

8. YOUR INIT FILE

Lisp has a number of global variables that control the environment. Global variables make it easy to customize the environment to fit your needs. One way to do this is to develop an `INIT` file. This is a file that is loaded when you start an image. You can use it to set variables, load files, define functions, and any other things that you want to do to make the Medley environment suit you.

Using the `USERGREETFILES` Variable

As described in File Variables section of Chapter 11, each program file has a global. Your `INIT` file could be called `INIT`, `INIT.LISP`, `INIT.USER`, or whatever the convention is at your site. There is no default name preferred by the system, it just looks for the files listed in the variable `USERGREETFILES` (see below). Check to see what the preference is at your site. Put this file in your directory. Your directory name should be the same as your login name. The `INIT` file is loaded by the function `GREET`. `GREET` is normally run when Medley is started. If this is not the case at your site, or you want to use the machine and Medley has already been started, you can run the function `GREET` yourself. If your user name was, for example, `TURING`, then you would type:

```
(GREET 'TURING)
```

This does a number of things, including undoing any previous greeting operation, loading the site init file, and loading your init file. Where `GREET` looks for your `INIT` file depends on the value of the variable `USERGREETFILES`. The value of this variable is set when the system's `SYSOUT` file is made, so check its value at your site! For example, its value could be:

```
80+ USERGREETFILES
({{DSK}INIT %. COM)
({{DSK}INIT- USER %. COM)
({{DSK}INIT- USER)
({{DSK}INIT))
```

Figure 8-1. Possible Value of `USERGREETFILES`

In each place you see `>USER>`, the argument passed to `GREET` is substituted into the path. This is your login name if you are just starting Medley. For example, the first value in the list would have the system check to see whether there was a `{DSK}<LISPFILES>TURING>INIT.LISP` file. No error is generated if you do not have an `INIT` file, and none of the files in `USERGREETFILES` are found.

Making an Init File

As described in File Variables section of Chapter 11, each program file has a global variable associated with it, whose name is formed by appending `COMS` to the end of the root filename. For any of the standard `INIT` file names, the variable `INITCOMS` is used. To set up an init file, begin by editing this variable. Type:

```
(DV INITCOMS)
```

An SEdit window will appear. This window is the same as the one called with the function `DF`, and described in the *Using the List Structure Editor* section in Chapter 7. This chapter assumes that you know how to use the SEdit structure editor.

The `COMS` variable is a list of lists. The first atom in each internal list specifies for the file package what types of items are in the list, and what it is to do with them. This section will deal with three types of lists: `VAR`, `FILE`, and `P`. Please read about others in Chapter 17 of the *IRM*.

Notice that inside the `vars` list, there is yet another list. The first item in the list is the name of the variable. It is bound to the value of the second item. There are many other variables that you can set by adding them to the `VAR` list. Some of these variables are described in Chapter 24, and many others can be found in the *IRM*.

If you want to automatically load files, that can be done in your init file also. For example, if you always want to load the Library file `SPY.LCOM`, you can load it by editing the `INITCOMS` variable to list the appropriate file in the list starting with `FILE`:

```
.
.
.
(FILE SPY)
.
.
.
```

Figure 8-2. `INITCOMS` Changed to Load `SPY.LCOM` File

Other files can also be added by simply adding their names to this `FILE` list.

Another list that can appear in a `COMS` list begins with `P`. This list contains Lisp expressions that are evaluated when the file is loaded. Do not put `DEFINEQ` expressions in this list. Define the function in the environment, and then save it on the file in the usual way (see Chapter 7).

One type of expression you might want to see here, however, is a `FONTCREATE` function (see Chapter 16). For example, if you want to use a Helvetica 12 BOLD font, and there is not a fontdescriptor for it normally in your environment, the appropriate call to `FONTCREATE` should be in the "P" list. The `INITCOMS` would look like this:

```
.
.
.
(FILE SPY)
(P (FONTCREATE 'HELVETICA 12 'BOLD))
.
.
.
```

Figure 8-3. `INITCOMS` Edited to Include a call to `FONTCREATE`

To quit, exit from SEdit in the usual way. When you run the function `MAKEFILES` (see Chapter 7), be sure that you are connected to the directory (see Chapter 4) where the `INIT` file should appear. Now when `GREET` is run, your `Init` file will be loaded.