Installation Manual: tcl-debug

Tae-hyung Kim
tk424@umr.edu

Department of Electrical and Computer Engineering
University of Missouri-Rolla

Last updated: Jan. 25, 2007

Table of Contents

Acknowledgements

I summarized the first draft version of this manual during my summer internship at BBN Technologies, Cambridge, MA, USA in 2005. I thank Regina Rosales Hain (BBN) and Prof. Larry D. Pyeatt (Texas Tech University) for their direct and indirect contributions to this manual. The manual for the ns release 2.30 has been tested by Fei Ren at the University of Missouri-Rolla. I thank him for his time and effort.

Tae-Hyung Kim

1. Installing tcl-debug for ns-allinone-2.30

As a debugger for Tcl, *tcl-debug* is recommended by the NS (Network Simulator) development team on the official NS website. This document is intended to provide a cookbook manual of "How to install *tcl-debug* to NS." This version of the manual is an update for the new release 2.30, ns-allinone-2.30. The commands have been tested on both the Debian Linux (sage) and cygwin (Linux emulator on Windows OS).

This manual only explains how to install *tcl-debug*. It is assumed that the (Debian) Linux (or cygwin) and NS are already installed.

1.1 Backward compatibility

Tcl-debug installation manuals for previous releases are available at the following links:
http://mailman.isi.edu/pipermail/ns-users/2006-January/053760.html for 2.29,
http://mailman.isi.edu/pipermail/ns-users/2005-June/050495.html for 2.28.

1.2 Errata for release 2.28

There is a typo in step 4.6 in the previous manual for release 2.28.

```
$ cp ns.exe ../lib/
```

The above command should be changed to

```
$ cp ns.exe ../bin/
```

Note: The manual for release 2.28 is targeted for cygwin users, assuming Linux users have less trouble on *tcl-debug* installation. Some commands of cygwin are slightly different from those of Linux. For example, the command for cygwin at step 2 is

```
$ tar -zxvf tcl-debug.tar.gz
```

while the command for my Debian machine is

```
$ tar zxvf tcl-debug.tar.gz
```

Nonetheless, the procedure to install *tcl-debug* on the Linux OS should be the same. I've been using this manual to install *tcl-debug* onto the Linux OS over and over, and it worked every time. ☺

1.3 Convention used in this manual

- A term NS is equivalent to NS-2. I'll use NS instead of NS-2 in this manual.

- I follow the convention in the NS manual to show the NS directory, *~ns*.
For example, *~ns* for release 2.30 is *~/ns-allinone-2.30/ns-2.30*.

- If, an NS user's account name is *username*, then the user's home directory will be */home/username*. Note that the "~" sign is synonymous with the home directory. If NS is installed on the home directory, then *~/ns-allinone-2.30/ns-2.30* is equal to */home/username/ns-allinone-2.30/ns-2.30*.

- There is a slight difference in the syntax between Linux OS and cygwin. In this manual, the default OS under consideration is the Linux OS (bash shell). If the command for Linux and cygwin is different, then the OS is denoted to the left of the dollar sign $. For example,

```
 Linux:   $ tar zxvf tcl-debug.tar.gz
 cygwin: $ tar -zxvf tcl-debug.tar.gz
```

"( )" is also used to show commands in cygwin. For example, the above two lines are equivalent to the single line below.

```
            $ tar zxvf tcl-debug.tar.gz  ( or cygwin: $ tar -zxvf tcl-debug.tar.gz )
```

1.4 Tested systems

This information is provided for your reference.

- NS version and download:
  release 2.30 all-in-one package, *ns-allinone-2.30.2.tar.gz*
  download at http://www.isi.edu/nsnam/ns/ or http://sourceforge.net/project/showfiles.php?group_id=149743&package_id=169689&release_id=371538

- *tcl-debug* version and download
  tcl-debug version 2.0
  download at http://expect.nist.gov/tcl-debug/tcl-debug.tar.gz

Note: I think the version of *tcl-debug* on the official NS website is a typo, as of Jan. 20, 2006. At http://www.isi.edu/nsnam/ns/ns-build.html#pieces, you will see the lines below at approximately the middle of this page.

> **tcl-debug** (optional, available for Tcl debugging help)
>
> Download source: version 1.7; web page: http://expect.nist.gov/tcl-debug/. This version will work with the Tcl/Tk 8.0 release.

When you actually download the tar file, *tcl-debug.tar.gz*, and install it, you will see a directory *tcl-debug-2.0*, not *tcl-debug-1.7*. This means the downloaded version of tcl-debug is actually 2.0, not 1.7.

- Linux package and download
  Debian Linux (sage)
  download at http://www.debian.org/

- Windows + Linux emulator
  Windows XP + cygwin
  The Cygwin DLL release version is 1.5.24-2
  download at http://www.cygwin.com or http://www.cygwin.com/setup.exe

1.5 List of the referred links

http://www.cygwin.com/
http://www.debian.org/
http://www.isi.edu/nsnam/ns/
http://expect.nist.gov/tcl-debug/tcl-debug.tar.gz
http://mailman.isi.edu/pipermail/ns-users/
http://mailman.isi.edu/pipermail/ns-users/2005-June/050495.html

2. How to install *tcl-debug*

There are four basic steps to install *tcl-debug*: (1) prepare installing *tcl-debug,* (2) compile *tcl-debug* and prepare (re)compiling ns, (3) compile ns, and (4) verify installation of *tcl-debug*. Details are presented in the following sections.

Step 1. Prepare installing *tcl-debug*

You will download the compressed file *tcl-debug.tar.gz* and extract it onto a directory parallel to the NS directory, *~ns*.

1.1 Download the *tcl-debug* from http://expect.nist.gov/tcl-debug/tcl-debug.tar.gz
This tar.gz file is a compressed file.

1.2 Move the downloaded tar.gz file to the NS root directory, *~/ns-allinone-2.30*, by typing

```
$ mv tcl-debug.tar.gz ~/ns-allinone-2.30
```

1.3 Change the directory to the NS root directory by typing

```
$ cd  ~/ns-allinone-2.30
```

1.4 Extract the tar.gz file by typing

```
 Linux:  $ tar zxvf tcl-debug.tar.gz
cygwin: $ tar -zxvf tcl-debug.tar.gz
```

Then, you will see a directory, tcl-debug-2.0, under ~/ns-allinone-2.30. Make sure the directory, tcl-debug-2.0, is PARALLEL to the NS directory, *~ns* or *~/ns-allinone-2.30/ns-2.30*. If so, then proceed to step 2; otherwise, go back to one of the previous steps.

This directory contains files and subdirectories. To check the content of this directory, type in

```
$ ls ~/ns-allinone-2.30/tcl-debug-2.0
```

Tip: If the above command for cygwin: $ tar -zxvf tcl-debug.tar.gz does not work, then you may try

```
cygwin:$ gunzip tcl-debug.tar.gz
cygwin:$ tar -xvf tcl-debug.tar
```

If you fail to extract the tar.gz file, then you have to come back to this step after learning more about how to extract compressed files in Unix/Linux. When you google with the keywords “Linux tar” or “Unix tar,” you will see more than 100,000 search results.

Tip: If you want to delete a directory with files and subdirectories in it, then you may type in

```
$ rm -rf directoryName
```

The option –rf means r(recursive) and f(force).

Step 2. Compile *tcl-debug* and prepare (re)compiling ns

Compile the *tcl-debug*, and move the compiled loadable library, *libtcldbg.a*, to the NS library directory, *~/ns-allinone-2.30/lib*, to prepare (re)compiling ns.

2.1 Configure and compile *tcl-debug*

2.1.1 Change the directory into the *tcl-debug* directory

```
$ cd ~/ns-allinone-2.30/tcl-debug-2.0
```

Check to see if configure and Makefile are located in this directory by typing

```
$ ls configure Makefile
```

2.1.2 Configure *tcl-debug* by typing

```
$ ./configure
```

If you see the following error message:

```
username:~/ns-allinone-2.30/tcl-debug-2.0$ ./configure
loading cache ./config.cache
checking for Tcl configuration... configure: warning: Can't find Tcl configuration definitions
./configure: line 630: #: No such file or directory
```

You can fix the error by modifying the file "*configure*" manually. One of the many ways is suggested in Appendix A, which explains how to change your "configure" manually. You may compare my "*configure*". Note that the Tcl version might be different. It is available at http://web.umr.edu/~tk424/CpE401_EE401_Spring_2006/etc/tcl-debug-2.0/configure.

Note: Do not forget to back up the original file "*configure,"* just in case.

2.1.3 Compile *tcl-debug* by typing

```
$ make
```

A loadable library, *libtcldbg.a*, is created after compilation. The file name indicates that this file is the loadable "library for tcl debug." Note that you get a loadable library after compilation, not an executable file.

To make sure libtcldbg.a is there, type in

```
$ ls *.a
```

or

```
$ ls ~/ns-allinone-2.30/tcl-debug-2.0/*.a
```

If you see "*libtcldbg.a,*" then please proceed to the next step; otherwise, go back to one of the previous steps.

2.1.4 Copy the loadable library to the NS library directory, *~ns/lib* or *~/ns-allinone-2.30/ns-2.30/lib*

```
$ cp *.a ~/ns-allinone-2.30/ns-2.30/lib
```

Make sure *libtcldbg.a* is copied properly by typing

```
$ ls ~/ns-allinone-2.30/ns-2.30/lib/*.a
```

or

```
$ cd ~/ns-allinone-2.30/ns-2.30/lib
$ ls
```

If you see "*libtcldbg.a,*" then please proceed to step 3; otherwise, go back to one of the previous steps.

Step 3. Compile the NS

This step is the most important of all. You will enable *tcl-debug* in NS, and then compile NS.

3.1 Move to the NS directory, *~ns* or *~/ns-allinone-2.30/ns-2.30*, by typing

```
$ cd ~/ns-allinone-2.30/ns-2.30
```

3.2 Configure NS with an option to enable *tcl-debug* in NS

3.2.1 Configure NS by typing

```
$ ./configure --with-tcldebug
```

3.2.2 Make sure *libtcldbg.a* is checked.

Scroll up and find the following line:

```
Checking dmallc... not requested with --with-dmalloc
```

Then read the line above it. If this configuration step is successful, then you should see the following line or a line similar to it:

```
Checking for libtcldbg... -L/home/username/ns-allinone-2.30/ns-2.30/ ../tcl-debug-2.0 -
ltcldbg
```

This line indicates the loadable library *libtcldbg.a* is recognized in the directory *tcl-debug-2.0*. If *libtcldbg.a* is failed to be recognized, then you will see,

```
Checking for libtcldbg... no
```

If this step fails, then go back to one of the previous steps.

3.3 Erase previous compilation record by typing

```
$ make clean
```

Note: This step is important.

3.4 Compile NS by typing

```
$ make
```

The compilation may take a while depending on your system. If the compilation is successful, then *tcl-debug* will be embedded to the NS executable. If fails, then you can come back to this step after fixing the bug. This bug fix may be time-consuming and/or painful. Errors during compilation are out of this manual's scope because they are related to NS installation.

3.5 Check to see if you can run the new NS executable by typing

```
$ ~/ns-allinone-2.30/ns-2.30/ns
```

A “%” sign should appear as follows:

```
$ ~/ns-allinone-2.30/ns-2.30/ns
 %
```

If you do not see the % prompt, then the compilation is failed. Go back to one of the previous steps. The “*ns*” command runs NS. To exit the Tcl shell “*%,*” you may type in

```
% exit
$
```

Tip: Ctrl+C also exits the Tcl shell %.

Note: The file name of the NS executable for Linux is “*ns.*” On the other hand, the file name for cygwin is “*ns.exe,*” not “*ns.*” Nevertheless, the command to execute NS is the same for both Linux OS and cygwin, which is “*ns.*”

3.6 Back up the original NS executable in directory *ns-allinone-2.30/bin* by typing

```
Linux:   $ mv ~/ns-allinone-2.30/bin/ns ~/ns-allinone-2.30/bin/ns.backup
```

or

```
Linux:   $ cd ~/ns-allinone-2.30/bin/
Linux:   $ mv ns ns.backup
```

In cygwin, type in

```
cygwin: $ mv ~/ns-allinone-2.30/bin/ns.exe ~/ns-allinone-2.30/bin/ns.backup
```

or

```
cygwin: $ cd ~/ns-allinone-2.30/bin/
cygwin: $ mv ns.exe ns.backup
```

3.7 Copy this NS executable *ns* to the *ns-allinone-2.30/bin* directory.

3.7.1 First, make sure that you are in the right directory by typing

```
$ cd ~/ns-allinone-2.30/ns-2.30
```

3.7.2 Now, copy the NS executable by typing

```
Linux:   $ cp ns ../bin/
cygwin: $ cp ns.exe ../bin/
```

Note: Unless specified, the NS executable in directory *~ns/bin* is run whenever you type in the command *ns*. This is because the directory ~ns/bin is set as the PATH in the *“.bashrc”* configuration file. (Editing*.bashrc* is out of our scope. You can search from the web.)

Step 4. Check if the *tcl-debug* works

In step 3, *tcl-debug* is embedded to the NS executable ns (or ns.exe). Step 4 is the last step of this manual. We will check to see if *tcl-debug* works properly in this step. You can skip this step if you are sure that *tcl-debug* is successfully embedded to the NS executable.

We will pick a sample Tcl script, such as *wireless-test.tcl*, and insert a line "debug 1;" into it. A command line "debug 1;" launches *tcl-debug* when the Tcl interpreter meets this line.

4.1 Pick a sample Tcl script in directory ~ns/tcl/ex or ~/ns-allinone-2.30/ns-2.30/tcl/ex, say wireless-test.tcl, and insert a line "debug 1;" to this Tcl script with a text editor.

4.1.1 Make sure you are in directory *~/ns-allinone-2.30/ns-2.30/tcl/ex/* by typing

```
$ cd ~/ns-allinone-2.30/ns-2.30/tcl/ex/
```

4.1.2 Open the Tcl script by typing

```
$  gedit wireless-test.tcl &
```

Lines 213 and 214 of the Tcl script should look like this:

```
set ns_  [new Simulator]
set chan [new $opt(chan)]
```

Note: *gedit* is used as the primary text editor in this manual. However, any kind of text editor can be used such as *vi*, *emacs*, *nedit*, and so on.

Note: If you do not open the Tcl script in *~/ns-allinone-2.30/ns-2.30/tcl/ex*, you may get an error message.

4.1.3 Put in a line "debug 1;" wherever you want to launch *tcl-debug*. In this manual, we insert this line between lines 213 and 214.

```
set ns_  [new Simulator]
debug 1;            # this line calls the tcl-debug
set chan [new $opt(chan)]
```

Note: The # sign is a comment. You can ignore the line after the "#" sign.

4.2 Run the Tcl script by typing

```
$ ns wireless-test.tcl
```

If *tcl-debug* works, then you will see a line similar to the following line:

```
2: lappend auto_path $dbg_library
dbg2.0>
```

Press Ctrl+C or type in the *exit* command to return to the original shell, and then you will see $.

```
dbg2.0> exit
$
```

If you do not see the *tcl-debug* shell "dbg2.0>", then go back to one of the previous steps.

Note: Not all errors in this step are because of failure in *tcl-debug*.

Note: If you decided to run a different sample Tcl script, that is also fine. Several sample Tcl scripts need a little bit more caution. Run them in the directory *~/ns-allinone-2.30/ns-2.30/tcl/ex/*. Otherwise, you will see an error message like the one below.

```
username: ~/ns-allinone-2.30/ns-2.30$ ns ./tcl/ex/wireless-test.tcl
couldn't read file "../lib/ns-mobilenode.tcl": no such file or directory
    while executing
"source.orig ../lib/ns-mobilenode.tcl"
    ("uplevel" body line 1)
    invoked from within
"uplevel source.orig [list $fileName]"
    invoked from within
"if [$instance_ is_http_url $fileName] {
set buffer [$instance_ read_url $fileName]
uplevel eval $buffer
} else {
uplevel source.orig [list $fileName]
..."
    (procedure "source" line 8)
    invoked from within
"source ../lib/ns-mobilenode.tcl"
    (file "./tcl/ex/wireless-test.tcl" line 183)
```

4.3 Installation is completed.
Congratulations if your *tcl-debug* installation is successful. Your next step is to learn how to "use" *tcl-debug.* Pedro Vale Estrela's website is an invaluable source for this purpose. You may find his website at http://inesc-0.tagus.ist.utl.pt/~pmsrve/ns2/. Don Libe's tutorial "A Debugger for Tcl Applications" is the first tutorial written by the developer of *tcl-debug*. The ps version is available at http://expect.nist.gov/doc/tcl-debug.ps. The pdf version is available at http://rds.yahoo.com/_ylt=A0geu9LoSsJFZhIAHeZXNyoA;_ylu=X3oDMTB2b2gzdDdtBGNvbG8DZQRsA1dTMQRwb3MDMQRzZWMDc3IEdnRpZAM-/SIG=12ctdv5ce/EXP=1170447464/**http%3a//www.ntu.edu.sg/home5/PG01053788/doc/tcl-debug.pdf

If these links are broken, you may search from the web. Good luck!

Yours truly,
Tae-hyung Kim

!!!
Friends are angels who lift us to our feet when
our wing have trouble remembering how to fly.
!!!

Appendix A. Hacking the bug in step 2.1.2

This appendix explains the steps to modify the file configure. I repeat step 2.1.2 for your convenience.

2.1.2 Configure *tcl-debug* by typing

```
$ ./configure
```

If you see the following error message:

```
username:~/ns-allinone-2.30/tcl-debug-2.0$ ./configure
loading cache ./config.cache
checking for Tcl configuration... configure: warning: Can't find Tcl configuration definitions
./configure: line 630: #: No such file or directory
```

You can fix the error by modifying the file "*configure*" manually. One of the many ways is suggested in Appendix A, which explains how to change your "configure" manually. You may compare my "*configure*" available at http://web.umr.edu/~tk424/CpE401_EE401_Spring_2006/etc/tcl-debug-2.0/configure. In my *"configure"*, the Tcl version might be different.

Note: Do not forget to back up the original file "*configure,"* just in case.

Bug fix

1. Open the file "*configure*" using a text editor. To use "*gedit,*" one of text editors, type in

```
$ gedit configure &
```

2. The file "*configure*" should look like below, around line 630 and line 1360. The line numbers are shown on the left for your convenience.

```
line625    echo "$ac_t""found $TCLCONFIG" 1>&6
line626   fi
line627 fi
line628
line629
line630 . $TCLCONFIG
...
line1359          fi
line1360          # see if one is installed
line1361          if test x"${ac_cv_c_tclh}" = x ; then
```

3. Insert the paths as follows:

```
line628 # ***
line629 TCLCONFIG="../tcl8.4.13/unix/tclConfig.sh"
line630   . $TCLCONFIG
...
line1360          # ***
line1361          ac_cv_c_tclh="../tcl8.4.13/generic"
line1362          # see if one is installed
```

In line 628 and line 1360, # *** is just a comment used to find the edited lines easily.

The idea of this step is to manually specify the file *tclConfig.sh* and the directory *generic* where tcl-debug should refer to. This process should be done automatically in *configure*, but something seems to be screwed up. If the Tcl version is changed, say tcl8.4.15, then make an appropriate correction. That is,

```
line629 TCLCONFIG="../tcl8.4.15/unix/tclConfig.sh"
line1361          ac_cv_c_tclh="../tcl8.4.15/generic"
```

instead of,

```
line629 TCLCONFIG="../tcl8.4.13/unix/tclConfig.sh"
line1361          ac_cv_c_tclh="../tcl8.4.13/generic"
```

4. Type in the command “$ .configure” again. If it works, then you will see messages such as the ones below. If not, then go back to one of the steps and correct the mistakes. If you canot solve the problem, then there might be other bugs. Ask questions to the NS mailing list.

```
username @:~/ns-allinone-2.30/tcl-debug-2.0$ ./configure
loading cache ./config.cache
checking for Tcl configuration... configure: warning: Can't find Tcl configuration definitions
checking Tcl build library... -L/home/ns2root/ns-allinone-2.30/tcl8.4.13/unix -ltcl8.4 -ldl  -lieee -lm
checking for Tcl_CreateCommand... yes
checking if Tcl library build specification is valid... yes
checking for gcc... gcc -pipe
checking whether the C compiler (gcc -pipe  ) works... yes
checking whether the C compiler (gcc -pipe  ) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc -pipe accepts -g... yes
checking whether the compiler (gcc -pipe) actually works... yes
checking for a BSD compatible install... /usr/bin/install -c
checking for ranlib... ranlib
checking how to run the C preprocessor... gcc -pipe -E
checking if running LynxOS... no
checking for Tcl private headers... found in ../tcl8.4.13/generic
checking if we are running Linux... yes
checking for strchr... yes
checking for stdlib.h... yes
checking type of library to build... unshared
updating cache ./config.cache
creating ./config.status
creating Makefile
creating pkgIndex
creating tcldbgcf.h
username @:~/ns-allinone-2.30/tcl-debug-2.0$
```

5. If *tcl-debug* is successfully configured, then you can proceed. Move forward to step 2.1.3.