

מדריך למתחיל ב-SVN

מאת: רועי מטרני roym@ee.technion.ac.il

המעבדה למערכות תכנה מרושתות

מבוא

לפני שמתחילים לקרוא את המדריך הזה, כדאי מאד לעבור על השקפים של הרצאת ההעשרה בנושא ניהול גרסאות תוכנה. וכן את המדריך לפתיחת פרוייקט ב-Redmine.

כפי שהוסבר במצגת, לרוב כלי הפיתוח, ובכלל זה כל כלי הפיתוח המובילים קיים תוסף של SVN Client. סביר להניח שתשתמשו בתוסף זה.

למשתמשי Windows קיימת אלטרנטיבה מצויינת בדמות TortoiseSVN. זהו ה-SVN Client המוביל והטוב ביותר. במדריך זה אנחנו נדגים שימוש בסיסי בעזרת TortoiseSVN, אולם הרוב המכריע של הכללים פה תופסים לכל SVN Client.

הורידו את התוכנה מכאן: <http://tortoisesvn.net/downloads> והתקינו אותה.

מדריכים נוספים

המדריך הזה מספק מידע על קצה המזלג כיצד להתחיל להשתמש במערכת. למי שיתעמק בנושא ויהפוך למשתמש קבוע בתוכנה כדאי מאד להיעזר במדריכים המקיפים יתר בשפה האנגלית ולהוסיף אותם לרשימת המועדפים.

המדריך של TortoiseSVN: http://tortoisesvn.net/docs/release/TortoiseSVN_en/index.html.

המדריך של SVN: <http://svnbook.red-bean.com/nightly/en/index.html>.

מונחים

תזכורת קצרה מאד לגבי המונחים במדריך:

Repository – ספריית המקור כולל כל היסטוריית הגרסאות, ששמורה בשרת ה-SVN.

Working Copy – עותק מקומי של ספריית המקור ששמורה במחשב מסוים.

Commit – פעולה שמבצעת הלכה למעשה את השינויים שביצענו ב-Working Copy אל תוך ה-Repository.

הקמת פרוייקט מבראשית

חלק זה מניח כי:

1. קיים לכם כבר פרוייקט ב-Redmine כולל SVN Repository ריק מקבצים. במידה ואין לכם, נא לפנות למהנדס המעבדה שלכם שידאג לנושא זה.

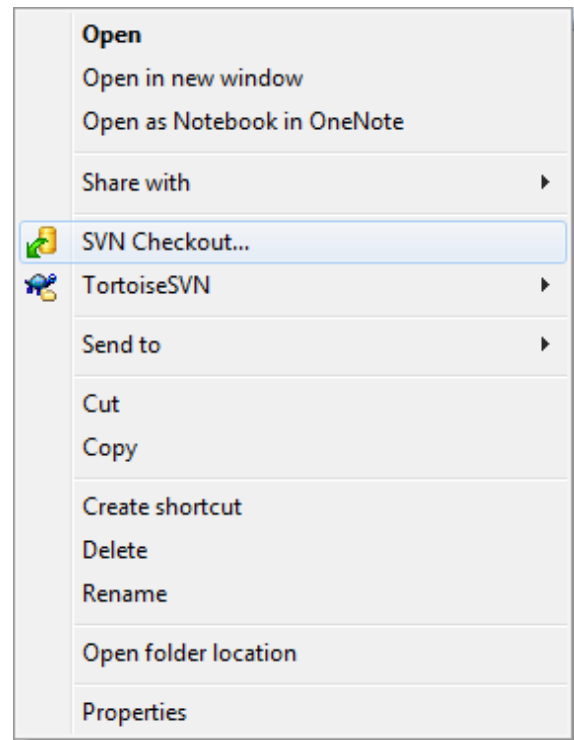
2. אין עדיין עץ פיתוח עם קבצים. אם כבר יש לכם עץ פיתוח, עברו לסעיף [הכנסת פרוייקט קיים ל-SVN](#).

יצירת Working Copy

כל תיקייה חדשה וכל קובץ חדש, ובכלל זה תיקיית ה-Root והקובץ הראשון בפרוייקט, נוצרים קודם מקומית במחשב של אחד המפתחים. לכן הצעד הראשון בפרוייקט חדש הוא יצירת תיקיית הפרוייקט אצלכם במחשב, והגדרת התיקיה החדשה בתור ה-Root Folder של הפרוייקט שלכם.

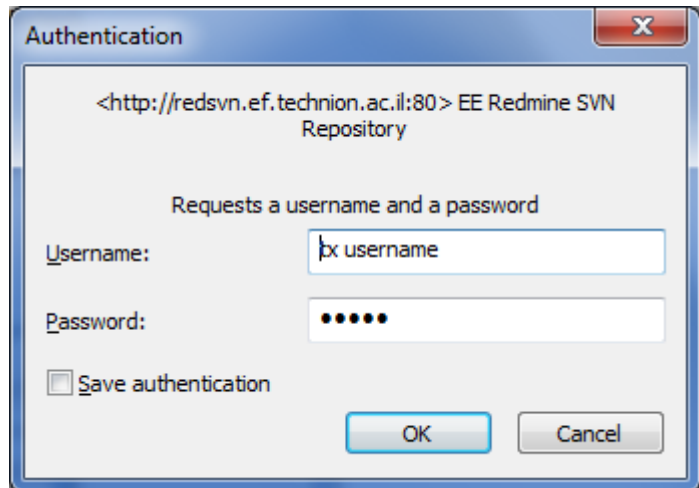
צרו תיקייה חדשה עבור הפרוייקט ותנו לה שם (למשל MyProject בדוגמא למטה).

בחרו עם העכבר את התיקיה החדשה. הקלקה ימנית על העכבר תפתח תפריט שמכיל פקודות של Tortoise.



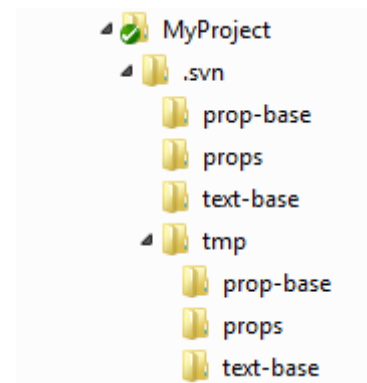
בחרו SVN Checkout... עקרונית הפעולה הזו נועדה לשפוך תוכן של פרוייקט מתוך ה-SVN לתוך תיקייה מסויימת במחשב שלכם (ראו סעיף...), אך מכיוון שה-Repository כרגע ריק, כל מה שהפעולה הזו עושה זה להגדיר את התיקיה כשורש של ה-Working Copy עבור הפרוייקט.

בחלון ה-Authentication הכניסו את שם המשתמש והסיסמא



לתיקייה שייצרתם נוסף סימן ירוק, ובתוכו נוספה תיקייה נוספת בשם (.svn). התיקייה הזו היא לשימוש התכנה ואתם לעולם לא תצטרכו לעיין בה או לשנות אותה.

אחרי שסיימתם עם הפעולה מצב התיקיות שלכם יהיה כך:



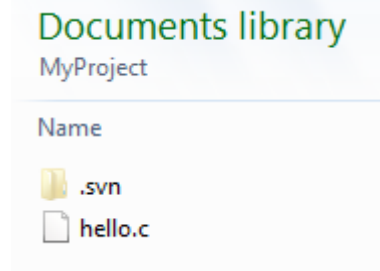
הוספת קובץ

את הקובץ מוסיפים קודם בספריה המקומית שלכם, כלומר ב-Working Copy. נניח שאנחנו מוסיפים את הקובץ .hello.c

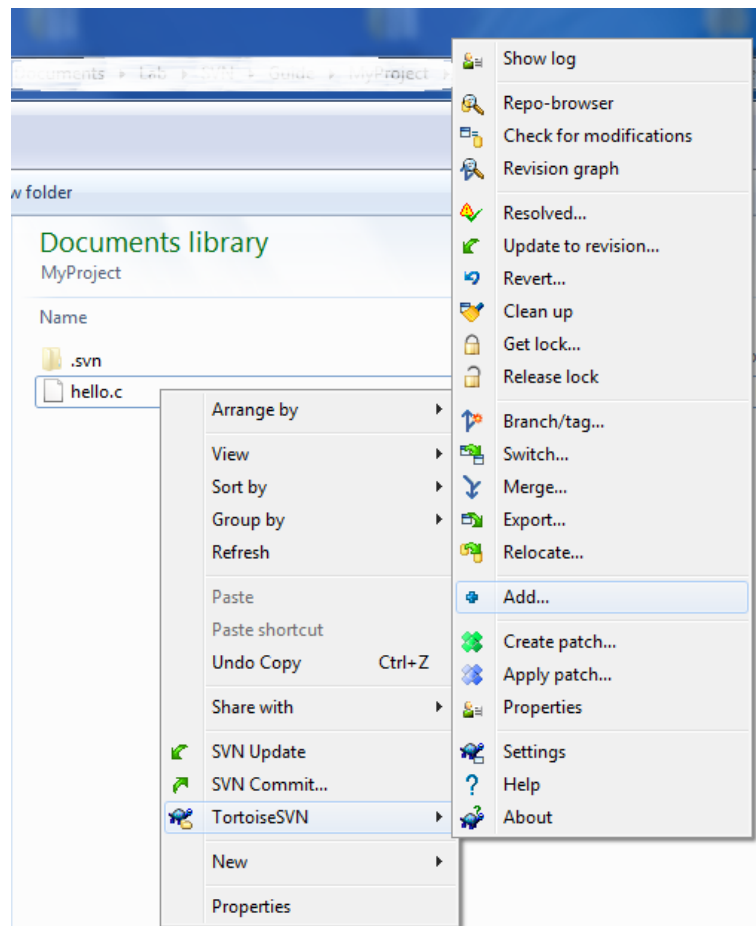
שלב א' – עריכה מקומית של הקובץ.

קודם כל נערוך קובץ חדש ונשמור אותו תחת הספריה MyProject.

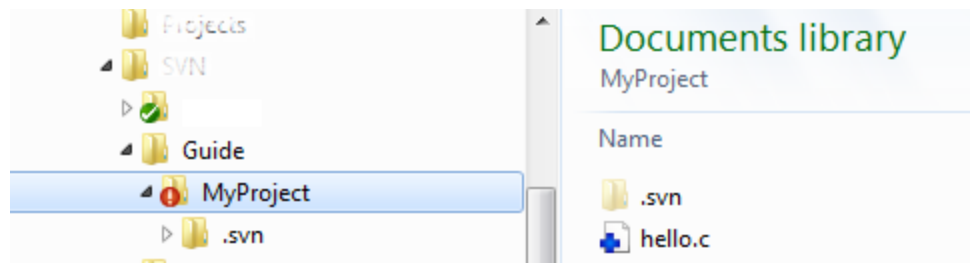
תיקיית הפרוייקט תיראה כך:



שלב ב' – פעולת Add
אחרי שהקובץ שמור, נסמן אותו כמועמד להוספה.



תיקיית הפרוייקט תיראה כך:

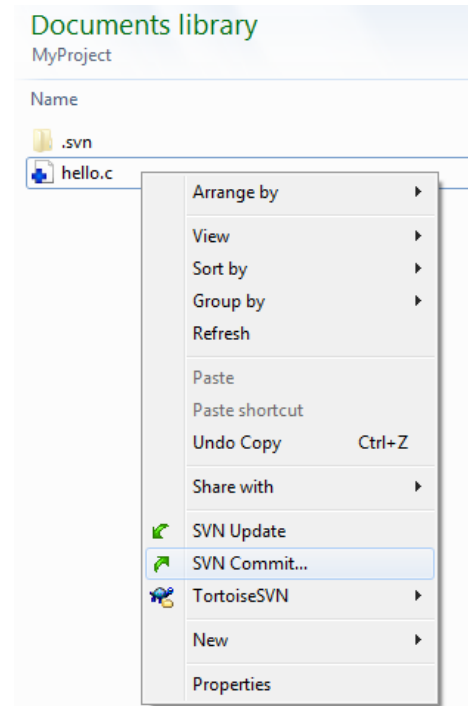


שימו לב לשינויים הבאים:

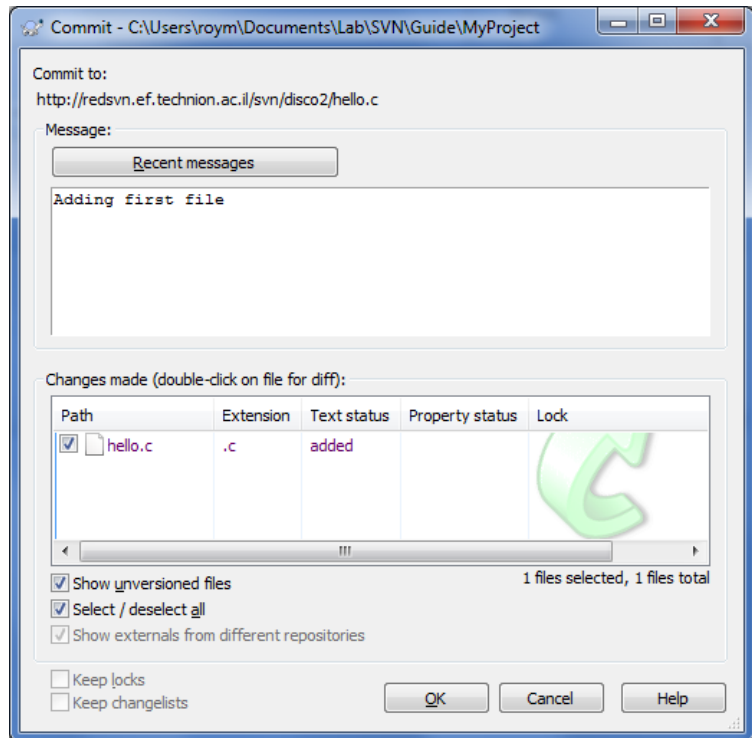
על הקובץ עצמו מסומן פלוס כחול, כלומר הקובץ מסומן להוספה אבל עדיין לא נמצא ב-Repository. ספריית הפרוייקט מסומנת באדום. המשמעות של זה היא שנעשו שינויים ב-Working Copy שעדיין לא נמצאים ב-Repository, כלומר לא נעשה עליהם Commit.

שלב ג' – פעולת Commit

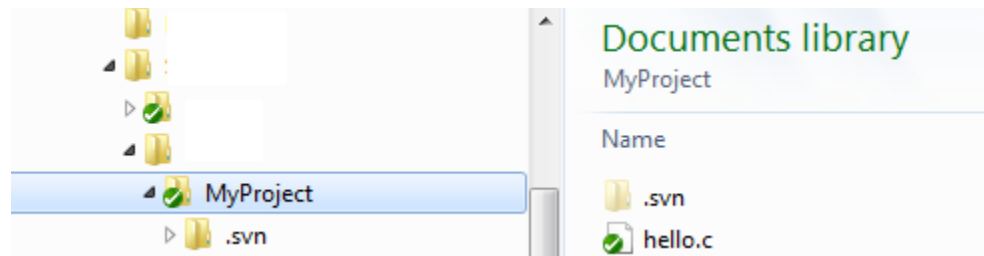
כאשר אנחנו בטוחים שהקובץ עובד ללא תקלות, נבצע פעולת Commit



נפתח החלון הבא:



כדאי לרשום הערה משמעותית בחלון ההערות וללחוץ OK. הפעולה הזו תשמור את הקובץ ב-Repository. תיקיית הפרוייקט תיראה כך:



תיקיית הפרוייקט הפכה לירוקה, וכך גם הקובץ החדש שהוספנו. עכשיו, כאשר הקובץ נמצא כבר בשרת SVN, ניתן לראות אותו גם דרך ה-Redmine.

Browser window showing the repository page for 'discovery2' at redsvn.ef.technion.ac.il/projects/disco2/repository. The page is titled 'discovery2' and shows the 'Repository' tab selected. The current path is 'root'. A table lists the file 'hello.c' with a size of 83 Bytes, revision 1, and author User1 LabAdmin. Below this, a 'Latest revisions' table shows a single revision (1) dated 10/07/2010 02:21 pm, also by User1 LabAdmin, with the comment 'Adding first file'. The page also includes navigation links like 'Home', 'My page', 'Projects', 'Help', and 'Sign out', and a search bar.

Home My page Projects Help

Logged in as u22233345 My account Sign out


discovery2

Search: discovery2

Overview Activity Issues New issue News Documents Wiki Files **Repository** Settings

root


Statistics | Revision:

Name	Size	Revision	Age	Author	Comment
 hello.c	83 Bytes	1	about 2 hours	User1 LabAdmin	Adding first file

Latest revisions

#	Date	Author	Comment
1	10/07/2010 02:21 pm	User1 LabAdmin	Adding first file

[View all revisions](#)

Also available in:  Atom

הכנסת פרוייקט קיים ל-SVN

קיימים 2 דרכים להכניס פרוייקט קיים לתוך SVN. הדרך המקובלת היא עם פקודת **Import**, אבל זו דרך בעייתית מבחינות מסוימות, ולכן נשתמש בדרך השנייה, שנקראת **Import In Place**. דרך זו דומה למדי להכנסת פרוייקט חדש.

בכל מקרה לפני שאתם מתחילים מומלץ לגבות את העבודה שלכם בקובץ ZIP במידה ומשהו משתבש.

הנחות עבודה:

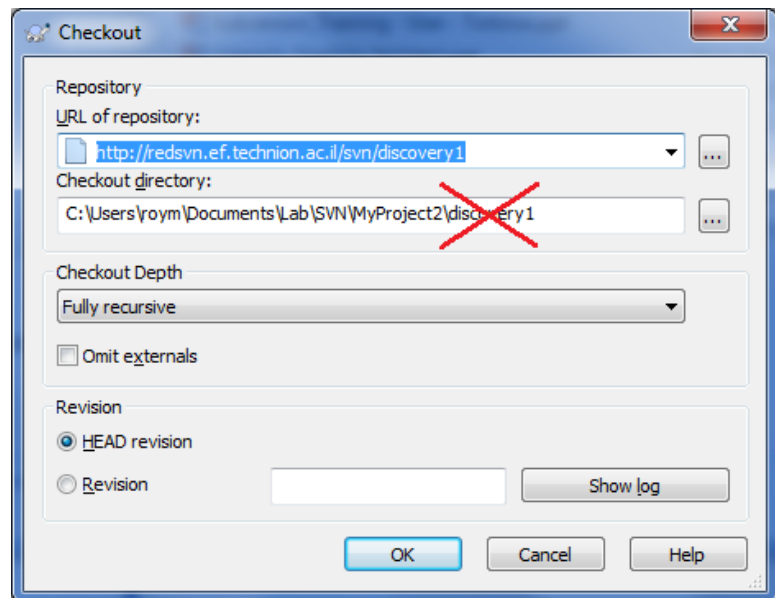
- יש עץ קבצים אחד שאותו רוצים להכניס ל-Repository
- ה-Repository כרגע ריק
- שם הפרוייקט במחשב שונה מהשם של ה-repository

לצורך הדוגמא, שם הפרוייקט הוא MyProject2 ושם ה-Repository הוא discovery1.

שלב א' – קישור בין מערכת הקבצים לבין ה-Repository

קליק ימני על תיקיית השורש של הפרוייקט (MyProject2 במקרה הזה) ובחרו בתפריט **SVN Checkout...**.

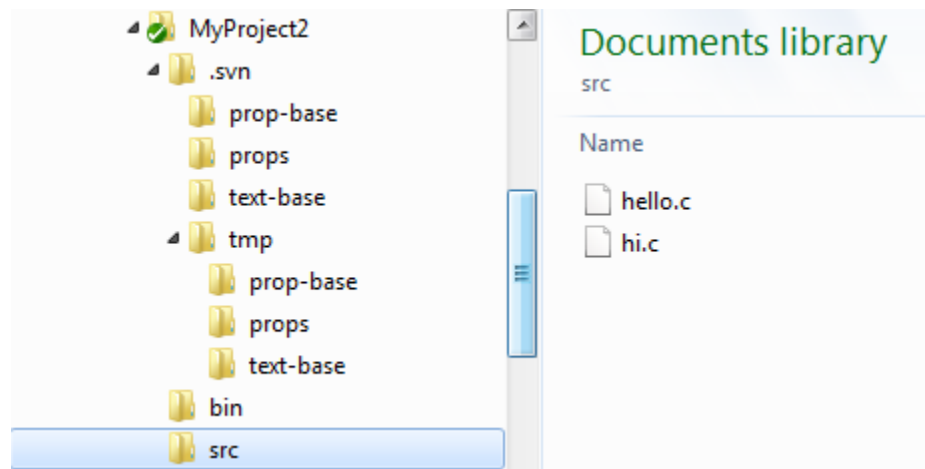
יופיע החלון הבא. שימו לב שהחלון מציע לכם ליצור תיקייה עם שם ה-Repository מתחת לתיקיית הפרוייקט. מחקו אותו והישארו עם התיקייה העליונה.



אחרי שלחצתם OK מופיעה הודעת אזהרה. תתעלמו ממנה.

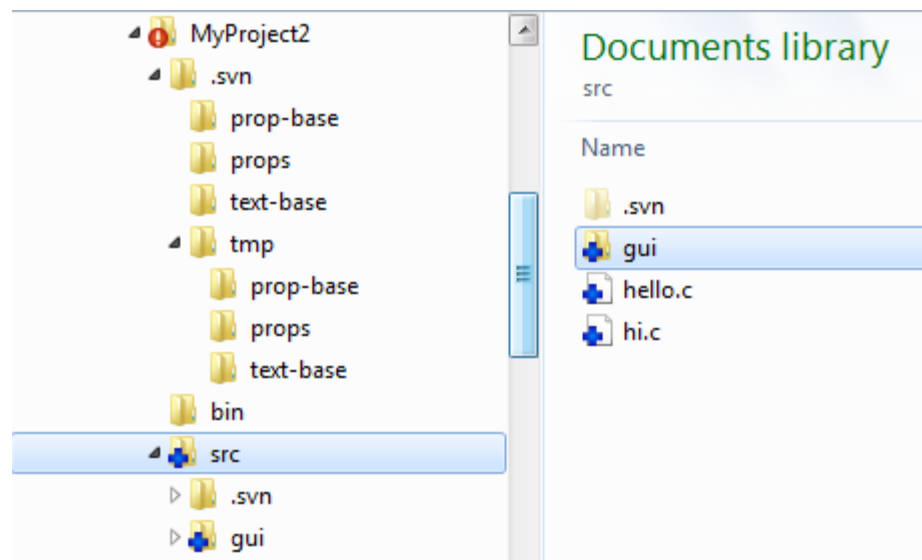
מכיוון שה-Repository היה ריק למעשה לא נעשה checkout לאף קובץ.

מבנה הקבצים שלכם נראה כך:



שלב ב' – הוספת הקבצים הרלוונטים

בסיפריה .svn. כרגיל לא נוגעים. תיקיית bin מכילה תוצרי קומפילציה וגם אותה לא נכניס. תיקיית src היא היחידה שמעניינת אותנו. עמדו על src, קליק ימני ובחרו TortoiseSVN->Add... בחרו בקבצים ותיקיות אותם אתם רוצים להכניס ולחצו OK. בתום השלב הזה הקבצים מסומנים עם + כחול. משמעות הדבר שהם מועמדים להוספה אבל עדיין לא בתוך ה-Repository.



בפרוייקט שלכם כמובן יכולות להיות כמה תיקיות וגם קבצים בתיקה העליונה. אתם צריכים להוסיף את כולם באותו האופן.

הכנסת פרוייקט קיים יכולה להיות לפעמים מורכבת. מכיוון שעץ הפרוייקט מכיל תיקיות וקבצים שלא צריכים להיכלל ב-Repository. כדאי להיות סבלנים ולבחור היטב מה מכניסים ומה לא. הדבר יכול לחסוך הרבה זמן מיותר בהמשך.

שלב ג' – Commit

פעולת ה-Commit תשלים את פעולת ההוספה ותעביר את הקבצים ל-Repository בשרת ה-SVN. קליק ימני על ראש הפרוייקט (בדוגמא MyProject2) ובחרו **SVN Commit...**. הכניסו הערה אינטליגנטית (למשל First revision in OK) ולחצו OK.

שינויים בפרוייקט קיים

בחלק הזה נעבור על פעולות בסיסיות, כגון שינוי, הזזה ומחיקה קבצים.

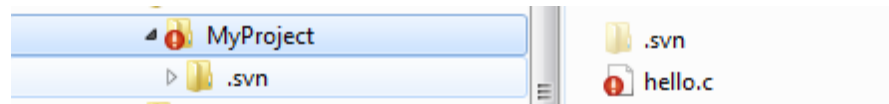
שינויים בקובץ קיים

כאשר רוצים לבצע שינוי בקובץ קיים, עורכים את השינוי אצלכם במחשב, ולאחר וידוא שהשינוי עובד טוב, מעדכנים את ה-Repository.

שינוי בגוף הקובץ

קובץ שעבר עריכה משנה את צבעו מירוק לאדום. גם התיקייה שמכילה אותו מאדימה, וכך הלאה בהיררכיה עד לשורש הפרוייקט.

לדוגמא:



אם אתם לא זוכרים מה השינויים שנעשו, אפשר לבחור את פקודת diff מתוך התפריט. השינויים שיוצגו הם בין הגירסא של ה-working copy אצלכם במחשב ובין הגירסא ממנה עשיתם checkout.

בפעם הבאה שתעשו SVN Commit... השינויים שביצעתם בקובץ יישמרו ב-Repository.

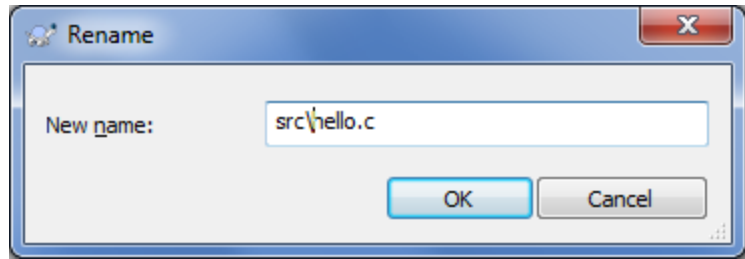
הערה - ניתן ורצוי לשנות את כל הקבצים שקשורים לאותה משימה לפני שמבצעים commit. אם לא תעשו כך, מישהו עלול לקחת את השינויים שלכם ולקבל מערכת קבצים שלא עוברת קומפילציה.

שינוי שם והעברת קובץ

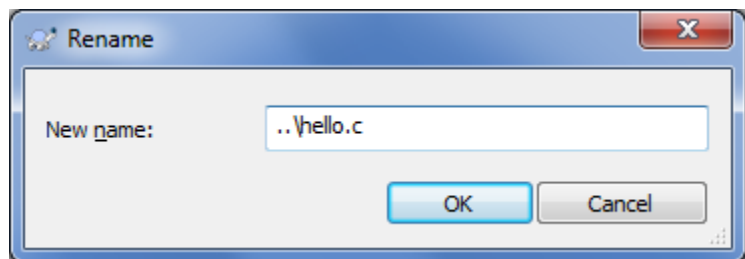
העברת קובץ או שינוי שם קובץ אשר מנוהל ב-SVN חייבים להיעשות ע"י פקודת SVN. אם לא תעשו כך, הקובץ ב-Repository יישאר במיקום או בשם הקודמים שלו, ותיכנסו למצב של חוסר עקביות בעץ הקבצים שלכם.

כל השינויים מהסוג הזה נעשים ע"י פקודת TortoiseSVN->Rename מתוך התפריט. במקרה שמשנים מיקום קובץ, יש להכניס את ה-Path היחסי מהמיקום הנוכחי של הקובץ.

למשל אם נעביר את הקובץ hello.c לתיקייה src נרשום את הפקודה הבאה:



ואם נרצה להחזיר אותו ספרייה אחת למעלה נרשום את הפקודה הבאה:



השינוי שביצעתם יכנס ל-Repository רק אחרי שתבצעו Commit.

הערה – הפעולה הזו מבצעת למעשה Delete ו-Add לקובץ, אבל ההיסטוריה שלו נשמרת.

עוד הערה – במקרה של הזזת קובץ אפשר לגרור את הקובץ עם הקלקה ימנית ולבחור באפשרות המתאימה.

מחיקת קובץ

מחיקת קובץ מתבצעת ע"י פעולת TortoiseSVN->Delete. גם כאן הפעולה תתפוס רק אחרי Commit.

עבודה בצוות

קיימות מספר גישות לעבוד בצוות. כל גישה טובה לסביבה אחרת, כתלות בסוגי הקבצים, אורך המשימות, גודל הפרוייקט ומספר אנשי הצוות.

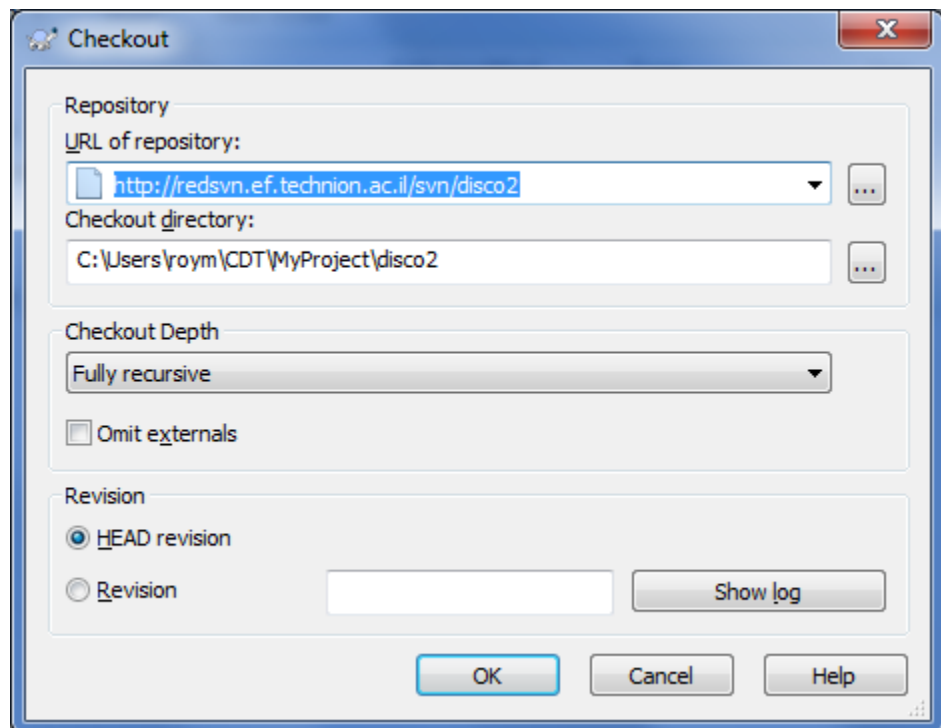
לצורך פרוייקט בתואר ראשון, מספיקה הגישה הפשטנית יותר, שבה אין branches או נעילות. המודל הזה נקרא Copy-Modify-Merge וניתן לקרוא עליו כאן: <http://svnbook.red-bean.com/en/1.0/ch02s02.html>.

הורדת פרוייקט קיים למחשב

צריך לזכור שתפקיד ה-SVN הוא ניהול קבצי מקור. זה אומר שאם אתם מתחברים לפרוייקט שכבר קיים, האחריות על הכנת הסביבה לפרוייקט היא עליכם וללא קשר לעבודה עם SVN.

אם אתם צריכים להוריד למחשב שלכם עותק של פרוייקט קיים (למשל השותף שלכם התחיל אותו, או שקיבלתם בירושה פרוייקט קודם), יש לבצע את הפעולות הבאות:

1. צרו תיקייה חדשה שתשמש כ-Root (למשל MyProject).
2. קליק ימני על התיקייה ובחרו בפקודת SVN Checkout...
3. שם הלינק ל-Repository הוא כפי שהוסבר במדריך ה-Redmine
4. אין צורך לשנות את שם התיקייה או את הערכים האחרים בחלון



ביצוע Commit במצב ללא קונפליקט

ביצוע Commit בעבודה בצוות מחייב update לפני ה-Commit, על מנת שלא יקרה מצב שבו דרסנו שינויים שמפתח אחר עשה.

דוגמא:

הקובץ hi.c שהיה בגרסא 9, שונה ע"י השותף (התווספה שורה 3). השותף ביצע Commit והקובץ עלה ל-Repository כגרסא 10. באותו הזמן אני שיניתי גם את הקובץ אבל במקום אחר (שורה 4).

הנה מפת הקבצים שנוצרה:

גרסא 9

```
1 int main(int argc, char *argv[]) {
2 // adding a debug print
3 printf("Hi and welcome\n");
4 return 1;
5 }
```

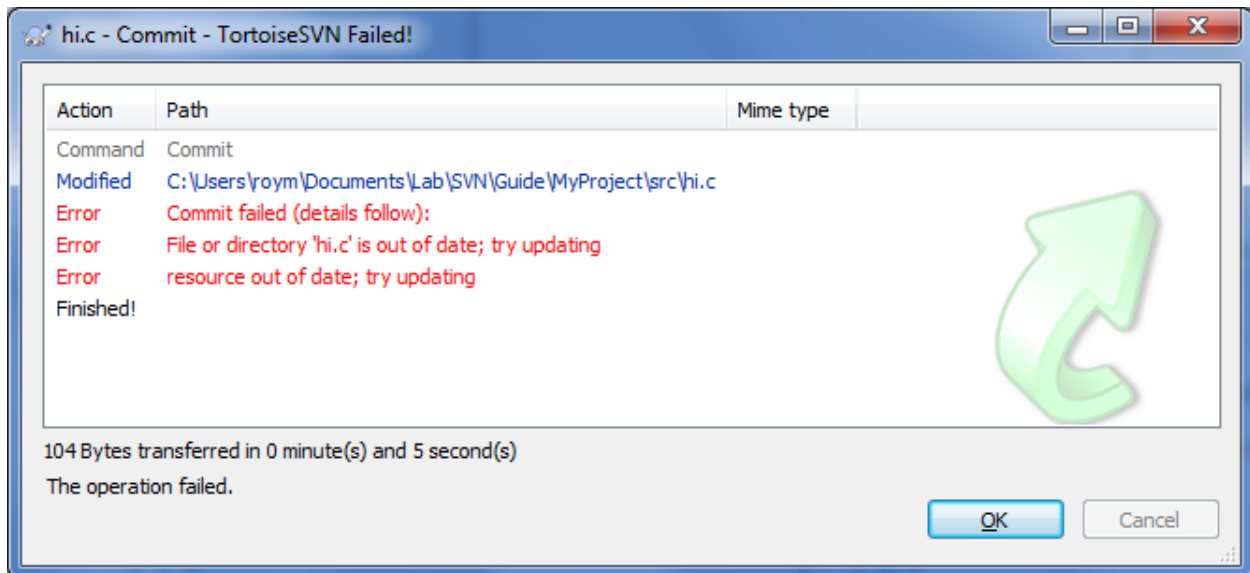
גרסת ה- Working copy

```
1 int main(int argc, char *argv[]) {
2 // adding a debug print
3 printf("Start\n");
4 printf("Hi and welcome\n");
5 return 1;
6 }
```

גרסא 10

```
1 int main(int argc, char *argv[]) {
2 // adding a debug print
3 printf("Hi and welcome\n");
4 printf("How are you? \n");
5 return 1;
6 }
```

אם אנסה לבצע Commit על הקובץ שלי ללא update, ה-SVN מזהה שהקובץ שלי, שיצא מתוך גרסא 9, מנסה לעדכן קובץ שכבר בגרסא 10 ויוציא את הודעת השגיאה הבאה:



במידה ואבצע update, אקבל הודעה שהקובץ מוזג אוטומטית.

הקובץ הממוזג נראה כך:

גרסא 11

```
1 int main(int argc, char *argv[]) {
2 // adding a debug print
3 printf("Start\n");
4 printf("Hi and welcome\n");
5 printf("How are you? \n");
6 return 1;
7 }
```

קונפליקט בין קבצים

כאשר אתם עושים update לקובץ, וה-SVN לא יודע כיצד למזג שינויים שלכם עם שינויים שהתבצעו ע"י איש צוות אחר, נוצר קונפליקט.

נסתכל על הדוגמא הבאה:

הקובץ hello.c שהיה בגרסא 6, שונה ע"י השותף ועלה ל repository כגרסא 7. במקביל אני שיניתי אותו באותה שורה (במקרה זה שנינו שינינו את שורה 4). כאשר עשיתי את פעולת ה-update לפני commit, ה-TortoiseSVN התריע על קונפליקט.

גרסא 6

```
1 int main(int argc, char *argv[]) {
2 // adding a debug print
3 printf("Hello and welcome\n");
4 return 1;
5 }
```

גרסת ה- Working copy

```
1 int main(int argc, char *argv[]) {
2 // adding a debug print
3 printf("Hello and welcome\n");
4 printf("It was nice meeting you\n");
5 return 1;
6 }
```

גרסא 7 (נוסף ע"י השותף)






```
1 int main(int argc, char *argv[]) {
2 // adding a debug print
3 printf("Hello and welcome\n");
4 printf("I am here\n");
5 return 1;
6 }
```

קובץ שנמצא בקונפליקט מסומן ע"י משולש צהוב, ונוצרים עבורו כמה קבצים זמניים שעוזים לפתור את הקונפליקט.

Documents library

src

Name

-  .svn
-  hello.c
-  hello.c.mine
-  hello.c.r6
-  hello.c.r7

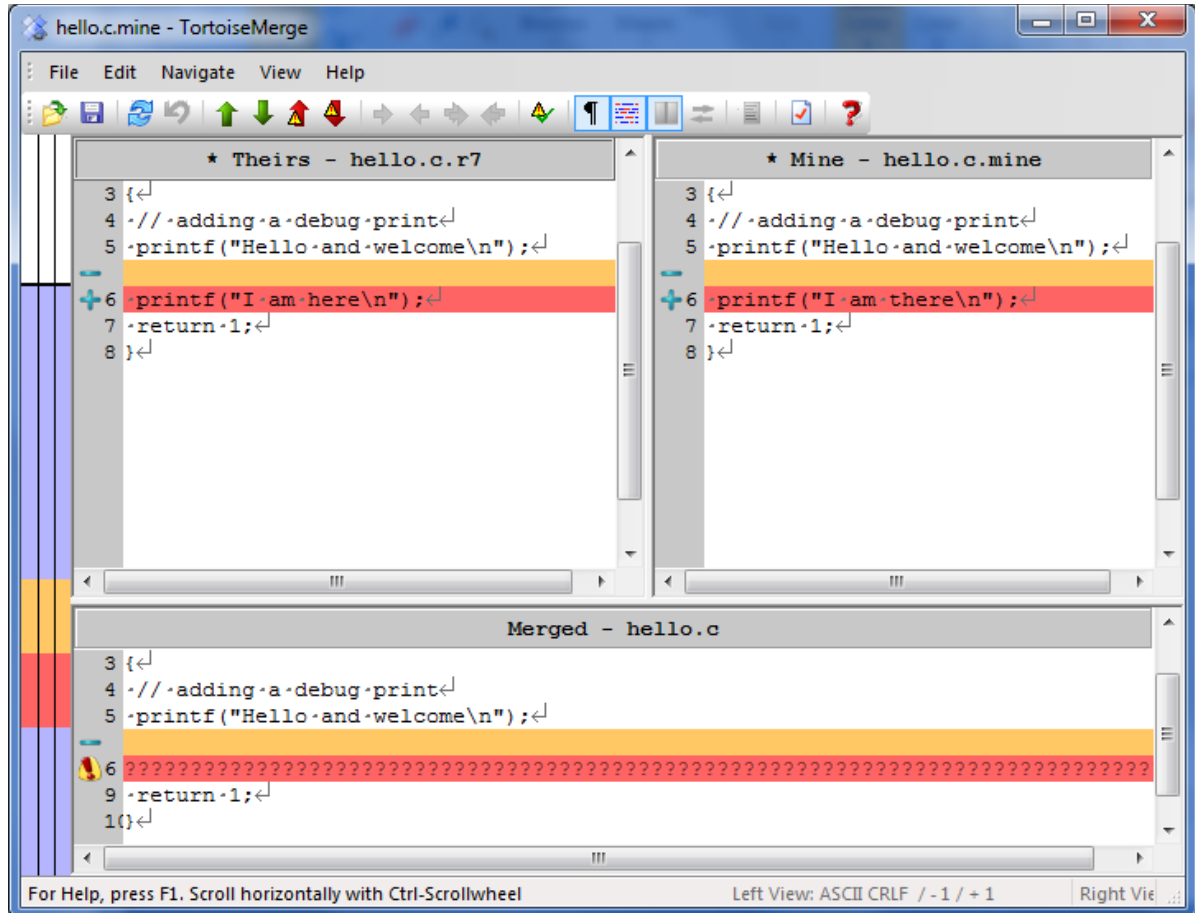
- r6 הוא הגרסא האחרונה המשותפת ל-2 הקבצים

r7 - הוא הגרסה החדשה ביותר ב-repository

mine - הוא הקובץ כפי ששונה ב-working copy

על מנת לפתור את הקונפליקט, יש לבחור בתפריט TortoiseSVN->Edit Conflicts

החלון הבא נפתח:



החלון השמאלי עליון מראה את הקובץ שקיים ב-Repository

החלון הימני עליון מראה את העותק המקומי של הקובץ

החלון התחתון מראה את תוצאת המיזוג בין שני הקבצים. הבלוקים שבהם יש קונפליקט צבועים באדום עם סימני שאלה. אפשר בעזרת תפריט ימני ליישב את הקונפליקט בכמה דרכים. להכניס בלוק אחד ואחריו את השני, או להתעלם מאחד מהשינויים, ולהישאר רק עם השני. אפשר גם לערוך את הקובץ הממוזג במידה וצריך.

אחרי שיישבו את כל הקונפליקטים, יש לשמור את הקובץ, ולבחור מתוך התפריט TortoiseSVN->Resolved. ב-Commit הבא הקובץ הממוזג ייכנס ל-Repository.

במקרה כזה כדאי למהר עם ה-Commit לפני שהשותף יבצע עוד שינוי בקובץ ותצטרכו לעשות את העבודה הזו שוב. הדרך הטובה ביותר היא לדחות פעולות כאלו של מיזוג עד לשלב שבו התכוונתם ממילא לעשות Commit.