The maproom development process

On Server

Root User
Create User
Link Bitbucket user
Get local copy
Get local copy
Make Changes
Commit Changes
Make Tarball
Migrate to www folder
Assign to account

Online

Create Bitbucket User
Clone template maproom
Fork to new maproom
Push to Bitbucket
Visualize on-line

Root User
Root User
Root User
Download PuTTY software for interaction with the server

Go to http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

PuTTY Download Page

Here are the PuTTY files themselves:

- PuTTY (the Telnet and SSH client itself)
- PSCP (an SCP client, i.e. command-line secure file copy)
- PSFTP (an SFTP client, i.e. general file transfer sessions much like FTP)
- PuTTYTel (a Telnet-only client)
- Plink (a command-line interface to the PuTTY back ends)
- Pageant (an SSH authentication agent for PuTTY, PSCP, PSFTP, and Plink)
- PuTTYgen (an RSA and DSA key generation utility).

LEGAL WARNING: Use of PuTTY, PSCP, PSFTP and Plink is illegal in countries where encryption is outlawed. I believe it is legal to us lawyer and so in doubt you should seek legal advice before downloading it. You may find this site useful (it's a survey of cryptography laws).

Use of the Telnet-only binary (PuTTYTel) is unrestricted by any cryptography laws.

There are cryptographic signatures available for all the files we offer below. We also supply cryptographically signed lists of checksums. To a Windows program to compute MD5 checksums, you could try the one at this site. (This MD5 program is also cryptographically signed by

Binaries

The latest release version (beta 0.63). This will generally be a version I think is reasonably likely to work well. If you have a problem with already fixed the bug, before reporting it to me.

For Windows on Intel x86

PuTTY: putty.exe
PuTTYTel: puttytel.exe

(or by FTP) (RSA sig) (DSA sig)

Open the PuTTY program and create a new session with these characteristics:

Host name: ip1.ceazamet.cl

Port: 10022

Connection type: SSH

Saved Session: ClimateDataLibrary

Then click ‘Save’
Then click ‘Open’ or double click on the recently created session.

When login appears: type ‘root’ (login as: root)

Type in the root password. You are now logged in as a root user.
Create new user in the Data Library

After entering as root user to the data library, you can create your user account:

To add a user:

>useradd kverbist2

To give it a password:

>passwd kverbist2

If you want to eliminate a user:

>userdel kverbist2

The creation of a user implicates that automatically the folder /home/user_name will be created, that will be assigned to this user
Create new user in Bitbucket or receive invitation

You can create a new user for the maproom code repository 'Bitbucket', entering directly into the website: https://bitbucket.org/

Or you can request an invitation from a user, which is the preferred option.


**Setup Bitbucket under your account**

**Introduce yourself to Git**

Introduce yourself to Git with your name and public email address. This information will be recorded in Git repository when you commit your changes. The easiest way to do so is:

```
>git config --global user.name "kverbist2"
>git config --global user.email "k.verbist@unesco.org"
```

**Generate and setup certificate**

1. Login as user

```
>su kverbist2
```

2. Generate certificate w/o passphrase (press Enter twice when asked for the passphrase):

```
$ ssh-keygen -f ~/.ssh/bitbucket -t rsa -C "k.verbist@unesco.org"
```

3. Create `~/.ssh/config`:

```
cat <<eos >>~/.ssh/config
Host bitbucket.org
HostName bitbucket.org
IdentityFile ~/.ssh/bitbucket
eos
chmod go-rwx ~/.ssh/config
```

```
[root@www kverbist2]$ cat <<eos >~/.ssh/config
> Host bitbucket.org
> HostName bitbucket.org
> IdentityFile ~/.ssh/bitbucket
> eos
[root@www kverbist2]$ chmod go-rwx ~/.ssh/config
```
4. Create a session in WinSCP with your new user

How to download and setup WinScp is explained in the previous manual and can be found here.

Now login this session as your new user.

5. Add your certificate to Bitbucket as follows:

- Open the file bitbucket.pub that is located under /home/yourname/.ssh/
- Copy the contents of ~/.ssh/bitbucket.pub to SSH Key field. Make sure that you copy the text of the public key exactly without extra spaces or carriage returns.
• Go to the location in bitbucket that holds all SSH keys: Bitbucket SSH location
• Set Label field to some unique value that identifies the key, e.g. work, home
• Paste the contents of ~/.ssh/bitbucket.pub to SSH Key field. Make sure that you paste the text of the public key exactly without extra spaces or carriage returns.
• Press Add key
5. Run a test to check if connection can be made:

```
>ssh -T -i ~/.ssh/bitbucket git@bitbucket.org
```

If successful, you should see something like this:

You can use git or hg to connect to Bitbucket.

**Setup command prompt (optional)**

You may set up your command prompt to show which branch you are on. Add the following line to your ~/.bashrc:

```bash
function parse_git_branch () {
    git branch 2> /dev/null | sed -e '/^[^*]/' -e 's/* \([^*]/\1\)/'
}
RED="\[033[0;31m\]
YELLOW="\[033[0;33m\]
GREEN="\[033[0;32m\]
NO_COLOUR="\[033[0m\]
PS1="$YELLOW\h:\w$RED\$(parse_git_branch)$NO_COLOUR$ \\
alias g=git
export PATH=$HOME/bin:/usr/local/miconf/bin:/usr/local/semantic_tools/bin:$PATH
```
Setup git aliases (optional)

Coapt paste the following code into the terminal:

```
cat <<eos>>/.gitconfig

[alias]
  l = log --graph --pretty=format:"%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr)
  %C(bold blue)<%an>%Creset"
  a = add
  b = branch
  ci = commit
  co = checkout
  sb = show-branch
  ds = diff --cached
  f = reflog
  r = remote -v
  sm = submodule
  smu = submodule update --init --recursive
  lgg = log --all --graph --decorate
  lgr = log --all --graph --decorate --oneline --simplify-by-decoration --no-merges
  lgt = log --format='%h %an %ar - %s'
  lg1 = log --oneline origin..HEAD
  lgf = log --oneline HEAD..origin/master
  h = help
  cl = clone --recursive
  cfd = clean -fd
  cffd = clean --fffd

eos
```
Cloning a local copy of a maproom to your account
As a first test, clone the template maproom:

1. Go to the bitbucket page of this maproom: https://bitbucket.org/kverbist/maproom_template_chile
And copy the SSH link

2. Go to the terminal and clone the repository recursively (including subfolders)
>git clone --recursive git@bitbucket.org:kverbist/maproom_template_chile.git
Initialized empty Git repository in /home/kverbi2t/maproom_template_chile/.git/
remote: Counting objects: 99, done.
remote: Compressing objects: 100% (51/51), done.
remote: Total 99 (delta 21), reused 0 (delta 0)
receiving objects: 100% (59/59), 20.48 KiB, done.
Resolving deltas: 100% (21/21), done.
Submodule 'maproomtools' (git@bitbucket.org:iriidl/maproomtools.git) registered for path 'maproomtools'
Submodule 'miconf' (git@bitbucket.org:iriidl/miconf.git) registered for path 'miconf'
Submodule 'pure' (git@bitbucket.org:iriidl/pure.git) registered for path 'pure'
Submodule 'ucore' (git@bitbucket.org:iriidl/ucore.git) registered for path 'ucore'
Initialized empty Git repository in /home/kverbi2t/maproom_template_chile/maproomtools/.git/
remote: Counting objects: 293, done.
remote: Compressing objects: 100% (200/200), done.
remote: Total 293 (delta 170), reused 163 (delta 94)
receiving objects: 100% (293/293), 16.70 KiB, done.
Resolving deltas: 100% (170/170), done.
Submodule path 'maproomtools': checked out '1470cb813a55de144f1a71709e46655864d015b50b'
Initialized empty Git repository in /home/kverbi2t/maproom_template_chile/miconf/.git/
remote: Counting objects: 352, done.
remote: Compressing objects: 100% (302/302), done.
remote: Total 352 (delta 150), reused 171 (delta 42)
receiving objects: 100% (352/352), 302.39 KiB | 514 KiB/s, done.
Resolving deltas: 100% (150/150), done.
Submodule path 'miconf': checked out 'e68eb47b5809baa958e6e462845af4c6c46ecdc'
Submodule 'lua' (git@bitbucket.org:iriidl/lua.git) registered for path 'lua'
Initialized empty Git repository in /home/kverbi2t/maproom_template_chile/miconf/lua/.git/
remote: Counting objects: 118, done.
remote: Compressing objects: 100% (118/118), done.
remote: Total 119 (delta 118), reused 293 (delta 118)
receiving objects: 100% (293/293), 502.38 KiB | 61 KiB/s, done.
Resolving deltas: 100% (118/118), done.
Submodule path 'lua': checked out 'a1e97d785a742f8d222ac48bea3b8e589cdd5d63'
Initialized empty Git repository in /home/kverbi2t/maproom_template_chile/pure/.git/
remote: Counting objects: 60, done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 60 (delta 16), reused 0 (delta 0)
receiving objects: 100% (60/60), 381.65 KiB | 42 KiB/s, done.
Resolving deltas: 100% (16/16), done.
Submodule path 'pure': checked out 'd43b2bad39a9da0c68103d76d2495826d1e16b9'
Initialized empty Git repository in /home/kverbi2t/maproom_template_chile/ucore/.git/
remote: Counting objects: 702, done.
remote: Compressing objects: 100% (570/570), done.
remote: Total 702 (delta 395), reused 472 (delta 222)
receiving objects: 100% (702/702), 246.08 KiB | 232 KiB/s, done.
Resolving deltas: 100% (395/395), done.
Submodule path 'ucore': checked out '2742be8f5be5d1f6ad4b80b979f71663f4bdb576'
Fork a new Maproom off of the original maproom_template_chile

Go to bitbucket and login

Click ‘Fork’ (NOT Fork of maproom_template)

Give a name and description and press ‘Fork Repository’

Go to administration:
In User Management, you can set user permissions and add/delete users.

Now select the SSH-key to the maproom and copy (Select and click ‘ctrl+C’).
Go to the terminal and go to your home directory:

Now clone the repository by typing `git clone --recursive` and paste the new repository ssh-key (click 'ctrl+V')

```bash
git clone --recursive git@bitbucket.org:kverbist2/maproom_test.git
```

Now go to the folder of the maproom and update the submodules using the command, to make sure the scripts that create the maproom are updated to the latest version:

```bash
cd maproom_test
```
You now have a personal copy of the template maproom, which you can modify and expand as wanted, without affecting any other users.

Make changes to the master, commit and push to the central repository
As a quick test to see if all is setup well, we make 1 small change to the maproom.

Go to the maproom section within maproom_test

> cd maproom

Check available files:

> ls

We see 1 xhtml file and 1 folder that holds additional xhtml and html files. As a test, we make a change to index.xhtml.en

Open the file in the WinSCP environment (make sure you are logged in as your user, and NOT as root).

Change the title >CEAZA map room< to >kverbist2 map room<
Now go to the terminal and check the status of the git repository:

```bash
>git status
```

A file has been modified from its original.

Now add the file to the files you want to add to the online repository ‘Bitbucket’.

```bash
>git add *
```

And commit the changes to the file sent to the repository

```bash
>git commit -a
```
A new screen will appear that requests to describe the changes made (comment).

Now enter i to start inserting text, the bottom changes to -- INSERT--

Now type the changes made: “Changed title to kverbist2 map room”
If finished, press ESC and type
:wt

to exit the editor (wq stands for write and quit). You will return to the prompt and see following message:

```
git config --global user.name "Your Name"
git config --global user.email you@example.com
```

If the identity used for this commit is wrong, you can fix it with:

```
git commit --amend --author='Your Name <you@example.com>'
```

1 files changed, 1 insertions(+), 1 deletions(-)

To check you are well connected to the repository, you can push your commitment to the repo

```
git push --up origin master
```

From now on, you can use git push to send commitments.

```
git push
```

```
Everything up-to-date
```

**Migrate the maproom to an online location**

Make sure you are in the main folder of the maproom `maproom_test`

```
pwd
```

```
/home/kverbist2/maproom_test
```

Now create a tarball (compressed version of the maproom):
>make tarball

```
make tarball
```

```
www:/maproom_test (master)$ make tarball
```

```
git-generated-version-info maproom.xml >maproom/version.xml
```

```
cd maproom; ./maproomtools/build_maproom.pl cwl-max-optimized;
```

```
Using ruleset 'owl-max-optimized'
```

```
Building maproomregistry.
```

```
mar oct 8 23:07:13 CLT 2013
```

```
Setting '/home/kverbist2/maproom_test/maproom/newmaproomcache' directory
```

```
Gathering rdfa triples
```

**IMPORTANT: open up a new console and login as root!!**

Then go to /var/www/html/

>cd /var/www/html/

Untar (de-inflate) the maproom in this folder (your version will vary)

> tar xvzf ~kverbist2/maproom_test/maproom-1.0.2-20-7a3f9551e3e89f04ee4808aa30103b.tgz

Now remove the current link to the maproom

> rm –f kverbist2

Now create a new virtual link to the maproom with freshly created maproom

(ln –s maproom_name link_name)

>ln –s maproom-1.0.2-20-7a3f9551e3e89f04ee4808aa30103b kverbist2

Now go to the online version of the maproom:

http://www.climatedatalibrary.cl/kverbist2/maproom/

You can see the title of the page reads now: **kverbist2 map room**
Congratulations: you implemented your first maproom!

CEAZA Map Room

The maproom is a collection of maps and other figures that monitor climate and societal conditions at present and in the recent past. The maps and figures can be manipulated and are linked to the original data. Even if you are primarily interested in data rather than figures, this is a good place to see which datasets are particularly useful for monitoring current conditions.

Examples
Climate affects sectors in society in a number of ways. These effects may be direct, as with heat stress, or indirect, as with infectious diseases such as malaria and meningitis.
How to add new maproom ‘accounts’

First go to /usr/local/squid/etc

First make a backup of the current (working) config file, using the current date:

>cp squid.conf squid.conf.20130517

Now edit the config file using

>vi squid.conf

And search for the line that states ‘acl iridlmaproom url_regex ^http://localiridl...’

You will find a list of already defined maproom names, such as UNEA and CAZALAC, to which you can add additional names, after using a pipe symbol ‘|’.

In order for the changes to come into effect, the squid needs to be reconfigured:

>../sbin/squid -k parse
>../sbin/squid -k reconfigure

Now stop and restart ingrid and squid
In order to check if Ingrid is running satisfactorily, you should check the service is running:

```bash
> ps -ef | grep ingrid
```

```
root 27173 1 0 21:48 pts/0 00:00:00 ./ingridd
root 27185 26871 0 21:49 pts/0 00:00:00 grep ingrid
```

**Making a new tag for the maproom**

First create a new tag

```bash
> git tag maproom_xyz-1.0.0
> git push --tags
```

Then adjust the Makefile with the following changes

VER = `$(shell miconf/scripts/git-generate-version-info maproom_xyz tag)`

VER_ID = `$(shell miconf/scripts/git-generate-version-info maproom_xyz id)`

**Make a new branch ‘cazalac’ based on master**

```bash
> git checkout --b cazalac master
```

Now you can make changes to the local version of the master under this new branch. If you’re finished:

```bash
> git status
> git add *
> git commit -a
```

Push the new branch and its first commit to the central repository (called ‘origin’):
>git push –u origin cazalac

Now create a tarball (compressed version of the maproom):

>make tarball

IMPORTANT: open up a new console and login as root

Then go to /var/www/html/

>cd /var/www/html/

Untar (de-inflate) the maproom in this folder (your version will vary)

>tar xvzf ~/kverbist/maproom_cazalac/maproom-1.3.0-19-ga1525f7ea2d8b9260fa4f54b0987e671cb

Now remove the current link to the maproom

>rm –f MAPROOM_NAME

Now create a new virtual link to the maproom with freshly created maproom

(ln –s maproom_name link_name)

>ln –s maproom-1.3.0-19-ga1525f7ea2d8b9260fa4f54b0987e671cb MAPROOM_NAME