



## 2<sup>η</sup> Σειρά Ασκήσεων

Αξία: 5% του τελικού σας βαθμού



### Bzr (Bazaar)

Το Bazaar (ή bzr) είναι ένα project ανοικτού κώδικα για τη δημιουργία ενός Distributed Version Control System. Με τον όρο Version Control System (VCS) εννοούμε ένα σύστημα το οποίο κρατάει ιστορικό των προηγούμενων εκδόσεων του πηγαίου κώδικα ενός project ή παρεμφερούς πληροφορίας (π.χ. txt αρχεία). Σκοπός του είναι να βοηθήσει στη ανάπτυξη λογισμικού τόσο σε ατομικό όσο και σε ομαδικό επίπεδο. Η έννοια του κατανεμημένου συστήματος, σημαίνει ότι δεν προϋποθέτει την ύπαρξη κάποιου κεντρικού server. Με τον τρόπο αυτό μπορούμε να έχουμε όλο το ιστορικό του κώδικα ενός project και να κάνουμε versioning του κώδικα τοπικά σε ένα υπολογιστή (ακόμα και σε ένα κοινό USB stick).

Περισσότερα μπορείτε να διαβάσετε στο <http://google.csd.uoc.gr/apache2-default/index.php/Bzr>

### Περιγραφή Άσκησης

Στην άσκηση αυτή καλείστε με βάση τις οδηγίες που δίδονται παρακάτω:

1. Να κάνετε login στο [google.csd.uoc.gr](http://google.csd.uoc.gr) χρησιμοποιώντας τα accounts που θα σας δοθούν και να φτιάξετε τις ρυθμίσεις για το bzr.
2. Να φτιάξετε ένα shared repository (exercise2) στην περιοχή σας και τα κατάλληλα branches (exercise2-mirror και addMyDetails) μέσα στο shared repository.
3. Στη συνέχεια στο branch addMyDetails θα φτιάξετε ένα directory με όνομα το login σας (αντί του test που χρησιμοποιείται στο παράδειγμα), το οποίο θα περιέχει τα αρχεία AM, NAME και EMAIL συμπληρωμένα με τις κατάλληλες πληροφορίες. Τέλος θα ενημερώσετε το bzr να κάνει versioning τα νέα αρχεία.
4. Έπειτα θα κάνετε commit τις αλλαγές σας στο branch addMyDetails.
5. Τέλος θα ενημερώσετε το κεντρικό branch (/spare/exercise2/exercise2-branch/) με τις αλλαγές σας. Φροντίστε να ακολουθήσετε με συνέπεια τις οδηγίες που περιγράφονται παρακάτω.

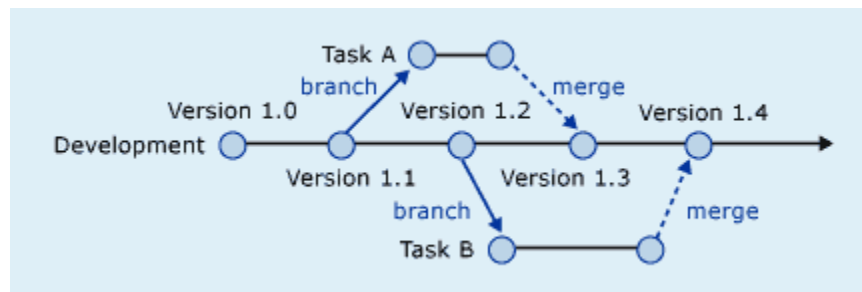
Φροντίστε τα branches που θα χρησιμοποιήσετε για την άσκηση αυτή, να παραμείνουν στην περιοχή σας. Εφόσον η άσκηση πραγματοποιηθεί επιτυχώς από όλους, στο τέλος το κεντρικό branch θα πρέπει να περιέχει τις πληροφορίες όλων των φοιτητών του μαθήματος. Για οποιαδήποτε απορία μη διστάσετε να στείλετε mail στη λίστα του μαθήματος.

## Ορολογία

- **Revision** – Μια εικόνα της κατάστασης των αρχείων και των καταλόγων με τα οποία δουλεύουμε, μία συγκεκριμένη έκδοσή τους.
- **Working tree** – Ο κατάλογος ο οποίος περιέχει τα αρχεία και τους υποκαταλόγους στους οποίους γίνεται διαχείριση εκδόσεων και που μπορεί ο χρήστης να τροποποιήσει. Ένα working tree συνδέεται με ένα branch.
- **Branch** – Ένα ταξινομημένο σύνολο revisions τα οποία περιγράφουν το ιστορικό ενός συνόλου αρχείων και καταλόγων. Τα branches μπορούν να χωρίστουν και στη συνέχεια να συνενωθούν (**merge**), δημιουργώντας ένα γράφο από revisions.
- **Repository** – ένα μέσο αποθήκευσης revisions. Στην απλή μορφή κάθε branch έχει το δικό του repository. Για εξοικονόμηση χώρου όμως υπάρχει και το **shared repository**, το οποίο μπορεί να αποθηκεύει πολλά branches.

## Παράδειγμα Ανάπτυξης με Χρήση VCS

Στην εικόνα φαίνεται ένα απλό παράδειγμα ενός project που χρησιμοποιεί ένα VCS. Ας θεωρήσουμε ότι έχουμε ένα branch που ονομάζεται Development, και βρίσκεται στην έκδοση 1.0. Έστω ότι γίνεται μία νέα αλλαγή, η οποία μας οδηγεί στην έκδοση 1.1. Εκείνη τη στιγμή εμείς αποφασίζουμε ότι θέλουμε να υλοποιήσουμε μία νέα λειτουργικότητα, έστω Task A. Οπότε φτιάχνουμε ένα νέο branch του Development, που το ονομάζουμε Task A και κάνουμε εκεί τις αλλαγές μας. Όσο εμείς εργαζόμαστε πάνω στο Task A, κάποιος άλλος κάνει μία νέα αλλαγή στο Development, και πλέον αυτό βρίσκεται στην έκδοση 1.2. Τώρα τα 2 branches, (Development και Task A), έχουν διαφορετικό ιστορικό. Εκείνη τη στιγμή κάποιος άλλος αποφασίζει να υλοποιήσει το task B, οπότε φτιάχνει και ένα νέο branch που ονομάζεται Task B. Έστω ότι εμείς τελειώνουμε τη δουλειά μας με το task A, και θέλουμε να την ενσωματώσουμε στο branch Development. Τότε πρέπει να γίνει η διαδικασία της συγχώνευσης (merging) αφού τα 2 branches έχουν διαφορετικό ιστορικό πλέον, και τελικά θα οδηγήσει σε μία νέα έκδοση του Development, την 1.3. Αντίστοιχα όταν τελειώσει η δουλειά στο Task B, η συγχώνευσή της στο Development θα οδηγήσει στην έκδοση 1.4.



## Σύνοψη bzd

Μια εντολή του bzd πρέπει να έχει την παρακάτω μορφή:

```
bzd command [ command_options ]  
bzd help  
bzd help command
```

Για παραπάνω πληροφορίες:

```
man bzc
```

## Settings

Αρχικά θα πρέπει να κάνετε login στο μηχάνημα google.csd.uoc.gr χρησιμοποιώντας το account και το password που θα σας δοθούν. Δημιουργούμε το directory “.bazaar” στην περιοχή μας και μέσα σε αυτό το αρχείο “bazaar.conf”, και εισάγουμε τα προσωπικά μας στοιχεία ανάλογα. Για το παράδειγμά μας, ας θεωρήσουμε το χρήστη Test Tester, με login test. Τα περιεχόμενα του αρχείου bazaar.conf θα μπορούσαν να είναι τα εξής:

```
email = Test Tester <test AT csd DOT uoc DOT gr>
editor = nano
mail_client = pine
```

Εκτελώντας πλέον την εντολή **whoami**

```
test@google:~/lala/exercise2$ bzc whoami
Test Tester <test AT csd DOT uoc DOT gr>
```

θα δείτε ότι πλέον το bzc γνωρίζει ποιος είστε!

## Αρχικοποίηση ενός bzc shared repository

Για να φτιάξουμε ένα **shared repository** που χρησιμοποιεί B+trees (1.9-rich-root) εκτελούμε την εντολή **init-repo** ως εξής:

```
test@google:~/lala$ bzc init-repo --1.9-rich-root exercise2
Shared repository with trees (format: 1.9-rich-root or development-wt5-rich-root)
Location:
  shared repository: exercise2
```

## Δημιουργία mirror και working branch

Για να δημιουργήσουμε τώρα ένα branch μέσα στο repository, με τα απαιτούμενα αρχεία για την άσκηση, αφού κάνουμε cd στο shared repository exercise2, εκτελούμε την εντολή **branch** ως εξής:

```
bzc branch /spare/exercise2/exercise2-branch/ exercise2-mirror
```

Πλέον έχουμε ένα νέο directory το οποίο ονομάζεται exercise2-mirror και το οποίο είναι ακριβές αντίγραφο του branch που υπάρχει στο /spare/test. **Αυτό το branch, δεν το πειράζουμε ποτέ.** Χρησιμοποιείται μόνο για να κατεβάζουμε τυχόν αλλαγές που υπάρχουν στο κεντρικό branch, και κατά τη διαδικασία όπου θέλουμε να ανεβάσουμε τις δικές μας αλλαγές στο κεντρικό branch, όπως θα περιγραφεί αργότερα.

Στη συνέχεια δημιουργούμε ένα νέο branch από το mirror, το οποίο είναι και το branch στο οποίο θα εργαστούμε και θα κάνουμε τις αλλαγές μας:

```
bzc branch exercise2-mirror addMyDetails
```

## Κάνοντας αλλαγές στο working branch

Για τις ανάγκες της άσκησης, πρέπει να δημιουργήσουμε ένα νέο directory με όνομα το login μας, στο

οποίο θα δημιουργήσουμε 3 αρχεία με ονόματα NAME, AM και EMAIL, και στα οποία θα εισάγουμε τα στοιχεία μας.

Το bzi μας παρέχει την εντολή **st** ή **status**, για να βλέπουμε το status του branch (π.χ αρχεία που μπορεί να έχουν γίνει modified, added ή removed) και την **diff** για να βλέπουμε τις αλλαγές που έχουν γίνει στα αρχεία και directories ενός branch, αντίστοιχα με την κλασσική diff εντολή του Unix.

π.χ. για ένα χρήστη test που έχει ακολουθήσει τα παραπάνω βήματα, έχουμε το εξής αποτέλεσμα:

```
test@google:~/exercise2/addMyDetails$ bzi st
unknown:
  test/
```

```
test@google:~/exercise2/addMyDetails$ bzi diff
```

Παρατηρήστε ότι το bzi μας λέει ότι δεν γνωρίζει το directory test που μόλις δημιουργήσαμε. Ο λόγος είναι ότι δεν του έχουμε πει ότι θέλουμε να κάνουμε versioning σε αυτό. Το bzi μας δίνει τις εντολές **add**, **rm**, **mv**, με τις οποίες μπορούμε να διαχειριστούμε τα αρχεία ενός branch. Οπότε για να κάνουμε versioning τα αρχεία που μόλις δημιουργήσαμε μέσα στο directory test, εκτελούμε:

```
test@google:~/exercise2/addMyDetails$ bzi add test/
adding test
adding test/AM
adding test/EMAIL
adding test/NAME
add completed
```

Η παραπάνω εντολή κάνει add αναδρομικά όλα τα αρχεία και directories που υπάρχουν μέσα στο directory test. Αν δεν τα θέλαμε όλα, θα έπρεπε να κάνουμε add κάθε ένα επιθυμητό αρχείο. Πλέον εκτελώντας τις εντολές st και diff έχουμε:

```
test@google:~/exercise2/addMyDetails$ bzi st
added:
  test/
  test/AM
  test/EMAIL
  test/NAME
```

```
test@google:~/exercise2/addMyDetails$ bzi diff
=== added directory 'test'
=== added file 'test/AM'
--- test/AM      1970-01-01 00:00:00 +0000
+++ test/AM      2009-02-25 11:16:10 +0000
@@ -0,0 +1,1 @@
+1293

=== added file 'test/EMAIL'
--- test/EMAIL   1970-01-01 00:00:00 +0000
+++ test/EMAIL   2009-02-25 11:16:19 +0000
@@ -0,0 +1,1 @@
+test@csd.uoc.gr

=== added file 'test/NAME'
--- test/NAME    1970-01-01 00:00:00 +0000
```

```
+++ test/NAME 2009-02-25 11:16:46 +0000
@@ -0,0 +1,1 @@
+Tester Test
```

Αν θέλουμε να αναιρέσουμε τυχόν αλλαγές που έχουμε κάνει στα αρχεία του branch, μπορούμε να εκτελέσουμε την εντολή `revert`. Η εντολή μπορεί να πάρει όρισμα τα αρχεία που θέλουμε να κάνουμε `revert` (να τα γυρίσουμε δηλαδή στην προηγούμενη συνεπή κατάσταση). Αν δεν ορίσουμε αρχεία τότε κάνει `revert` όλες τις αλλαγές στα versioned αρχεία.

Είμαστε έτοιμοι πια να πούμε στο `bzi` να αποθηκεύσει τις αλλαγές, να φτιάξει δηλαδή ένα νέο **revision**. Αυτό μπορεί να γίνει χρησιμοποιώντας την εντολή **commit**, όπου θα μας ζητηθεί να εισάγουμε ένα μήνυμα που περιγράφει τις αλλαγές που έχουμε κάνει π.χ. “Add my personal information”. Οπότε εκτελώντας:

```
test@google:~/exercise2/addMyDetails$ bzi commit
Committing to: /home/test/exercise2/addMyDetails/
added test
added test/AM
added test/EMAIL
added test/NAME
Committed revision 3.
```

Η εντολή `commit` μπορεί να πάρει σαν όρισμα συγκεκριμένα αρχεία που θέλουμε να κάνουμε `commit`. Αλλιώς κάνει `commit` όλα τα αρχεία των οποίων το status έχει αλλάξει. Σε περίπτωση κάποιου μη επιθυμητού `commit`, το `bzi` παρέχει και την εντολή **uncommit**, η οποία γυρνάει το history του branch, ένα revision πίσω.

### **Βλέποντας το history ενός branch**

Μπορούμε να δούμε το ιστορικό ενός branch χρησιμοποιώντας την εντολή **log**:

```
test@google:~/exercise2/addMyDetails$ bzi log
-----
revno: 3
committer: Test Tester <test AT csd DOT uoc DOT gr>
branch nick: addMyDetails
timestamp: Wed 2009-02-25 13:30:37 +0200
message:
  Add my personal information
-----
revno: 2
committer: Panagiotis Papadakos <papadako AT csd DOT uoc DOT gr>
branch nick: test
timestamp: Wed 2009-02-25 12:09:57 +0200
message:
  Rename
-----
revno: 1
committer: Panagiotis Papadakos <papadako AT csd DOT uoc DOT gr>
branch nick: test
timestamp: Wed 2009-02-25 11:56:07 +0200
message:
  Add file holding names
```

## Ενημερώνοντας το κεντρικό branch με τις αλλαγές μας

Προτού ανεβάσουμε τις αλλαγές μας στο κεντρικό branch έτσι ώστε να είναι διαθέσιμες σε όλους τους χρήστες, πρέπει να ενημερώσουμε το mirror branch (exercise2-mirror) με τυχόν αλλαγές που μπορεί να έχουν γίνει στο κεντρικό. Αυτό γίνεται χρησιμοποιώντας την εντολή **pull**. Έστω ότι έχει προστεθεί ένα νέο αρχείο lala στο κεντρικό branch. Τότε:

```
test@google:~/exercise2/exercise2-mirror$ bzip pull
Using saved parent location: /spare/exercise2/exercise2-branch/
+N lala
All changes applied successfully.
Now on revision 3.
```

Πλέον το mirror branch έχει ενημερωθεί με αυτή και επομένως πρέπει να ενημερώσουμε και το branch addMyDetails, το οποίο περιέχει τις δικές μας αλλαγές. Για να το κάνουμε αυτό, χρησιμοποιούμε την εντολή merge, μιας και τα 2 branches (exercise2-mirror και addMyDetails) έχουν διαφορετικό ιστορικό πια, ως εξής:

```
test@google:~/exercise2/addMyDetails$ bzip merge ../exercise2-mirror/
+N lala
All changes applied successfully.
```

Πλέον έχουν γίνει update τα αρχεία στο branch addMyDetail, και μένει να κάνουμε commit για να ενημερωθεί το history του branch. Για αυτό το commit συνήθως χρησιμοποιούμε ένα comment του στυλ “Merge upstream”.

```
test@google:~/exercise2/addMyDetails$ bzip commit .
Committing to: /home/test/exercise2/addMyDetails/
added lala
Committed revision 4.
```

Τρέχοντας πλέον την εντολή log, έχουμε το εξής output:

```
test@google:~/exercise2/addMyDetails$ bzip log|more
-----
revno: 4
committer: Test Tester <test AT csd DOT uoc DOT gr>
branch nick: addMyDetails
timestamp: Wed 2009-02-25 15:41:00 +0200
message:
  Merge upstream
-----
  revno: 2.1.1
  committer: Panagiotis Papadakos <papadako AT csd DOT uoc DOT gr>
  branch nick: test
  timestamp: Wed 2009-02-25 15:34:26 +0200
  message:
    Add new file
-----
revno: 3
committer: Test Tester <test AT csd DOT uoc DOT gr>
branch nick: addMyDetails
timestamp: Wed 2009-02-25 13:30:37 +0200
```

```

message:
  Add my personal information
-----
revno: 2
committer: Panagiotis Papadakos <papadako AT csd DOT uoc DOT gr>
branch nick: test
timestamp: Wed 2009-02-25 12:09:57 +0200
message:
  Rename
-----
revno: 1
committer: Panagiotis Papadakos <papadako AT csd DOT uoc DOT gr>
branch nick: test
timestamp: Wed 2009-02-25 11:56:07 +0200
message:
  Add file holding names

```

Έπειτα απομένει να κάνουμε merge και commit τις αλλαγές από το working branch στο mirror, έτσι ώστε στο τέλος να το κάνουμε **push** στο κεντρικό, να ανεβάσουμε δηλαδή τις αλλαγές μας στο κεντρικό. Για το commit αυτό χρησιμοποιούμε ένα μήνυμα του στυλ “Add my personal information”. Διαδοχικά έχουμε:

```

test@google:~/exercise2/exercise2-mirror$ bzip merge ../addMyDetails/
+N test/
+N test/AM
+N test/EMAIL
+N test/NAME
All changes applied successfully.

test@google:~/exercise2/exercise2-mirror$ bzip commit .
Committing to: /home/test/exercise2/exercise2-mirror/
added test
added test/AM
added test/EMAIL
added test/NAME
Committed revision 4.

test@google:~/exercise2/exercise2-mirror$ bzip push /spare/exercise2/exercise2-branch/
All changes applied successfully.
Pushed up to revision 4.

```

Τρέχοντας πλέον την εντολή log στο κεντρικό branch, βλέπουμε ότι το κεντρικό branch έχει γίνει update με τις αλλαγές μας:

```

test@google:~/exercise2/exercise2-mirror$ bzip log /spare/exercise2/exercise2-branch/
-----
revno: 4
committer: Test Tester <test AT csd DOT uoc DOT gr>
branch nick: exercise2-mirror
timestamp: Wed 2009-02-25 15:47:38 +0200
message:
  Add my personal information
-----

```

revno: 2.1.2  
committer: Test Tester <test AT csd DOT uoc DOT gr>  
branch nick: addMyDetails  
timestamp: Wed 2009-02-25 15:41:00 +0200  
message:  
Merge upstream

-----  
revno: 2.1.1  
committer: Test Tester <test AT csd DOT uoc DOT gr>  
branch nick: addMyDetails  
timestamp: Wed 2009-02-25 13:30:37 +0200  
message:  
Add my personal information

-----  
revno: 3  
committer: Panagiotis Papadacos <papadako AT csd DOT uoc DOT gr>  
branch nick: test  
timestamp: Wed 2009-02-25 15:34:26 +0200  
message:  
Add new file

-----  
revno: 2  
committer: Panagiotis Papadacos <papadako AT csd DOT uoc DOT gr>  
branch nick: test  
timestamp: Wed 2009-02-25 12:09:57 +0200  
message:  
Rename

-----  
revno: 1  
committer: Panagiotis Papadacos <papadako AT csd DOT uoc DOT gr>  
branch nick: test  
timestamp: Wed 2009-02-25 11:56:07 +0200  
message:  
Add file holding names





# Bazaar

## Quick Start Card

## More Information

`bzr help`  
<http://bazaar-fvcs.org>

### Initialization

```
New project  
bzr init myproject  
Existing project  
cd myproject  
bzr init  
bzr add .  
New branch  
bzr branch mp myproject  
New checkout  
bzr checkout mp myproject  
New "lightweight" checkout  
bzr checkout --lightweight mp \  
myproject
```

### File Manipulation

```
Add/"version" files  
bzr add foo.py  
bzr add bar/  
Remove/"unversion" files  
bzr remove --keep foo.py  
Remove and delete files  
bzr remove --force foo.py
```

### Version Control

```
Commit  
bzr commit foo.py -m "foo"  
bzr commit -m "the rest"  
Undo last commit  
bzr uncommit  
Revert changes  
bzr revert
```

### Merging

```
Merge two branches  
cd myproject  
bzr merge ../myproject-foo  
bzr commit  
Pull changes from myproject  
cd myproject-foo  
bzr pull ../myproject  
Update a checkout  
bzr update  
Auto-detect resolved conflicts  
bzr resolve  
Specify resolved conflict  
bzr resolve foo.py
```

### Information

```
Working tree status  
bzr status  
Revision log  
bzr log  
bzr log foo.py  
Working tree changes  
bzr diff  
foo.py changes  
bzr diff foo.py  
foo.py changes between  
revisions 1 and 3  
bzr diff -r1..3 foo.py  
Missing revisions  
bzr missing  
Branch information:  
bzr info  
Contents of foo.py at revision 3  
bzr cat -r3 foo.py
```

## Concepts

**Branch:** line of development for a project  
**Working tree:** version controlled directory  
**Repository:** store for Bazaar revisions  
**Revision:** version of the source code committed to the repository  
**Tag:** named revision  
**Related branches:** branches having a common ancestor  
**Merge:** the operation of applying to a branch all the changes introduced by another one

## Supported URL Prefixes

```
aftp:// Access using active FTP  
bzr:// fast access using the Bazaar smart server  
bzr+ssh:// fast access using the Bazaar smart server over SSH.  
file:// Access using the standard filesystem (default)  
ftp:// Access using passive FTP  
http:// Read-only access of branches exported on the web.  
https:// Read-only access of branches exported on the web  
using SSL.  
sftp:// Access using SFTP (most SSH servers provide SFTP).
```

### Publishing

```
Push revisions  
bzr push ../myproject-fo1  
Push revisions remotely  
bzr push sftp://host/myproject-fo1  
Email a merge directive  
bzr send  
Create a merge directive  
bzr send -o ../base.patch  
Export current revision as a  
directory  
bzr export ../myproject-dist/  
Export current revision as an  
archive  
bzr export ../myproject-dist.zip
```