



Migrazione di un progetto da CVS a GIT

Per migrare un progetto dal sistema di versioning CVS a git sono disponibili diverse utility, ma già con quella inclusa nel normale client git ([cvsimport](#)) è possibile effettuare l'operazione.

Questo strumento per funzionare richiede un accesso diretto al repository CVS da filesystem, poiché alcune operazioni necessarie per la conversione non possono essere effettuate lato client.

L'operazione è ripetibile nel caso in cui vengano effettuati nuovi commit sul progetto CVS anche dopo la migrazione a git: questi nuovi commit saranno integrati all'interno del repository git.

Migrazione manuale

Per effettuare la migrazione manualmente bisogna effettuare i seguenti passaggi:

- creazione di un nuovo repository git vuoto

```
git --bare init /path/to/new/repo/<CVSMODULE>
```

- conversione da CVS a git

```
git --bare cvsimport -d:local:<CVSROOT> -v -k -m -i [-A <AUTHORFILE>] -o master -C  
/path/to/new/repo/<CVSMODULE> <CVSMODULE>
```

dove:

- <CVSROOT> indica il path assoluto sul file system del repository CVS
- /path/to/new/repo indica la locazione su file system del nuovo repository git
- <AUTHORFILE> indica un file contenente il mapping tra gli utenti CVS e il nome esteso + mail previsto da git. Ogni riga del file appare come

```
ncognome-3di=Nome Cognome <ncognome@3di.it>
```

- <CVSMODULE> indica il nome del modulo CVS (ovvero del progetto)

Utilizzando questi comandi, verrà prima inizializzato un repository vuoto (bare) all'interno di /path/to/new/repo/<CVSMODULE> e poi verrà importato tutto il codice presente nel module CVS <CVSMODULE> all'interno di /path/to/new/repo/<CVSMODULE>/.git.

Si riporta un esempio sul progetto mailArchiver. Il progetto mailArchiver è attualmente un modulo con lo stesso nome sotto la CVSROOT /opt/cvs/docway3.

```
git init --bare /opt/git/migrazione/docway3/mailArchiver  
git --bare cvsimport -d:local:/opt/cvs/docway3 -v -k -m -i -A /root/bin/cvs2git.AUTHORS -o master  
-C /opt/git/migrazione/docway3/mailArchiver mailArchiver
```

Al termine della migrazione è possibile clonare il repository git appena creato mediante comandi tipo:

```
git clone ssh://user@cvs.3di.it:/opt/git/migrazione/docway3/mailArchiver/
```

oppure, sfruttando l'esposizione tramite HTTP del repository¹⁾

```
git clone http://gittest.bo.priv/migrazione/docway3/mailArchiver/.git
```

Migrazione batch

È stato approntato uno script sul nostro server CVS **vegezio** che automatizza le operazioni necessarie alla migrazione di un singolo progetto (modulo in CVS) oppure di tutti i progetti (moduli) all'interno della stessa CVSROOT (per es., tutti i progetti sotto docway3, docway4, progetti ecc.).

Lo script è posto in /root/bin/cvs2git.sh e se ne riporta il codice:

[cvs2git.sh](#)

```
#!/bin/bash  
  
set -e  
  
function usage () {  
  
cat << EOF  
Usage:
```



```
cvs2git.sh -d CVSR00T [-m MODULE] -o DESTDIR
```

where

CVSR00T	path to the cvs root of a project (e.g. /opt/cvs/docway4)
MODULE	(Optional) module to import (e.g. code)
DESTDIR	destination directory for the new git repository. If no MODULE is set, every module found will be put inside this directory

EOF

```
}
```

```
# Conversione di un singolo modulo CVS in git.
```

```
# Argomenti:
```

```
# $1 CVSR00T
```

```
# $2 MODULE
```

```
# $3 DESTDIR
```

```
function convert_module () {
```

```
    mkdir -p "$3/$2"
```

```
    (cd "$3/$2" && git init --bare "$3/$2")
```

```
    git cvsimport -d :local:"$1" -C "$3/$2" -v -k -m -i -A /root/bin/cvs2git.AUTHORS -o  
master "$2"
```

```
}
```

```
# Conversione di tutti i moduli all'interno di una CVSR00T
```

```
# Argomenti:
```

```
# $1 CVSR00T
```

```
# $2 DESTDIR
```

```
function convert_all_modules () {
```

```
    for module in "$1"/*
```

```
    do
```

```
        if [ -d "$module" -a "$module" != "CVSR00T" ]
```

```
        then
```

```
            CVS_MODULE_NAME=${module##*/}
```

```
            convert_module "$1" "$CVS_MODULE_NAME" "$2"
```

```
        fi
```

```
    done
```

```
}
```

```
while getopts "d: m: o:" OPTION
```

```
do
```

```
    case "$OPTION" in
```

```
    d)
```

```
        SRC_CVSR00T="$OPTARG"
```

```
        ;;
```

```
    m)
```

```
        SRC_CVSMODULE="$OPTARG"
```

```
        ;;
```

```
    o)
```

```
        DESTDIR="$OPTARG"
```

```
        ;;
```

```
    esac
```

```
done
```

```
if [ x"$SRC_CVSR00T" == x ]
```

```
then
```

```
    echo "No CVSR00T specified. Exiting."
```

```
    usage
```

```
    exit 1
```

```
fi
```

```
if [ x"$DESTDIR" == x ]
```

```
then
```

```
    echo "No DESTDIR specified. Exiting."
```



```
usage
exit 2

fi

if [ x"$SRC_CVSMODULE" == x ]
then
    convert_all_modules "$SRC_CVSRROOT" "$DESTDIR"
else
    convert_module "$SRC_CVSRROOT" "$SRC_CVSMODULE" "$DESTDIR"
fi
```

Questo script utilizza un file di specifica degli “autori” su CVS, che effettua un mapping tra gli utenti registrati nei commit su CVS e gli utenti git. Il contenuto di tale file è il seguente:

```
gmgelli-3di=Gregorio Massimo Gelli <gmgelli@3di.it>
sstagni-3di=Simone Stagni <sstagni@3di.it>
writelli-3di=Walter Ritelli <writelli@3di.it>
rtirabassi-3di=Roberto Tirabassi <rtirabassi@3di.it>
lsopko-3di=Ladislav Sopko <lsopko@3di.it>
mmariotti-3di=Michele Mariotti <mmariotti@3di.it>
czappavigna-3di=Carmin Zappavigna <czappavigna@3di.it>
mballerini-3di=Massimiliano Ballerini <mballerini@3di.it>
fcappelli-3di=Francesco Cappelli <fcappelli@3di.it>
mstancikova-3di=Miroslava Stancikova <mstancikova@3di.it>
stancredi-3di=Stefano Tancredi <stancredi@3di.it>
mgelli-3di=Mauro Gelli <mgelli@3di.it>
rpierpaoli-3di=Roberto Pierpaoli <rpierpaoli@3di.it>
mbernardini-3di=Mirko Bernardini <mbernardini@3di.it>
vcapoccia-3di=Valerio Capoccia <vcapoccia@3di.it>
abarducci-3di=Andrea Barducci <abarducci@3di.it>
aalberghini-3di=Alan Alberghini <aalberghini@3di.it>
ganastasi-3di=Giosuè Anastasi <ganastasi@3di.it>
fgazzillo-3di=Fabio Gazzillo <fgazzillo@3di.it>
ncandelora-3di=Nicola Candelora <ncandelora@3di.it>
sdifelice-3di=Sarah Di Felice <sdifelice@3di.it>
spassarotto-3di=Siro Passarotto <spassarotto@3di.it>
dpranteda-3di=Domenico Pranteda <dpranteda@3di.it>
mciullo-3di=Marco Ciullo <mciullo@3di.it>
```

Migrazione da CVS a Gitlab

Per migrare i progetti CVS su Gitlab è necessario effettuare comunque il passaggio di conversione descritto nei punti precedenti. Una volta terminata la migrazione del progetto o dei progetti, è possibile importarli su Gitlab in diversi modi.

Importazione singolo progetto via HTTP

Questo tipo di importazione è indicato per singoli progetti. È necessario creare un nuovo progetto²⁾ ed indicare di importare un qualsiasi progetto git mediante la sua URL (ultimo bottone). La URL da indicare è composta nel seguente modo:

```
http://gittest.bo.priv/PATH/PROGETTO/.git
```

tenendo conto che PATH è già relativo al percorso /opt/git sul server vegezio.

Importazione singolo progetto mediante pull e push

Questo tipo di importazione (la più generica in assoluto) prevede l'utilizzo di una qualsiasi macchina client che faccia da ponte tra il repository migrato e Gitlab.

Per poter effettuare questo tipo di migrazione, è necessario che:

- il progetto sia stato migrato in un percorso raggiungibile dal client mediante ssh³⁾
- il progetto di destinazione sia già stato creato su Gitlab
- si sia caricata la propria chiave pubblica all'interno del proprio profilo su Gitlab

Le operazioni da effettuare sono le seguenti:

1. clonazione del repository migrato sulla propria macchina:



```
git clone ssh://user@cvs.3di.it/path/to/repo
```

2. rimozione del remote "origin" attuale (che punterà a git.3di.it):

```
git remote remove origin
```

3. impostazione del nuovo remote "origin" su Gitlab⁴⁾:

```
git remote add origin git://gitlab.bo.priv/group/project
```

Batch import

Questa procedura permette di importare in un colpo solo diversi repository git migrati da CVS. I passi da seguire sul server Gitlab sono i seguenti:

1. creazione del gruppo/i che conterranno i nuovi progetti:

```
sudo -u git mkdir /var/opt/gitlab/git-data/repositories/GRUPPO
```

dove GRUPPO rappresenta il gruppo/namespace all'interno del quale si vuole che siano posti i repository. Ovviamente si possono creare più gruppi in questa fase.

2. copia dei repository all'interno di /var/opt/gitlab/git-data/repositories/GRUPPO⁵⁾

```
cp -r /path/to/source/repos /var/opt/gitlab/git-data/repositories/GRUPPO
```

3. Aggiustamento del proprietario gruppo (devono essere git:git):

```
chown -R git:git /var/opt/gitlab/git-data/repositories/GRUPPO/
```

4. Aggiornamento DB dei repository ed effettiva importazione in Gitlab:

```
gitlab-rake gitlab:import:repos
```

A questo punto i repository importati dovrebbero comparire nell'interfaccia di Gitlab all'interno dei rispettivi gruppi.

Riferimenti esterni

[Guida base per git cvsimport](#)

[| man page di git cvsimport](#)

[| flag utili per git cvsimport](#)

[| Accesso HTTP all'attuale repository git su vegezio \(/opt/git\)](#)

[Importazione di repository "bare" in Gitlab](#)

¹⁾

effettuata mediante un reverse proxy su vegezio stesso

²⁾

magari già in un gruppo consono

³⁾

controllare i permessi relativi al gruppo gitusers su vegezio

⁴⁾

copiare la URL fornita da Gitlab stesso quando si visualizza il nuovo progetto

⁵⁾

o dei vari gruppi