Sprinx
THE DOERS

INSTALLATION GUIDE

SHOCK VML Portal

Date: 14th August 2015

CONTENTS

INTRODUCTION **3**

PURPOSE OF THIS DOCUMENT 3

SW REQUIREMENTS 3

INSTALLATION **4**

COMPONENTS 4

PREREQUISITIES 4

APACHE SERVER 5

DATABASE SERVER - POSTGRESQL 6

3RD PARTY LIBRARIES 9

PYTHON AND ITS LIBRARIES 11

WEB AND RENDER APPLICATION 12

SFPT 14

ONLINE PYTHON 16

BASIC SETTINGS 18

INTRODUCTION

PURPOSE OF THIS DOCUMENT

This document is intended for administrators, developers and users who want to install the whole solution of the **Virtual Mission Laboratory** (hereinafter only **VML**). The document describes the procedure how to make the VML application alive including basic setup and installation of the 3rd party libraries and applications which are necessary for smooth running of the VML.

SW REQUIREMENTS

For proper and full functionality of VML portal, you need to use a web browser with JavaScript and HTML5.

Recommended web browsers:

- Mozilla Firefox (version 19 or higher)
- Chrome (version 25 or higher)
- Internet Explorer (version 10 or higher)

The development of the VML has been intended for computers with **linux operating system**. That is the reason why this guide describes how to perform the installation for linux operating systems only.

Recommended operating systems:

- Linux OpenSUSE 13.2 (32-bit)

Requirements:

- **Administrator privileges (root)**

Important note:

It is possible to use whichever distribution and version of linux operating system, but some steps of the installation from this guide might differ from other versions or distributions.

INSTALLATION

COMPONENTS

The whole solution of the VML portal consists of following parts:

- Prerequisites
- Apache server
- Database server - PostgreSQL
- 3rd party libraries (CDF and Boost library)
- Python and its libraries
- Web and render application
- SFTP server
- Online python convertor

PREREQUISITIES

By the term **prerequisites** it is meant all libraries and software components which are required for successful installation of the VML. All prerequisites should be provided by operating system and its distribution. Before you start with the installation of the VML, please make sure you have successfully installed all of them or make an installation again.

List of prerequisites:

- blas
- freetype
- freetype2-devel
- freetype-devel
- gcc
- gcc-c++
- gcc-fortran
- hdf5
- hdf5-devel
- lapack-devel
- libblas3
- liblapack3
- libpng-devel
- make
- openssl
- zlib-devel

zypper install blas freetype freetype2-devel freetype-devel gcc gcc-c++ gcc-fortran hdf5 hdf5-devel lapack-devel libblas3 liblapack3 libpng-devel make openssl zlib-devel

Start the installation process with extracting of the installation package (***VML-install-package.tar.gz***)

Locate VML-install-package archive file, e.g.:

```
cd /home/user
```

Extract the file to directory of your choice, e.g.:

```
tar -xzf VML-install-package.tar.gz -C /home/user/
```

APACHE SERVER

1) Install Apache server from the distribution package

```
zypper install apache2 apache2-mod_php5 php5 php5-pgsql
```

Note: It is recommended to install also packages which will be suggested after this command

2) Turn on special server features

   Modify the configuration file *(/etc/apache2/default-server.conf*) on the **line 22**. Change the value *"Options None"* to the value *"**Options All**"*.

3) Enable php5 mode

```
a2enmod php5
```

4) Restart apache server

```
systemctl restart apache2.service
```

DATABASE SERVER - POSTGRESQL

1) Install **PostgreSQL server** from the distribution package

```
zypper install postgresql-server postgresql postgresql-devel postgresql-contrib
```

Note: It is recommended to install also packages which will be suggested after this command

2) Turn the PostgreSQL server on

```
rcpostgresql start
```

3) Switch user to postgres

```
su - postgres
```

4) Run *psql* and set the **default password** for *postgres* user

```
psql
```

write the following commands:

```
ALTER USER postgres WITH PASSWORD 'postgres';
```

```
\q
```

5) Set the server to listen to all addresses, not only from localhost

Modify the configuration file (*/var/lib/pgsql/data/postgresql.conf*) on the **line 59**. Change the value *"#listen_addresses = 'localhost'"* to ***"listen_addresses = '*'"***.

Modify the configuration file (*/var/lib/pgsql/data/pg_hba.conf*) on the **line 79**. Change the method value from *"ident"* to ***"trust"***, see the following example.

```
# "local" is for Unix domain socket connections only
local           all         all                               trust
# IPv4 local connections:
host            all         all         127.0.0.1/32          trust
# IPv6 local connections:
host            all         all         ::1/128               trust
```

6) Switch back to the root user and restart the server

```
exit

rcpostgresql restart
```

7) Create the database and the data structure required for the VML

Switch user to postgres

```
su - postgres
```

Locate VML-install-package folder, e.g.:

```
cd /home/user/VML-install-package
```

Create a new **database** named **esa**

```
createdb esa
```

Run SQL scripts for creating of the data structure and basic dataset

```
psql -U postgres -f dump.sql esa

psql -U postgres -f dump2.sql esa
```

Switch back to root user

```
exit
```

8) Instal thel web tool ***phpPgAdmin*** for online access to the database

Locate VML-install-package folder, e.g.:

```
cd /home/user/VML-install-package
```

Extract *phpPgAdmin* tar file to the location */srv/www/htdocs/*

```
tar -xzf ./libraries/phpPgAdmin-5.1.tar.gz -C /srv/www/htdocs/
```

Modify the configuration file (*/srv/www/htdocs/phpPgAdmin-5.1/conf/ config.inc.php*) on the **line 93**. Change the value *"$conf['extra_login_security'] = true;"* to ***"$conf['extra_login_security'] = false;".***

3RD PARTY LIBRARIES

1) Install **Boost library** from the provided archive file

Locate VML-install-package folder, e.g.:

```
cd /home/user/VML-install-package
```

Extract the archive file

```
tar -xzf ./libraries/boost_1_53_0.tar.gz -C ./extract
```

Locate the boost folder, e.g.:

```
cd /home/user/VML-install-package/extract/boost_1_53_0
```

Run the installation

```
sh ./bootstrap.sh
./b2 install
```

2) Install **CDF library** from the provided archive file

Locate VML-install-package folder, e.g.:

```
cd /home/user/VML-install-package
```

Extract the archive file

```
tar -xzf ./libraries/cdf34_1-dist-cdf.tar.gz -C ./extract
```

Locate the CDF extract folder, e.g.:

```
cd /home/user/VML-install-package/extract/cdf34_1-dist
```

Run the installation

```
make OS=linux ENV=gnu CURSES=no FORTRAN=no UCOPTIONS=-O2 SHARED=yes all

make INSTALLDIR=/usr/local/cdf install
```

PYTHON AND ITS LIBRARIES

1) Install **Python** from the distribution package

```
zypper install python3-devel python-pip
```

2) Make sure that python include folder is included as a default folder for searching header files

```
C_INCLUDE_PATH=/usr/include/python
export C_INCLUDE_PATH

CPLUS_INCLUDE_PATH=/usr/include/python
export CPLUS_INCLUDE_PATH
```

3) Install **Numpy** library to the python via pip install

```
pip install numpy
```

4) Install **Matplotlib** library to the python via pip install

```
pip install matplotlib
```

5) Install **Scipy** library to the python via pip install

```
pip install scipy
```

6) Install **Spacepy** library to the python via pip install

```
pip install Cython

pip install spacepy
```

7) Install **Pillow** library to the python via pip install

```
pip install pillow
```

WEB AND RENDER APPLICATION

1) Locate VML-install-package folder, e.g.:

```
cd /home/user/VML-install-package
```

2) Extract the archive file ***esa.tar.gz*** to the destination */home/*

```
tar -xzf esa.tar.gz -C /home
```

3) To active a view over a web browser make a link to the application

```
ln -s /home/esa /srv/www/htdocs/esa
```

4) Modify the file (/etc/**ld.so.conf**), add the following line and save the file

```
/usr/local/cdf/lib
```

5) Activate the changes in the previous file by the following command

```
ldconfig
```

6) Locate the cpp_vml_code folder in the home/esa directory:

```
cd /home/esa/cpp_vml_code
```

7) Make sure that are necessary flags are included in the makefile

Change the **line 10** of the **Makefile** *(/home/esa/cpp_vml_code/Makefile*) according to the output of the following command. Check also the **line 39** whether python include directories match or need some adjustment. If they are the same, no action is required.

```
python3.4-config --cflags
```

Note: This command might differ according to your python version

Change the **line 30** of the **Makefile** *(/home/esa/cpp_vml_code/Makefile*) according to the output of the following command. If all flags are included, no action is required.

```
python3.4-config --ldflags
```

Note: This command might differ according to your python version

Repeat this whole step with the **Makefile** for the importer tool located in */home/esa/cpp_vml_code/importer/Makefile*

8) Run the make file and build the **VML** tool and the **importer** tool

```
make >log.txt 2>log_e.txt

cd /home/esa/cpp_vml_code/importer

make >log.txt 2>log_e.txt
```

Check if the last line of the file **log.txt** (in the both directories) includes the word "*Done*" (on the third line). If so you successfully built the application and you can continue with the next step. In another case, please check the error log (**log_e.txt**) and look for errors. Here we cannot provide detailed instructions how to exactly fix this state. Please check the internet to look up some solutions according to errors.

9) Copy built binary files into the bin folder (*/home/esa/bin*)

```
cp /home/esa/cpp_vml_code/vml.out /home/esa/bin/

cp /home/esa/cpp_vml_code/importer/importer.out /home/esa/bin/
```

SFPT

1) Create a new usergroup named **sftponly**

```
groupadd -r sftponly
```

2) Create a new linux user named **wwwrun** and add him to the group

```
usermod -a -G sftponly wwwrun
```

Note: If user wwwrun does not exist, use command: **useradd** -G sftponly wwwrun

3) Create a new directory named **sftponly** in the root directory

```
mkdir /sftponly
```

Set the ownership of sftponly and esa folders to the user wwwrun

```
chown root:root /sftponly

chmod 0750 /sftponly

chown -R wwwrun /home/esa
```

4) Create a special loop mounted partition for /sftponly mountpoint. It is needed to prevent privilege escalation and to (optionally) setup user quotas (e.g. 10GB)

```
dd if=/dev/zero of=/home/sftponly_image.ext4 bs=1G count=10

mkfs.ext4 /home/sftponly_image.ext4
```

5) Add the following line to the end of the file (*/etc/init.d/boot.local*)

```
mount -o loop,nodev,nosuid /home/sftponly_image.ext4 /sftponly
```

Run the file to apply changes

```
/etc/init.d/boot.local
```

6) Modify the file */etc/ssh/sshd_config*. Add the following lines to the end of the file

```
Match group sftponly
ForceCommand internal-sftp
ChrootDirectory %h
X11Forwarding no
AllowTcpForwarding no
```

7) Locate VML-install-package folder, e.g.:

```
cd /home/user/VML-install-package
```

and copy the script **newuser.sh** to the folder */srv/www/cgi-bin/*

```
cp newuser.sh /srv/www/cgi-bin/newuser.sh
```

8) Add scripts used for maintenance to the **cron**

```
crontab -u wwwrun crontab.wwwrun
```

ONLINE PYTHON

1) Modify the file **sudoers** (*/etc/sudoers*)

   Find the variable named **Defaults env_keep** (line 43) and make sure that the following values are included

```
Defaults env_keep = "MPLCONFIGDIR SPACEPY PYTHON_EGG_CACHE CDF_LIB CDF_BASE CDF_TMP LANG LC_ADDRESS ...."
```

Note: Do not change already presented values, add new ones only

   Add the following lines to the end of the file

```
wwwrun  ALL=(ALL) NOPASSWD: /usr/sbin/useradd, /usr/sbin/chpasswd, /bin/ln, /bin/chown, /bin/chmod, /usr/bin/find, /bin/cat, /bin/rm, /bin/mkdir, /bin/cp, /usr/bin/rpython
```

2) Configure apparmor for the python with restricted privileges (**rpython**)

   Copy the python binary file and rename it to **rpython**

```
cp /usr/bin/python3.4 /usr/bin/rpython
```

Note: This command might differ according to your python version

3) Create a file named **usr.bin.rpython** in the location /etc/apparmor.d/ and add the following lines to the file

```
#include <tunables/global>
/usr/bin/rpython {
  #include <abstractions/base>
  #include <abstractions/fonts>
  #include <abstractions/python>
  /proc/** r,
  /etc/ssl/** r,
  /tmp/** rw,
  /usr/local/ rm,
  /usr/local/** rm,
  owner /sftponly/** rw,
}
```

4) Activate rpython rules

```
apparmor_parser /etc/apparmor.d/usr.bin.rpython
```

5) Locate VML-install-package folder, e.g.

```
cd /home/user/VML-install-package
```

Copy the script **run_restricted** to the directory */usr/local/bin*

```
cp run-restricted /usr/local/bin
```

BASIC SETTINGS

Before you start using the VML you might need to make some basic settings. Please pay attention especially to the following:

File **LocalSettings.php** in the location */home/esa/LocalSettings.php*

- hostname (**line 59**)

File **DefaultSettings.php** in the location */home/esa/includes/DefaultSettings.php*

- hostname (**line 97**)
- text of the email for newly registered user (**line 109**)
- email sender name (**line 115, 116**)

Now everything should be ready for using of the VML portal. **Start with the registration of a new admin user:**

1) Type address to your browser (e.g.: localhost/esa) and register a new future admin user
2) Log out the current user
3) Log in with the following credentials: **esa@esa.esa**, password **esa**
4) Change the role of the previous registered user via tab **manage users**
5) **Deactivate** the user **esa@esa.esa**
6) Log in as an administrator

For more information, please see the User manual for admin users.