# ToolDaemon 2.5
# User Guide

July 2000.
v2.5 © 1999-2000, DouWère.
Program by Frédéric Blondiau.

The latest version of ToolDaemon is available at http://www.douwere.com/shareware/td/

## This guide is not a exhaustive tutorial to the shell!...

...because a very complete and easy to read documentation about the shell is available at the Apple Developer Tools page at http://developer.apple.com/tools/

Look for the "Introduction to MPW", "MPW Command Reference" and "ToolServer Reference" books, all available if pdf format.

## The ToolDaemon login

First, ToolDaemon requires a username.

```
Username:
```

You need to enter a valid user name, as defined in the"File Sharing" control panel — or in the "Users and Groups" control panel, before Mac OS 9.

This user must also be allowed to use ToolDaemon; this is done by creating a dedicated folder with the user name in the "Users" subfolder, inside the "ToolDaemon folder", as described in the Read-Me document.

If there is no answer to this prompt, the session is aborted after 15 seconds.

Since version 2, there may be a small delay if some previous identification failed; every second, a dot is sent to the user.  Then, ToolDaemon requires the user's password.

```
Password:
```

If there is no answer to this prompt, the session is aborted after 15 seconds.

If everything is OK, ToolDaemon starts ToolServer after printing some useful information.

```
Username:Nonos
Password:****
Authentication failed (error 22).
Username:Nonos
```

```
...
Password:********
Last login on Tuesday, November 9, 1999 at 10:19 PM from a.computer.domain.
1 login attempt failed.
ToolDaemon 2.5. (c)1999-2000, DouWere.  Program by Frederic Blondiau.
ToolServer 3.5, Copyright Apple Computer, Inc. 1990-99. All rights reserved.
ToolDaemon>
```

**The ToolDaemon prompt**

The ToolDaemon prompt invites you to enter MPW commands, with `options` (beginning with -) and/or `parameters` (separated by spaces or tabs).

These commands can be *build in commands*, as this one to get the current date in the short format...

```
ToolDaemon>Date -s
```

...or *external commands* (called `Tools`), as this one to select another network printer...

```
ToolDaemon>Choose -pr -type LaserWriter '*:LW 12/640 PS'
```

...or *names of text files* with a collection of commands (called `Scripts`), as this one, executed at startup by ToolServer...

```
ToolDaemon>Startup
```

...or even *names of native MacOS applications*, as this one to launch and switch to `SimpleText` and let it open the `myFile.txt` file, in the current directory.

```
ToolDaemon>'Simple Text' myFile.txt
```

Note: grouping by single quotes avoids the execution of the `Simple` command, with parameters `Text` and `myFile.txt`.

To get a list and a small explanation of the (some of the) existing commands, just enter `Help` and press the `Enter` Key; you should get this answer.

```
MPW Help Summaries -- ToolDaemon version

Original work copyright Apple Computer, Inc. 1986-1998
 All rights reserved.

 Help summaries are available for each of the MPW commands.
 To see the list of all commands enter "Help Commands".  To see
 a partial list of commands, e.g., those relating to the file
 system, enter "Help FileSystem".  Some commands may appear in
 more than one category.
 In addition, brief descriptions of Expressions, Patterns,
 Selections, Characters, Shortcuts, and Variables are also
 included.

 To see Help summaries, enter a command such as

 Help Commands       # a list of all commands
 Help Editing        # a list of commands useful for editing
 Help FileSystem     # a list of commands relating to files,
                     #   directories, and volumes
```

```
Help Scripting      # a list of commands useful primarily within
                    #    scripts
Help System         # a list of commands that perform system
                    #    functions
Help Miscellaneous  # a list of utilities and aids to software
                    #    construction
Help Launch         # information on launching applications from
                    #    the Shell
Help Variables      # summary of the standard MPW shell
                    #    variables
Help Characters     # summary of MPW Shell special characters
Help Expressions    # summary of expressions
Help Selections     # summary of selections
Help Patterns       # summary of patterns (regular expressions)
```

A lot of variables are already defined by the shell (the boot disk, the path for commands, etc), and the shell lets you define your own variables...

```
ToolDaemon>Set myVariable 'Hello, there!'
```

To replace a variable by its own value, you need to put it into brackets, like

```
ToolDaemon>Set anotherVariable {myVariable}
```

The output of many commands can be used as new commands, use and abuse of the copy/paste menu items, if your telnet client allows it.  To replace a command by its own output, you need to put it into backquotes, like

```
ToolDaemon>Set aThirdVariable `Count -l 'ToolDaemon Logon'`
```

Note: as mentioned before, the use of grouping by quotes are often required to make separate items a single parameter...  To let variable evaluation and command substitution occur inside quote, use double quotes!

As the `Set` command requires a single parameter, a better writing of the previous command is

```
ToolDaemon>Set anotherVariable "{myVariable}"
ToolDaemon>Set aThirdVariable "`Count -l 'ToolDaemon Logon'`"
```

All commands are processed asynchronously.  Multiple commands can be separated by `;` (or by pipe operators) as in

```
ToolDaemon>`Backup -from 'My Disk:' -to 'My Mirror:' -sync`;Shutdown
```

The return of the `ToolDaemon>` prompt indicates that the last command completed... but you can enter new commands before the prompt returns.  For example, if you enter this...

```
ToolDaemon>WhereIs 'my lost file'
```

...the `ToolDaemon>` prompt does not come back immediately, as the search may take a while.  To know the status (i.e. the name of the executing command or script), just press enter.  This can also be done by sending the 'Are You There' telnet command.

```
WhereIs>
```

To abort a running command or script, enter a single dot followed by enter. This can also be done by sending the 'Interrupt process' telnet command.

```
WhereIs>.
### ToolServer - WhereIs aborted.
### ToolServer - Execution of MPW.Script terminated.
ToolDaemon>
```

Closing your current telnet session does not abort the pending commands, so you can initiate long batch processes running unattended (but `ToolServer` will quit once no more users are using it and the last command completed).

Each user can have a custom login and/or logout. These behaviors are defined in the `ToolDaemon logon` and `ToolDaemon logoff` scripts, in the user's folder.

**Authentication errors**

If, after identification, you get an error code greater than zero, there is a problem related to the user authentication.

3 (ugULInfoRecNotCreated)
 There is a problem with the users and groups file. Delete it and recreate it.

11 (ugNoMoreUsers)
 The user was not found in the users and groups file.

13 (ugBadParamErr)
 The length of the administrator password is greater than 8.

16 (ugServerRunning)
 Server is running.

17 (ugAdminRunning)
 Admin is running.

21 (ugBadUID)
 ugAccessID is wrong. The users and groups file may not be open.

22 (ugInvalidPasswd)
 The password is not the correct password for this user (or the Admix password is incorrect, when using AppleShare server).

23 (ugEOF)
 End of file.

98 (ugDefNameErr)
 Default name of users and groups file could not be found.

99 (ugFatalError)
 There is a problem with the users and groups file. Delete it and recreate it.

If the error code is smaller than zero (when a user exists in the the users and groups file, and a correct password is entered), then it is a classical MacOS error.

-120 (dirNFErr)
 There is no "Users" folder.  ToolDaemon is not installed correctly : try reinstalling the product.

-43 (fnfErr)
 There is no file for the identified user, i.e. you forgot to open the "Users" subfolder, inside the "ToolDaemon folder", make a copy (or an alias) of the "@ user session" item and, rename the copy with the user name, to allow this user to connect via Telnet.

## System requirements

ToolDaemon has been designed to run on any Macintosh computer (thanks to PowerPlant and CodeWarrior from MetroWerks).

It runs native on any PowerPC computer,  but will run on a 68020 or better processor.
It uses the native Open Transport stack, but will run on MacTCP configurations.
It uses the Appearance Manager of MacOS 8, but will run on System 7.1 or better (the Thread Manager is required).

## The memory requirements

The default memory requirement for ToolServer is 1024 K.  The memory usage for a session is about 50K per session.  If you need to get more than 4 simultaneous sessions, please consider increasing the memory requirement in the Finder information panel.

The default memory requirement for a ToolServer session is 1536 K, but this can be lowered to 512 K while small tools (read non-compiler tools) are used.

## MPW Users

If you already have the MPW development tools installed on your computer, you may prefer to make the alias with the user name point to your current MPW folder instead of the "@ user session" folder in order to retrieve your own scripts and configuration.

Be aware that these extra files are required if order to run ToolDaemon successfully :  `ToolDaemon Logon`  and  `ToolDaemon Logoff`  to copy from the "@ user session" folder into your MPW folder, and  `Startup`  in the the "TS Startup Items" folder.

You may also want to copy another file,  `Processes`  from the ToolDaemon Commands folder.

Finally, the process portion of the ToolDaemon modified  `MPW.Help`  file could also be integrated in your original  `MPW.Help`  file.

## Contact

Please, send your impressions, suggestions and/or the description of problems you've had with this program to Frédéric Blondiau at frederic.blondiau@douwere.com and be sure to check for latest versions and more info on http://www.douwere.com

DouWère,
Petit Bioleux, 24
B-4122   Plainevaux
Belgium

Fax: +32 4 372 16 30

English is a language I don't speak natively; I apologize for any mistakes. Long live to MacOS!