ADD WFS Development Notes

Peter L. Pulsifer

Jeff McKenna

A. Paul R. Cooper

November 2004

Geomatics and Cartographic Research Centre, Carleton University

British Antarctic Survey

Introduction:

Objectives

Limitations:

Feature Catalogue

Software Installation

Intended to be a general guide
Detailed instructions can be found at the URLs listed

PostGresql

http://www.postgresql.org

PostGIS

http://postgis.refractions.net

Geoserver

Installing a Java SDK

1. download "J2SE v 1.4.2_06 SDK" from http://java.sun.com/j2se/1.4.2/download.html
2. rename this file to 'j2sdk-1_4_2_06-linux-i586-rpm.bin'
3. # sudo chmod a+x j2sdk-1_4_2_06-linux-i586-rpm.bin
4. # ./j2sdk-1_4_2_06-linux-i586-rpm.bin
5. # sudo rpm -iv j2sdk-1_4_2_06-linux-i586.rpm
6. java is installed into /usr/java/j2sdk1.4.2_06
7. set JAVA_HOME environment variable to root directory of java install (JAVA_HOME=/usr/java/j2sdk1.4.2_06)
8. add the '/usr/java/j2sdk1.4.2_06/bin' folder to your PATH
9. reread the profile with command 'source .bash_profile'
10. test your environment variables with 'env'

Install Tomcat Servlet v5.0.28

(note that the latest GeoServer release comes packaged with Jetty, another Open Source java Servlet)

1. note that there are newer releases of Tomcat, however several users recommend this version because it is stable
2. # wget http://apache.mirror.secondchapter.info/jakarta/tomcat-5/v5.0.28/bin/jakarta-tomcat-5.0.28.tar.gz
3. # sudo tar xzvf jakarta-tomcat-5.0.28.tar.gz -C /usr/local/

4. # sudo chown -R nobody:www jakarta-tomcat-5.0.28/
5. add the following to your .bash_profile, "to avoid stability problems" on Redhat 9.0:
    - # export LD_ASSUME_KERNEL=2.2.5
    - then re-read profile with "source .bash_profile" command
6. start Tomcat by executing the following in /usr/local/jakarta-tomcat-5.0.28/bin/:
    - # sudo ./startup.sh
    - you should see a message such as:
    - `Using CATALINA_BASE:   /usr/local/jakarta-tomcat-5.0.28`
    - `Using CATALINA_HOME:   /usr/local/jakarta-tomcat-5.0.28`
    - `Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-5.0.28/temp`
    - `Using JAVA_HOME:       /usr/java/j2sdk1.4.2_06`
7. verify that Tomcat has successfully started by going to http://localhost:8080 in a browser (you should see a Tomcat welcome page)
    - if you are installing remotely, install 'links' program (a text-based browser) from http://links.sourceforge.net/ and type at the commandline:
        - # links http://localhost:8080 (you should see the Tomcat welcome page in text format)
8. optionally look into starting Tomcat on reboot (not covered in these instructions)

Connect Tomcat to Apache with mod_jk2 connector

1. see mod_jk howto if u need more info: http://jakarta.apache.org/tomcat/connectors-doc/jk2/jk/aphowto.html
2. obtain jk2 connector source from http://apache.mirror.mcgill.ca/jakarta/tomcat-connectors/jk2/ (because no existing binaries exist for Redhat 9.0)
    - # wget http://apache.mirror.mcgill.ca/jakarta/tomcat-connectors/jk2/jakarta-tomcat-connectors-jk2-src-current.tar.gz
3. # sudo tar xzvf jakarta-tomcat-connectors-jk2-src-current.tar.gz -C /home/src
4. build from source:
    - # cd /home/src/jakarta-tomcat-connectors-jk2-2.0.4-src/jk/native2
    - # sudo ./configure --with-apxs2=/usr/local/apache2/bin/apxs
    - # sudo make
    - # cd ../build/jk2/apache2
    - # sudo /usr/local/apache2/bin/apxs -n jk2 -i mod_jk2.so
5. build should place the mod_jk2.so file in /usr/local/apache2/modules/mod_jk2.so
6. change permissions on mod_jk2.so if necessary
7. add the following to Apache's httpd.conf file:
    - LoadModule jk2_module modules/mod_jk2.so
8. create a 'workers2.properties" file in /usr/local/apache2/conf containing the following:
        - `[channel.socket:localhost:8009]`
        - `port=8009`
        - `host=127.0.0.1`
        - `[ajp13:localhost:8009]`
        - `channel=channel.socket:localhost:8009`

        - `[uri:/servlets-examples/*]`
        - `worker=ajp13:localhost:8009`
9. change permissions for this file if necessary
10. restart apache
11. goto http://127.0.0.1/servlets-examples/ in a browser (note that port 8080 is no longer required)

Install GeoServer v1.2.3

1. download geoserver.war.zip from http://sourceforge.net/project/showfiles.php?group_id=25086
2. extract the contents of the zipfile
3. move the .war file to Tomcat's webapps directory (if not using Tomcat move it to your servlet's web-accessible directory)
    - # cp geoserver.war /usr/local/jakarta-tomcat-5.0.28/webapps/
4. change the permissions on geoserver.war file if necessary (for tomcat/apache)
5. for Tomcat: open yourpathto/apache2/conf/workers2.properties and add the following:
        - `[uri:/geoserver/*]`
        - `worker=ajp13:localhost:8009`
6. restart apache
7. goto http://127.0.0.1/geoserver/
8. wait while pages are generated for first time
9. set admin user/password

Starting Services:

PostgreSQL:

PostgreSQL file tree is typically located at: /usr/local/pgsql
Database clusters would typically be located at /usr/local/pgsql/data/*clustername*

Stopping Server:

- if u used postmaster command just CTL-C
- if u used pg_ctl:

% pg_ctl stop

Database Creation

Establishing a Database Cluster:

PostgreSQL Documents:

http://www.postgresql.org/docs/7.4/interactive/creating-cluster.html
http://www.postgresql.org/docs/7.4/interactive/app-initdb.html

Example

[postgres]$ initdb –D /usr/local/pgsql/data/add

Run-Time Configuration

Once the database cluster has been initialized, a number of run-time parameters that affect the behaviour of the database system can be changed by modifying the *postgresql.conf* file found in the root of your database cluster directory (i.e. /usr/local/pgsql/data/add)

Particularly important are:

Tcpip_socket = true|false : server will accept TCP/IP socket connections

Port = integer : TCP/IP port 5432 is default. When using a port other than default, it is important to specify when using client applications such a psql or mapserver.

Shared_buffers = int Affects performance

Some of these can be toggled or modified using the postmaster application when starting the server.

Starting Server:

PostgreSQL Documents:
http://www.postgresql.org/docs/7.4/interactive/postmaster-start.html
http://www.postgresql.org/docs/7.4/interactive/app-postmaster.html

- switch to the postgres user or to user intended to own the server process

% su postgres password: *****

Set configuration parameters by editing postgresql.conf as described above. Then start server using one of two methods:

Postmaster

Example:

```
$ postmaster -D /usr/local/pgsql/data/add >logfile 2>&1 &
```

pg_ctl wrapper utility

http://www.postgresql.org/docs/7.4/interactive/app-pg-ctl.html

% pg_ctl start –D /usr/local/pgsql/data/add -l /usr/local/pgsql/log/add_postmaster_5433.log

Creating a Database in your Database Cluster:

- Create the database:

```
% createdb –p 5433 add
```

```
Using the –p parameter creates the database using an alternate to the default port (5432). Not required if the default port is being used.
```

Create PostGIS Extension Language:

```
See section 2.4 of PostGIS document

Example:

$ createlang –p 5433 plpgsql add

enable the PL/pgSQL extension in your database (instructions taken from http://postgis.refractions.net/docs/x83.html):
```

Load the postgis object and function defs into your db

$ psql –p 5433 -f /usr/local/pgsql/share/contrib/postgis.sql add

you should see a series of CREATE statements and notices

```
NOTE:

GEOS and PROJ install to /usr/local/lib by default. You may see errors
mentioning their libraries when enabling a DB for PostGIS. This is
likely due to /usr/local/lib not being in the default library path. In
linux (Fedora at least) you can add "/usr/local/lib" to /etc/ld.so.conf
(or create an entry in ld.so.conf.d) and run ldconfig to update the
system library paths.

The other older way to do it (and maybe still the way on Solaris) is to
append /usr/local/lib to your LD_LIBRARY_PATH environment variable. The
variable may not exist already and defaults may be in use, so be
careful that you haven't "replaced" the defaults.
```


```
GEOS and PROJ install to /usr/local/lib by default. You may see errors
mentioning their libraries when enabling a DB for PostGIS. This is
likely due to /usr/local/lib not being in the default library path. In
linux (Fedora at least) you can add "/usr/local/lib" to /etc/ld.so.conf
(or create an entry in ld.so.conf.d) and run ldconfig to update the
system library paths.

The other older way to do it (and maybe still the way on Solaris) is to
append /usr/local/lib to your LD_LIBRARY_PATH environment variable. The
variable may not exist already and defaults may be in use, so be
careful that you haven't "replaced" the defaults.
```


Load the EPSG identifiers into your db

```
$ psql -p 5433 -f /usr/local/pgsql/share/contrib/spatial_ref_sys.sql
add
```

Customize EPSG code if code is not included (i.e. SRID 3031)

PostGIS contains a table (spatial_ref_sys) that defines projection/SRID (EPSG) codes. Not all projections are supported by this standard table. For example, the default table does not include a polar stereographic projection. New projections can be supported by INSERTING data into the spatial_ref_sys table. In this example, the following data was save to a .sql file and loaded into PostGIS at the command prompt as follows:

```
$ psql -p 5433 -f
/usr/local/pgsql/share/contrib./spatial_ref_sys_ps.sql add
```

where spatial_ref_sys_ps.sql is the name of the text file containing the INSERT statement (below) and add is the name of the PostGIS database. NOTE: this file has been copied the /usr/local/pgsql/share/contrib folder. The scale05 database has been updated.

INSERT Statement -- -- Polar Stereographic (from British Antarctic Survey ADD ESRI .prj file)

```
INSERT INTO "spatial_ref_sys" (srid, auth_name, auth_srid, srtext, proj4text) VALUES (3031,'ESRI',
3031,'PROJCS["Polar Stereographic",GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS
84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],TOWGS84[0,0,0,0,0,0,0],AUTHORITY["EPSG","63
26"]],PRIMEM["Greenwich",0,AUTHORITY["EPSG","8901"]],UNIT["degree",0.0174532925199433,AUTHORITY
["EPSG","9108"]],AXIS["Lat",NORTH],AXIS["Long",EAST],AUTHORITY["EPSG","4326"]],PROJECTION["Polar_
Stereographic"],PARAMETER["latitude_of_origin",-
71],PARAMETER["central_meridian",0],PARAMETER["scale_factor",1],PARAMETER["false_easting",0],PARA
METER["false_northing",0],UNIT["METERS",1]]','+proj=stere +lat0=-90 +lat_ts=-71 +long_0=0 +x_0=0 +y_0=0
+ellps=WGS84 +datum=WGS84 +units=m no_defs'); COMMIT;
```

The data in the *strext* field was optained from an ESRI .prj file associated with a the ADD coast10 shapefile. In this case, the OGR2OGR program created the .prj file when converting from the original Arcinfo binary coverage files.

LOADING DATA with shp2pgsql loader

1. obviously convert data to ESRI shapefiles if necessary (using ogr2ogr, or FME)
2. use the shp2pgsql loader (utility compiled by postgres install)
    - located in /usr/local/pgsql/bin/
3. make sure directory is writeable by postgres user

Multi-step process

1. execute loader to create sql loader file:

```
% shp2pgsql -d coast05_arc coast05arc scale05 > coast05arc.sql
```

2. run the sql file created

```
% psql -d scale05 -f coast05arc.sql
```

3. you should see a bunch of INSERT output

Problems encountered

- Use of sed command to find and replace problem characters in field names
- use the "-d" switch in shp2pgsql to drop the table cleanly (battled error message "duplicate key.." before using this switch) . Also useful when reloading after an error. Removes relevant record from **** system table
- "numeric field overflow" error when loading coast05_poly -> expanded AREA and PERIMETER fields to numeric(30,9)

PostGIS and Mapserver

If you will be querying your layers using Mapserver you will also need an "oid index".

Mapserver requires unique identifiers for each spatial record when doing queries, and the PostGIS module of Mapserver uses the PostgreSQL `oid` value to provide these unique identifiers. A side-effect of this is that in order to do fast random access of records during queries, an index on the `oid` is needed.

To build an "oid index", use the following SQL:

```
CREATE INDEX [indexname] ON [tablename] ( oid );
```

More Details:

For queries to be fast, you must have a unique key for your spatial table and you must have an index on that unique key.

You can specify what unique key for mapserver to use with the `USING UNIQUE` clause in your `DATA` line:

```
DATA "the_geom FROM geotable USING UNIQUE gid"
```

If your table does not have an explicit unique column, you can "fake" a unique column by using the PostgreSQL row "oid" for your unique column. "oid" is the default unique column if you do not declare one, so enhancing your query speed is a matter of building an index on your spatial table oid value.

```
postgis# CREATE INDEX geotable_oid_idx ON geotable (oid);
```

Advanced Usage

The `USING` pseudo-SQL clause is used to add some information to help mapserver understand the results of more complex queries. More specifically, when either a view or a subselect is used as the source table (the thing to the right of "FROM" in a `DATA` definition) it is more difficult for mapserver to automatically determine a unique identifier for each row and also the SRID for the table. The `USING` clause can provide mapserver with these two pieces of information as follows:

```
DATA "the_geom FROM (SELECT table1.the_geom AS the_geom, table1.oid AS
oid, table2.data AS data
 FROM table1 LEFT JOIN table2 ON table1.id = table2.id) AS new_table
USING UNIQUE oid USING SRID=-1"
```

USING UNIQUE <uniqueid>

Mapserver requires a unique id for each row in order to identify the row when doing map queries. Normally, it would use the oid as the unique identifier, but views and subselects don't automatically have an oid column. If you want to use Mapserver's query functionality, you need to add a unique column to your view or subselect, and declare it with `USING UNIQUE`. For example, you could explicitly select one of the table's oid values for this purpose, or any other column which is guaranteed to be unique for the result set.

The `USING` statement can also be useful even for simple `DATA` statements, if you are doing map queries. It was previously recommended to add an index on the oid column of tables used in query-able layers, in order to speed up the performance of map queries. However, with the `USING` clause, it is possible to tell mapserver to use your table's primary key as the identifier for map queries, and then it is no longer necessary to have an additional index.

Note:
"Querying a Map" is the action of clicking on a map to ask for information about the map features in that location. Don't confuse "map queries" with the SQL query in a `DATA` definition.

USING SRID=<srid>

PostGIS needs to know which spatial referencing system is being used by the geometries in order to return the correct data back to mapserver. Normally it is possible to find this information in the "geometry_columns" table in the PostGIS database, however, this is not possible for tables which are created on the fly such as subselects and views. So the `USING SRID=` option allows the correct SRID to be specified in the `DATA` definition.

Warning

The parser for Mapserver PostGIS layers is fairly primitive, and is case sensitive in a few areas. Be careful to ensure that all SQL keywords and all your `USING` clauses are in upper case, and that your `USING UNIQUE` clause precedes your `USING SRID` clause.

Examples

Lets start with a simple example and work our way up. Consider the following Mapserver layer definition:

```
LAYER
 CONNECTIONTYPE postgis
 NAME "roads"
 CONNECTION "user=theuser password=thepass dbname=thedb host=theserver"
 DATA "the_geom FROM roads"
 STATUS ON
 TYPE LINE
 CLASS
  COLOR 0 0 0
```

```
 END
END
```

This layer will display all the road geometries in the roads table as black lines.

Now lets say we want to show only the highways until we get zoomed in to at least a 1:100000 scale - the next two layers will acheive this effect:

```
LAYER
 CONNECTION "user=theuser password=thepass dbname=thedb host=theserver"
 DATA "the_geom FROM roads"
 MINSCALE 100000
 STATUS ON
 TYPE LINE
 FILTER "road_type = 'highway'"
 CLASS
  COLOR 0 0 0
 END
END

LAYER
 CONNECTION "user=theuser password=thepass dbname=thedb host=theserver"
 DATA "the_geom FROM roads"
 MAXSCALE 100000
 STATUS ON
 TYPE LINE
 CLASSITEM road_type
 CLASS
  EXPRESSION "highway"
  SIZE 2
  COLOR 255 0 0
 END
 CLASS
  COLOR 0 0 0
 END
END
```

The first layer is used when the scale is greater than 1:100000, and displays only the roads of type "highway" as black lines. The `FILTER` option causes only roads of type "highway" to be displayed.

The second layer is used when the scale is less than 1:100000, and will display highways as double-thick red lines, and other roads as regular black lines.

So, we have done a couple of interesting things using only mapserver functionality, but our `DATA` SQL statement has remained simple. Suppose that the name of the road is stored in another table (for whatever reason) and we need to do a join to get it and label our roads.

```
LAYER
 CONNECTION "user=theuser password=thepass dbname=thedb host=theserver"
 DATA "the_geom FROM (SELECT roads.oid AS oid, roads.the_geom AS
the_geom, road_names.name as name
   FROM roads LEFT JOIN road_names ON roads.road_name_id =
```

```
road names.road name id) AS named roads
   USING UNIQUE oid USING SRID=-1"
 MAXSCALE 20000
 STATUS ON
 TYPE ANNOTATION
 LABELITEM name
 CLASS
  LABEL
   ANGLE auto
   SIZE 8
   COLOR 0 192 0
   TYPE truetype
   FONT arial
  END
 END
END
```

This annotation layer adds green labels to all the roads when the scale gets down to 1:20000 or less. It also demonstrates how to use an SQL join in a `DATA` definition.

Converting Data Using ogr2ogr:

See URLs:

Utility programs
http://www.flids.com:16080/websites/ogr/ogr_utilities.html

Details of PostGIS format
http://www.flids.com:16080/websites/ogr/drv_pg.html

Ogr2ogr –f "PostgreSQL" PG:dbname=add

-a_srs 3031
-lco PRECISION=NO
-lco LAUNDER=YES
-nln "table_name"

- GDAL must be compiled with PostgreSQL support
- try the following for a list of supported formats:

  ```
  % ogrinfo --formats
  ```

- if PostgreSQL not listed, u must recompile GDAL --with-pg=/usr/local/pgsql/bin/pg_config

- use PRECISION=NO to avoid numeric field precision errors such as:

```
     ERROR 1: INSERT command for new feature failed.
     ERROR:  numeric field overflow
     DETAIL:  The absolute value is greater than or equal to 10^13 for
field with
     precision 18, scale 5.

     ERROR 1: Terminating translation prematurely after failed
     translation of layer PAL
     example:
     % ogr2ogr -f PostgreSQL PG:dbname=scale05 coast05 PAL -nln
"coast05polyogr" -lco PRECISION=NO
```

- or with PRECISION=YES:

```
     % ogr2ogr -f PostgreSQL PG:dbname=scale05 cont05 -nln "cont05ogr"

ogr2ogr -f "PostgreSQL" PG: user=ezs dbname=myGISdb host=localhost
password=geheim port=5432"
D:\klusjes\LVE\R2058\data\basis\shp\nl_gemeente.shp overwrite=yes
```

Must ensure that EPSGcode parameters are included in proj folder – use locate epsg

Data Loaded Using shp2pgsql:

Scale10

```
#PostgreSQL/PostGIS Bulk Loading script for the Antarctic Digital Database v. 4.1
#Author:  Peter L. Pulsifer
#Organization: Geomatics and Cartographic Research Centre, Carleton University, Ottawa, Canada
#Date:  November, 2004

#Notes:
#A pipe to sed supports the removel of PostgreSQL 'unfriendly' characters like # and - found in the source data.
#The sed /g parameter is used to replace more than one instance of # found in field names of linestring data.
#The sed /g parameter is not used for removing the single instance of the - character.  Using /g can also remove - signs in coordinate
geometry


echo Bulk loading ADD4.1 Scale 10 Data
echo ...loading COAST10 POINT
shp2pgsql -d -s 3031 scale10/coast10_shp/LAB.shp cst10_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add
echo ...loading  COAST10 LINESTRING
shp2pgsql -d -s 3031 scale10/coast10_shp/ARC.shp cst10_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add
echo ...loading  COAST10 POLYGON
shp2pgsql -d -s 3031 scale10/coast10_shp/PAL.shp cst10_polygon add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add
echo ...loading  CONT10 LINESTRING
shp2pgsql -d -s 3031 scale10/cont10_shp/ARC.shp cnt10_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add
echo ...loading  ROCK10 POINT
shp2pgsql -d -s 3031 scale10/rock10_shp/LAB.shp rck10_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add
echo ...loading  ROCK10 LINESTRING
shp2pgsql -d -s 3031 scale10/rock10_shp/ARC.shp rck10_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add
echo ...loading  ROCK10 POLYGON
shp2pgsql -d -s 3031 scale10/rock10_shp/PAL.shp rck10_polygon add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add
```

Scale05

```
#PostgreSQL/PostGIS Bulk Loading script for the Antarctic Digital Database v. 4.1
#Author:  Peter L. Pulsifer
#Organization: Geomatics and Cartographic Research Centre, Carleton University, Ottawa, Canada
#Date:  November, 2004

#Notes:
#A pipe to sed supports the removel of PostgreSQL 'unfriendly' characters like # and - found in the source data.
#The sed /g parameter is used to replace more than one instance of # found in field names of linestring data.
#The sed /g parameter is not used for removing the single instance of the - character.  Using /g can also remove - signs in coordinate
geometry

echo ...bulk loading ADD4.1 Scale 05 Data
echo ...loading COAST05 POINT Data
shp2pgsql -d -s 3031 scale05/coast05_shp/LAB.shp cst05_point add | sed s/#/_/ | sed s/-id/_id/g1 | psql -p 5433 add

echo ...loading COAST05 LINESTRING Data
shp2pgsql -d -s 3031 scale05/coast05_shp/ARC.shp cst05_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ...loading COAST05 POLYGON Data
shp2pgsql -d -s 3031 scale05/coast05_shp/PAL.shp cst05_polygon add | sed s/#/_/ | sed s/-id/_id/ | sed s/19,9/30,9/g | psql -p 5433 add

echo ...loading CONT05 LINESTRING Data
shp2pgsql -d -s 3031 scale05/cont05_shp/ARC.shp cnt05_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ...loading GFLOW05 LINESTRING Data
shp2pgsql -d -s 3031 scale05/gflow05_shp/ARC.shp flw05_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ...loading ROCK05 POINT Data
shp2pgsql -d -s 3031 scale05/rock05_shp/LAB.shp rck05_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ...loading ROCK05 LINESTRING Data
shp2pgsql -d -s 3031 scale05/rock05_shp/ARC.shp rck05_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ...loading ROCK05 POLYGON Data
shp2pgsql -d -s 3031 scale05/rock05_shp/PAL.shp rck05_polygon add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add
```

Scale01

```
#PostgreSQL/PostGIS Bulk Loading script for the Antarctic Digital Database v. 4.1
#Author:  Peter L. Pulsifer
#Organization: Geomatics and Cartographic Research Centre, Carleton University, Ottawa, Canada
#Date:  November, 2004

#Notes:
#A pipe to sed supports the removel of PostgreSQL 'unfriendly' characters like # and - found in the source data.
#The sed /g parameter is used to replace more than one instance of # found in field names of linestring data.
#The sed /g parameter is not used for removing the single instance of the - character.  Using /g ca# This script loads the scale01 data
into the postgres db name add

echo ... processing CLIFF01 LINESTRING Data
shp2pgsql -d -s 3031 scale01/cliff01_shp/arc.shp clf01_point add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ... processing COAST01 POINT Data
shp2pgsql -d -s 3031 scale01/coast01_shp/lab.shp cst01_point add | sed s/#/_/ | sed s/-id/_id/g1 | psql -p 5433 add

echo ... processing COAST01 LINESTRING Data
shp2pgsql -d -s 3031 scale01/coast01_shp/arc.shp cst01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ... processing COAST01 POLYGON Data
shp2pgsql -d -s 3031 scale01/coast01_shp/pal.shp cst01_polygon add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ... processing CONT01 LINESTRING Data
shp2pgsql -d -s 3031 scale01/cont01_shp/arc.shp cnt01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ... processing ELEVN01 POINT Data
```

```
shp2pgsql -d -s 3031 scale01/elevn01_shp/lab.shp elv01_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ... processing FANAU01 POINT Data
shp2pgsql -d -s 3031 scale01/fauna01_shp/lab.shp fna01_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ... processing GFLOW01 LINESTRING Data
shp2pgsql -d -s 3031 scale01/gflow01_shp/arc.shp flw01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ... processing GMARG01 LINESTRING Data
shp2pgsql -d -s 3031 scale01/gmarg01_shp/arc.shp gmg01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ... processing HUMAN01 POINT Data
shp2pgsql -d -s 3031 scale01/human01_shp/lab.shp hmn01_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ... processing IDOME01 POINT Data
shp2pgsql -d -s 3031 scale01/idome01_shp/lab.shp icd01_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

# **** LAKES AND MORAN COVERAGES CURRENTLY NOT AVAILABLE 11-15-2004
#echo ... processing LAKES01 POINT Data
#shp2pgsql -d -s 3031 scale01/lakes01_shp/lab.shp lak01_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

#echo ... processing LAKES01 LINESTRING Data
#shp2pgsql -d -s 3031 scale01/lakes01_shp/arc.shp lak01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

#echo ... processing LAKES01 POLYGON Data
#shp2pgsql -d -s 3031 scale01/lakes01_shp/pal.shp lak01_polygon add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

#echo ... processing MORAN01 POINT Data
#shp2pgsql -d -s 3031 scale01/moran01_shp/lab.shp mrn01_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

#echo ... processing MORAN01 LINESTRING Data
#shp2pgsql -d -s 3031 scale01/moran01_shp/arc.shp mrn01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

#echo ... processing MORAN01 POLYGON Data
#shp2pgsql -d -s 3031 scale01/moran01_shp/pal.shp mrn01_polygon add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ... processing ROCK01 POINT Data
shp2pgsql -d -s 3031 scale01/rock01_shp/lab.shp rck01_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ... processing ROCK01 LINESTRING Data
shp2pgsql -d -s 3031 scale01/rock01_shp/arc.shp rck01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ... processing ROCK01 POLYGON Data
shp2pgsql -d -s 3031 scale01/rock01_shp/pal.shp rck01_polygon add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ... processing STRMS01 LINESTRING Data
shp2pgsql -d -s 3031 scale01/strms01_shp/arc.shp str01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add

echo ... processing TRANSPORT01 POINT Data
shp2pgsql -d -s 3031 scale01/transport01_shp/lab.shp trn01_point add | sed s/#/_/ | sed s/-id/_id/ | psql -p 5433 add

echo ... processing TRAV01 LINESTRING Data
shp2pgsql -d -s 3031 scale01/trav01_shp/arc.shp trv01_linestring add | sed s/#/_/g | sed s/-id/_id/ | psql -p 5433 add
```

Scale00

```
#PostgreSQL/PostGIS Bulk Loading script for the Antarctic Digital Database v. 4.1
#Author:  Peter L. Pulsifer
#Organization: Geomatics and Cartographic Research Centre, Carleton University, Ottawa, Canada
#Date:  November, 2004

#Notes:
#A pipe to sed supports the removel of PostgreSQL 'unfriendly' characters like # and - found in the source data.
#The sed /g parameter is used to replace more than one instance of # found in field names of linestring data.
#The sed /g parameter is not used for removing the single instance of the - character.  Using /g ca#
#This script loads the scale00 data into the postgres db name add
```

```
echo ... processing CLIFF00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/cliff00_shp/ARC.shp clf00_point add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing COAST00 POINT Data
shp2pgsql -d -s 3031 scale00/coast00_shp/LAB.shp cst00_point add | sed s/#/_/ | sed s/-ID/_ID/g1 | psql -p 5433 add

echo ... processing COAST00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/coast00_shp/ARC.shp cst00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing COAST00 POLYGON Data
shp2pgsql -d -s 3031 scale00/coast00_shp/PAL.shp cst00_polygon add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing CONT00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/cont00_shp/ARC.shp cnt00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing ELEVN00 POINT Data
shp2pgsql -d -s 3031 scale00/elevn00_shp/LAB.shp elv00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing FANAU00 POINT Data
shp2pgsql -d -s 3031 scale00/fauna00_shp/LAB.shp fna00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing GFLOW00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/gflow00_shp/ARC.shp flw00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing GMARG00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/gmarg00_shp/ARC.shp gmg00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing HUMAN00 POINT Data
shp2pgsql -d -s 3031 scale00/human00_shp/LAB.shp hmn00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing IDOME00 POINT Data
shp2pgsql -d -s 3031 scale00/idome00_shp/LAB.shp icd00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

# **** LAKES AND MORAN COVERAGES CURRENTLY NOT AVAILABLE 11-15-2004
echo ... processing LAKES00 POINT Data
shp2pgsql -d -s 3031 scale00/lakes00_shp/LAB.shp lak00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing LAKES00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/lakes00_shp/ARC.shp lak00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing LAKES00 POLYGON Data
shp2pgsql -d -s 3031 scale00/lakes00_shp/PAL.shp lak00_polygon add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing MORAN00 POINT Data
shp2pgsql -d -s 3031 scale00/moran00_shp/LAB.shp mrn00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing MORAN00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/moran00_shp/ARC.shp mrn00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing MORAN00 POLYGON Data
shp2pgsql -d -s 3031 scale00/moran00_shp/PAL.shp mrn00_polygon add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing REEF00 POINT Data
shp2pgsql -d -s 3031 scale00/reef00_shp/LAB.shp ref00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing ROCK00 POINT Data
shp2pgsql -d -s 3031 scale00/rock00_shp/LAB.shp rck00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing ROCK00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/rock00_shp/ARC.shp rck00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing ROCK00 POLYGON Data
shp2pgsql -d -s 3031 scale00/rock00_shp/PAL.shp rck00_polygon add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing STRMS00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/strms00_shp/ARC.shp str00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add

echo ... processing TRANSPORT00 POINT Data
shp2pgsql -d -s 3031 scale00/transport00_shp/LAB.shp trn00_point add | sed s/#/_/ | sed s/-ID/_ID/ | psql -p 5433 add
```

```
echo ... processing TRAV00 LINESTRING Data
shp2pgsql -d -s 3031 scale00/trav00_shp/ARC.shp trv00_linestring add | sed s/#/_/g | sed s/-ID/_ID/ | psql -p 5433 add
```

Indexing Data:

Scale10

```
create index cst10_point_gidx on cst10_point using gist (the_geom gist_geometry_ops);
create index cst10_linestring_gidx on cst10_linestring using gist (the_geom gist_geometry_ops);
create index cst10_polygon_gidx on cst10_polygon using gist (the_geom gist_geometry_ops);
create index cnt10_linestring_gidx on cnt10_linestring using gist (the_geom gist_geometry_ops);
create index rck10_point_gidx on rck10_point using gist (the_geom gist_geometry_ops);
create index rck10_linestring_gidx on rck10_linestring using gist (the_geom gist_geometry_ops);
create index rck10_polygon_gidx on rck10_polygon using gist (the_geom gist_geometry_ops);
```

Scale05

```
create index cst05_point_gidx on cst05_point using gist (the_geom gist_geometry_ops);
create index cst05_linestring_gidx on cst05_linestring using gist (the_geom gist_geometry_ops);
create index cst05_polygon_gidx on cst05_polygon using gist (the_geom gist_geometry_ops);
create index cnt05_linestring_gidx on cnt05_linestring using gist (the_geom gist_geometry_ops);
create index flw05_linestring_gidx on flw05_linestring using gist (the_geom gist_geometry_ops);
create index rck05_point_gidx on rck05_point using gist (the_geom gist_geometry_ops);
create index rck05_linestring_gidx on rck05_linestring using gist (the_geom gist_geometry_ops);
create index rck05_polygon_gidx on rck05_polygon using gist (the_geom gist_geometry_ops);
```

Scale01

```
create index clf01_point_gidx on clf01_point using gist (the_geom gist_geometry_ops);
create index cst01_point_gidx on cst01_point using gist (the_geom gist_geometry_ops);
create index cst01_linestring_gidx on cst01_linestring using gist (the_geom gist_geometry_ops);
create index cst01_polygon_gidx on cst01_polygon using gist (the_geom gist_geometry_ops);
create index cnt01_linestring_gidx on cnt01_linestring using gist (the_geom gist_geometry_ops);
create index elv01_point_gidx on elv01_point using gist (the_geom gist_geometry_ops);
create index fna01_point_gidx on fna01_point using gist (the_geom gist_geometry_ops);
create index flw01_linestring_gidx on flw01_linestring using gist (the_geom gist_geometry_ops);
create index gmg01_linestring_gidx on gmg01_linestring using gist (the_geom gist_geometry_ops);
create index hmn01_point_gidx on hmn01_point using gist (the_geom gist_geometry_ops);
create index icd01_point_gidx on icd01_point using gist (the_geom gist_geometry_ops);
create index rck01_point_gidx on rck01_point using gist (the_geom gist_geometry_ops);
create index rck01_linestring_gidx on rck01_linestring using gist (the_geom gist_geometry_ops);
create index rck01_polygon_gidx on rck01_polygon using gist (the_geom gist_geometry_ops);
create index str01_linestring_gidx on str01_linestring using gist (the_geom gist_geometry_ops);
create index trn01_point_gidx on trn01_point using gist (the_geom gist_geometry_ops);
create index trv01_linestring_gidx on trv01_linestring using gist (the_geom gist_geometry_ops);
```

Scale00

```
create index clf00_point_gidx on clf00_point using gist (the_geom gist_geometry_ops);
create index cst00_point_gidx on cst00_point using gist (the_geom gist_geometry_ops);
create index cst00_linestring_gidx on cst00_linestring using gist (the_geom gist_geometry_ops);
create index cst00_polygon_gidx on cst00_polygon using gist (the_geom gist_geometry_ops);
```

```
create index cnt00_linestring_gidx on cnt00_linestring using gist (the_geom gist_geometry_ops);
create index elv00_point_gidx on elv00_point using gist (the_geom gist_geometry_ops);
create index fna00_point_gidx on fna00_point using gist (the_geom gist_geometry_ops);
create index flw00_linestring_gidx on flw00_linestring using gist (the_geom gist_geometry_ops);
create index gmg00_linestring_gidx on gmg00_linestring using gist (the_geom gist_geometry_ops);
create index hmn00_point_gidx on hmn00_point using gist (the_geom gist_geometry_ops);
create index icd00_point_gidx on icd00_point using gist (the_geom gist_geometry_ops);
create index lak00_point_gidx on lak00_point using gist (the_geom gist_geometry_ops);
create index lak00_linestring_gidx on lak00_linestring using gist (the_geom gist_geometry_ops);
create index lak00_polygon_gidx on lak00_polygon using gist (the_geom gist_geometry_ops);
create index mrn00_point_gidx on mrn00_point using gist (the_geom gist_geometry_ops);
create index mrn00_linestring_gidx on mrn00_linestring using gist (the_geom gist_geometry_ops);
create index mrn00_polygon_gidx on mrn00_polygon using gist (the_geom gist_geometry_ops);
create index ref00_point_gidx on ref00_point using gist (the_geom gist_geometry_ops);
create index rck00_point_gidx on rck00_point using gist (the_geom gist_geometry_ops);
create index rck00_linestring_gidx on rck00_linestring using gist (the_geom gist_geometry_ops);
create index rck00_polygon_gidx on rck00_polygon using gist (the_geom gist_geometry_ops);
create index str00_linestring_gidx on str00_linestring using gist (the_geom gist_geometry_ops);
create index trn00_point_gidx on trn00_point using gist (the_geom gist_geometry_ops);
create index trv00_linestring_gidx on trv00_linestring using gist (the_geom gist_geometry_ops);
```

The SCAR Feature Catalogue

Load data (not tested):

Associating with ADD:

Creating Views to Serve WFS:

Configuring Geoserver:

Go To:

http://mywfs.org:8080/geoserver/welcome.do

Configuration files typically stored in:

/usr/local/Jakarta-tomcat-5.5.4/webapps/geoserver

/usr/local/Jakarta-tomcat-5.5.4/webapps/geoserver/WEB-INF/
services.xml
catalog.xml

/usr/local/Jakarta-tomcat-5.5.4/webapps/geoserver/data/featureTypes

/usr/local/Jakarta-tomcat-5.5.4/webapps/geoserver/data/styles/

Using the WFS

http:// mywfs.org:8080/geoserver/wfs?SERVICE=WFS&VERSION=1.1.1&REQUEST=GetCapabilities

http:// mywfs.org:8080/geoserver/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities

http://
mywfs.org:8080/geoserver/wfs?service=wfs&request=GetFeature&Version=1.0.0&TypeName=cst01_polygon&bbox=-2661827,-2489793,2750918,2321608&srs=EPSG:3031&styles=normal&outputFormat=GML2

http:// mywfs.org:8080/geoserver/wms?request=GetMap&layers=cst01_polygon&bbox=-2661827,-2489793,2750918,2321608&width=800&height=800&srs=EPSG:3031&styles=normal&Format=image/svg+xml