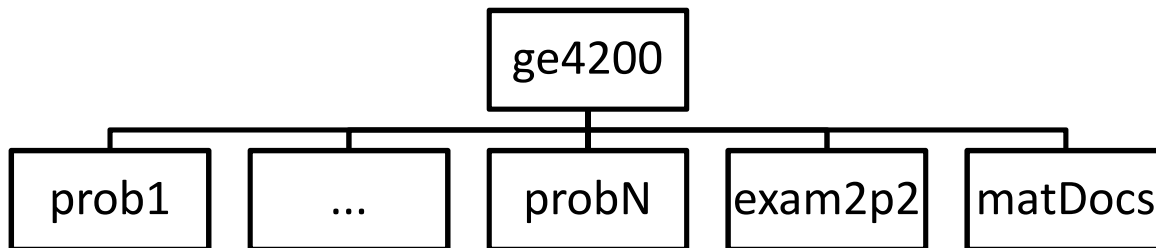


A Modern Approach to .m File Documentation (v. 1.0)  
GE-4200, Winter 2011-12, Dr. C. S. Tritt

Modern programming languages and coding styles generally involve internal, or self, documentation. The Java approach of putting each function in its own file and generating documentation automatically using the *javadoc* program is an example of this. There are many variations on this theme. Most organizations have a style guide or policy document outline their recommended or required practices. Together with my other handouts, this document is intended to simulate industrial practice.

I now required the installation and use of two programs, *m2html* (a Matlab script) and Graphviz (a free dot language processor) that is used by *m2html*. Download and installation instructions for these programs follow in appendices. The purpose of *m2html* is to create an interlinked collection of html files documenting your Matlab code. *Graphviz* is used by *m2html* to produce dependency (calling relationship) diagrams.

To accommodate the way *m2html* works and my approach to grading, you now must organize your work in a particular folder structure. I recommend you create a *ge4200* folder in which to place all your work for this course. My *ge4200* folder is a subfolder within my *d:\classes* folder, but yours can be anywhere you want it to be. You should put the files associate with each project in their own subfolder within *ge4200*. For example, you could put you problem 3 files in a subfolder called *prob3*. You will also need a documentation subfolder in your *ge4200* folder, where *m2html* will put its output. I recommend you use *m2html* such that the documentation for each assignment is placed in a subfolder within your documentation folder. A good name for your documentation subfolder, and the one I use, is *matDocs*. This is all shown schematically below.



You are still required to write a memo covering the purpose, use and design of your program. See my "Advanced Matlab Program Documentation Requirements" (non-GUI Version 3.0 or higher) handout for details. **Your memo should be placed in the same folder as your Matlab source code.**

I have also created a Matlab script, *zip2grade.m* that combines you source files, memo and *m2html* generated html documents into a single zip archive. You can then e-mail this zip file to me for grading. The *zip2grade* script reads your last name and document folder name from a configure file and prompts you for the source code folder name. The configuration file must be called *z2gconfig.txt*. You can download a zip archive, *z2g.zip*, containing the script and a sample configuration file from <http://people.msoe.edu/~tritt/ge4200/examples.html>. You should place its contents in your *ge4200* folder. You should then edit the *z2gconfig.txt* file using the Matlab editor or any other text editor. The first line in this file is a comment and is ignored. The next line should be your last name. This name will be used to create the submission zip file names. The last line should be your documentation folder name. I will discuss and demonstrate the use of this program later in the Standalone Application slide show.

## Using m2html

The document generation program *m2html* must be launched in the Matlab *Current Folder*, one directory level above the folder for which you want documentation generated. For example if your Matlab code is in the directory *d:\ge4200\prob3*, then you need set *ge4200* as the current folder before running *m2html*. This can be done using the *Current Folder* setting in the command window or manually by entering

```
>>cd d:\ge4200
```

You can the launch *m2html* with a command like:

```
>> m2html('mfiles', 'prob3', 'htmldir', 'matDocs', 'recursive', 'on', 'global', 'on', 'graph', 'on');
```

This will cause *m2html* to process all the m-files within the *prob3* directory and its subdirectories (the result of the *recursive on* option) and place the resulting documentation into the *ge4200\matDocs* folder. In this case, the *matDocs* subfolder will contain a *prob3* subfolder with an html file for each M-file in the original *prob3* subfolder and an *index.html* table of contents file. The documentation produced will indicate linkages (calling relationships among all the processed files (the result of the *global on* option)). An overall *index.html* file will also be placed in the *matDocs* directory. This file will contain an index to the files in the folders of *ge4200docs* directory.

## Appendix 1 – Installing m2html

Download the latest version of m2html from <http://www.artefact.tk/software/matlab/m2html>.

Unzip *m2html.zip* into the appropriate folder (for example `d:\ge4200\m2html`)

Add m2html folder to matlab search path for all future sessions (until explicitly changed). See the specific instructions for doing this in Appendix 3.

## Appendix 2 – Installing Graphviz (Graph Visualization Software & dot language processor)

Download the latest version of Graphviz from <http://www.graphviz.org>.

There is a warning of the Graphviz website that there have been reports of the Graphviz installer destroying the Windows path variable. I didn't experience this problem when I installed it, but they recommend you create a Windows restore point (what Graphviz calls a checkpoint) before executing the graphviz msi file. To do this:

Open the Windows *System* dialog by clicking the *Start* button, right-clicking *Computer*, and then clicking *Properties*.

In the left pane, click *System Protection*. If you're prompted for an administrator password or confirmation, type the password or provide confirmation.

Click the *System Protection* tab, and then initiate the creation of a restore point by clicking on *Create...*

In the *System Protection* dialog box, type a description, and then click *Create*.

After creating a restore point run the graphviz msi file by double clicking on it.

**Note: you must restart windows after installing the Graphviz software for the *dot* command needed by *m2html* to be available.**

### Appendix 3 – Changing and Saving the Matlab Search Path

To add folders to the search path for the current session and future sessions, use the `addpath` function, or follow these steps (in particular, note step 4, b for saving changes between sessions):

1. From the Current Folder browser, select *File > Set Path...*. The Set Path dialog box will appear.
2. Click one of these buttons:
  - a. *Add Folder*
  - b. *Add with Subfolders*
3. In the *Browse For Folder* dialog box, select the folder to add to the search path, and then click *OK*. MATLAB adds the specified folder to the top of the search path. If you do not want the folder at the top of the search path, see *Changing the Order of Folders on the Search Path*.
4. Keep or cancel the search path changes:
  - a. To use the newly modified search path only in the current session, click *Close*.
  - b. **To reuse the newly modified search path in the current session and future sessions, click *Save*, and then click *Close*. For details on where to save the file, see *Saving Changes to the Search Path*.**
  - c. To undo your changes, click *Revert*, and then click *Close*.
  - d. To restore the default search path, click *Default*, and then click *Close*. See *Restoring the Default Search Path*.

### Appendix 4 – A Few Comments about Paths

An important concept that is often skipped when introducing computer applications and programming languages is that of the search path. Programs must be able to find their component parts. On MS Windows system this is done by a combination of environment variables (particularly the path variable) and registry keys (the more modern, but still not universal approach). Unix and related operating systems (including Apple's OS X Lion) do not appear to use Window's registry approach.

Here is a short list of Matlab path related topics: Matlab Current Folder (see *cd*, and related topics in Matlab Help), Matlab User Path (see *userpath*, and related topics in Matlab Help). Matlab Path (see *path*, and related topics in Matlab Help) and Startup file, folder (see *startup*, *matlabrc*, and related topics in Matlab Help).