

Instruction for installing CADLIVE Simulator

A table of contents

1. Abstract	2
2. Installing Redhat7.1	2
3. Stopping unnecessary services	2
4. Uninstalling old version.....	2
5. Installing apache.....	2
6. Installing PostgreSQL.....	3
6.1 Creating group and management user	3
6.2 Creating the directory for postgresQL and installation.....	3
6.3 Environment setting for postgresQL	4
6.4 Initialization and setting database	4
6.5 Setting automatic start	5
6.6 Registration for users.....	5
7. Installing php	5
7.1 Installing php.....	5
7.2 Setting change of apache	6
7.3 Function confirmation.....	6
7.4 Installing phplib	6
8. Installing Java2 SDK	7
9. Installing Xerces2 Java Parser	7
10. Installing mpich.....	8
11. Installing CADLIVE Simulator	8
11.1 Installing CADLIVE Simulator.....	8
11.2 Setting apache	8
11.3 Installing batch job	8
11.4 Setting php.....	9
11.5 Setting database	9
11.6 Others	9

1. Abstract

Install Redhat7.1.
Stop the services that are not necessary for the simulator.
Uninstall old versions of apache, postgresSQL, and php.
Install new versions of apache, postgresSQL, php, and phplib.
Install Java2SDK, Xerces2, and Jaka Parser.
Install mpich.

The commands preceded by # are carried out on the root account, and those indicated by \$ are done on another account.

2. Installing Redhat7.1

Full install Redhat 7.1.

3. Stopping unnecessary services

Login at root, and select the startup services at setup.
Stop apmd, atd, isdn, pcmcia, and sendmail that startup as defaults.
Add ntpd.

In order to adjust the time of all the master and slaves, in the file,/etc/ntp.conf, read it as follows,
server = "IP_address of host"

4. Uninstalling old version

Uninstall apache, postgresSQL, and php, if they have been already installed and remain in the system.

```
#rpm -e --nodeps php  
#rpm -e --nodeps apache  
#rpm -e --nodeps postgresql
```

For the tools associated with apache, postgresql, and php, specify and delete them.

```
#rpm -qa | grep apache  
#rpm -qa | grep postgresql  
#rpm -qa | grep php
```

5. Installing apache

Download the source file of the apache with the version of 1.3.20 from
<http://httpd.apache.org/dist/httpd/>.

Create the directory of /usr/local/src at the root account.
Put the downloaded file at the directory of /usr/local/src.

```
#cd /usr/local/src  
#tar -xvzf apache_1.3.20.tar.gz
```

The downloaded file is expanded, and the directory of apache_1.3.20 is created.

```
#cd apache_1.3.20  
#OPTIM="-O2" ./configure --enable-rule=SHARED_CORE --enable-module=so >&config.log &  
#make >& make.log &  
#make install >& install.log &
```

Installing apache is completed.

>& is the command to write standard output in the following file. & at last enables one to carry the next operations without waiting for the execution of the command.

If one wants to see the content of `***.log`.

```
#tail -f ***.log
```

The file is closed by Ctrl+C.

To start apache automatically, the file of `/etc/rc.d/init.d/httpd` is edited as follows.

```
#!/bin/sh
#description: Apache
#chkconfig: 345 80 30
. /etc/rc.d/init.d/functions
APACHE="/usr/local/apache/bin/apachectl"

if [ -x $APACHE ]; then
    $APACHE $1
fi
```

This file is registered as follows.

```
#chmod +x /etc/rc.d/init.d/httpd
#/sbin/chkconfig --add httpd
#/sbin/chkconfig httpd on
```

6. Installing PostgreSQL

The source file of PostgreSQL with the version of 7.1.2 is downloaded from <ftp://ftp.jaist.ac.jp/pub/dbms/PostgreSQL/>.

The installation is carried out at the root account. The downloaded file is copied in the directory of `/usr/local/src`. Create the group named as "postgres". Register the account of "postgres" that belongs to the group, whose name is the same as the group, as a management user.

6.1 Creating group and management user

Delete the user named as postgres, which has been created by installing Redhat, and create a new group and accounts.

(Uninstalling Postgresql may delete the account.)

```
#userdel postgres
#groupadd postgres
#useradd -g postgres -d /home/postgres -m -s /bin/bash postgres
#passwd postgres
```

6.2 Creating the directory for postgresQL and installation

Installing postgresQL is carried out not at the root but at postgres. At the root, make the directory for postgresQL, and expand the source file there. Transfer it to the group of postgres, and install it at postgres.

```
#mkdir /usr/local/pgsql
```

```
#chown postgres.postgres /usr/local/pgsql
#cd /usr/local/src
#tar -xvzf postgresql-7.1.2.tar.gz
```

The file is expanded and the directory of postgresql-7.1.2 is created.

```
#chown -R postgres.postgres postgresql-7.1.2
```

The following process is carried out at postgres.

```
#su - postgres
$cd /usr/local/src/postgresql-7.1.2
$./configure --enable-multibyte=EUC_JP --enable-syslog >& config.log &
$make >& make.log &
$make install >& install.log &
$exit
#
```

Installing PostgreSQL is completed.

6.3 Environment setting for PostgreSQL

The following process is carried out at the root.

```
#su -
```

The environment variables for PostgreSQL are registered in the file of /etc/profile and /etc/ld.so.conf.

Add the following lines to the end of /etc/profile

```
export POSTGRES_HOME=/usr/local/pgsql
export PATH=$PATH:$POSTGRES_HOME/bin
export PGLIB=$POSTGRES_HOME/lib
export PGDATA=$POSTGRES_HOME/data
export MANPATH=$MANPATH:$POSTGRES_HOME/man
```

Add the following line to the end of /etc/ld.so.conf

```
/usr/local/pgsql/lib
```

Add the following line to reflect the changes

```
#!/sbin/ldconfig -v
```

6.4 Initialization and setting database

The following processes are carried out at postgres.

```
#su - postgres
```

Initialization is done by:

```
$/usr/local/pgsql/bin/initdb
```

In order to set PostgreSQL, the file of /usr/local/pgsql/data/postgresql.conf is edited as follows.

The number in the left indicates the line number whose content is changed.

```
29: tcpip_socket = true           (from false to true)
144: ifdef ENABLE_SYSLOG         (delete comment out)
145: syslog = 0 # range 0-2      (delete comment out)
146: syslog_facility = 'LOCAL0'  (delete comment out)
147: syslog_ident = 'postgres'   (delete comment out)
148: endif                       (delete comment out)
```

When the database is made in another server, access it through internet. In this case, the file of /usr/local/pgsql/data/pg_hba.conf is edited as follows,

```
181: host    all    ***.***.***.***    255.255.255.0    trust
```

where *** indicates the IP address that you like to refer as the database server. In addition, the referred server has to be edited as follows.

```
181: host    all    ***.***.***.0    255.255.255.0    trust
```

This enables the server to access the database within the network.

6.5 Setting automatic start

It is possible to start database by

```
pg_ctl -w start
```

at postgres.

In order to start the database automatically at the same time of machine startup, edit at root as follow.

```
#cd /etc/rc.d/init.d
```

```
#cp /usr/local/src/postgresql-7.1.2/controlib/start-scripts/linux postgres
```

Add the following lines to the copied file of postgres

```
2: #description: postgresSQL
```

```
3: #chkconfig: 345 98 02
```

```
4: . /etc/rc.d/init.d/functions
```

Register the postgres file as follows.

```
#chmod +x postgres
```

```
#/sbin/chkconfig --add postgres
```

```
#/sbin/chkconfig postgres on
```

6.6 Registration for users

Skip this section if you do not set any local database.

After confirming that PostgreSQL is started up, the following procedures are carried out at postgres, where the user of nobody will be registered.

```
$/usr/local/pgsql/bin/pg_ctl -w start
```

```
$createuser nobody
```

```
$n
```

```
$n
```

When you are asked regarding the permission of database creation and the addition of users, input (n).

7. Installing php

The source file of php (version 4.0.6) is downloaded from

<http://http.php.net/download.php>.

7.1 Installing php

The downloaded file is placed and expanded at the directory of /usr/local/src, resulting in creating the directory of php_4.0.6. The following process is carried out at the root.

```
#cd /usr/local/src
```

```
#tar -xvzf php-4.0.6.tar.gz
```

```
#cd php-4.0.6
```

```
#./configure --enable-mbstring --enable-mbstr-enc-trans
--with-apxs=/usr/local/apache/bin/apxs
--with-pgsql=/usr/local/pgsql
--with-gd=/usr
--with-jpeg-dir=/usr >& config.log & (No return key is required)
#make >& make.log &
#make install >& install.log &
```

The installation is completed, and set environments.

```
#cp php.ini-dist /usr/local/lib/php.ini
```

Add the following lines to the end of the copied file of php.ini.

```
[mbstring]
mbstring.internal_encoding = EUC-JP
mbstring.http_input = auto
mbstring.http_output = pass
mbstring.detect_order = auto
mbstring.substitute_character = none
```

7.2 Setting change of apache

In order to use php, the setting file for apache, /usr/local/apache/conf/httpd.conf, should be edited as follows.

The number on the left indicates the line that should be edited.

```
363or364:  DirectoryIndex index.html index.php
765or768:  AddType application/x-httpd-php .php
766or769:  AddType application/x-httpd-php-source .phps
```

7.3 Function confirmation

Create info.php where <? phpinfo() ?> is just written at the directory of /usr/local/apache/htdocs. Restart apache by:

```
#/usr/local/apache/bin/apachectl restart
```

, or reboot the machine.

From browser, input
http://machine_name or IP machine_address/info.php,
If various data regarding php is displayed, php functions well.

7.4 Installing phplib

The source file of phplib (version 7.2d) from
<http://prdownloads.sourceforge.net/phplib/phplib-7.2d.tar.gz?download>

Expand the file at the directory of /usr/local/src.
The following process is carried out at the root.

```
#cd /usr/local/src
#tar -xvzf phplib-7.2d.tar.gz
```

Next, create the symbolic link to the directory for apache as follows:

```
#cd /usr/local/apache
#ln -s ../src/phplib7.2d phplib
```

Next, convert the extension of the files of the directories (/usr/local/apache/phplib/php, /usr/local/apache/phplib/pages, /usr/local/apache/phplib/pages/abmin) from "php3" to "php".

```
# mv ~.php3 ~.php
```

```
#cd /usr/local/apache/phplib/php
```

Here, edit prepend.php as follows.

```
13: $_PHPLIB["libdir"] = "";
```

```
15: require($_PHPLIB["libdir"] . "db_pgsql.inc");
```

The installation is completed.

8. Installing Java2 SDK

The source file of Java2 SDK (version1.4.1) is downloaded from <http://java.sun.com/products/archive/index.html>

The file is placed at /root/, where the installation will be carried out.
The following process is carried out at the root.

```
#cd
#chmod +x j2sdk-1_4_1_03-linux-i586-rpm.bin
#./j2sdk-1_4_1_03-linux-i586-rpm.bin
```

When the license agreement is asked, click "yes" to make .rpm file.

```
#rpm -iv j2sdk-1_4_1_03-fcs-linux-i586.rpm
```

The file is installed at /usr/java/j2sdk1.4.1_03. Create the symbolic link of jdk to /usr/java.

```
#cd /usr/java
#ln -s j2sdk1.4.1_03 jdk
```

Installing Java is completed.

9. Installing Xerces2 Java Parser

The source file of **Xerces2 Java Parser** (2.2.1) is downloaded from <http://xml.apache.org/dist/xerces-j/>.

The file is placed at /root/, where the installation is carried out.
Expand the file, which creates the directory of xerces-2_2_1.
The following process is carried out at the root.

```
#cd
#tar -xvzf Xerces-J-bin.2.2.1.tar.gz
```

The expanded files are transferred to /usr/local/, and create the symbolic link.

```
#mv xerces-2_2_1 /usr/local/
#cd /usr/local/
#ln -s xerces-2_2_1 xerces
```

The installation is completed.

10. Installing mpich

The source file of mpich with the version 1.2.2.2 is downloaded from <http://www-unix.mcs.anl.gov/mpi/mpich/>

Expand mpich.tar.gz at root.

```
#cd
#tar -xvzf mpich.tar.gz
```

The directory of mpich-1.2.2.2 is automatically created.

```
#cd mpich-1.2.2.2
#./configure --prefix=/usr/local/mpich --fc=g77 --cc=gcc >& configure.log &
#make >& make.log &
#make install >& install.log &
```

The file is installed at the directory of /usr/local/mpich.

11. Installing CADLIVE Simulator

11.1 Installing CADLIVE Simulator

Using the root account, make the directory of /home/CADLIVE, and expand CADLIVE_DATA.tgz at the directory of /home/CADLIVE

```
#mkdir /home/CADLIVE
#cd /home/CADLIVE
#tar -xvzf CADLIVE_DATA.tgz
```

The directories necessary for the simulator are created automatically. In the directory of ./Session, one operates jobs via browser. Actually, since one operates jobs at the account of nobody, set the simulator so that "nobody" can read and write.

```
#chmod 777 Session
```

11.2 Setting apache

PC clients are required to access /home/CADLIVE/Life via a browser. Thus, the file of /usr/local/apache/conf/httpd.conf is reedited as follows,

```
284: DocumentRoot "/home/CADLIVE"

309: <Directory "/home/CADLIVE">
319:     Options MultiViews
326:     AllowOverride FileInfo Options
331:     Order allow, deny
332:     Allow from all
333: </Directory>
```

In the above case, a PC browser is able to access the simulator by <http://machine%name/Life>.
(Use ID:kris Passwd:test)

11.3 Installing batch job

CADLIVE/Tools is the directory for batch jobs. Create the load modules and libraries necessary for

system operations by the following commands:

```
#cd /home/CADLIVE/Tools
#make depend
#make install
```

11.4 Setting php

Since php controls batch jobs, one changes the environment setting according to the installation directory or machine environments. The file of CADLIVE/Chemi/define.inc is changed as follows:

CADLIVE/Chemi/define.inc

```
2: // definition of path
3: $CADLIVE_home      = "/home/CADLIVE";
4: $dirBin             = $CADLIVE_home."/Tools/bin/";
5: $dirPhpSource       = $CADLIVE_home."/Chemi/";

13: // parallel computing
14: $num_cpu           = 8;
15: $mpirun             = "/usr/local/mpich/bin/mpirun -np ";
```

\$CADLIVE_home is the absolute path for the installation directory. \$num_cpu indicates the number of PCs controlled by MPI.

Since the setting of apache is altered, restart.

```
#/usr/local/apache/bin/apachectl restart
```

Installing CADLIVE is completed.

11.5 Setting database

One has to set database.

Brief instruction of postgresQL (at postgres)

```
START:          pg_ctl -w start
SHUTDOWN        pg_ctl stop
                (If one fails, pg_ctl -m immediate stop)
LIST DB         psg -l
CREATE DB        createdb name_database
DELETE DB        dropdb name_database
FORMAT DB        psql -e name_database < definition_file
BACKUP DB        pg_dump name_database > BackUp_file
RESTORE DB       psql < BackUp_file
```

Create the database for CADLIVE,

```
$create life_db
```

Next, define the format type for DB. The following two files are employed as the definition files.

```
$psql -e life_db /home/CADLIVE/Life/create_database.pgsq
$psql -e life_db /home/CADLIVE/Life/create_mydb.pgsq
```

11.6 Others

One has to set the file of local.inc.

At user account, edit /home/CADLIVE/Life/local.inc as follows.

```
12: var $Host = "****.***.***.***";  
13: var $Database = "name_database";
```

"name_database" is the same that has been created previously. In the line 12, when one makes DB at local, Host is localhost. When DB is referred from outside, Host is the its own host name. When one refers to another host, Host is the host name that one wants to refer.

For example,
var \$Host = "kurata01.bse.kyutech.ac.jp";