Data Handbook

Table of Contents

Setting Up Databases and Data

Setting up MySQL user

Log in

```
mysql -h vm-lnx-cpccfmysql -u first.last -p
```

Then enter your password

Changing password

```
SET PASSWORD = PASSWORD('newPass');
```

Setup config files

1. Go into the Verification's input directory:

   ```
   cd $VERIF_HOME/input
   ```

   Note: The environment variable $VERIF_HOME needs to be defined or the scripts won't run
2. Copy the example file to a real config file:

   ```
   cp verif_data.conf.example verif_data.conf
   ```

3. Change the file permissions so only you can see it (contains a password!):

   ```
   chmod 600 verif_data.conf
   ```

4. Edit the file, following the directions in the file

Updating the dataSettings table

The dataSettings table under the cpc_reference database specifies all the necessary information about each dataset, basically all the information found in the dataSettings Google Doc.

In order to get changes from the Google Doc into the dataSettings database table:

Step 1 – Save the Google Doc as a CSV file

Click File –> Download as –> CSV

Put in $VERIF_HOME/trunk/scripts.

Step 2 – Run db_setup_updateDataSettings.pl

Run the db_setup_updateDataSettings.pl script with the following options:

-f [csv_file] -t dataSettings

Use the -h option to print the script usage.

Importing Data

All data import scripts are named db_import_*

Importing Data From a Single Forecast

To import data from a single forecast or observation (but over any range of dates), run the script db_import_data.pl. Running it with no options or with the -h flag will give you the usage details.

This script reads in ASCII data (found based on the dataSettings database table under the cpc_reference database) and writes the data to the appropriate forecast or observation table. If the file wasn't found, then NULLs will be written for that day so there will always be some kind of value in the table for each day. If after importing the data from the ASCII file one or more locations are missing on that day, then they are filled with NULLs, again so that there will always be some kind of value each day.

For example, to import 1 week of the manual forecast:

```
> $VERIF_HOME/scripts/db_import_data.pl -validdate -d forecast -v temp
-a 05d -g stn -s 20100401 -e 20100407 -f manual -l 08d
```

To import 6 months of monthly observations:

```
> $VERIF_HOME/scripts/db_import_data.pl -validdate -d observation -v
temp -a 01m -g cd -s 201003 -e 201008
```

Importing Data From All Forecasts

To import data from all existing forecasts and observations (also over any range of dates), run the script db_import_daily.pl. This scripts is basically a wrapper script that runs db_import_data.pl for each forecast and observation. Running it with no options or with the -h flag will give you the usage details.

Importing Climate Phenomena Data

To import climate phenomena data from all existing forecasts (also over any range of dates), run the file db_import_climPhenom.pl. Running it with no options or with the -h flag will give you the usage details.

This script reads the climate phenomena data from an ASCII file (either local or on a web server). The latest ENSO values can be found here:

http://www.cpc.ncep.noaa.gov/data/indices/oni.ascii.txt

For example, to import all the latest historical seasonal ENSO values:

```
> ./db_import_climPhenom.pl -i http://www.cpc.ncep.noaa.gov/data/
indices/oni.ascii.txt -c ENSO -t seasonal -m 1 -y 2 -v 3 -f MMM
```

Testing Databases

Missing Data

The script used to find missing data is db_test_findMissing.pl. Run it with no options or with the -h option to print the script usage.

For example, to search for missing observations in all long range climate division observation database tables:

```
./db_test_findMissing.pl -d observation -g cd -s 199501 -e 201204 -f
longRange -loglevel error -t '(temp|precip)_0[13]m_cd'
```

The -t option allows you to narrow down the list of tables that are searched using regular expressions.

The -n option can be used to count NULLs in the database as valid rows and just include rows that are *completely missing* in the total missing row count. This is useful to see if the import script had issues when running, since the import script will place NULLs in the database when a certain location doesn't have a value.

For example, to search for missing forecasts in only the manual forecast:

```
./db_test_findMissing.pl -d forecast -g cd -s 199501 -e 201204 -f
longRange -loglevel error -t '(temp|precip)_manual_.*'
```

The -loglevel option specifies what level of log messages you want from the script. If you only want to see script output when missing data was found, use a loglevel of "error". See the script usage for more information.