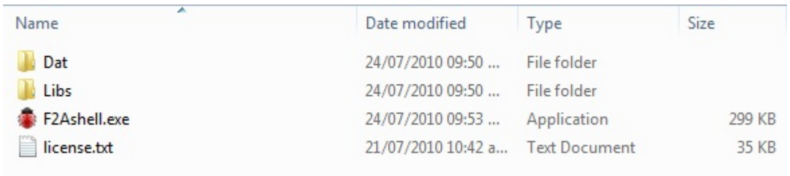
F2A Shell

User Manual

How to run F2A shell in your PC

The first step is to download the last version of F2A shell. Usually, the application is in a RAR file.

Extract the entire F2A shell's folder form RAR file into the folder assigned by you.

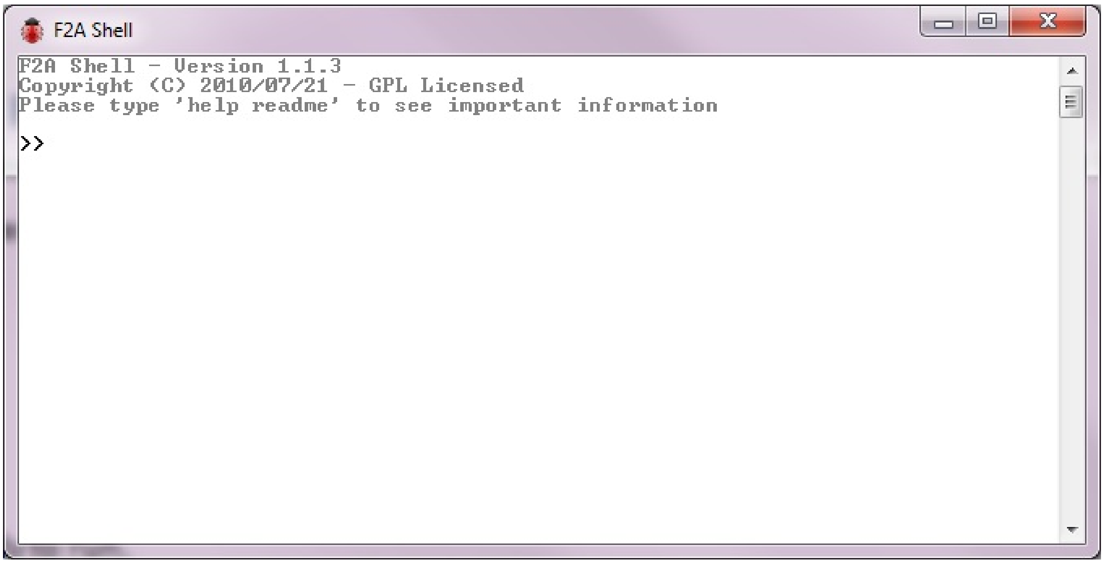


Name	Date modified	Type	Size
Dat	24/07/2010 09:50 ...	File folder	
Libs	24/07/2010 09:50 ...	File folder	
F2Ashell.exe	24/07/2010 09:53 ...	Application	299 KB
license.txt	21/07/2010 10:42 a...	Text Document	35 KB

F2A shell folder installed.

Well, F2A is ready to run.

Do double-click in F2Ashell.exe. The next image shows you how the windows must look.



```
F2A Shell - Version 1.1.3
Copyright (C) 2010/07/21 - GPL Licensed
Please type 'help readme' to see important information

>>
```

F2A shell is running.

Now F2A shell is ready.

Console

The interface of F2A Shell consists in a console to introduce text commands and parameters.

Once you run the application, a prompt should appear. In normal conditions of starting, this prompt is '>>', but it can change depending of the conditions of the shell. The entire format of the prompt is:

$[*user name*]**>**[*domain*]**::**[*subdomain*]**>**

Ahead, in this document, I will explain you what fields of the prompt are.

Next to prompt, the place to write the command and parameters (if they are necessary) is suited. The general format to introduce a command line is:

[*command*] [*parameter 1*] [*parameter 2*] […]

Then, you must press enter to execute the command line into the shell. Between command and next parameter, and between parameters, an space character must be placed.

Examples:

```
>> help
>> domain example
>example> domain .
>>
```

Domains

One of the main things to understand, it's the meaning of the domains. A domain is a context space where a set of commands are allowed. So, a specific command could be valid in one domain and doesn't exist in other one. However, there are some commands which are always valid; those commands are Master Port's Commands.

The list of the Master Port's Commands is:

- domain
- exit
- login
- logout
- mount
- script
- subdomain
- system
- unmount

Master Port's Commands

domain

This command sets the command's context (group of commands allowed). You can choose different domains without restrictions.

Syntaxes:

domain [*domain name*]

sets *domain name* as a domain.

domain .

comes back to default domain.

There are three domains pre-loaded into the shell:

- root
- system
- lib

In **root** domain you can find commands related with the shell. In **system** domain you can find the commands related with your operative system. At least, the **lib** domain is the default domain for the commands from the different libraries' interfaces which were not been mounted in a specific domain (this will be explain in detail later).

The commands of root domains are:

- show

exit

If you want to get out of the shell, you must execute **exit** command.

help

Shell command to show helpful information about commands available and particular command's information.

Syntaxes:

help

shows you the commands available: Master Port's commands and specific commands allowed into the actual domain.

help [*command name*]

shows you detailed particular command's information .

login

Different libraries could need information about the user who is using the console; for that, the shell has a command to introduce that information. You can **login** by the shell using this command.

Syntaxes:

login

the *user name* and *password* will be ask to you.

login [*user name*]

the *password* will be ask to you.

login [*user name*] [*password*]

no extra information is required to log.

logout

With this command you clean the user information.

mount

The process to load a library into the shell is conformed executing the **mount** command. You have to specify the DLL (interface) library associated to the new set of commands. You can specify the new (or not) domain where the commands of this library will be allowed.

Syntaxes:

mount [*library name*]

the default domain is lib.

mount [*library name*] [*domain name*]

In the parameter *library name* you can specify the entire path to reach the library from the folder's F2Ashell. In addition, you can mount more than one library in a same domain and mount a same library in different domains; all of this will depends of your convenience.

Examples:

```
>> mount newLib.dll
>> mount newLib.dll Test
>> mount Libs/newLib2.dll Test2
```

script

If you need to execute a set of commands and you don't want to write them one by one, you can execute them using a text file where each line represents a command line, and then, use the **script** command to execute the entire file. It's a convention use ***ssf*** extensions for these types of files. The last line of these files has to be empty.

Syntaxes:

script [*script text file name*]

When you start the shell application, the shell application searches for the **open.ssf** file (inside **Dat** folder) and executes it. In the closing process, when you execute the exit command, the **close.ssf** file is executes as well. You can use these files to automate different commands that you need execute each time you start and close the shell.

subdomain

A library could need discriminate domains inside a domain, that's the reason because subdomains exist. To set a subdomain you have to execute **subdomain** command as the same way as **domain** command.

Syntaxes:

subdomain [sub*domain name*]

sets *subdomain name* as a subdomain.

subdomain .

comes back to default subdomain.

No subdomains are preloaded.

system

This command executes a system command from your operative system. For example:

```
>> system help
>> system dir
```

unmount

To release a library loaded into a domain's shell (or several domains) you have to use the **unmount** command.

Syntaxes:

unmount [*library name*]

releases *library name* from the shell.

Root's Commands

show

It shows specific information about shell.

Syntaxes:

show ports

shows you all ports created into the shell. The format is:

[*port number*] **-**[*port name*]**- @**[*domain of the port*] **/lib:** [*library associated*]