

## Table of Contents

Table of Contents	1
PGSQL-RUNNER	2
1. WHAT IS THIS	2
2. INSTALLATION AND CONFIGURATION	2
2.1. Prerequisites	2
2.2. Fetch the source	2
2.3. Build the first pgsql-runner instance	2
2.4. Create you local conf file	3
2.5. Start hacking	3

# PGSQL-RUNNER

## 1. WHAT IS THIS

A bash script tool to run your postgres sql script files and log it's actions nicely ... As well as :  
A generic swiss knife wanna be bash / perl centric application pgsql-runner for quicky packaging and deploying your tools, create new tools out of your existing ones , generating code for additonal functions, search and replace in both file paths and contents ... and all the rest not mentioned actions in the sfw/bash/pgsql-runner/tests/all-pgsql-runner-tests.lst file ...

## 2. INSTALLATION AND CONFIGURATION

The pgsql-runner instances and clones have been running on \*Nix boxes with bash > 3.0 ... , but some actions require 4.0 and older ...  
Cygwin has been tested also ...

### 2.1. Prerequisites

The must have binaries are:

bash, perl, zip

The nice to have are:

tmux, vim ,ctags

The examples are for Ubuntu - use you OS package manager ...

```
apt-get autoclean
apt-get install --only-upgrade bash

sudo apt-get install -y perl

# optionally
sudo apt-get install -y excuberant-ctags
sudo apt-get install -y 7z

apt-get upgrade
```

### 2.2. Fetch the source

Fetch the source from git hub as follows:

```
# create your product dir:
mkdir -p /opt/csited/pgsql-runner
cd /opt/csited/pgsql-runner/

# fetch the source
git clone git@github.com:YordanGeorgiev/pgsql-runner.git
```

### 2.3. Build the first pgsql-runner instance

When doing for first time do exactly as shown bellow, otherwise no joy ...

Each pgsql-runner instance has it's own version, environmnt type and owner. For now just follow the instruction - after half an hour you will be hacking this ...

```
# build your product version dir - a kind of "this instance of the thingy dir"
mv -v /opt/csitedata/pgsql-runner/pgsql-runner /opt/csitedata/pgsql-runner/pgsql-runner.1.1.5.dev.$USER
```

## 2.4. Create you local conf file

The default conf file provides only limited functionality ( this is by design ) , thus copy and configure the configuration file for your host

```
# go to the product version dir
cd /opt/csitedata/pgsql-runner/pgsql-runner /opt/csitedata/pgsql-runner/pgsql-runner.1.1.5.dev.$USER

mv -v sfw/bash/pgsql-runner/pgsql-runner.set-your-host.conf sfw/bash/pgsql-runner/pgsql-runner.`hostname -s`.conf
```

## 2.5. Start hacking

Start hacking ... or wait check at least the test call running all the functions of the tool ...

```
# optionally if you are in the vim camp open the "project relative files list file"
vim meta/.dev.pgsql-runner

# Ctrl + Z ,
bash sfw/bash/pgsql-runner/test-pgsql-runner.sh

# now clone your own instance
bash sfw/bash/pgsql-runner/pgsql-runner.sh -a to-app=my-tool
```