

## **MENUMASTER rev 1.0**

### ***Summary***

Menumaster is written in TI basic for the TI-89 family. Its purpose is to permit easy management of multiple sets of custom menus (toolbars). Menumaster consists of two parts – the main program, ‘menumast’, and a table of menu sets ‘menutabl’. The menu table is easily maintained with TI’s built-in Data/Matrix Editor.

### ***Installation***

Copy the two elements of this program, ‘menumast’ and ‘menutabl’, to your ‘main’ folder. Menumaster can then be run from the command line with

```
menumast ( ) <enter>
```

It is highly recommended that you assign ‘menumast()’ to a diamond number key. For example, to assign it to diamond-8 create this program.

```
:kbdprgm8 ( )  
:Prgm  
:menumast ( )  
:EndPrgm
```

If you want to store menumaster in a folder other than ‘main’ or run it from a folder other than ‘main’ see the program notes at the end of this document.

### ***Operation***

#### **Menumast()**

To load a custom menu, invoke ‘menumast()’ from the command line or from a diamond key, as above. Menumaster displays a dialog box with a drop-down menu of the custom menu sets found in ‘menutabl’. Select a menu set, then <enter><enter>. That’s all there is to it!

#### **Menutabl**

Use the built-in Data/Matrix Editor to maintain your custom menu sets in ‘menutabl’. It permits you to enter new menu sets, edit existing menu sets, delete old menu sets, and move menu sets and tab (key) definitions around.

Within ‘menutabl’ each row corresponds to a menu set. The first column of each row contains a string which is the name of the menu set as displayed in the dialog box pull-down when you run ‘menumast()’.

Columns 2 through n contain strings that define the menu tabs for each menu set. Each string contains a menu tab title followed by the items under the tab. You can have quite a few tabs, and so many

columns, but I don't know what the upper limit actually is. The tab title and items are delimited (default) by the '■' character. (That's <2<sup>nd</sup>><CHAR><4><3>. Hint: copy it and then paste it wherever you need it.) So, for example, a tab definition with the title of 'birds' and three items, 'robbin', 'wren', and 'owl' will look like:

```
"birds■robbin■wren■owl"
```

A menu set with description 'animals' and two tabs 'mammals' and 'birds', might look like:

```

      C1                C2                C3
Rn | "animals" | "mammals■cat■dog" | "birds■robbin■wren■owl"
```

A small yet complete menutabl with two menu sets, 'animals' and 'cars', might look like:

```

      C1                C2                C3
R1 | "animals" | "mammals■cat■dog" | "birds■robbin■wren■owl"
R2 | "cars" | "domestic■ford■chevy■dodge" | "import■bmw■toyota■yugo"
```

## Program notes

If you want to install 'menumast' and/or 'menutabl' to another folder besides 'main', or if you want to run it when your default directory is not the same one that 'menumast' is in, keep two things in mind: First, make sure any diamond number key (kbdprgm) assignment points to the correct location of 'menumast'. For example, if 'menumast' is in folder 'myfolder'

```

:kbdprgm8( )
:Prgm
:myfolder\menumast( )
:EndPrgm
```

Second, make sure that 'menumast' is looking for 'menutabl' in the correct folder. The first few lines of program 'menumast' contain a define statement that sets local variable 't' to the location of 'menutabl'. For example, if 'menutabl' is in folder 'myfolder' change the define statement to:

```
:Define t="myfolder\menutabl"
```

If for some perverted reason you want to use a delimiter character in the tab definitions other than '■' change it in the define statement that sets local variable 'd' to the delimiter character. For example, if you want to use '@' as the delimiter:

```
:Define d="@"
```

Finally, if you find that in creating your menu sets in the Data/Matrix Editor you have complete rows or columns of undefined ('undef') cells you should delete those rows or columns to save space and prevent unwanted program behavior.

## Wrap-up

Menumaster is free for anyone's use, free of any license or copyright. I claim the usual disclaimer, that is, I ain't responsible for nothin'. I hope this program proves useful. It has been tested on a TI-89 Titanium but not on any other models. Please let me know of any bugs. –Warren wseverin@warsev.com