SVN tutorial

Chris Pu CSCE-315

What's SVN

- SVN is a development tool for keeping track of all your changes in code.
 - SVN will be used for all of your team projects in this semester
 - Other similar programs: clearcase, CVS...
 - Our department has a SVN **Repository** server

How to use SVN

- TA/Instructor create the repository for a project:
 - https://svn.cse.tamu.edu/[repository name]/[subtree]
- To use SVN in Visual studio, students need to install SVN client on their Windows
 - SVN client: <u>VisualSVN</u> and <u>TortoiseSVN</u> in combination with Visual Studio
 - For linux user, you can use install a SVN client with GPL, and use SVN command line

• First download <u>TortoiseSVN</u> (64-bit)

<u>http://tortoisesvn.net/downloads.html</u>

- And install it.
- Then download VisualSVN and install it
 - <u>http://www.visualsvn.com/visualsvn/download/</u>

Now, open your visual studio, you will see VisualSVN on Visual studio's menu bar.

File Edit View Project Build E	ebug Team Data Tools	VisualSVN Test Window Help		
i 🎁 • 🛅 • 📂 🛃 🥔 🐰 🛍 😫	19-C-J-11	5 Show Changes	· 1009	- 🔍 🕾 🖬 🛠 📾 🌾 🛃 🛍 🗆 - 🖕
🖪 🗞 🖕 🔺 👘 👍 🚍 '	200000000	🕼 Update		
Solution Explorer 🔹 🛙 🗙	minute X	Pt Schmitter		-
	miaes.c X	≦⊫ Show Log		
	(Global Scope)	Disk-Browser	•	*
⊿ Source Files	⊡/*	Dana Province		ŧ
Ct mraes.c	* Implementation	repo-browser	rd	A
er mralloc.c	* Could/Should be	Create Patch	me critical loops in	
e mrarth0.c	* aes_ecb_encrypt	Apply Patch	no oraciona roopo in	
mrarth1.c	*	P Branch		
mrarth2.c	* Note that the 1	Sill Switch	! Nonetheless	
miartris.c	* we are jumping	- Switchin	of the favourites	
mibits.c	* KIJNDAEL, part.	V Merge		
en mibricke	* We were right!	Revert Changes	2000	
mrcore.c	*	Release Lock		
CH mrcrt.c	* Copyright (c) :	update To Revision		
en mrcurve.c	*/			
C mrdouble.c	#211-211-31 (1-1-21-21-1)	Add Solution to Subversion		
mrebrick.c	#include <stallb f<="" th=""><th>Get Solution from Subversion</th><th></th><th></th></stallb>	Get Solution from Subversion		
🕶 mrec2m.c	#Include allacin	Upgrade Working Copy		
🕶 mrecn2.c	#define MR_WORD mi	Set Working Copy Root		
🕶 mrfast.c		Cleanup Working Copy		
🕶 mrflash.c	/* this is fixed *	a the second second		
🚰 mrflsh1.c	#define NB 4	Registration		
🥶 mrflsh2.c 🛛 ≡	/* rotates x one 1	About VisualSVN		
mrflsh3.c				
Ci mrflsh4.c	#define ROTL(x) (((x)>>7) ((x)<<1))			
C ¹ mrfrnd.c				
C ¹ mrgcd.c	/* Rotates 32-bit word left by 1, 2 or 3 byte */			
mrgt2m.c	#define ROIL8(x) (((x)<(8) ((x)>>24))			
mrio1.c	#define ROTL16(x) (((x)<16) ((x)>16))			
miloz.c	#define ROTL24(x) (((x)<<24) ((x)>>8))			
mijack.c				
en mmonty c	static const MR_BYTE InCo[4]={0xB, 0xD, 0x9, 0xE}: /* Inverse Coefficients */			
en mmuldy.c static const MR BVTF ntab.[]=				
C mrpi.c	[1, 3, 5, 15, 17, 51, 82, 55, 26, 46, 114, 150, 161, 248, 19, 53,			
€1 mrpower.c 100 % + 4				

- Clicck 'Add Solution to Subversion',
- a popup will come up, check 'I will set working copy root manually' and click 'Next'
- Select the correct folder that your project is in, then click 'Next'
- Choose 'Exisiting Repository' and click 'Next'
- In the 'Destination URL:' box, type in the project location url from above (Here use https://svn.cse.tamu.edu/[repository name]/[subtree])
- It will prompt you for username/password.
- After verification, it will ask you if you want to import your new files (For the first time, the repository is empty) click 'Import' this may take a bit
- And... your done

SVN commands

- Concept: Check In, Check Out, Commit
 - Check In: also known as **Import**
 - First, create a directory for it in your repository.
 - \$ svn mkdir https://svn.cse.tamu.edu/[repository name]/teamxx_projx
 - Next, import project files. Change the current directory to the project's directory, and run *svn import*
 - \$ cd /home/name/[...]/myproj
 - \$ svn import https://svn.cse.tamu.edu/[repository name]/teamxx_projx

Check Out

- the repository is stored in the svn directory which you won't deal with. To work on your files, first you need to check a working copy out of the repository. To do so, use svn checkout:
 - svn checkout https://svn.cse.tamu.edu/[repository name]/teamxx_projx

Commit

 Once you're done and you want to store the new revision in your repository, run svn commit in the checked-out myproj directory:

– \$ svn commit

Working with Revisions

Check Status

– \$ svn status <filename>

Compare different Revisions

- \$ svn compare -r R1:R2 <filename>
- (Replace R1 and R2 with actual revision numbers you want to compare)

Revert Local Edits

- \$ svn revert <filename>
- Revert to Previous Revisions
 - \$ svn update -r R
 - (Replace R with an actual revision number)

Final Words

- \$svn help <command> to get a help message on <command>:
 - \$ svn help import
 - <u>http://svnbook.red-bean.com/</u>
- Questions?