INSTALLATION NOTES FOR RIF4.0 DATABASE

Contents

[1. Install Postgres Software. 2](#_Toc384304243)

[1 Windows 2](#_Toc384304244)

[1.1.2 OpenSSL on Windows 4](#_Toc384304245)

[1.1.3 PL/R on Windows 4](#_Toc384304246)

[1.1.4 Creating a Windows Postgres Database Cluster 5](#_Toc384304247)

[1.2 Linux 5](#_Toc384304248)

[1.2.1 Redhat 5](#_Toc384304249)

[1.2.2 Ubuntu 5](#_Toc384304250)

[1.2.3 Building from source on Linux 5](#_Toc384304251)

[1.2.4 Set up Logging 5](#_Toc384304252)

[1.2.5 Creating a Linux Postgres Database Cluster 6](#_Toc384304253)

[2. Creating the SAHSUland database 7](#_Toc384304254)

[2.1 Create the SAHSUland tablespace 7](#_Toc384304255)

[2.2 Run db\_create.sql 8](#_Toc384304256)

[2.2.1 Test Connection to Postgres 11](#_Toc384304257)

[3. RIF Database Development Environment 12](#_Toc384304258)

[Appendix A - Postgres Networking 13](#_Toc384304259)

[A.1 Windows Postgres laptop 13](#_Toc384304260)

[A.2 Linux Postgres server 13](#_Toc384304261)

[A.3 Windows Postgres server 13](#_Toc384304262)

[A.3.1 pg\_hba.conf 13](#_Toc384304263)

[A.3.2 pg\_ident.conf 15](#_Toc384304264)

[A.4 Postgres password file (pgpass) 16](#_Toc384304265)

[Appendix B – Postgres Server Configuration 17](#_Toc384304266)

[Appendix C – SAHSULand db\_create.sql script 25](#_Toc384304267)

# 1. Install Postgres Software.

The new RIF database is being targeted as PostGres 9.3; and will use features requiring this version or later.

Windows and most Linux distributions have pre built packages.

Postgres is usually setup in one of three ways:

* Standalone mode on a Windows firewalled laptop. This uses local database MD5 passwords and no SSL and is not considered secure for network use.
* Secure mode on a Windows server and Active directory network. This uses remote database connections using SSL; with SSPI (Windows GSS connectivity) for psql and secure LDAP for Java connectivity.
* Secure mode on a Linux server and Active directory network. This uses remote database connections using SSL; with GSSAPI/Kerberos for psql and secure LDAP for Java connectivity.

Postgres can proxy users (see ident.conf examples in Appendix A - Postgres Networking). Typically this is used to allow remote postgres administrator user authentication and to logon as the schema owner (rif40).

## 1 Windows

Postgres is best downloaded from Enterprise DB: <http://www.enterprisedb.com/products-services-training/pgdownload>. The Postgres installer then runs stack builder to download and install the additional packages. The following additional packages need to be installed:

* PostGres (database, PG Admin III administration tool, and common extensions)
* PostGIS (Geospatial integration)
* pgAgent (batch engine)
* pgJDBC (Java database connector for PostGres)
* pgODBC (ODBC database connector for PostGres

The following are optional:

* pgBouncer (load balancer; for use if you have a synchronous or near synchronous replica database)

The below scripts will install Enterprise DB Postgres 9.3 if the packages have been downloaded using stackbuilder:

REM Install postgres

REM Download from http://www.enterprisedb.com/products-services-training/pgdownload

postgresql-9.3.2-3-windows-x64.exe --mode unattended --superpassword XXXXXXXXXX

REM Add C:\Program Files\PostgreSQL\9.3\bin to path in future

REM Test Connect to DB

"C:\Program Files\PostgreSQL\9.3\bin\PGadmin3.exe"

REM install

REM "C:\Program Files\PostgreSQL\9.3\bin\stackbuilder.exe"

REM

REM Download: pgagent, pgbouncer, pgjdbc, psqlODBC (32/64 bit), Postgis 64 bit

REM

REM postgis\_2\_1\_pg93.exe --unattendedmodeui none --superpassword XXXXXXXXXX

edb\_pgagent.exe --unattendedmodeui none --pgpassword XXXXXXXXXX --systempassword XXXXXXXXXX --pgAgent Administrator

edb\_pgbouncer.exe --unattendedmodeui none --pgpassword XXXXXXXXXX

edb\_pgjdbc.exe --unattendedmodeui none

edb\_psqlodbc.exe --unattendedmodeui none

Notes:

* The PG Agent installer creates a separate service account and therefore requires the Windows administrator password.
* Be careful to use 32/64 bit installers as appropriate.
* Postgres is installed to: C:\Program Files\PostgreSQL\9.3
* Most configuration files are in: C:\Program Files\PostgreSQL\9.3\data
* System psqlrc (psql command line interface startup script used to setup the RIF) is in: C:\Program Files\PostgreSQL\9.3\etc
* PGpass user password file is found on Microsoft Windows at: *%APPDATA%\postgresql\pgpass.conf.* An example is at: ?.
* GDAL: <http://gis4free.wordpress.com/2011/03/10/how-to-install-and-configure-postgis-raster-on-windows/>
* Postgres software: <http://www.postgresql.org/download/windows/>
* Windows equivalent of SYSTRACE ProcessMonitor: <http://technet.microsoft.com/en-gb/sysinternals/bb896653.aspx>
* Fixing Windows Code page errors

H:\>psql

psql (9.3.2)

WARNING: Console code page (850) differs from Windows code page (1252)

8-bit characters might not work correctly. See psql reference

page "Notes for Windows users" for details.

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

Type "help" for help.

Type:

cmd.exe /c chcp 1252

Or: modify the cmd shortcut to run cmd.exe /k chcp 1252

* Set System pane environment variable defaults: PGDATABASE=sahsuland; PGHOST=WPEA-MRCADMIN1 if not localhost (i.e. remotely connecting)
* Create RIF users:

CREATE USER pch NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN NOREPLICATION;

GRANT rif\_manager TO pch;

GRANT rif\_user TO pch;

CREATE SCHEMA pch AUTHORIZATION pch;

GRANT ALL ON SCHEMA pch TO pch;

GRANT CONNECT ON DATABASE traffic to pch;

ALTER USE pch PASSWORD '<password>';

***WARNING: You must create users (e.g. peterh/federicof/keving) first before import or the objects will end up being owned by the default schema/role***

* Setup pg\_hba.conf, pg\_ident.conf as per A.1 Windows Postgres laptop or A.3 Windows Postgres server. In the second case, an extra users created will need to be mapped in pg\_ident.conf.
* Setup postgresql.conf. As a minum the rif40 custom variable classes must be set:

#------------------------------------------------------------------------------

# CUSTOMIZED OPTIONS

#------------------------------------------------------------------------------

custom\_variable\_classes = 'rif40' # list of custom variable class names

#

rif40.debug = '' # functionA:levelA, functionB:levelB, ...

# level is DEBUG1...4

rif40.send\_debug\_to\_info = off # Send debug output to info (i.e. screen)

#

# Eof

A server example is given at: Appendix B – Postgres Server Configuration

Errors are found in the Windows server application event log file

* Check Postgres (the database administrator user) can log on using psql and PGAdminIII

### 1.1.2 OpenSSL on Windows

Postgres uses OpenSSL to secure connections using TLS. Open SSL (from <http://www.openssl.org/related/binaries.html>); see http://vibhorkumar.wordpress.com/2011/07/17/how-to-enable-ssl-in-postgresqlppas/ for instructions.

Notes:

* WARNING: can't open config file: /usr/local/ssl/openssl.cnf - see: <http://jaspreetchahal.org/warning-cant-open-config-file-usrlocalsslopenssl-cnf/>

### 1.1.4 Creating a Windows Postgres Database Cluster

This is usually done by the Enterprise DB installer. To be added.

## 1.2 Linux

### 1.2.1 Redhat

### 1.2.2 Ubuntu

Ubuntu will install from pre-built packages; see: <http://trac.osgeo.org/postgis/wiki/UsersWikiPostGIS20Ubuntu1204>. Essentially you need to add unstable GIS libraries. The URL also contains build instructions. As of December 2013 Postgres 9.2 is the latest version;

Ubuntu install packages:

* postgis postgresql-9.2 postgresql-9.2-postgis postgresql-client-9.2 postgresql-client-common postgresql-common postgresql-contrib-9.2 postgresql-server-dev-9.2

### 1.2.3 Building from source on Linux

The script build.sh will build Postgres and PostGIS from the sources. This was what was done on the private network. The following are eequired libraries. You will probably have to build them:

* GEOS 3.3.5
* PROJ 4.8.0 (--without-jni option)
* GDAL 1.9.1
* JSON
* R

Also required:

* Perl, Python, TCL, dtrace, Kerberos, GSSAPI, PAM, OpenSSL, XML, XSLT

Oracle FDW - needs Oracle - this is the cause of the --without-ldap flag as Oracle has these compiled it so any client of Oracle has to use its libraries. You will get a dynamic link error if you try when the oracle\_fdw extension loads.

### 1.2.4 Set up Logging

postgresql.conf is set to log to LOCAL3; logs are processed by either

* SYSLOG

Edit /etc/syslog.conf

LOCAL3.\* /var/log/postgres

Restart the syslogd service.

* RSYSLOD

Create /etc/rsyslog.d/70-postgres.conf and add:

LOCAL3.\* /var/log/postgres

Restart the rsyslogd service. Note that RSYSLOG can log to postgres. This allows an audit trail to be created on a remote Postgres database (e.g. a replication cluster)

See: <http://www.rsyslog.com/doc/rsyslog_pgsql.html>

### 1.2.5 Creating a Linux Postgres Database Cluster

CD into the configuration directory:

* Unbuntu distro: /etc/postgresql/9.1/main
* Build from source: /usr/local/pgsql/share and PG\_DATA

Save old postgresql.conf; merged supplied server version (Appendix B – Postgres Server Configuration) over original, restart PostGres.

The distribution will have hard coded the 'ConfigDir'; normally PGDATA is set in a script (environment on Ubuntu). If you do change this Postgres will not start complaining it cannot find the cluster database.

e.g.

root@ph-laptop:/etc/postgresql/9.3/main# service postgresql restart

Error: Invalid data directory

\* No PostgreSQL clusters exist; see "man pg\_createcluster"

If you use pg\_ctl and not pg\_clusterctl (you get one or the other) to start PostGres PGDATA can be set in the init script:

PGDATA=/var/lib/postgresql/9.3/main

export PGDATA

Or you can use pg\_ctl.conf:

pg\_ctl\_options = '-D /var/lib/postgresql/9.3/main'

If you use pg\_clusterctl it is best to edit postgresql.conf - change 'ConfigDir' in data\_directory, hba\_file, ident\_file

e.g. On turing (old SAHSU Private network development server) PGDATA is */oradisk1/rif/data*

root@ph-laptop:/etc/postgresql/9.3/main# service postgresql restart

\* Restarting PostgreSQL 9.1 database server

\* The PostgreSQL server failed to start. Please check the log output.

[fail]

If it fails, see the log: */var/log/postgres*

root@ph-laptop:/etc/postgresql/9.1/main# tail /var/log/postgres

Mar 15 10:14:42 ph-laptop postgres[20965]: [1-1] 20965 2013-03-15 10:14:42.005 GMT FATAL: XX000: could not create shared memory segment: Invalid argument

Mar 15 10:14:42 ph-laptop postgres[20965]: [1-2] 20965 2013-03-15 10:14:42.005 GMT DETAIL: Failed system call was shmget(key=5432001, size=572121088, 03600).

Mar 15 10:14:42 ph-laptop postgres[20965]: [1-3] 20965 2013-03-15 10:14:42.005 GMT HINT: This error usually means that PostgreSQL's request for a shared memory segment exceeded your kernel's SHMMAX parameter. You can either reduce the request size or reconfigure the kernel with larger SHMMAX. To reduce the request size (currently 572121088 bytes), reduce PostgreSQL's shared memory usage, perhaps by reducing shared\_buffers or max\_connections.

Mar 15 10:14:42 ph-laptop postgres[20965]: [1-4] #011If the request size is already small, it's possible that it is less than your kernel's SHMMIN parameter, in which case raising the request size or reconfiguring SHMMIN is called for.

Mar 15 10:14:42 ph-laptop postgres[20965]: [1-5] #011The PostgreSQL documentation contains more information about shared memory configuration.

Mar 15 10:14:42 ph-laptop postgres[20965]: [1-6] 20965 2013-03-15 10:14:42.005 GMT LOCATION: InternalIpcMemoryCreate, pg\_shmem.c:185

* In this case the shared memory parameters are wrong

root@ph-laptop:/etc/postgresql/9.1/main# sysctl -a | grep shm

kernel.shm\_rmid\_forced = 0

kernel.shmall = 2097152

kernel.shmmax = 33554432

kernel.shmmni = 4096

vm.hugetlb\_shm\_group = 0

I.e. SHMMAX is 32M

Edit /etc/sysctl.conf, add kernel.shmmax = 67108864 and reload with sysctl -p

root@ph-laptop:/etc/postgresql/9.3/main# sysctl -p

kernel.shmmax = 67108864

Start the service

root@ph-laptop:/etc/postgresql/9.1/main# service postgresql restart

\* Restarting PostgreSQL 9.1 database server [ OK ]

Find where psqlrc is located:

peter@ph-laptop:~$ strace -o psql.out psql -U peter -d sahsuland

psql (9.1.7)

Type "help" for help.

sahsuland=> \q

peter@ph-laptop:~$ grep psqlrc p.out

grep: p.out: No such file or directory

peter@ph-laptop:~$ grep psqlrc psql.out

access("/etc/postgresql-common/psqlrc-9.1.7", R\_OK) = -1 ENOENT (No such file or directory)

access("/etc/postgresql-common/psqlrc", R\_OK) = -1 ENOENT (No such file or directory)

access("/home/peter/.psqlrc-9.1.7", R\_OK) = -1 ENOENT (No such file or directory)

access("/home/peter/.psqlrc", R\_OK) = -1 ENOENT (No such file or directory)

* Copy the supplied psqlrc into (in this case) /etc/postgresql-common/psqlrc

# 2. Creating the SAHSUland database

## 2.1 Create the SAHSUland tablespace

* Check login (as postgres)

root@ph-laptop:/etc/postgresql/9.1/main# su - postgres

postgres@ph-laptop:~$ psql

psql (9.1.7)

Type "help" for help.

postgres=#

* Check you are postgres

postgres=# select user;

current\_user

--------------

postgres

(1 row)

* Create the SAHSUland tablespace. The database must have full control over this directory; on Windows Postgres runs (somewhat insecurely) as *NETWORK SERVICE*; on Linux as *postgres*.

CREATE TABLESPACE sahsuland LOCATION '/var/lib/postgresql/9.1/sahsuland';

Or:

CREATE TABLESPACE sahsuland LOCATION 'C:\\PostgresDB\\sahsuland';

## 2.2 Run db\_create.sql

* Run db\_create.sql. This wil also re-create the database.The password is the encrypted version:

psql -U postgres -d postgres -v encrypted\_password=md57bf0096f3dde481e6802bd534959821c -e -f db\_create.sql

C:\Users\pch\Documents>psql -U postgres -d postgres -v encrypted\_password=md57bf0096f3dde481e6802bd534959821c -e -f db\_create.sql

You are connected to database "postgres" as user "postgres" on host "localhost" at port "5432".

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

DO LANGUAGE plpgsql $$

DECLARE

c1 CURSOR FOR

SELECT p.proname

FROM pg\_proc p, pg\_namespace n

WHERE p.proname = 'rif40\_startup'

AND n.nspname = 'rif40\_sql\_pkg'

AND p.proowner = (SELECT oid FROM pg\_roles WHERE rolname = 'rif40')

AND n.oid = p.pronamespace;

--

c1\_rec RECORD;

BEGIN

OPEN c1;

FETCH c1 INTO c1\_rec;

CLOSE c1;

--

IF c1\_rec.proname = 'rif40\_startup' THEN

PERFORM rif40\_sql\_pkg.rif40\_startup();

ELSE

RAISE INFO 'RIF startup: not a RIF database';

END IF;

END;

$$;

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: RIF startup: not a RIF database

DO

psql:db\_create.sql:119: INFO: db\_create.sql() encrypted password="md57bf0096f3dde481e6802bd534959821c"

psql:db\_create.sql:134: INFO: db\_create.sql() User check: postgres

psql:db\_create.sql:158: INFO: db\_create.sql() RIF required Postgres version 9.3 or higher OK; current version: PostgreSQL 9.3.2, co

mpiled by Visual C++ build 1600, 64-bit

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: adminpack V1.0 is installable

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: plperl V1.0 is installable

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: postgis V2.1.1 is installable

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: postgis\_topology V2.1.1 is installable

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: pgcrypto V1.0 is installable

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: sslinfo V1.0 is installable

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: xml2 V1.0 is installable

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: dblink V1.1 is installable

psql:db\_create.sql:191: INFO: db\_create.sql() RIF required extension: plr V8.3.0.15 is installable

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* WARNING !!!

\*

\* This script will drop sahsuland, re-create it and reset the passwords to the

\* default

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Press any key to continue or <control-C> to interrupt:

Creating users/roles...

psql:db\_create.sql:309: INFO: SQL> ALTER USER postgres ENCRYPTED PASSWORD 'md57bf0096f3dde481e6802bd534959821c';

psql:db\_create.sql:309: INFO: db\_create.sql() RIF schema user rif40 exists

psql:db\_create.sql:309: INFO: db\_create.sql() RIF schema user gis exists

psql:db\_create.sql:309: INFO: db\_create.sql() RIF schema user pop exists

psql:db\_create.sql:309: INFO: db\_create.sql() RIF schema user peterh exists

psql:db\_create.sql:309: INFO: db\_create.sql() RIF schema role rif\_user exists

psql:db\_create.sql:309: INFO: db\_create.sql() RIF schema role rif\_manager exists

psql:db\_create.sql:309: INFO: db\_create.sql() RIF schema role rif\_no\_suppression exists

psql:db\_create.sql:309: INFO: db\_create.sql() RIF schema role rifupg34 exists

psql:db\_create.sql:309: INFO: SQL> REVOKE CREATE ON SCHEMA public FROM PUBLIC;

Creating SAHSULAND...

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

You are now connected to database "sahsuland" as user "postgres".

psql:db\_create.sql:357: INFO: db\_create.sql() User check: postgres

Restoring database...

pg\_restore : pg\_restore: connecting to database for restore

At line:1 char:1

+ pg\_restore -d sahsuland -U postgres -v sahsuland.dump 2>&1 | tee ('pg\_restore{0 ...

+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

+ CategoryInfo : NotSpecified: (pg\_restore: con...ase for restore:String) [], RemoteException

+ FullyQualifiedErrorId : NativeCommandError

pg\_restore: creating COMMENT sahsuland\_dev

pg\_restore: creating SCHEMA gis

pg\_restore: creating SCHEMA pch

pg\_restore: creating SCHEMA peterh

pg\_restore: creating SCHEMA pop

pg\_restore: creating SCHEMA public

pg\_restore: creating COMMENT SCHEMA public

...

pg\_restore: setting owner and privileges for FK CONSTRAINT t\_rif40\_studies\_geography\_fk

pg\_restore: setting

owner and privileges for FK CONSTRAINT t\_rif40\_

study\_sql\_sid\_line\_fk

pg\_restore: setting owner and privileges for FK CONSTRAINT t\_rif40\_study\_sqllog\_stdid\_fk

pg\_restore: setting owner and privileges for FK CONSTRAINT t\_rif40\_studyareas\_study\_id\_fk

* Check database restore errors. The following are innocuous.

pg\_restore: [archiver (db)] could not execute query: ERROR: could not load library "C:/Program Files/PostgreSQL/9.3/lib/plperl.dll": The specified module could not be found.

pg\_restore: [archiver (db)] could not execute query: ERROR: extension "plperl" does not exist

pg\_restore: [archiver (db)] could not execute query: ERROR: could not open extension control file "C:/Program Files/PostgreSQL/9.3/share/extension/oracle\_fdw.control": No such file or directory

pg\_restore: [archiver (db)] could not execute query: ERROR: extension "oracle\_fdw" does not exist

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_delete" for relation "geometry\_columns" already exists

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_insert" for relation "geometry\_columns" already exists

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_update" for relation "geometry\_columns" already exists

Linux restore example (this assumes a Bourne shell)

pg\_restore -d sahsuland -v sahsuland.dump 2>&1 | tee restore.log

There will usually be a few errors when transferring data from other systems (i.e. loading the supplied database)

WARNING: errors ignored on restore: 13

Search the restore log for errors

egrep 'archiver|Command' restore.log

pg\_restore: [archiver (db)] Error while PROCESSING TOC:

pg\_restore: [archiver (db)] Error from TOC entry 15; 2615 307691 SCHEMA peterh peterh

pg\_restore: [archiver (db)] could not execute query: ERROR: role "peterh" does not exist

Command was: ALTER SCHEMA peterh OWNER TO peterh;

pg\_restore: [archiver (db)] Error from TOC entry 290; 3079 17745 EXTENSION plperl

pg\_restore: [archiver (db)] could not execute query: ERROR: could not open extension control file "/usr/share/postgresql/9.1/extension/plperl.control": No such file or directory

Command was: CREATE EXTENSION IF NOT EXISTS plperl WITH SCHEMA pg\_catalog;

pg\_restore: [archiver (db)] Error from TOC entry 4762; 0 0 COMMENT EXTENSION plperl

pg\_restore: [archiver (db)] could not execute query: ERROR: extension "plperl" does not exist

Command was: COMMENT ON EXTENSION plperl IS 'PL/Perl procedural language';

pg\_restore: [archiver (db)] Error from TOC entry 295; 3079 10872087 EXTENSION oracle\_fdw

pg\_restore: [archiver (db)] could not execute query: ERROR: could not open extension control file "/usr/share/postgresql/9.1/extension/oracle\_fdw.control": No such file or directory

Command was: CREATE EXTENSION IF NOT EXISTS oracle\_fdw WITH SCHEMA public;

pg\_restore: [archiver (db)] Error from TOC entry 4766; 0 0 COMMENT EXTENSION oracle\_fdw

pg\_restore: [archiver (db)] could not execute query: ERROR: extension "oracle\_fdw" does not exist

Command was: COMMENT ON EXTENSION oracle\_fdw IS 'foreign data wrapper for Oracle access';

pg\_restore: [archiver (db)] Error from TOC entry 206; 1259 13000430 TABLE t\_rif40\_num\_denom peterh

pg\_restore: [archiver (db)] could not execute query: ERROR: role "peterh" does not exist

Command was: ALTER TABLE peterh.t\_rif40\_num\_denom OWNER TO peterh;

pg\_restore: [archiver (db)] Error from TOC entry 289; 1259 13007943 VIEW rif40\_num\_denom peterh

pg\_restore: [archiver (db)] could not execute query: ERROR: role "peterh" does not exist

Command was: ALTER TABLE peterh.rif40\_num\_denom OWNER TO peterh;

pg\_restore: [archiver (db)] Error from TOC entry 288; 1259 13007938 VIEW rif40\_num\_denom\_errors peterh

pg\_restore: [archiver (db)] could not execute query: ERROR: role "peterh" does not exist

Command was: ALTER TABLE peterh.rif40\_num\_denom\_errors OWNER TO peterh;

pg\_restore: [archiver (db)] Error from TOC entry 207; 1259 13000450 VIEW rif40\_user\_version peterh

pg\_restore: [archiver (db)] could not execute query: ERROR: role "peterh" does not exist

Command was: ALTER TABLE peterh.rif40\_user\_version OWNER TO peterh;

pg\_restore: [archiver (db)] Error from TOC entry 4369; 2618 10871427 RULE geometry\_columns\_delete postgres

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_delete" for relation "geometry\_columns" already exists

Command was: CREATE RULE geometry\_columns\_delete AS ON DELETE TO geometry\_columns DO INSTEAD NOTHING;

pg\_restore: [archiver (db)] Error from TOC entry 4367; 2618 10871425 RULE geometry\_columns\_insert postgres

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_insert" for relation "geometry\_columns" already exists

Command was: CREATE RULE geometry\_columns\_insert AS ON INSERT TO geometry\_columns DO INSTEAD NOTHING;

pg\_restore: [archiver (db)] Error from TOC entry 4368; 2618 10871426 RULE geometry\_columns\_update postgres

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_update" for relation "geometry\_columns" already exists

Command was: CREATE RULE geometry\_columns\_update AS ON UPDATE TO geometry\_columns DO INSTEAD NOTHING;

pg\_restore: [archiver (db)] Error from TOC entry 4750; 0 0 ACL peterh peterh

pg\_restore: [archiver (db)] could not execute query: ERROR: role "peterh" does not exist

Command was: REVOKE ALL ON SCHEMA peterh FROM PUBLIC;

To redo the restore DROP DATABASE sahsuland and then CREATE again using the last two commands in the *db\_create.sql* script

### 2.2.1 Test Connection to Postgres

1. Test connection to Postgres using psql:

P:\Github\rapidInquiryFacility\rifDatabase\Postgres\psql\_scripts>psql

Active code page: 1252

You are connected to database "sahsuland\_dev" as user "pch" on host "wpea-rif1" at port "5432".

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_log\_setup() send DEBUG to INFO: off; debug function list: []

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): SQL> SET search\_path TO pch,rif40, public, topology, gis, pop, rif40\_sql\_pkg,rif\_studies;

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): PostGIS extension V2.1.1 (POSTGIS="2.1.1 r12113" GEOS="3.4.2-CAPI-1.8.2 r3924" PROJ="Rel. 4.8.0, 6 March

2012" GDAL="GDAL 1.10.0, released 2013/04/24" LIBXML="2.7.8" LIBJSON="UNKNOWN" TOPOLOGY RASTER)

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): FDW functionality disabled - FDWServerName, FDWServerType, FDWDBServer RIF parameters not set.

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): V$Revision: 1.11 $ DB version $Revision: 1.11 $ matches

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): V$Revision: 1.11 $ rif40\_geographies, rif40\_tables, rif40\_health\_study\_themes exist for user: pch

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): search\_path: rif40, public, topology, gis, pop, rif40\_sql\_pkg, rif\_studies, reset: rif40, public, topolo

gy, gis, pop, rif40\_sql\_pkg

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: WARNING: GLOBAL is deprecated in temporary table creation at character 8

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): Created temporary table: g\_rif40\_study\_areas

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: WARNING: GLOBAL is deprecated in temporary table creation at character 8

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): Created temporary table: g\_rif40\_comparison\_areas

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: INFO: rif40\_startup(): Deleted 0, created 2 tables/views/foreign data wrapper tables

DO

psql (9.3.2)

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

Type "help" for help.

sahsuland\_dev=>

1. Test you can access RIF40\_NUM\_DENOM (the automatic numerator/denominator pairs table)

sahsuland=> SELECT \* FROM rif40\_num\_denom;

geography | numerator\_table | numerator\_description | theme\_description | denominator\_table | denominator\_description | automatic

-----------+------------------+----------------------------+------------------------------------------+-------------------+-------------------------+-----------

SAHSU | SAHSULAND\_CANCER | Cancer cases in SAHSU land | SAHSU land cancer incidence example data | SAHSULAND\_POP | SAHSU land population | 1

(1 row)

Common issues with the psqlrc logon script - to be handled better in startup:

1. No rif40.\* custom variable parameters set in postgresql.conf

C:\Users\pch\Documents\database\postgres\psql\_scripts>psql

You are connected to database "sahsuland" as user "pch" on host "localhost" at port "5432".

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: ERROR: unrecognized configuration parameter "rif40.send\_debug\_to\_info"

psql (9.3.2)

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

Type "help" for help.

1. RIF40 not in search path

RIF40 not in search path

* Set search path (should be done by db\_create.sql)

ALTER DATABASE sahsuland SET search\_path TO rif40,public,topology,gis,pop,rif40\_sql\_pkg;

1. Schema "pch" does not exist

psql:C:/Program Files/PostgreSQL/9.3/etc/psqlrc:36: ERROR: schema "pch" does not exist

* User owned schema not created

# 3. RIF Database Development Environment

The principal build script is v4\_0\_create\_sahsuland.sql in the *rapidInquiryFacility* Github directory rifDatabase\Postgres\psql\_scripts (<https://github.com/kgarwood/rapidInquiryFacility/blob/master/rifDatabase/Postgres/psql_scripts/v4_0_create_sahsuland.sql>).

The script must be run as the schema owner (rif40) on the *sahsuland\_dev* Postgres database only. sahusland is always created afresh from an empty database using *pg\_restore*. E.g

cd rifDatabase\Postgres\psql

psql -U postgres -d Postgres

* Create the SAHSULAND\_DEV tablespace:

CREATE TABLESPACE sahsuland\_dev LOCATION 'C:\\PostgresDB\\sahsuland\_dev';

* (Re-)create *sahsland\_dev* database. This requires *sahsuland\_dev.dump*, otherwise edit *create\_sahsuland\_dev\_db.sql* to build direct from scripts:

psql -U postgres -d postgres -w -e -f create\_sahsuland\_dev\_db.sql

* Check sahsuland builds from the scripts:

psql -U rif40 -d sahsuland\_dev -w -e -f v4\_0\_create\_sahsuland.sql

To create sahsuland.dmp, excluding UK91, EW01 shapefiles from non dev dumps:

pg\_dump -U postgres -w -F custom -T '\*x\_uk\*' -T '\*.x\_ew01\*' -v sahsuland > C:\Users\pch\sahsuland.dump

As of March 2014, the Java test programs (dumpdata) and the installation notes and scripts are not in the repository. They will be added when tidied some more.

# Appendix A - Postgres Networking

## A.1 Windows Postgres laptop

Standalone mode on a Windows firewalled laptop. This uses local database MD5 passwords and no SSL and is not considered secure for network use.

## A.2 Linux Postgres server

Secure mode on a Windows server and Active directory network. This uses remote database connections using SSL; with SSPI (Windows GSS connectivity) for psql and secure LDAP for Java connectivity.

## A.3 Windows Postgres server

Secure mode on a Linux server and Active directory network. This uses remote database connections using SSL; with GSSAPI/Kerberos for psql and secure LDAP for Java connectivity.

This is not yet complete

### A.3.1 pg\_hba.conf

# PostgreSQL Client Authentication Configuration File

# ===================================================

#

# Refer to the "Client Authentication" section in the PostgreSQL

# documentation for a complete description of this file. A short

# synopsis follows.

#

# This file controls: which hosts are allowed to connect, how clients

# are authenticated, which PostgreSQL user names they can use, which

# databases they can access. Records take one of these forms:

#

# local DATABASE USER METHOD [OPTIONS]

# host DATABASE USER ADDRESS METHOD [OPTIONS]

# hostssl DATABASE USER ADDRESS METHOD [OPTIONS]

# hostnossl DATABASE USER ADDRESS METHOD [OPTIONS]

#

# (The uppercase items must be replaced by actual values.)

#

# The first field is the connection type: "local" is a Unix-domain

# socket, "host" is either a plain or SSL-encrypted TCP/IP socket,

# "hostssl" is an SSL-encrypted TCP/IP socket, and "hostnossl" is a

# plain TCP/IP socket.

#

# DATABASE can be "all", "sameuser", "samerole", "replication", a

# database name, or a comma-separated list thereof. The "all"

# keyword does not match "replication". Access to replication

# must be enabled in a separate record (see example below).

#

# USER can be "all", a user name, a group name prefixed with "+", or a

# comma-separated list thereof. In both the DATABASE and USER fields

# you can also write a file name prefixed with "@" to include names

# from a separate file.

#

# ADDRESS specifies the set of hosts the record matches. It can be a

# host name, or it is made up of an IP address and a CIDR mask that is

# an integer (between 0 and 32 (IPv4) or 128 (IPv6) inclusive) that

# specifies the number of significant bits in the mask. A host name

# that starts with a dot (.) matches a suffix of the actual host name.

# Alternatively, you can write an IP address and netmask in separate

# columns to specify the set of hosts. Instead of a CIDR-address, you

# can write "samehost" to match any of the server's own IP addresses,

# or "samenet" to match any address in any subnet that the server is

# directly connected to.

#

# METHOD can be "trust", "reject", "md5", "password", "gss", "sspi",

# "krb5", "ident", "peer", "pam", "ldap", "radius" or "cert". Note that

# "password" sends passwords in clear text; "md5" is preferred since

# it sends encrypted passwords.

#

# OPTIONS are a set of options for the authentication in the format

# NAME=VALUE. The available options depend on the different

# authentication methods -- refer to the "Client Authentication"

# section in the documentation for a list of which options are

# available for which authentication methods.

#

# Database and user names containing spaces, commas, quotes and other

# special characters must be quoted. Quoting one of the keywords

# "all", "sameuser", "samerole" or "replication" makes the name lose

# its special character, and just match a database or username with

# that name.

#

# This file is read on server startup and when the postmaster receives

# a SIGHUP signal. If you edit the file on a running system, you have

# to SIGHUP the postmaster for the changes to take effect. You can

# use "pg\_ctl reload" to do that.

# Put your actual configuration here

# ----------------------------------

#

# If you want to allow non-local connections, you need to add more

# "host" records. In that case you will also need to make PostgreSQL

# listen on a non-local interface via the listen\_addresses

# configuration parameter, or via the -i or -h command line switches.

# TYPE DATABASE USER ADDRESS METHOD

# IPv4, IPv6 local connections:

#

host all postgres 127.0.0.1/32 md5

host all postgres ::1/128 md5

hostssl all postgres 146.179.138.xxx 255.255.255.255 md5

#

# Allow local connections as schema owner (usually use a proxy)

#

#hostssl sahsuland pop 127.0.0.1/32 md5

#hostssl sahsuland pop ::1/128 md5

#hostssl sahsuland gis 127.0.0.1/32 md5

#hostssl sahsuland gis ::1/128 md5

#hostssl sahsuland rif40 127.0.0.1/32 md5

#hostssl sahsuland rif40 ::1/128 md5

#

# Active directory GSSAPI connections with pg\_ident.conf maps for schema accounts

#

hostssl sahsuland all 127.0.0.1/32 sspi map=sahsuland

hostssl sahsuland all ::1/128 sspi map=sahsuland

hostssl sahsuland\_dev all 127.0.0.1/32 sspi map=sahsuland\_dev

hostssl sahsuland\_dev all ::1/128 sspi map=sahsuland\_dev

#

# Allow remote access from specified IP addresses by:

#

# a) SSPI (Windows native GSS [Kerberos] machanism

#

hostssl sahsuland all 146.179.138. xxx 255.255.255.255 sspi map=sahsuland

hostssl sahsuland\_dev all 146.179.138. xxx 255.255.255.255 sspi map=sahsuland\_dev

#

# b) LDAP (to be fixed – need to use different server

#

# hostssl sahsuland\_dev all 146.179.138.157 255.255.255.255 ldap ldapurl="ldaps:// xxx.ic.ac.uk/basedn;cn=;,o=Imperial College,c=GB"

#

# No LDAP URLs or username map on Windows

#

# 2014-03-12 13:44:24 GMT LOG: 00000: LDAP login failed for user "cn=pch,o=Imperial College,c=GB" on server " xxx.ic.ac.uk": Invalid Credentials 2014-03-12 13:44:24 GMT LOCATION: CheckLDAPAuth, src\backend\libpq\auth.c:2321

#

#host sahsuland\_dev all 146.179.138.157 255.255.255.255 ldap ldapserver= xxx.ic.ac.uk ldapprefix="uid=" ldapsuffix=",ou=phs,o=Imperial College,c=GB"

#

# 2014-03-12 13:50:33 GMT LOG: 00000: LDAP login failed for user "pch@IC.AC.UK" on server " xxx.ic.ac.uk": Invalid DN Syntax 2014-03-12 13:50:33 GMT LOCATION: CheckLDAPAuth, src\backend\libpq\auth.c:2321

#

#host sahsuland\_dev all 146.179.138.157 255.255.255.255 ldap ldapserver= xxx.ic.ac.uk ldapprefix= ldapsuffix="@IC.AC.UK"

#host sahsuland\_dev all 146.179.138.157 255.255.255.255 ldap ldapserver= xxx.ic.ac.uk ldapprefix= ldapsuffix=",o=Imperial College,c=GB"

#

# Other databases

#

hostssl traffic all 127.0.0.1/32 sspi

hostssl traffic all ::1/128 sspi

hostssl traffic all 146.179.138. xxx 255.255.255.255 sspi

#

#host all all 127.0.0.1/32 md5

#host all all ::1/128 md5

# Allow replication connections from localhost, by a user with the

# replication privilege.

#host replication postgres 127.0.0.1/32 md5

#host replication postgres ::1/128 md5

### A.3.2 pg\_ident.conf

# PostgreSQL User Name Maps

# =========================

#

# Refer to the PostgreSQL documentation, chapter "Client

# Authentication" for a complete description. A short synopsis

# follows.

#

# This file controls PostgreSQL user name mapping. It maps external

# user names to their corresponding PostgreSQL user names. Records

# are of the form:

#

# MAPNAME SYSTEM-USERNAME PG-USERNAME

#

# (The uppercase quantities must be replaced by actual values.)

#

# MAPNAME is the (otherwise freely chosen) map name that was used in

# pg\_hba.conf. SYSTEM-USERNAME is the detected user name of the

# client. PG-USERNAME is the requested PostgreSQL user name. The

# existence of a record specifies that SYSTEM-USERNAME may connect as

# PG-USERNAME.

#

# If SYSTEM-USERNAME starts with a slash (/), it will be treated as a

# regular expression. Optionally this can contain a capture (a

# parenthesized subexpression). The substring matching the capture

# will be substituted for \1 (backslash-one) if present in

# PG-USERNAME.

#

# Multiple maps may be specified in this file and used by pg\_hba.conf.

#

# No map names are defined in the default configuration. If all

# system user names and PostgreSQL user names are the same, you don't

# need anything in this file.

#

# This file is read on server startup and when the postmaster receives

# a SIGHUP signal. If you edit the file on a running system, you have

# to SIGHUP the postmaster for the changes to take effect. You can

# use "pg\_ctl reload" to do that.

# Put your actual configuration here

# ----------------------------------

# MAPNAME SYSTEM-USERNAME PG-USERNAME

#

sahsuland pch pop

sahsuland pch gis

sahsuland pch rif40

sahsuland pch pch

#

sahsuland\_dev pch pop

sahsuland\_dev pch gis

sahsuland\_dev pch rif40

sahsuland\_dev pch pch

sahsuland\_dev pch postgres

#

# Eof

## A.4 Postgres password file (pgpass)

One line per host, database and account. Order is:

* Host
* Port
* Database (usually \*)
* User
* Password

localhost:5432:\*:postgres:XXXXXXX

localhost:5432:\*:peterh: XXXXXXX

wpea-pch:5432:\*:peterh: XXXXXXX

wpea-rif1:5432:\*:postgres: XXXXXXX

wpea-rif1:5432:\*:pch: XXXXXXX

# Appendix B – Postgres Server Configuration

# -----------------------------

# PostgreSQL configuration file

# -----------------------------

#

# This file consists of lines of the form:

#

# name = value

#

# (The "=" is optional.) Whitespace may be used. Comments are introduced with

# "#" anywhere on a line. The complete list of parameter names and allowed

# values can be found in the PostgreSQL documentation.

#

# The commented-out settings shown in this file represent the default values.

# Re-commenting a setting is NOT sufficient to revert it to the default value;

# you need to reload the server.

#

# This file is read on server startup and when the server receives a SIGHUP

# signal. If you edit the file on a running system, you have to SIGHUP the

# server for the changes to take effect, or use "pg\_ctl reload". Some

# parameters, which are marked below, require a server shutdown and restart to

# take effect.

#

# Any parameter can also be given as a command-line option to the server, e.g.,

# "postgres -c log\_connections=on". Some parameters can be changed at run time

# with the "SET" SQL command.

#

# Memory units: kB = kilobytes Time units: ms = milliseconds

# MB = megabytes s = seconds

# GB = gigabytes min = minutes

# h = hours

# d = days

#------------------------------------------------------------------------------

# FILE LOCATIONS

#------------------------------------------------------------------------------

# The default values of these variables are driven from the -D command-line

# option or PGDATA environment variable, represented here as ConfigDir.

#data\_directory = 'ConfigDir' # use data in another directory

# (change requires restart)

#hba\_file = 'ConfigDir/pg\_hba.conf' # host-based authentication file

# (change requires restart)

#ident\_file = 'ConfigDir/pg\_ident.conf' # ident configuration file

# (change requires restart)

# If external\_pid\_file is not explicitly set, no extra PID file is written.

#external\_pid\_file = '(none)' # write an extra PID file

# (change requires restart)

#------------------------------------------------------------------------------

# CONNECTIONS AND AUTHENTICATION

#------------------------------------------------------------------------------

# - Connection Settings -

listen\_addresses = '\*' # what IP address(es) to listen on;

# comma-separated list of addresses;

# defaults to 'localhost', '\*' = all

# (change requires restart)

port = 5432 # (change requires restart)

max\_connections = 100 # (change requires restart)

# Note: Increasing max\_connections costs ~400 bytes of shared memory per

# connection slot, plus lock space (see max\_locks\_per\_transaction).

#superuser\_reserved\_connections = 3 # (change requires restart)

#unix\_socket\_directory = '' # (change requires restart)

#unix\_socket\_group = '' # (change requires restart)

#unix\_socket\_permissions = 0777 # begin with 0 to use octal notation

# (change requires restart)

#bonjour = off # advertise server via Bonjour

# (change requires restart)

#bonjour\_name = '' # defaults to the computer name

# (change requires restart)

# - Security and Authentication -

#authentication\_timeout = 1min # 1s-600s

ssl = on # (change requires restart)

#ssl\_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers

# (change requires restart)

#ssl\_renegotiation\_limit = 512MB # amount of data between renegotiations

#password\_encryption = on

#db\_user\_namespace = off

# Kerberos and GSSAPI

#krb\_server\_keyfile = ''

#krb\_srvname = 'postgres' # (Kerberos only)

#krb\_caseins\_users = off

# - TCP Keepalives -

# see "man 7 tcp" for details

#tcp\_keepalives\_idle = 0 # TCP\_KEEPIDLE, in seconds;

# 0 selects the system default

#tcp\_keepalives\_interval = 0 # TCP\_KEEPINTVL, in seconds;

# 0 selects the system default

#tcp\_keepalives\_count = 0 # TCP\_KEEPCNT;

# 0 selects the system default

#------------------------------------------------------------------------------

# RESOURCE USAGE (except WAL)

#------------------------------------------------------------------------------

# - Memory -

shared\_buffers = 512MB # min 128kB

# (change requires restart)

temp\_buffers = 256MB # min 800kB

#max\_prepared\_transactions = 0 # zero disables the feature

# (change requires restart)

# Note: Increasing max\_prepared\_transactions costs ~600 bytes of shared memory

# per transaction slot, plus lock space (see max\_locks\_per\_transaction).

# It is not advisable to set max\_prepared\_transactions nonzero unless you

# actively intend to use prepared transactions.

work\_mem = 256MB # min 64kB

#maintenance\_work\_mem = 16MB # min 1MB

#max\_stack\_depth = 2MB # min 100kB

# - Kernel Resource Usage -

#max\_files\_per\_process = 1000 # min 25

# (change requires restart)

#shared\_preload\_libraries = '' # (change requires restart)

# - Cost-Based Vacuum Delay -

#vacuum\_cost\_delay = 0ms # 0-100 milliseconds

#vacuum\_cost\_page\_hit = 1 # 0-10000 credits

#vacuum\_cost\_page\_miss = 10 # 0-10000 credits

#vacuum\_cost\_page\_dirty = 20 # 0-10000 credits

#vacuum\_cost\_limit = 200 # 1-10000 credits

# - Background Writer -

#bgwriter\_delay = 200ms # 10-10000ms between rounds

#bgwriter\_lru\_maxpages = 100 # 0-1000 max buffers written/round

#bgwriter\_lru\_multiplier = 2.0 # 0-10.0 multipler on buffers scanned/round

# - Asynchronous Behavior -

#effective\_io\_concurrency = 1 # 1-1000. 0 disables prefetching

#------------------------------------------------------------------------------

# WRITE AHEAD LOG

#------------------------------------------------------------------------------

# - Settings -

#wal\_level = minimal # minimal, archive, or hot\_standby

# (change requires restart)

#fsync = on # turns forced synchronization on or off

#synchronous\_commit = on # synchronization level; on, off, or local

#wal\_sync\_method = fsync # the default is the first option

# supported by the operating system:

# open\_datasync

# fdatasync (default on Linux)

# fsync

# fsync\_writethrough

# open\_sync

#full\_page\_writes = on # recover from partial page writes

#wal\_buffers = -1 # min 32kB, -1 sets based on shared\_buffers

# (change requires restart)

#wal\_writer\_delay = 200ms # 1-10000 milliseconds

#commit\_delay = 0 # range 0-100000, in microseconds

#commit\_siblings = 5 # range 1-1000

# - Checkpoints -

#checkpoint\_segments = 3 # in logfile segments, min 1, 16MB each

#checkpoint\_timeout = 5min # range 30s-1h

#checkpoint\_completion\_target = 0.5 # checkpoint target duration, 0.0 - 1.0

#checkpoint\_warning = 30s # 0 disables

# - Archiving -

#archive\_mode = off # allows archiving to be done

# (change requires restart)

#archive\_command = '' # command to use to archive a logfile segment

#archive\_timeout = 0 # force a logfile segment switch after this

# number of seconds; 0 disables

#------------------------------------------------------------------------------

# REPLICATION

#------------------------------------------------------------------------------

# - Master Server -

# These settings are ignored on a standby server

#max\_wal\_senders = 0 # max number of walsender processes

# (change requires restart)

#wal\_sender\_delay = 1s # walsender cycle time, 1-10000 milliseconds

#wal\_keep\_segments = 0 # in logfile segments, 16MB each; 0 disables

#vacuum\_defer\_cleanup\_age = 0 # number of xacts by which cleanup is delayed

#replication\_timeout = 60s # in milliseconds; 0 disables

#synchronous\_standby\_names = '' # standby servers that provide sync rep

# comma-separated list of application\_name

# from standby(s); '\*' = all

# - Standby Servers -

# These settings are ignored on a master server

#hot\_standby = off # "on" allows queries during recovery

# (change requires restart)

#max\_standby\_archive\_delay = 30s # max delay before canceling queries

# when reading WAL from archive;

# -1 allows indefinite delay

#max\_standby\_streaming\_delay = 30s # max delay before canceling queries

# when reading streaming WAL;

# -1 allows indefinite delay

#wal\_receiver\_status\_interval = 10s # send replies at least this often

# 0 disables

#hot\_standby\_feedback = off # send info from standby to prevent

# query conflicts

#------------------------------------------------------------------------------

# QUERY TUNING

#------------------------------------------------------------------------------

# - Planner Method Configuration -

#enable\_bitmapscan = on

#enable\_hashagg = on

#enable\_hashjoin = on

#enable\_indexscan = on

#enable\_material = on

#enable\_mergejoin = on

#enable\_nestloop = on

#enable\_seqscan = on

#enable\_sort = on

#enable\_tidscan = on

# - Planner Cost Constants -

#seq\_page\_cost = 1.0 # measured on an arbitrary scale

#random\_page\_cost = 4.0 # same scale as above

#cpu\_tuple\_cost = 0.01 # same scale as above

#cpu\_index\_tuple\_cost = 0.005 # same scale as above

#cpu\_operator\_cost = 0.0025 # same scale as above

effective\_cache\_size = 1024MB

# - Genetic Query Optimizer -

#geqo = on

#geqo\_threshold = 12

#geqo\_effort = 5 # range 1-10

#geqo\_pool\_size = 0 # selects default based on effort

#geqo\_generations = 0 # selects default based on effort

#geqo\_selection\_bias = 2.0 # range 1.5-2.0

#geqo\_seed = 0.0 # range 0.0-1.0

# - Other Planner Options -

#default\_statistics\_target = 100 # range 1-10000

#constraint\_exclusion = partition # on, off, or partition

#cursor\_tuple\_fraction = 0.1 # range 0.0-1.0

#from\_collapse\_limit = 8

#join\_collapse\_limit = 8 # 1 disables collapsing of explicit

# JOIN clauses

#------------------------------------------------------------------------------

# ERROR REPORTING AND LOGGING

#------------------------------------------------------------------------------

# - Where to Log -

log\_destination = 'stderr, eventlog' # Valid values are combinations of

# stderr, csvlog, syslog, and eventlog,

# depending on platform. csvlog

# requires logging\_collector to be on.

# This is used when logging to stderr:

logging\_collector = on # Enable capturing of stderr and csvlog

# into log files. Required to be on for

# csvlogs.

# (change requires restart)

# These are only used if logging\_collector is on:

#log\_directory = 'pg\_log' # directory where log files are written,

# can be absolute or relative to PGDATA

#log\_filename = 'postgresql-%Y-%m-%d\_%H%M%S.log' # log file name pattern,

# can include strftime() escapes

#log\_file\_mode = 0600 # creation mode for log files,

# begin with 0 to use octal notation

#log\_truncate\_on\_rotation = off # If on, an existing log file with the

# same name as the new log file will be

# truncated rather than appended to.

# But such truncation only occurs on

# time-driven rotation, not on restarts

# or size-driven rotation. Default is

# off, meaning append to existing files

# in all cases.

#log\_rotation\_age = 1d # Automatic rotation of logfiles will

# happen after that time. 0 disables.

#log\_rotation\_size = 10MB # Automatic rotation of logfiles will

# happen after that much log output.

# 0 disables.

# These are relevant when logging to syslog:

#syslog\_facility = 'LOCAL0'

#syslog\_ident = 'postgres'

#silent\_mode = off # Run server silently.

# DO NOT USE without syslog or

# logging\_collector

# (change requires restart)

# - When to Log -

client\_min\_messages = 'log' # values in order of decreasing detail:

# debug5

# debug4

# debug3

# debug2

# debug1

# log

# notice

# warning

# error

log\_min\_messages = 'info' # values in order of decreasing detail:

# debug5

# debug4

# debug3

# debug2

# debug1

# info

# notice

# warning

# error

# log

# fatal

# panic

#log\_min\_error\_statement = error # values in order of decreasing detail:

# debug5

# debug4

# debug3

# debug2

# debug1

# info

# notice

# warning

# error

# log

# fatal

# panic (effectively off)

#log\_min\_duration\_statement = -1 # -1 is disabled, 0 logs all statements

# and their durations, > 0 logs only

# statements running at least this number

# of milliseconds

# - What to Log -

#debug\_print\_parse = off

#debug\_print\_rewritten = off

#debug\_print\_plan = off

#debug\_pretty\_print = on

log\_checkpoints = on

log\_connections = on

log\_disconnections = on

#log\_duration = off

log\_error\_verbosity = 'verbose' # terse, default, or verbose messages

log\_hostname = on

log\_line\_prefix = '%t ' # special values:

# %a = application name

# %u = user name

# %d = database name

# %r = remote host and port

# %h = remote host

# %p = process ID

# %t = timestamp without milliseconds

# %m = timestamp with milliseconds

# %i = command tag

# %e = SQL state

# %c = session ID

# %l = session line number

# %s = session start timestamp

# %v = virtual transaction ID

# %x = transaction ID (0 if none)

# %q = stop here in non-session

# processes

# %% = '%'

# e.g. '<%u%%%d> '

#log\_lock\_waits = off # log lock waits >= deadlock\_timeout

#log\_statement = 'none' # none, ddl, mod, all

log\_temp\_files = 1024 # log temporary files equal or larger

# than the specified size in kilobytes;

# -1 disables, 0 logs all temp files

#log\_timezone = '(defaults to server environment setting)'

#------------------------------------------------------------------------------

# RUNTIME STATISTICS

#------------------------------------------------------------------------------

# - Query/Index Statistics Collector -

#track\_activities = on

#track\_counts = on

#track\_functions = none # none, pl, all

#track\_activity\_query\_size = 1024 # (change requires restart)

#update\_process\_title = on

#stats\_temp\_directory = 'pg\_stat\_tmp'

# - Statistics Monitoring -

#log\_parser\_stats = off

#log\_planner\_stats = off

#log\_executor\_stats = off

#log\_statement\_stats = off

#------------------------------------------------------------------------------

# AUTOVACUUM PARAMETERS

#------------------------------------------------------------------------------

autovacuum = off # Enable autovacuum subprocess? 'on'

# requires track\_counts to also be on.

#log\_autovacuum\_min\_duration = -1 # -1 disables, 0 logs all actions and

# their durations, > 0 logs only

# actions running at least this number

# of milliseconds.

#autovacuum\_max\_workers = 3 # max number of autovacuum subprocesses

# (change requires restart)

#autovacuum\_naptime = 1min # time between autovacuum runs

#autovacuum\_vacuum\_threshold = 50 # min number of row updates before

# vacuum

#autovacuum\_analyze\_threshold = 50 # min number of row updates before

# analyze

#autovacuum\_vacuum\_scale\_factor = 0.2 # fraction of table size before vacuum

#autovacuum\_analyze\_scale\_factor = 0.1 # fraction of table size before analyze

#autovacuum\_freeze\_max\_age = 200000000 # maximum XID age before forced vacuum

# (change requires restart)

#autovacuum\_vacuum\_cost\_delay = 20ms # default vacuum cost delay for

# autovacuum, in milliseconds;

# -1 means use vacuum\_cost\_delay

#autovacuum\_vacuum\_cost\_limit = -1 # default vacuum cost limit for

# autovacuum, -1 means use

# vacuum\_cost\_limit

#------------------------------------------------------------------------------

# CLIENT CONNECTION DEFAULTS

#------------------------------------------------------------------------------

# - Statement Behavior -

#search\_path = '"$user",public' # schema names

#default\_tablespace = '' # a tablespace name, '' uses the default

#temp\_tablespaces = '' # a list of tablespace names, '' uses

# only default tablespace

#check\_function\_bodies = on

#default\_transaction\_isolation = 'read committed'

#default\_transaction\_read\_only = off

#default\_transaction\_deferrable = off

#session\_replication\_role = 'origin'

#statement\_timeout = 0 # in milliseconds, 0 is disabled

#vacuum\_freeze\_min\_age = 50000000

#vacuum\_freeze\_table\_age = 150000000

#bytea\_output = 'hex' # hex, escape

#xmlbinary = 'base64'

#xmloption = 'content'

# - Locale and Formatting -

datestyle = 'iso, dmy'

#intervalstyle = 'postgres'

#timezone = '(defaults to server environment setting)'

#timezone\_abbreviations = 'Default' # Select the set of available time zone

# abbreviations. Currently, there are

# Default

# Australia

# India

# You can create your own file in

# share/timezonesets/.

#extra\_float\_digits = 0 # min -15, max 3

#client\_encoding = sql\_ascii # actually, defaults to database

# encoding

# These settings are initialized by initdb, but they can be changed.

lc\_messages = 'English\_United Kingdom.1252' # locale for system error message

# strings

lc\_monetary = 'English\_United Kingdom.1252' # locale for monetary formatting

lc\_numeric = 'English\_United Kingdom.1252' # locale for number formatting

lc\_time = 'English\_United Kingdom.1252' # locale for time formatting

# default configuration for text search

default\_text\_search\_config = 'pg\_catalog.english'

# - Other Defaults -

#dynamic\_library\_path = '$libdir'

#local\_preload\_libraries = ''

#------------------------------------------------------------------------------

# LOCK MANAGEMENT

#------------------------------------------------------------------------------

#deadlock\_timeout = 1s

#max\_locks\_per\_transaction = 64 # min 10

# (change requires restart)

# Note: Each lock table slot uses ~270 bytes of shared memory, and there are

# max\_locks\_per\_transaction \* (max\_connections + max\_prepared\_transactions)

# lock table slots.

#max\_pred\_locks\_per\_transaction = 64 # min 10

# (change requires restart)

#------------------------------------------------------------------------------

# VERSION/PLATFORM COMPATIBILITY

#------------------------------------------------------------------------------

# - Previous PostgreSQL Versions -

#array\_nulls = on

#backslash\_quote = safe\_encoding # on, off, or safe\_encoding

#default\_with\_oids = off

#escape\_string\_warning = on

#lo\_compat\_privileges = off

#quote\_all\_identifiers = off

#sql\_inheritance = on

#standard\_conforming\_strings = on

#synchronize\_seqscans = on

# - Other Platforms and Clients -

#transform\_null\_equals = off

#------------------------------------------------------------------------------

# ERROR HANDLING

#------------------------------------------------------------------------------

#exit\_on\_error = off # terminate session on any error?

#restart\_after\_crash = on # reinitialize after backend crash?

#------------------------------------------------------------------------------

# CUSTOMIZED OPTIONS

#------------------------------------------------------------------------------

custom\_variable\_classes = 'rif40' # list of custom variable class names

#

rif40.debug = '' # functionA:levelA, functionB:levelB, ...

# level is DEBUG1...4

rif40.send\_debug\_to\_info = off # Send debug output to info (i.e. screen)

#

# Eof

# Appendix C – SAHSULand db\_create.sql script

--

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--

-- GIT Header

--

-- $Format:Git ID: (%h) %ci$

-- $Id$

-- Version hash: $Format:%H$

--

-- Description:

--

-- Rapid Enquiry Facility (RIF) - Comment Postgres tables/views

--

-- Copyright:

--

-- The Rapid Inquiry Facility (RIF) is an automated tool devised by SAHSU

-- that rapidly addresses epidemiological and public health questions using

-- routinely collected health and population data and generates standardised

-- rates and relative risks for any given health outcome, for specified age

-- and year ranges, for any given geographical area.

--

-- Copyright 2014 Imperial College London, developed by the Small Area

-- Health Statistics Unit. The work of the Small Area Health Statistics Unit

-- is funded by the Public Health England as part of the MRC-PHE Centre for

-- Environment and Health. Funding for this project has also been received

-- from the Centers for Disease Control and Prevention.

--

-- This file is part of the Rapid Inquiry Facility (RIF) project.

-- RIF is free software: you can redistribute it and/or modify

-- it under the terms of the GNU Lesser General Public License as published by

-- the Free Software Foundation, either version 3 of the License, or

-- (at your option) any later version.

--

-- RIF is distributed in the hope that it will be useful,

-- but WITHOUT ANY WARRANTY; without even the implied warranty of

-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

-- GNU Lesser General Public License for more details.

--

-- You should have received a copy of the GNU Lesser General Public License

-- along with RIF. If not, see <http://www.gnu.org/licenses/>; or write

-- to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor,

-- Boston, MA 02110-1301 USA

--

-- Author:

--

-- Peter Hambly, SAHSU

--

\set ECHO OFF

\set ON\_ERROR\_STOP ON

\o db\_create.rpt

BEGIN;

--

-- Check command line parameters

--

--\prompt 'Password: ' password

\set npassword '''XXXX':encrypted\_password''''

SET rif40.password TO :npassword;

DO LANGUAGE plpgsql $$

DECLARE

c1 CURSOR FOR SELECT CURRENT\_SETTING('rif40.password') AS password;

c1\_rec RECORD;

BEGIN

OPEN c1;

FETCH c1 INTO c1\_rec;

CLOSE c1;

--

IF UPPER(c1\_rec.password) = 'XXXX' THEN

RAISE EXCEPTION 'db\_create.sql() C209xx: No -v encrypted\_password=<encrypted password> parameter';

ELSE

RAISE INFO 'db\_create.sql() encrypted password="%"', SUBSTR(c1\_rec.password, 5);

END IF;

END;

$$;

SET rif40.password TO :encrypted\_password;

--

-- Check user is postgres on postgres

--

DO LANGUAGE plpgsql $$

BEGIN

IF user = 'postgres' AND current\_database() = 'postgres' THEN

RAISE INFO 'db\_create.sql() User check: %', user;

ELSE

RAISE EXCEPTION 'db\_create.sql() C209xx: User check failed: % is not postgres on postgres database (%)',

user, current\_database();

END IF;

END;

$$;

--

-- Check DB version

--

DO LANGUAGE plpgsql $$

DECLARE

c1 CURSOR FOR

SELECT version() AS version, SUBSTR(version(), 12, 3)::NUMERIC as major\_version;

c1\_rec RECORD;

--

BEGIN

OPEN c1;

FETCH c1 INTO c1\_rec;

CLOSE c1;

--

IF c1\_rec.major\_version < 9.3 THEN

RAISE EXCEPTION 'db\_create.sql() C902xx: RIF requires Postgres version 9.3 or higher; current version: %',

c1\_rec.version::VARCHAR;

ELSE

RAISE INFO 'db\_create.sql() RIF required Postgres version 9.3 or higher OK; current version: %',

c1\_rec.version::VARCHAR;

END IF;

END;

$$;

--

-- Check availability of extensions

--

DO LANGUAGE plpgsql $$

DECLARE

c1 CURSOR(l\_name VARCHAR) FOR

SELECT name, default\_version

FROM pg\_available\_extensions

WHERE name = l\_name;

c1\_rec RECORD;

--

namelist VARCHAR[]:=ARRAY['adminpack', 'plperl', 'postgis', 'postgis\_topology', 'pgcrypto', 'sslinfo', 'xml2', 'dblink', 'plr'];

x VARCHAR;

i INTEGER:=0;

BEGIN

FOREACH x IN ARRAY namelist LOOP

OPEN c1(x);

FETCH c1 INTO c1\_rec;

IF c1\_rec.name IS NULL THEN

RAISE WARNING 'db\_create.sql() RIF required extension: % is not installable', x;

i:=i+1;

ELSE

RAISE INFO 'db\_create.sql() RIF required extension: % V% is installable', c1\_rec.name, c1\_rec.default\_version;

END IF;

CLOSE c1;

END LOOP;

IF i > 0 THEN

RAISE EXCEPTION 'db\_create.sql() C209xx: % RIF required extensions are not installable',

i::VARCHAR USING HINT='See previous warnings';

END IF;

END;

$$;

\echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\echo \*

\echo \* WARNING !!!

\echo \*

\echo \* This script will drop sahsuland, re-create it and reset the passwords to the

\echo \* default

\echo \*

\echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\prompt 'Press any key to continue or <control-C> to interrupt: ' ans

CREATE EXTENSION IF NOT EXISTS plr;

DROP EXTENSION plr;

--

-- This will fail if:

--

-- a) PLR (http://www.joeconway.com/plr) is not installed as per instructions

--

/\*

In the instructions below:

PostgreSQL is installed to <pgdir>

R is installed to <rdir>

For example, these directories might be:

<pgdir> = C:\PostgreSQL\9.3

<rdir> = C:\R\R-3.0.2

I recommend you ensure there are no spaces in the pathname to either <pgdir> or <rdir>.

Correct quoting of pathnames with spaces is left as an exercise for the reader ;-)

The following files are contained in this zip file:

---------------------------------------------------

README.txt: place in <pgdir>\doc\extension

plr.dll: place in <pgdir>\lib

plr.sql: place in <pgdir>\share\extension

plr.control: place in <pgdir>\share\extension

plr--8.3.0.15.sql place in <pgdir>\share\extension

plr--unpackaged--8.3.0.15.sql place in <pgdir>\share\extension

Ensure the following environment variables are set \*prior\* to starting PostgreSQL:

---------------------------------------------------

PATH=<pgdir>\bin;<rdir>\bin\x64;$PATH

R\_HOME=<rdir

\*/

--

-- See also: http://www.bostongis.com/PrinterFriendly.aspx?content\_name=postgresql\_plr\_tut01

-- http://www.joeconway.com/plr/doc/plr-install.html

--

-- b) Postgres is NOT restarted

--

\echo Creating users/roles...

--

-- Create users and roles

--

DO LANGUAGE plpgsql $$

DECLARE

c1 CURSOR(l\_name VARCHAR) FOR

SELECT \* FROM pg\_user WHERE usename = l\_name;

c2 CURSOR FOR

SELECT CURRENT\_SETTING('rif40.password') AS password;

c3 CURSOR(l\_name VARCHAR) FOR

SELECT \* FROM pg\_roles WHERE rolname = l\_name;

c1\_rec RECORD;

c2\_rec RECORD;

c3\_rec RECORD;

--

userlist VARCHAR[]:=ARRAY['rif40', 'gis', 'pop', 'peterh'];

rolelist VARCHAR[]:=ARRAY['rif\_user', 'rif\_manager', 'rif\_no\_suppression', 'rifupg34'];

x VARCHAR;

sql\_stmt VARCHAR;

BEGIN

OPEN c2;

FETCH c2 INTO c2\_rec;

CLOSE c2;

sql\_stmt:='ALTER USER postgres ENCRYPTED PASSWORD '''||c2\_rec.password||'''';

RAISE INFO 'SQL> %;', sql\_stmt::VARCHAR;

PERFORM sql\_stmt;

--

FOREACH x IN ARRAY userlist LOOP

OPEN c1(x);

FETCH c1 INTO c1\_rec;

CLOSE c1;

IF c1\_rec.usename IS NOT NULL THEN

RAISE INFO 'db\_create.sql() RIF schema user % exists', c1\_rec.usename::VARCHAR;

ELSE

sql\_stmt:='CREATE ROLE '||x||

' NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN NOREPLICATION ENCRYPTED PASSWORD '''||c2\_rec.password||'''';

RAISE INFO 'SQL> %;', sql\_stmt::VARCHAR;

EXECUTE sql\_stmt;

END IF;

END LOOP;

--

FOREACH x IN ARRAY rolelist LOOP

OPEN c3(x);

FETCH c3 INTO c3\_rec;

CLOSE c3;

IF c3\_rec.rolname IS NOT NULL THEN

RAISE INFO 'db\_create.sql() RIF schema role % exists', c3\_rec.rolname::VARCHAR;

ELSE

sql\_stmt:='CREATE ROLE '||x||

' NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT NOLOGIN NOREPLICATION';

RAISE INFO 'SQL> %;', sql\_stmt::VARCHAR;

EXECUTE sql\_stmt;

END IF;

END LOOP;

--

-- Revoke PUBLIC

--

sql\_stmt:='REVOKE CREATE ON SCHEMA public FROM PUBLIC';

RAISE INFO 'SQL> %;', sql\_stmt::VARCHAR;

EXECUTE sql\_stmt;

END;

$$;

END;

\echo Creating SAHSULAND...

--

-- Create SAHSULAND database (THIS DOES NOT WORK in PL/pgsql)

--

DO LANGUAGE plpgsql $$

DECLARE

c1 CURSOR(l\_name VARCHAR) FOR

SELECT \* FROM pg\_tablespace WHERE spcname = l\_name;

c1\_rec RECORD;

--

sql\_stmt VARCHAR;

BEGIN

OPEN c1('sahsuland');

FETCH c1 INTO c1\_rec;

CLOSE c1;

--

IF c1\_rec.spcname IS NULL THEN

sql\_stmt:='CREATE TABLESPACE sahsuland LOCATION ''C:\\PostgresDB\\sahsuland''';

RAISE INFO 'SQL> %;', sql\_stmt::VARCHAR;

EXECUTE sql\_stmt;

END IF;

--

END;

$$;

\set echo ALL

DROP DATABASE IF EXISTS sahsuland;

CREATE DATABASE sahsuland WITH OWNER rif40 TABLESPACE sahsuland;

COMMENT ON DATABASE sahsuland IS 'RIF V4.0 PostGres SAHSULAND Example Database';

\c sahsuland postgres localhost

--

-- Check user is postgres on sahsuland

--

\set ECHO OFF

DO LANGUAGE plpgsql $$

BEGIN

IF user = 'postgres' AND current\_database() = 'sahsuland' THEN

RAISE INFO 'db\_create.sql() User check: %', user;

ELSE

RAISE EXCEPTION 'db\_create.sql() C209xx: User check failed: % is not postgres on postgres database (%)',

user, current\_database();

END IF;

END;

$$;

\set echo ALL

CREATE EXTENSION IF NOT EXISTS plr;

RESET rif40.password;

\echo Restoring database...

--

-- Shell pg\_restore

--

--\! pg\_restore -d sahsuland -U postgres -v sahsuland.dump > pg\_restore.txt 2>&1

--

-- Powershell version

--

\! powershell -command "pg\_restore -d sahsuland -U postgres -v sahsuland.dump 2>&1 | tee ('pg\_restore{0}.log' -f (Get-Date -format 'yyyy.MM.dd-HH.mm'))"

--

-- Display errors

--

\echo Database restore errors...

\! findstr "ERROR:" pg\_restore.txt

/\*

Errors like this can be ignored - others may not!

pg\_restore: [archiver (db)] could not execute query: ERROR: could not load library "C:/Program Files/PostgreSQL/9.3/lib/plperl.dll"

: The specified module could not be found.

pg\_restore: [archiver (db)] could not execute query: ERROR: extension "plperl" does not exist

pg\_restore: [archiver (db)] could not execute query: ERROR: could not open extension control file "C:/Program Files/PostgreSQL/9.3/

share/extension/oracle\_fdw.control": No such file or directory

pg\_restore: [archiver (db)] could not execute query: ERROR: extension "oracle\_fdw" does not exist

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_delete" for relation "geometry\_columns" already

exists

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_insert" for relation "geometry\_columns" already

exists

pg\_restore: [archiver (db)] could not execute query: ERROR: rule "geometry\_columns\_update" for relation "geometry\_columns" already

exists

\*/

ALTER DATABASE sahsuland SET search\_path TO rif40,public,topology,gis,pop,rif40\_sql\_pkg;

\c sahsuland

\set VERBOSITY terse

DO LANGUAGE plpgsql $$

DECLARE

sql\_stmt VARCHAR;

BEGIN

--

-- Force rebuild of user objects

--

sql\_stmt:='DROP VIEW IF EXISTS '||USER||'.rif40\_user\_version';

PERFORM rif40\_sql\_pkg.rif40\_ddl(sql\_stmt);

--

PERFORM rif40\_log\_pkg.rif40\_add\_to\_debug('rif40\_ddl\_checks:DEBUG3');

PERFORM rif40\_log\_pkg.rif40\_log\_setup();

PERFORM rif40\_log\_pkg.rif40\_send\_debug\_to\_info(TRUE);

PERFORM rif40\_sql\_pkg.rif40\_startup();

IF USER != 'rif40' THEN

PERFORM rif40\_sql\_pkg.rif40\_ddl\_checks();

END IF;

PERFORM rif40\_log\_pkg.rif40\_remove\_from\_debug('rif40\_ddl\_checks');

END;

$$;

\echo SAHSUland installed.

--

-- Eof