Cheng-Wei Wu 11/7/2005

# Use Emacs Editing Cache Routines

### Updated History

### 11/2/2011

8/15/2012 – about using SSH, rename ‘mump’ to ‘mumps’, etc

### History

It started at the dark age of “^%ZMSP” (and “^%ZMSP” only in 1997). I have learned to use emacs starting 1990, and after that I become addicted to this editor. On the first day to work at Epic, I stared at this ^%ZMSP, and then felt that I become a “newbee to computer” that there are things I used to be able to do but I don’t know how in ^%ZMSP.

Well, now we have Epic Studio which provides a lot of nice feature. I use Epic Studio -- Dan Demp did a good job and it is officially supported by Epic and keeping adding nice features and EMC2). But I still use emasc every day, and still commonly use it for editing/diff M routines, and all kinds of other work (UTF encoding, binary editing). There are still things Epic Studio does not have (like text wrapping mode, multi-frames, etc).

The idea: use emacs to start a cache process (*mumps-shell*) that talks to emacs. With combination of a set of LOOKITT that I have developed/collected, it can save/load routines to a server file that will be automatically loaded in emacs for editing. When requested, the changes can be loaded back to the cache env just like editing with %ZMSP. The result is the mumps-shell. Surely it would be nice if there is a programming mode that works for mump (M, cache) since emacs has very nice color highlighting ability. That’s the idea to create the mumps-mode.

In beginning, I found there is emacs available on our unix server (great, after all, that is not the end of the world), but unfortunately that is a version does not compiled with X. So I started used it as text editor (without any coloring, but at least much familiar way of editing ability). Later I found a free X server that I can run from PC, then I installed a X-supported emacs running from unix. Perhaps, around 1999, I found there is Windows emacs available. That is excellent, so I modified the mumps-shell that can use telnet and the ability to editing files across server (through ftp behind the scene) from the window emacs. Later, I asking is there ability to NFS mount the server directory so our workstations can access it, then there is samba to make this possible. So again I modified the part of editing file across server via ftp (open file as “/user@server:~/file”) to directly use the NFS path. That’s pretty much the current status now.

Actually, the end result is a combination of using unix script, emacs, and Lookitt macros to help making the development process. Some of them now can be done from the current EMC2/Epic Studio.

Major features includes:

* automatically login
* load/save routine between the cache env and unix server
* command to execute cache macro from mumps-shell
* lock/unlock the routine through cache process via mumps-shell
* display lock count (since this is one inconvenient thing is this package); you will need my cache macro though
* use with version control RCS (there may need some extra setup)

Major features it lacks:

* Hyperlink to other tag, routine – emacs is capable but need work to make this happen (maybe one day I’d like to look into this????)
* Definitely a lot others…

### Things Required

You will need:

* windows emacs (sorry, not xemacs) - get the newest Windows emacs from [Gnu](http://ftp.gnu.org/gnu/emacs/windows/) (currently 23.4.1)
* a modified version of telnet (so it can accept run time parameters): (12/23/2011: this is no longer required if you use the new function `mumps-shell-ssh' (which uses secure shell SSH) instead of the original `mumps-shell-win' (which uses ntelnet) – see appendix about emacs/Tramp/ssh for more detail reference. You may have to install PUTTY (steps 9-11 in the appendix). To use tramp, basically you will need the following 2 lines in your emacs init file:

(require 'tramp) ;to use tramp and ssh

(setq tramp-default-method "plink")

* my elisp files (mumps-shell.el, mumps-mode.el, mumps-shell-win.el, etc)
  + - mumps-mode.el – the programming mode for emacs editing cache
    - mumps-shell.el – core for cache session from emacs
    - mumps-shell-win.el – wrapper of mumps-shell for windows to cache server
    - telnet-mode.el – handles telnet mode
    - checksum.el – provides checksum like EMC2
    - tint-instances.el – a package do extra search/highlight (TODO-replace with standard highlight feature in hi-lock.el)
  + Optional: diff program so you can use ediff packages in emacs – the best way now is probably install cgywin and make sure you install diff.

Installation:

* Install emacs if you have not done so
* Copy the folder [F:\personal\cwwu\ToNew\tmp\WS](file:///F:\personal\cwwu\ToNew\tmp\WS) to your workstation c:/Documents and Settings/<your user>/My Documents/WS
* The above steps will copy all my elisp packages that are needed for mumps-shell/mode usage and emacs setup files which copes with the path that will search the elisp packages and the modified telnet program (I got it from internet and renamed it to ntelnet).

### Setting

**Windows setup**

* + Windows environment variables setup: Set user environment variables: (Start >> Settings >> Control Panel >> System >> Advanced >> Environment Variables):
  + HOME as “c:\Documents and Settings\<user>\My Documents\WS “ - used for emacs
  + PATH as “%PATH%;%HOME%\bin” - for stuff in %HOME%\bin, this is optional
* You should exam your copy of c:/Documents and Settings/<user>/My Documents/WS/.emacs-init.el. Especially if you have your own setup/customization for emacs.

**Unix Setup**

* + Create directory path under your home like the following:

~/work/dev

~/work/sbin

* + Add ~/work/sbin to your PATH (depending on which shell you use)
  + Change directory to ~/work/sbin
  + cp ~cwwu/work/sbin/AE AE
  + ln –s AE CDE
  + ln –s AE cde
  + ln –s AE IDE
  + ln –s AE ide

**Cache Setup (Lookitt) - optional**

* Copy my lookitt macros (user 592)
* Useful macros: emrsave, emrload, chksum, lts

**Mumps-shell Setup (reference the document at the beginning of mumps-shell.el, mumps-shell-win.el)**

There are some default settings that you want to change for your convenience. You can use emacs custom setting or in your emacs initialing file to override it (learn emacs!!), or you can modify directly the default values in the following files.

mumps-shell.el:

(defvar mumps-shell-cmd "fnd"

"\*The shell command to start a cache session: [fnd/cde/ide/res]")

;-> "cde" for CDE access

(defvar mumps-shell-default-userID "592"

"\*The default cache user ID")

; -> your cache user ID

(defvar mumps-shell-default-dir "DEV"

"\*The default logon cache logical directory for the cache logon directory

prompt. DEV for FND, EMP for CDE. This often changes.")

; -> "EMP" for CDE

mumps-shell-win.el:

(set-variable 'mumps-shell-server "epic-fnd") ;the unix server running cache: override the one in mumps-shell

; -> "epic-devel" for CDE

(set-variable 'mumps-shell-file-network-directory (concat "//epic-nfs/")) ;\*CW+2 12/2002 now samba is readable/writable now

(set-variable 'mumps-shell-file-server-directory "/net\_home/")

(set-variable 'mumps-shell-file-local-directory "cwwu/work/dev/")

; replace “cwwu” with your unix user and create folders work/dev in your home directory

Once complete the modification. Compile the .el file (to binary file .elc) for better efficiency (`byte-compile-file’).

**Additional emacs setup – optional**

I have these setting in my emacs (in .emacs or any of your emacs user setting file):

;to use mumps-shell-ssh, instead of mumps-shell-win (which needs ntelnet)

(require 'tramp) ;to use tramp and ssh

;;mump (M) mode setup

(setq auto-mode-alist

(append

'(("\\.\\(mump\\|ROU\\)\\(\\.\\sw\*\\)?$" . mumps-mode)

("ARD-.[^\_]+-.+\\.txt$" . mumps-mode)

) auto-mode-alist))

(autoload 'mumps-mode "mumps-mode")

;; mumps-shell mode setup

;(autoload 'mumps-shell "mumps-shell" "mumps-shell-mode" t)

;just temporary replaced by following for mumps-shell-win development

(autoload 'mumps-shell-ssh "mumps-shell-win" "\

Start a mump shell process using ssh from a MS window system."

t)

(autoload 'mumps-shell-win "mumps-shell-win" "\

Start a mump shell process from a MS window system."

t)

(autoload 'mumps-shell-set-fnd "mumps-shell" "\

Set mumps-shell for FND dev"

t)

(autoload 'mumps-shell-set-cde "mumps-shell" "\

Set mumps-shell for CDE"

t)

### Usage

* In emacs, “M-x mumps-shell-ssh” or “M-x mumps-shell-win” (may need ntelnet) to start the login process
* ;; Some useful commands in mumps-shell mode/buffer, see mumps-shell-map

;; for more detail:

;;

;; "\C-cf" 'mumps-shell-checkin-routine-with-version-lock

;; "\C-c\C-f" 'mumps-shell-find-routine

;; "\C-c\C-r" 'mumps-shell-find-routine-read-only

;; "\C-c\C-v" 'mumps-shell-find-file

;; "\C-cl" 'mumps-shell-routine-lock

;; "\C-cr" 'mumps-shell-routine-unlock

;; "\C-cc" 'mumps-shell-checksum

;;

;; "\C-ck" 'mumps-shell-show-lock-count

;; "\C-cm" 'mumps-shell-lookitt-macro

;;

;; "\C-ce" 'mumps-shell-change-env

;; "\C-cd" 'mumps-shell-change-dir

;; "\C-c\C-d" 'mumps-shell-change-dev

;; "\C-c\C-e" 'mumps-shell-change-envdir

### Things to Watch out

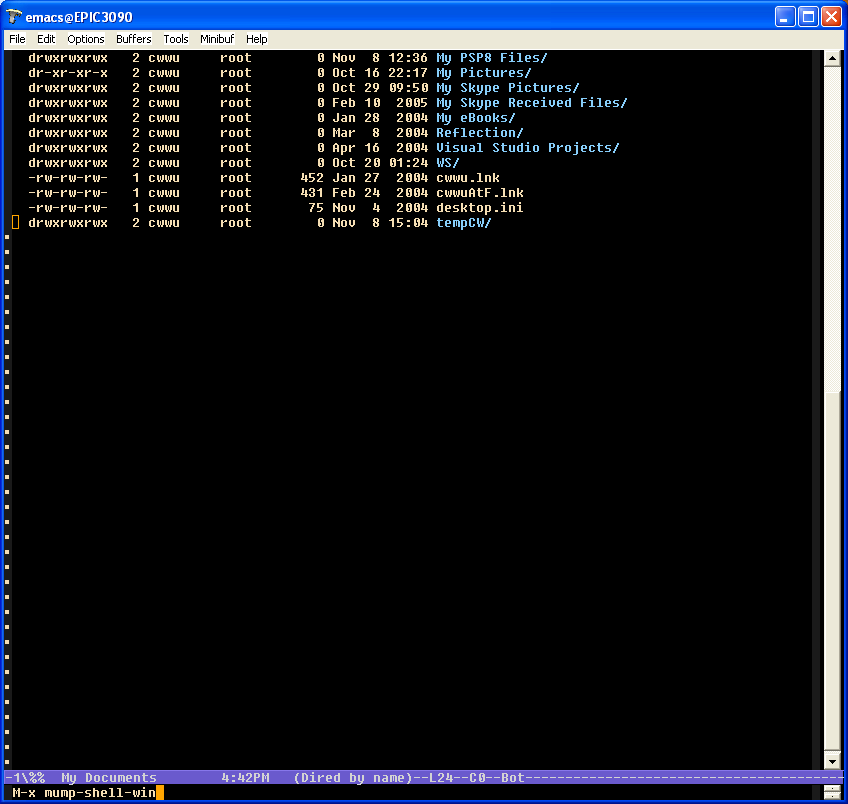
* Remember, there is a temporarily server file (in ~/work/dev) in additional to the copy in the cache.
* Advantage of this is you can continue work if cache is down (but of course not if unix is down).
* Disadvantage: watch out overriding/lost changes. Mumps-shell provides the way to lock the routine through a cache process, but it depends on how you do it. Especially be careful if you are sharing object with others (but the problem is some people do not tell you first before they officially shared the object).
* That’s why I use checksum/diff a lot when I doing development/PQA. It’s a good habit anyway before you move your log.
* It allows you switch among some environment (limited, like pdar/sde/ide). But if you do that, remember which env you are loading/saving the routine!! Well, this is no longer option is CDE/FND anymore since they no longer share the same manager directory.

### Other Extension:

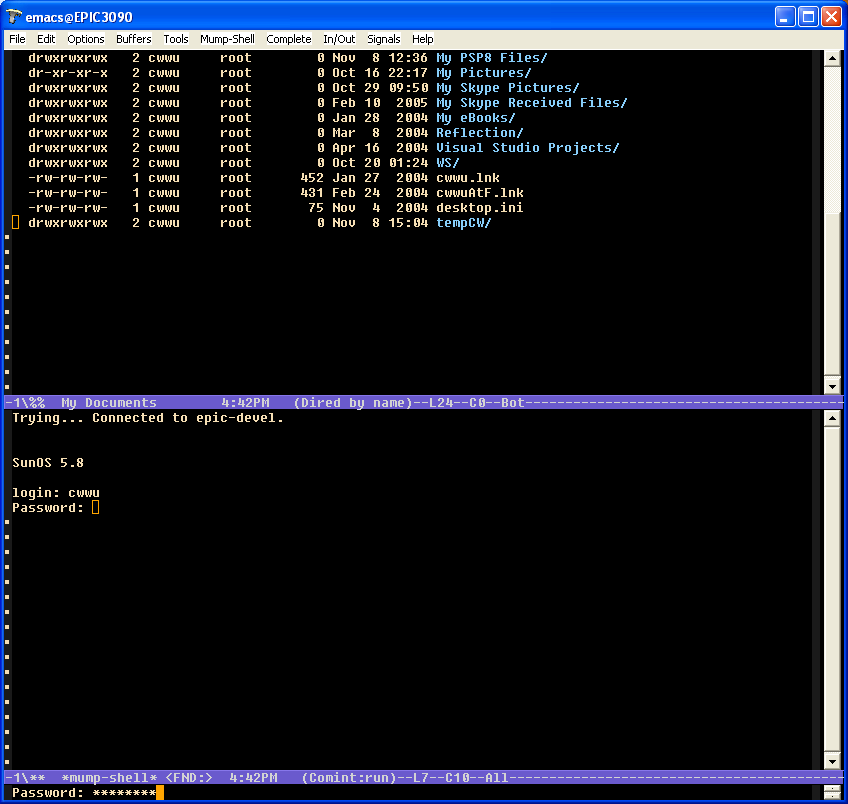
* Use emacs (+diff) for EMC2 source diffing
* Use emacs to diff VB source (from VB)
* It turns out to be a convenient way to do IU/SU – with the help of the diff on emacs and selectively change them in the process of diffing.
* Cygwin: <http://www.cygwin.com/>
* Ntemacs: <http://www.gnu.org/software/emacs/windows/faq2.html>

### Screen Shoots:

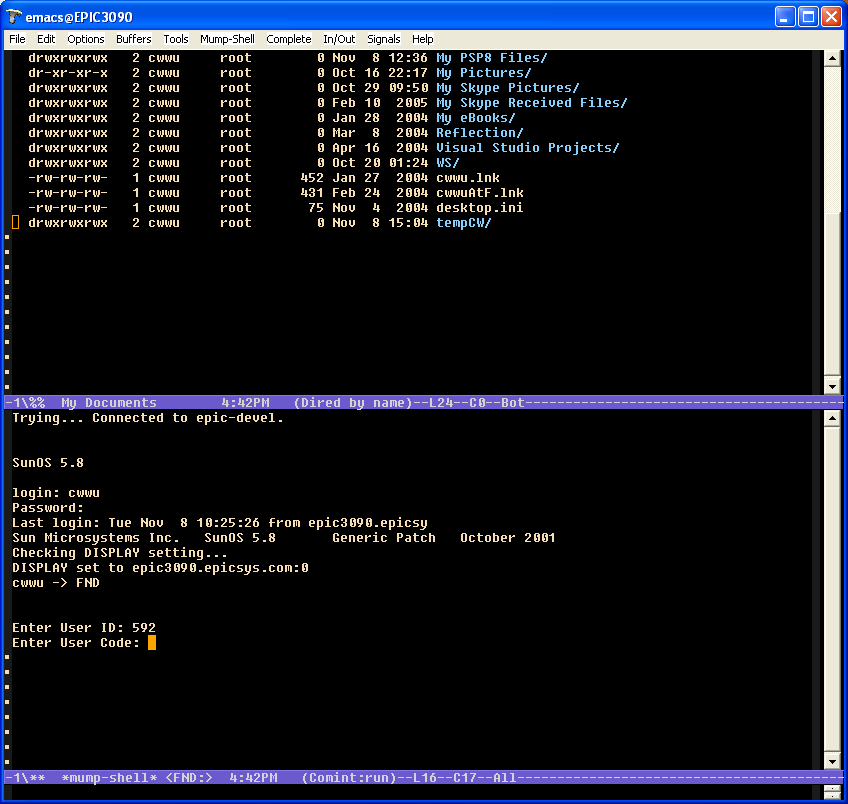
Starting mumps-shell:

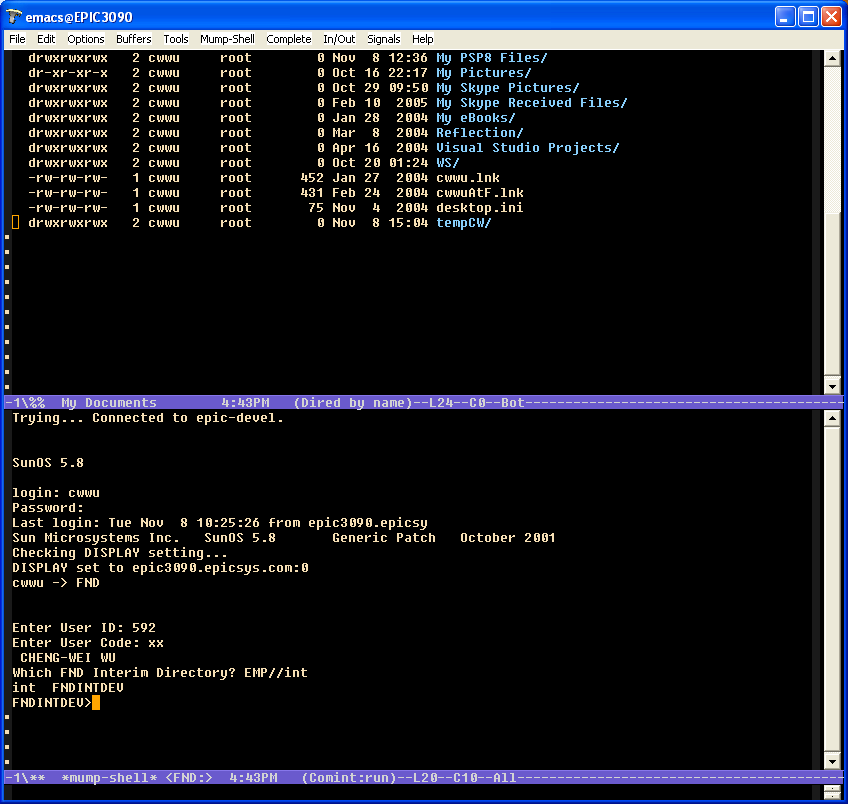


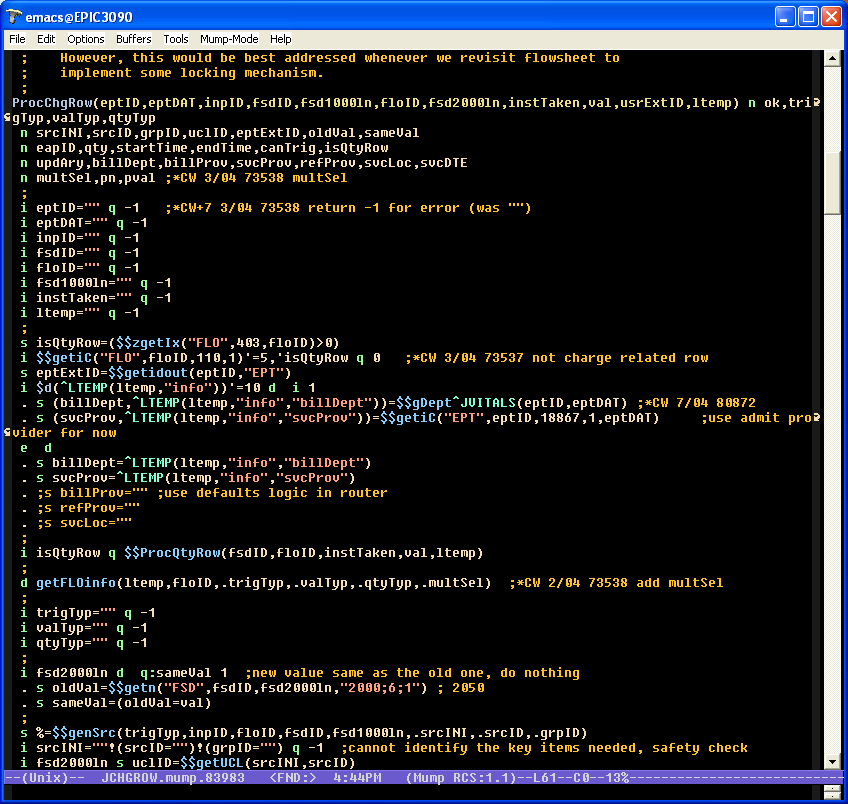
Unix password prompt:



Cache ID/code prompt:



End of logon process (mumps-shell):  


A cache routine in mumps-mode:  


**Appendix**

[**emacs Tramp on Windows**](http://alecthegeek.wordpress.com/2008/05/15/emacs-tramp-on-windows/)

Posted by [Alec The Geek](http://alecthegeek.wordpress.com/) on 15 May 2008

Updated Oct/08

I am using emacs with the tramp extension to edit files on a remote server using ssh. I had to pull in various hacks to get it to work so I hope these notes might help others.

1. Install [Emacs for Windows](http://www.gnu.org/software/emacs/windows/ntemacs.html) and [Cygwin](http://cygwin.com/) with OpenSSH
2. Upgrade emacs [Tramp](http://www.gnu.org/software/tramp/) to the latest version. This is best done using Cygwin bash and Cygwin make (in particular ensure that –with-lispdir and –infodir options are set correctly when running configure).
   * Handy tip — leave the tramp build directory hanging around. When you next upgrade Emacs it’s just make install (providing you use same directory names)
3. Ensure that the Cygwin bin directory is on the Windows system path
4. At the Cygwin bash prompt test ssh connection to your host
5. [Generate public/private keys and upload to the host](http://mah.everybody.org/docs/ssh)
6. Test ssh again to see if public authentication works (it did not for me because it had been disabled by the UNIX admin)
7. Configure emacs to use tramp  
   (require 'tramp)  
   ;;(setq tramp-debug-buffer t)  
   ;;(setq tramp-verbose 10)  
   (setq tramp-auto-save-directory "c:\\tmp")  
   (setq tramp-default-method "plink")
8. Start emacs and test connection (C-x C-f) using cygwin ssh/scp/sftp. If it works skip to step 14. I had issues so I continued with step 9
9. [Install](http://www.jeremyenglish.org/docs/tramp_ntmacs.pdf) the Putty program suite
10. Test the connection using putty
11. Ensure the Putty install directory is on the system PATH
12. In emacs visit the host (C-x C-f) using the [method](http://www.gnu.org/software/tramp/#External-transfer-methods) ‘plink’. If that does not work you are on your own (but see troubleshooting below)
13. If the public keys worked in ssh then import your OpenSSH keys into Putty
14. Use [emacs bookmarks](http://marc-abramowitz.com/archives/2006/03/12/emacs-tramp-bookmarks-awesome/) to save your common host locations

Troubleshooting:

1. Getting rid of the “Couldn’t find exit status of `test -e …” error message by deleting ~/emacs.d/tramp file. This clears cached settings as documented at <http://www.nabble.com/Emacs-tramp-troubles-with-old-Sun-tt13607411.html>.
2. If Tramp and Emacs does not work for you try these alternatives:
   1. [Vim](http://vim.org) has a [netrw](http://vim.sourceforge.net/scripts/script.php?script_id=1075) plugin
   2. Both [FileZilla](http://filezilla-project.org/) and [WinSCP](http://winscp.net/eng/docs/introduction) have options to edit a file from the remote file system which then invokes a local Windows editor of your choice.
   3. [jEdit](http://www.jedit.org) has an [ftp](http://plugins.jedit.org/list.php?category=7) plugin that supports sftp and bookmarks

YMMV

With many thanks to all the folks on the web who documented their experiences and Michael Albinus on the tramp-devl mailing list.

Pasted from <<http://alecthegeek.wordpress.com/2008/05/15/emacs-tramp-on-windows/>>

* <http://www.gnu.org/s/tramp/>
* <http://www.gnu.org/software/emacs/windows/big.html#Network-access>
* Epic and SSH <http://wiki.epicsys.com/main/SSH_access>