

# 调试 PostgreSQL 源码

张文升

<http://ode.cnblogs.com>

Email:wensheng.zhang@foxmail.com

本文使用 Xming、Putty 和 VMWare 几款工具，在 Windows 桌面环境调用 Ubuntu 虚拟机上的图形界面进行 PostgreSQL 源码的调试。文中的所有步骤适合真机 Ubuntu 系统，只是所有图形界面的调用是在 Windows 桌面环境下。这样做，和直接在 Ubuntu 虚拟机中进行调试，并没有什么区别，只是我个人比较喜欢 Windows 下的一些工具的界面，如 VISIO2013，Office2013，比起 Libreoffice 还是要方便美观很多。

## 一、配置 Eclipse 和 JAVA 环境变量

1、下载 jdk 并解压，之后配置系统环境变量

```
export JAVA_HOME=/opt/jdk1.7.0_60
export JRE_HOME=$JAVA_HOME/jre
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```

2、下载 Eclipse 并解压

在尝试运行 eclipse 时，当遇到 “No Java virtual machine” 的错误时，首先确定 JAVA 环境变量是否正确配置并使之生效了，如果问题依旧，可以通过在 Eclipse 目录中建立一个 jre 的软连接到 jdk 的 jre 目录即可。

```
ln -s /opt/jdk1.7.0_60/jre /opt/eclipse/jre
```

在 Putty 中配置 X11，配置完成后尝试运行 Eclipse，当遇到 “cannot open display” 类似的错误时，回到虚拟机的图形界面，执行 `sudo xhost +`，问题应该就可以解决了。在 Windows 中看到 Eclipse 界面，但是 Eclipse 的所有字体都非常小，可以通过在 xming 快捷方式的启动参数里加上一个参数：`-dpi 108`。windows 默认的 dpi 是 96 像素，建议

设置为 96 的 125%或 150%。

如：

```
"C:\Program Files (x86)\Xming\Xming.exe" :0 -clipboard -multiwindow -dpi 108
```

## 二、配置 PostgreSQL 源码

1、安装 PostgreSQL 源码调试所需要的包

```
sudo apt-get install make gcc g++ gdb tar libreadline6-dev zlib1g-dev bison flex
```

```
sudo apt-get install openssl libssl-dev libpam-dev libxml2 libxml2-dev libxslt-dev tcl tcl-dev libperl-dev python python-dev systemtap systemtap-sdt-dev
```

如果要调试 PostgreSQL 源码，需要安装 gdb：sudo apt-get install gdb。

2、添加 PostgreSQL 的系统用户

```
sudo adduser postgres
```

3、在 PostgreSQL 源码目录中执行 configure

如果只打开--enable-debug 选项，在 gdb 里进行调试时，会有一些变量的值无法打印出

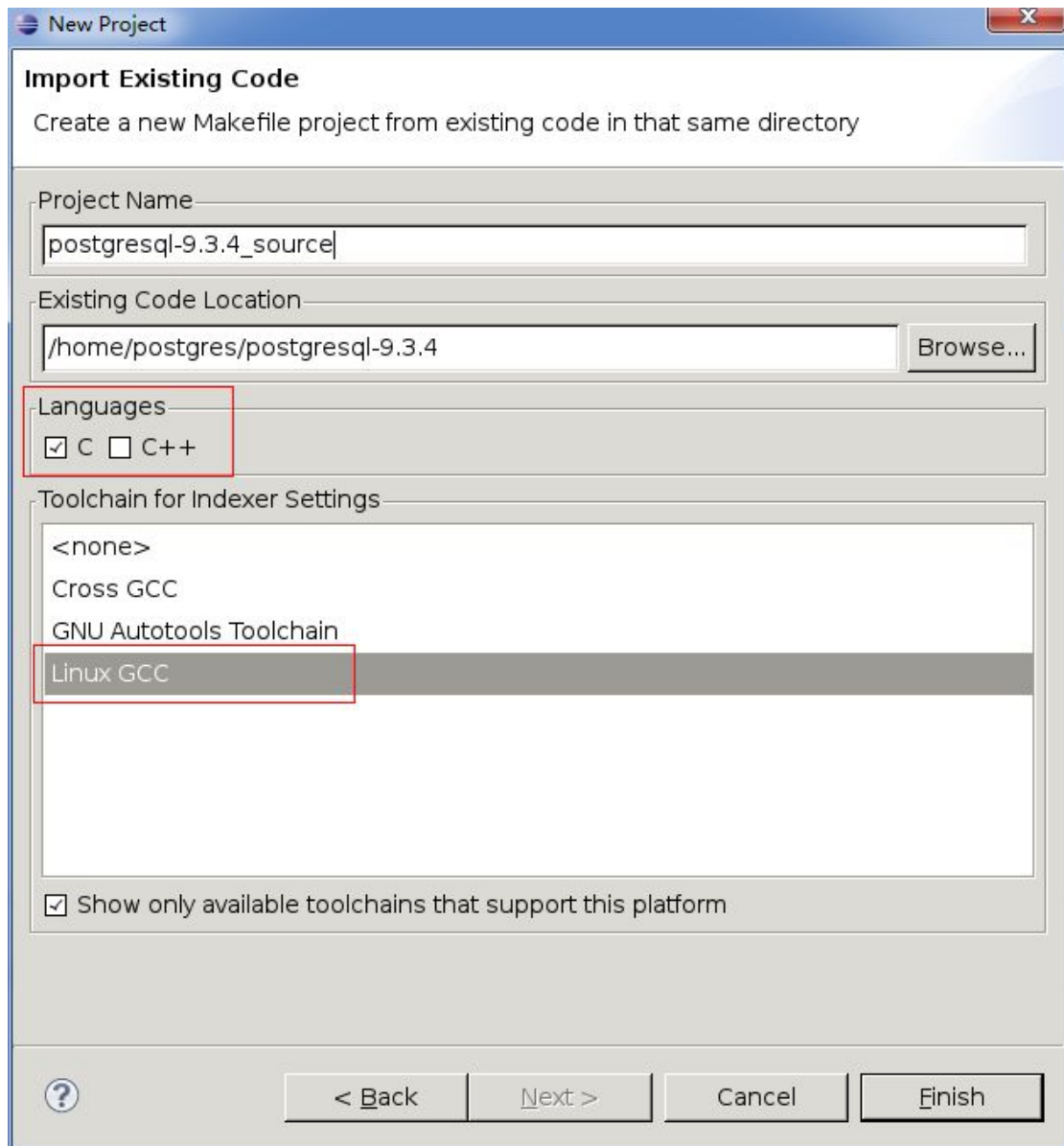
来，这是因为 gcc 在进行编译代码时，会进行一定程度的优化。为了能够在调试跟踪时看

到所有变量的值的变化情况，我们有必要降低甚至取消优化选项：CFLAGS=-o0

```
--enable-depend --enable-cassert --enable-debug  
./configure CFLAGS=-O0 --prefix=/opt/PostgreSQL/9.3.4/ --with-perl  
--with-python --with-tcl --with-openssl --with-pam --without-ldap --with-libxml  
--with-libxslt --enable-thread-safety --enable-dtrace --enable-depend  
--enable-cassert --enable-debug
```

## 三、在 Eclipse 中建立 PostgreSQL 源码工程

1、新建一个 Project 并导入 PostgreSQL 源码



Language: 选择 C, 必须去掉"C++"复选框

Toolchain for Indexer Settings: 选择 Linux GCC

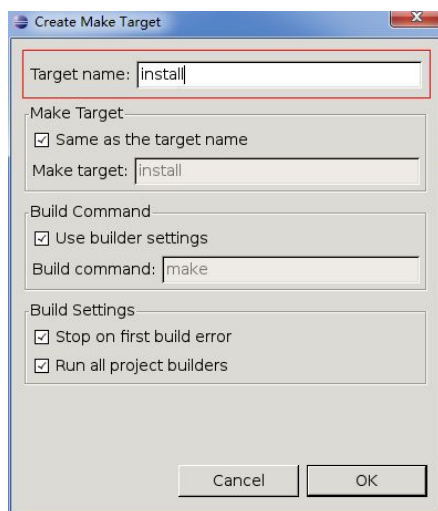
单击"Finish"按钮。

## 2、设置 Eclipse 的 Make Target

在调试 PostgreSQL 之前, 首先需要将其安装到某个目录中, 在 Eclipse 中是通过建立一个 Make Target 实现的。

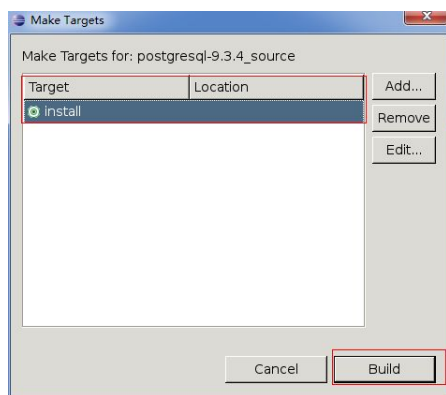
在工程单击右键, 选择 Make Targets-Create, 在弹出的"Create Make Target"对话框

中，Target name 文本控件的值输入：install，实际上它要做的就是我们通过源码编译安装时执行 make install 命令，单击 OK 按钮完成。



### 3、编译安装 PostgreSQL 源码到目标目录

在 PostgreSQL 源码工程上单击右键，选择 Make Targets-Build，选择 install，单击 Build。



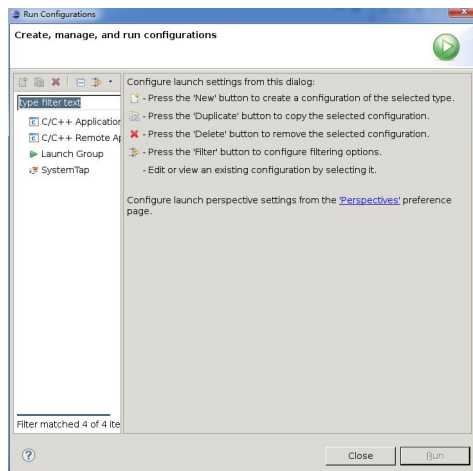
直到 console 窗口出现 "PostgreSQL installation complete."，表示安装完成。

此时，在 prefix 参数指定的安装目录下，会生成 bin、include、lib 和 share 目录。

观察 console 窗口：

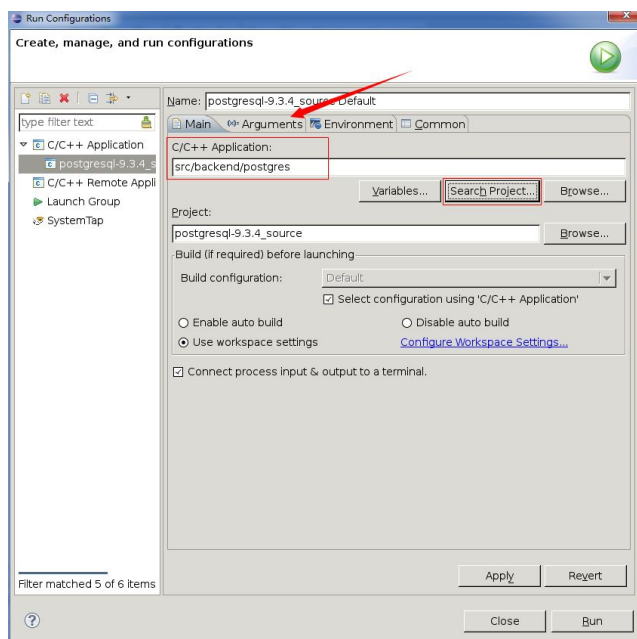


## Configurations

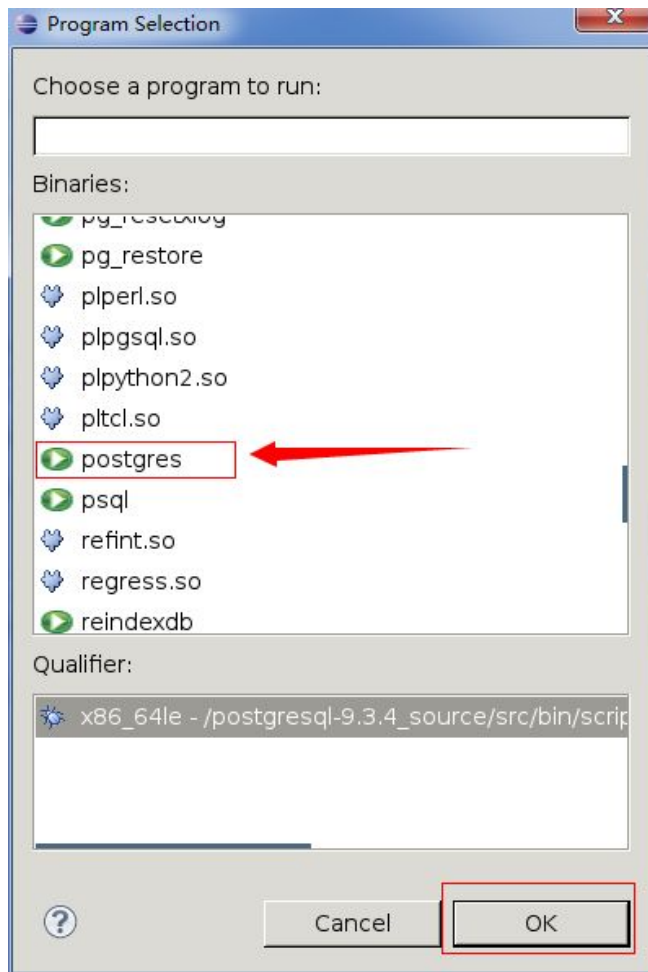


双击 C/C++ Application, 会创建一个名为 postgresql-source-9.3.4 Default 的配置界

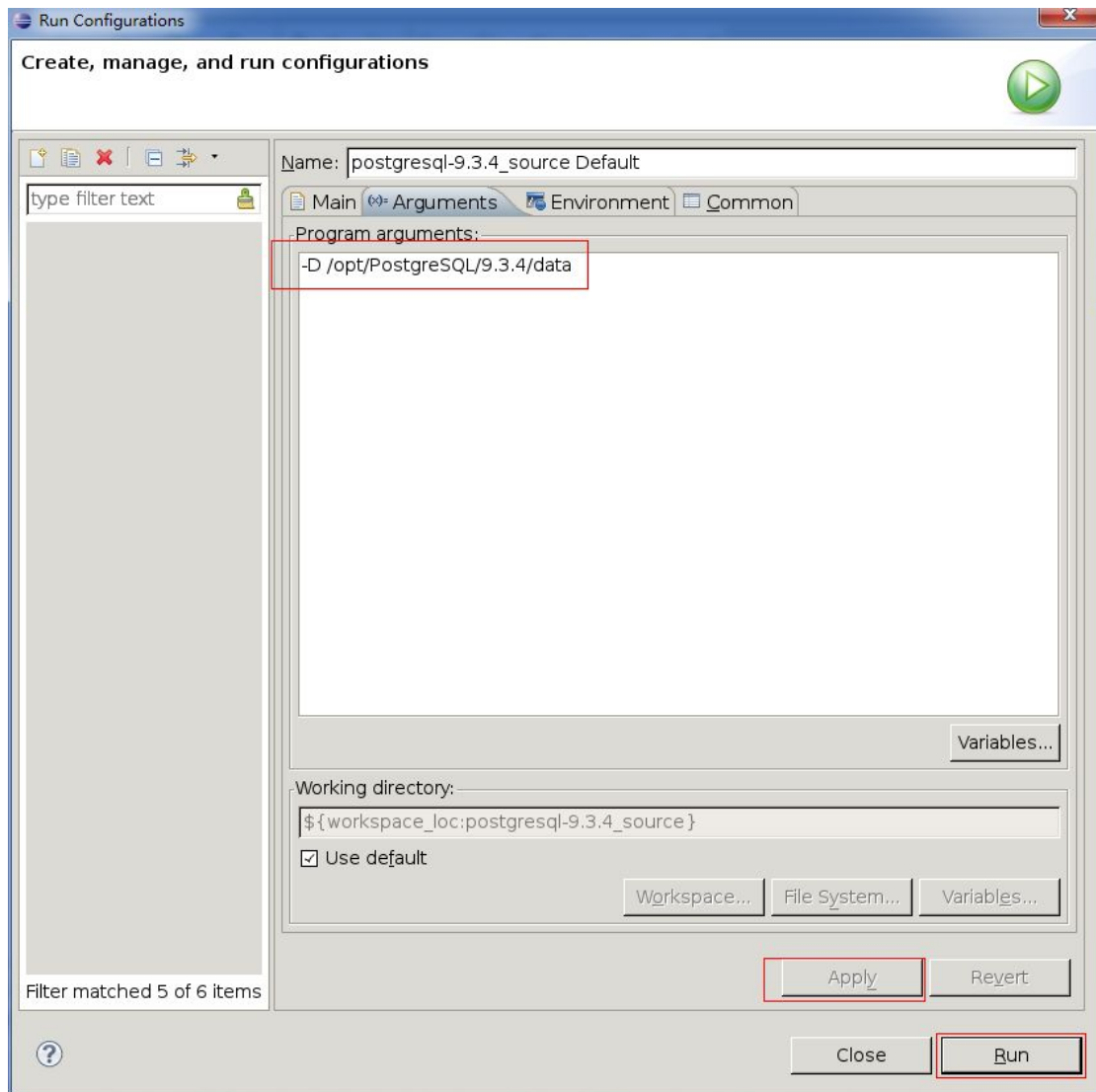
面, 在 C/C++ Application 文本控件中输入 src/backend/postgres



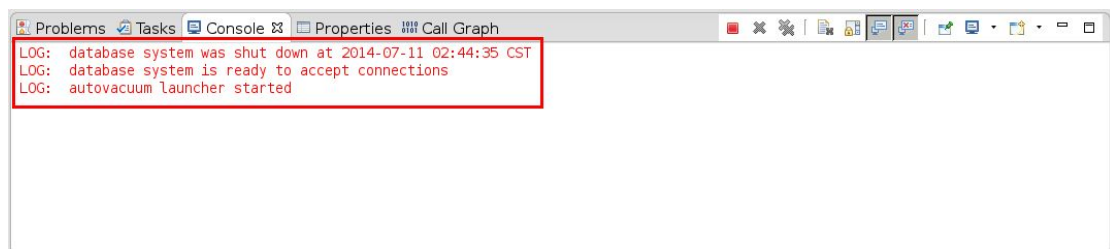
或者点击 Search Project...按钮, 选择 postgres



两种方法都可以。完成后，切换到上图中红色箭头所指的 Arguments 选项卡，在 Program arguments 文本控件中输入启动实例参数 -D 和数据库的数据目录所在：-D /opt/PostgreSQL/9.3.4/data，如下图所示：



单击 Apply，然后单击 Run，上图中因为已经点击过 Apply 按钮，所以 Apply 按钮是禁用状态。如果这一步遇到 Run 按钮是禁用状态，那可能你使用的不是 postgres 用户运行 Eclipse，可以切换到 postgres 用户试试看。



LOG: database system was shut down at 2014-07-11 02:44:35 CST  
LOG: database system is ready to accept connections  
LOG: autovacuum launcher started

看到这些信息，表示数据库已成功启动。



查看 postgres 进程和相关后台进程: `ps -ef | grep postgres`

```
postgres@devpc:~$ ps -ef | grep postgres
postgres 2708 2622 0 02:19 pts/1 00:00:00 su - postgres
postgres 2716 2708 0 02:19 pts/1 00:00:00 -su
postgres 2924 1 0 02:22 pts/1 00:00:00 dbus-launch --autolaunch=b60e99248f8a07c588a96c90000007 --binary-syntax --close-stderr
postgres 2923 1 0 02:22 ? 00:00:00 //bin/dbus-daemon --fork --print-pid 5 --print-address 7 --session
root 2934 660 0 02:24 ? 00:00:00 sshd: postgres [priv]
postgres 3099 2954 1 02:24 ? 00:00:30 sshd: postgres@pts/2
postgres 3100 3099 0 02:24 pts/2 00:00:00 -bash
postgres 3370 1886 0 02:32 pts/0 00:00:00 su - postgres
postgres 3377 3370 0 02:32 pts/0 00:00:00 -su
postgres 14695 3100 0 02:54 pts/2 00:00:00 ./eclipse
postgres 14696 14695 75 02:54 pts/2 00:03:00 /opt/eclipse/jre/bin/java -Dosgi.requiredJavaVersion=1.6 -XX:MaxPermSize=256m -Xms40m -Xmx512m -jar /opt/eclipse/plugins/org.eclipse.equinox.launcher_1.3.0.v20130327-1440.jar -os linux -ws gtk -arch x86_64 -showsplash /opt/eclipse/plugins/org.eclipse.platform_4.3.2.v20140221-1700/splash.bmp -i launcher /opt/eclipse/eclipse -name Eclipse --launcher.library /opt/eclipse/plugins/org.eclipse.equinox.launcher.gtk.linux.x86_64_1.1.200.v20140116-2212/eclipse_1508.so -st arcup /opt/eclipse/plugins/org.eclipse.equinox.launcher_1.3.0.v20130327-1440.jar --launcher.appendVMargs -exitdata d8003 -product org.eclipse.epp.package.cpp.product -vm /opt/eclipse/jre/bin/java -vmargs -Dosgi.requiredJavaVersion=1.6 -XX:MaxPermSize=256m -Xms40m -Xmx512m -jar /opt/eclipse/plugins/org.eclipse.equinox.launcher_1.3.0.v20130327-1440.jar
postgres 14972 14696 0 02:56 pts/2 00:00:00 /home/postgres/postgresq1-9.3.4/src/backend/postgres -D /opt/PostgreSQL/9.3.4/data
postgres 14978 14972 0 02:56 ? 00:00:00 postgres: checkpoint process
postgres 14979 14972 0 02:56 ? 00:00:00 postgres: writer process
postgres 14980 14972 0 02:56 ? 00:00:00 postgres: wal writer process
postgres 14981 14972 0 02:56 ? 00:00:00 postgres: autovacuum launcher process
postgres 14982 14972 0 02:56 ? 00:00:00 postgres: stats collector process
postgres 14983 3377 0 02:58 pts/0 00:00:00 ps -ef
postgres 14984 3377 0 02:58 pts/0 00:00:00 grep --color=auto postgres
postgres@devpc:~$
```

至此, 数据库已经安装, 启动成功, 可以通过 psql 连接到数据库进行测试。

`postgres@devpc:~$ /opt/PostgreSQL/9.3.4/bin/psql -U postgres postgres`

点击 Console 面板右侧的红色 Terminate 图标, 看到以下输出即可停止数据库服务了。

```
LOG: received smart shutdown request
LOG: autovacuum launcher shutting down
```

## 五、调试 PostgreSQL 代码

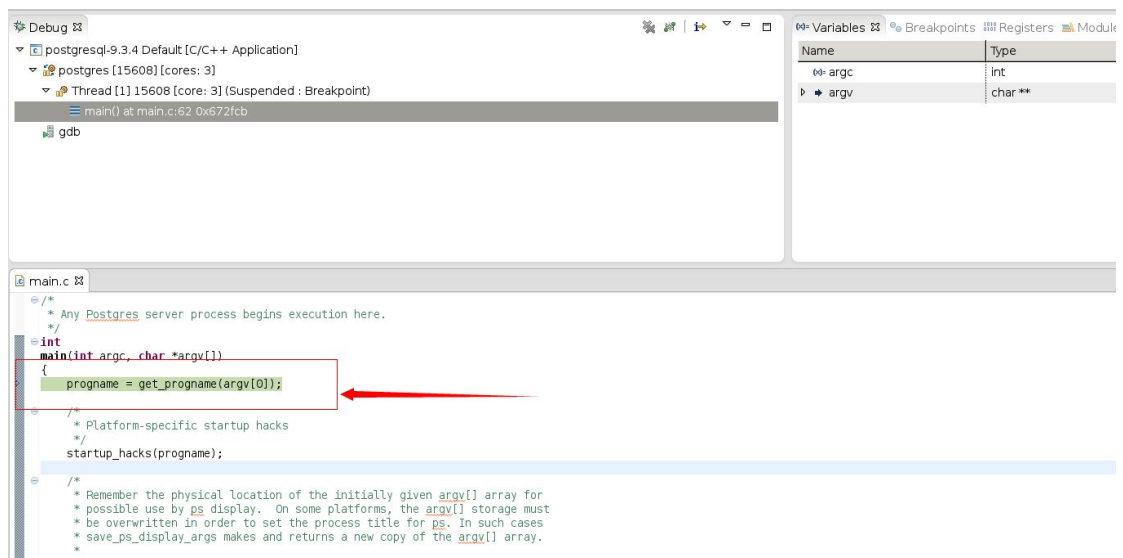
### 1、调试主进程

数据库运行成功后, 一定要关闭已经开启的 Postmaster 进程才能进行调试, 否则会提示

"Postmaster 已经存在"的错误。

单击工程右键, Debug As-Local C/C++ Application, 选择 "postgres" 作为调试程序, 单

击 OK, 即可进入调试模式。



此时程序自动会在 main 函数处停下,所以要继续点击工具栏上的"Resume"按钮(Eclipse 默认的快捷键 F8 ) 让主进程运行完,否则无法连接数据库。Debug 运行完后会有和运行成功时一样的提示信息。

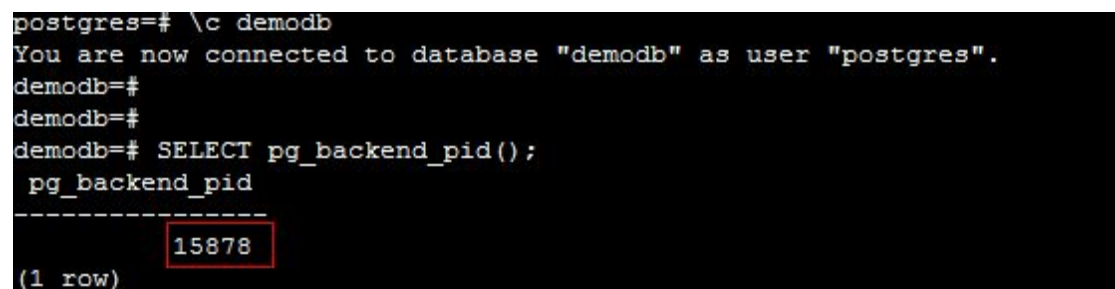
这时的 console 窗口会有以下输出:

```
LOG: database system was interrupted; last known up at 2014-07-12 14:37:56 CST
LOG: database system was not properly shut down; automatic recovery in progress
LOG: record with zero length at 0/178A4A0
LOG: redo is not required
LOG: autovacuum launcher started
LOG: database system is ready to accept connections
```

## 2、服务进程(postgres 子进程)的调试

使用 psql 连接数据库, 查看当前的进程号: `select pg_backend_pid();`

注意: 记住这个进程号, 如 15878。



```
postgres=# \c demodb
You are now connected to database "demodb" as user "postgres".
demodb=#
demodb=#
demodb=# SELECT pg_backend_pid();
 pg_backend_pid
-----
              15878
(1 row)
```

打开另一个终端, 输入 `ps -ef | grep sql`

发现此时, 多了一个进程:

```
lq      20579 20558  0 22:13 ?          00:00:00 postgres: lq DemoDB [local]
idle
```

回到 Eclipse 调试界面, 单击菜单 Run->Debug Configure, 在弹出的 Debug Configurations 界面, 双击 C/C++ Attach to Application:

配置界面:

Name: postgres

Main 选项卡:

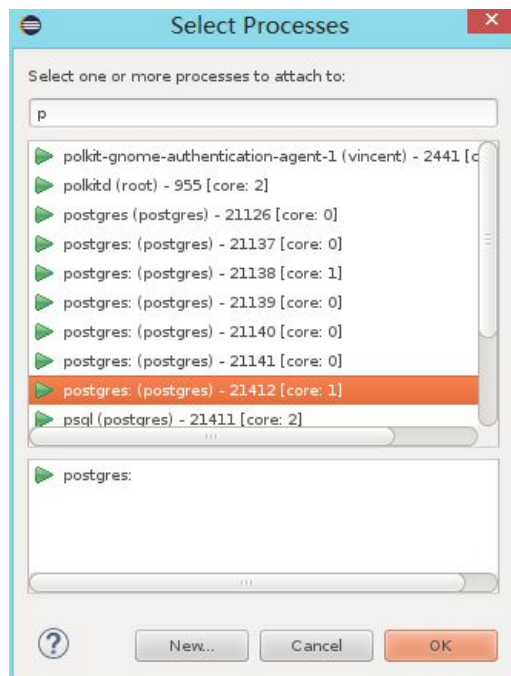
C/C++ Application: src/backend/postgres

Project: pgsq1

Build configuration: 取消"Select configuration...",选择"Use Active"

单击 Debug, 然后会出现一个进程选择窗口, 输入"post"在程序的进程列表中, 选择进程号(21706)对应的 postgres 进程。

单击 OK。



在调试子进程的过程中遇到以下错误:

Could not attach to process. If your uid matches the uid of the target process, check the setting of /proc/sys/kernel/yama/ptrace\_scope, or try again as the root user. For more details, see /etc/sysctl.d/10-ptrace.conf

解决方法:

1.可以通过下面的命令对这个限制进行临时修改, 但是重启后限制依旧存在。

```
echo 0 | sudo tee /proc/sys/kernel/yama/ptrace_scope
```

2.可以通过修改配置文件来解除限制(我选择了这种方式)

```
sudo vim /etc/sysctl.d/10-ptrace.conf
```

修改前:

```
kernel.yama.ptrace_scope = 1
```

修改后:

```
kernel.yama.ptrace_scope = 0
```

重启之后可以生效。

修改之后重复上述的步骤既可看到已经将调试工具挂上了这个 postgres 进程。

## 六、 PostgreSQL WIKI 推荐的 5 本好书:

- 1、 An Introduction to Database Systems, by C.J. Date, Addison, Wesley
- 2、 A Guide to the SQL Standard, by C.J. Date, et. al, Addison, Wesley
- 3、 Fundamentals of Database Systems, by Elmasri and Navathe
- 4、 Transaction Processing, by Jim Gray and Andreas Reuter, Morgan Kaufmann
- 5、 Transactional Information Systems, by Gerhard Weikum and Gottfried Vossen, Morgan Kaufmann

## 七、 参考

[https://wiki.postgresql.org/wiki/Working\\_with\\_Eclipse](https://wiki.postgresql.org/wiki/Working_with_Eclipse)

<http://my.oschina.net/tashi/blog/222887>

<http://blog.sciencenet.cn/blog-419883-306975.html>