as a template and created my own. In the response dir is a file created by Martin that is called aeic_cmg5t.
In the sensor dir is a file created by Martin called cmg5t.Guralp equipment response can vary based on cabling between sensor and digitizer. A sensor directly coupled to the digitizer in the stacked configuration (no visible cables) uses “single ended” response. Sensors and digitizers connected via a cable are considered “differential”. Guralp manuals handle this on the sensor side with a 2x multiplier, an incompatibility with dbbuild. Instead /2 on the digitizer side is appropriate for dbbuild. This allows a sensor originally delivered for use as single ended to retain the same response when used as differential.

Appendix 2 – Original notes by M. LaFevers

Overview of steps for database update.
Make local copies of appropriate database(s).
Maintain short_period database.
Maintain dataless seed_database for imported stations.
Maintain dbbuild batch file.
Merge dataless_seed with short_period.
Build complete database on merged short_period using dbbuild with batch file.
Hand edit tables, network, schanloc, snetsta, affiliation.
Deliver database with Response and stage dirs to Mitch. Mitch will install new master_stations database, Response and stage dirs to the appropriate systems.
Copy necessary files back to the database repository.
HINT: BACKUP YOUR WORK AT EVERY STEP OF THE PROCESS!

Level 1 update:
1. Copy the short_period database and Response dir to your working directory. Copy all_datalessdb and Response dir to another working directory. Also the master_stations-dbubild batch file to a third directory, along with master_stations.affiliation, network, schanloc, and snetsta. Rename the master_stations tables to backup names to be set aside for later.
2. Make the necessary hand edits or additions to the short_period database rows.
3. Run dbverify.
4. Copy your work to be later placed in the database repository.
5. Merge short_period with all_datalessdb.
6. Run dbverify.
7. Run dbbuild in batch mode on the merged db, reference Level 3 update below.
8. Run dbverify.
9. Send mitch an E-mail telling him the location of your final database with proper Response files. When he confirms it's done, make a directory with today's date in the name in the check out area. Create two subdirectories, short_period, and sp_with_dataless.Copy short_period and sp_with dataless into the subdirectories. Be sure to include the Response and stage dirs in the copies.

Level 2 update:
These updates will be rather rare. And it's hard to even know when they are needed. The best you can hope for is that a network manager has let you know that there has been a response change or addition for an imported station. Or quarterly checks by downloading dataless seed volumes from IRIS and comparing with installed databases. Always compare on and offdates of the database you create from the dataless seed volume with existing dates. Some networks do not keep historically accurate databases.
1. Create your dataess seed volume. You can either merge your new station with your existing dataless seed volume for the network, or download all dataless seed volumes for your network.
2. Make a copy in a safe place that will be later copied to the dataless seed network repository.
3. Merge with all the rest of the dataless seed volumes to get one big dataless seed db.
4. Run dbverify on the db.
5. Make a copy of the merged database and Response dir to be later copied to the dataless seed all network repository.

6. Copy short_period db from the repository to a working dir.
7. Merge dataless_seed db with short_period db.
8. Run dbverify on the db.
9. Make a copy of the db and Response dir to be later placed in the sp_with_datalessdb repository.
10. Copy master_stations-dbbuild to your working directory.
11. Run dbbuild in batch mode on the sp_with_datalessdb. Reference **Level 3 update** below.
12. Run dbverify.
13. Follow step 10 from Level 1 update section. Also create dated dirs in the dataless_networks dir and copy your dataless seed volumes and db's to the dir.

Level 3 update:
The easy one.
1. Make a copy of sp_with_dataless, the Response dir and the master_stations-dbbuild file to your working dir.
2. Rename snetsta, network, affiliation, and schanloc to a backup. They will cause problems with dbbuild. See **Non-Standard tables** note below.
3. Edit the batch file to suit your needs. Run dbbuild in batch mode on your db. Redirect the screen output to a file you can inspect.
4. Hand edit the tables mentioned in the **Non-Standard tables** section below. Copy the tables to match your master_stations database name structure.
5. Run dbverify on your database.
6. Check the database and Response dir in with Mitch.
7. Copy master_stations-dbbuild file to the database repository in a dir with a date.