

DELTA
Version 2
User's Manual

OPENetwork

Brooklyn, New York

Copyright © OPENetwork, Inc. 1994-2001

All Rights Reserved. No part of this document may be reproduced without the express written consent of OPENetwork, Inc., except for brief excerpts for the purpose of critical review.

License

`delta` used to be under a proprietary license. It will be soon released under an open source license. Currently the source is not yet released but the binaries distributed with this manual are free for unlimited non-commercial use.

This documentation was prepared with L^AT_EX on a PC, with a 600 dpi Postscript printer. Mi-Shell was instrumental in speeding up the cycle of programming and viewing the results.

Contents

Preface	1
1 Quick Start	3
1.1 Overview	3
1.2 System Requirements	5
1.3 What you can get	5
1.4 Installation	6
1.5 Setting up your video	6
1.6 Navigating the menus	6
1.7 Setting up your editors	6
1.8 You must read this	6
2 Fundamentals	9
2.1 General Conventions	9
2.2 The screens of <code>delta</code>	10
2.3 The entry screen	11
2.4 What's in that file? (the file browser)	12
2.5 Comparing directories	14
2.6 Comparing files	16
2.7 Printing with <code>delta</code>	18
2.8 Changing the options	18
2.9 The configuration file	21
3 Smart Backups	23
3.1 Many full backups?	23
3.2 Two kinds of files:	23
3.3 "PROGRAM" changes:	24
3.4 "DATA" changes:	25
3.5 How I keep several PCs in sync.	25
3.6 Comparing computers	26
4 Advanced use of <code>delta</code>	29
4.1 Connecting computers inexpensively	29
4.2 Command line options...	30

4.3	Running <code>delta</code> in Windows	32
4.4	Wildcard expansion	32
4.5	Are you a <code>delta</code> wizard yet?	33
A	OPENetwork	35
A.1	Contacting OPENetwork	35
A.2	More about <code>delta</code>	36
A.3	OPENetwork's Other Products	39
A.4	OPENetwork's Team	41

“An effort is never wasted.”
Elie Chetrit

Preface

Ever since 1976, unix programmers have used “diff” for their file or directory comparisons. MS-DOS users had to do with inferior tools until around 1989 when MKS and OPENetwork both ported unix commands to MS-DOS under the names of MKS Toolkit and The Berkeley Utilities.

When shortly thereafter, a friend repeatedly failed to understand the output of “diff,” I realized there was room for a simple and intuitive comparison tool. John Lowenthal undertook the project and came up with a brilliant addition: the artificial intelligence engine which works so smoothly in the background that `delta` users hardly ever notice it. Since then, `delta` has been the leading comparison tool for MS-DOS-based platforms.

Since 1994, Jean Michel, who had written the Berkeley Utilities “diff,” completely revamped the product, speeding it up, simplifying its user interface and adding many features.

Still `delta` is a product in transition and we need your feedback to make it what we want: the indispensable programmer’s companion. We want to know who you are and how you work. The author welcomes any queries at `jmichel@math.jussieu.fr`

Jean-Claude Chetrit
Fall 1996

“Vive la difference!”
A frenchman

Chapter 1

Quick Start

`delta` is a comparison tool for serious users of MS-DOS compatible systems such as Windows and OS/2 and particularly for programmers and power users. It has now also been ported to the LINUX operating system.

- Its primary function is to allow an easy visual comparison of 2 directories or 2 files.
- A closely related function is keeping 2 directories or files in sync. (See section 3.5)

You can start using `delta` right away because it is easy to use. However, there are probably some features you will not discover unless you use the on-line help or read the manual.

1.1 Overview

There are 3 steps for using `delta`:

1. Select either 2 files or 2 directories to be compared To do so,
 - type `delta object1 object2` on the command line, OR
 - type `delta` on the command line, then chose or type `object1` and `object2` on the entry screen, OR
 - type `delta` on the command line, then chose both objects in the entry screen by typing their name or browsing through directories.
2. The comparison itself:
 - If you compare 2 files, `delta` will show you the files aligned with each other and highlight the differences.

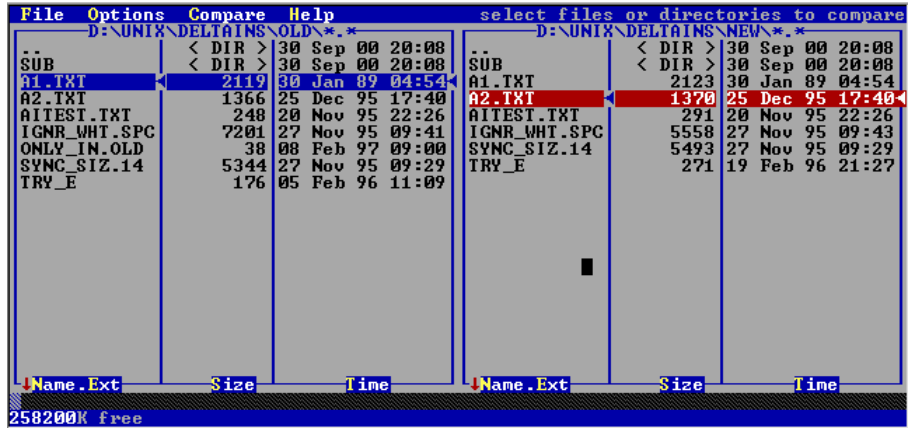


Figure 1.1: The Entry Screen



Figure 1.2: The File Comparison Screen



Figure 1.3: The Directory Comparison

- If you compare 2 directories, `delta` will display both directories, with equal files or subdirectories separated by = and different files or subdirectories separated by < (less recent) or > (more recent). Using the directory comparison, you can compare 2 different files (with the same name) by moving the cursor to that filename and hitting `Enter`.

3. Change the comparison options:

F6 or Options menu, submenu Comparison Options allows you to change `delta`'s options such as ignore whitespace, ignore case, etc... then F10 to save the options for the current session. You can also save these options in a permanent configuration file.

1.2 System Requirements

A computer equipped with color EGA, VGA or SVGA, running MS-DOS 3.x or higher, or any version of Windows. A hard disk and 640K are highly desirable although not necessary.

1.3 What you can get

- `delta.exe` (or just `delta` under LINUX) which is self-sufficient.
- The manual you are reading, in pdf or html format.
- Some sample directories to show features of comparison in `demo.zip`.

1.4 Installation

Installation just consists in putting the executable and the manual in some convenient place.

1.5 Setting up your video

`delta` can use any screen width and height in a MS-DOS box or in full-screen MS-DOS, or any X-Windows xterm, provided it has at least 80 columns and 25 rows. You may want to chose a wider display to show two files side-by-side. Under MS-DOS or windows you may or ask `delta` to change the display, either by typing `delta -l nn` where nn is 25, 43 or 50 specifying how many lines you want. Or you should find which text modes are supported by your video chip in the disk or manual which came with your video card. Or if not, you may be able to get the information from the Internet or from the card manufacturer's BBS. Then use `delta -Vn` where n is the mode you wish to try. For instance, with the Tseng ET4000 chip,

```
delta -V33
```

gives you 132 columns by 60 lines text mode. After a few experiments, make the best video choice your default by saving it in `delta.cfg`.

1.6 Navigating the menus

It is not necessary to have a mouse to use `delta`. You can get around the menus using the keyboard: `Alt` + the highlighted letter in the menu name or entry name goes there. Once inside a menu, you can also navigate with the cursor up and down keys, and chose an entry by hitting `Enter`. Most menus have contextual help which shows on the bottom display line.

1.7 Setting up your editors

From the comparison screen, go to the options menu down to the entries "editor" and "edit both." Type in the name of your ASCII editor. Go down to "save options" and save your changes to the file the program will suggest. More details will be given in section 2.8.

1.8 You must read this

We know you are in a hurry: assuming you got and unpacked `demo.zip` somewhere, go there and compare directories `old` and `old` by typing:

```
delta old new
```

Now, make things as visible as possible: go to the `Options` menu, submenu `Color`, `v` to get to the “visible” color scheme, and hit `Enter` to accept this choice. You can also type `Alt0-c-v-Enter`.

These are the files in `OLD` and `NEW`:

- `A1.TXT`: this simply demonstrates the use of the `Grey +` or `Grey -` key.
- `A2.TXT`: you should toggle the “ignore whitespace” option (`F6` or `Options` menu, submenu `Comparison Options`); assuming you chose the “visible” color scheme, you will see line 16 on the first file and line 17 on the second one marked different or not.
- `AITEST.TXT`: this shows the artificial intelligence (ai) engine. Line 7 on the left matches up with line 10 on the right which is good (ai is on). Turn ai off (`F6` or `Options` menu, submenu `Comparison Options`) and watch line 7 match up with line 8 on the right (bad).
- `IGNR_WHT.SPC`: the comparison looks great or terrible depending on whether ignore white space is on or off.
- `HELLO`: these files are equal and separated by an '=' sign.
- `MENU.ANS`: this shows the use of the 'a' toggle for ANSI sequences (this is not an option, but a display toggle: just type 'a').
- `SYNC_SIZ.14`: the comparison is not great with an ai sync size of 4 (the default) and looks better (line 4 matches line 5) with an ai sync size of 14 (`F6` or `Options` menu, submenu `Comparison Options`).
- `TRY_E`: make sure that ignore white space is off. Then toggle `E` and `T` to see their purpose.

“Faithfulness to fundamentals seems to be a common thread linking professionalism in all areas.” *Toshiro Kageyama, Lessons in the Fundamentals of Go*

Chapter 2

Fundamentals

2.1 General Conventions

Some of the conventions used in this manual are typical in software manuals, while others are not:

- Notation

`^U` stands for Control-U and `F1` for the first function key.

- Help

`F1` will get you help in every screen. In addition, the command line options will be displayed if you type at the MS-DOS prompt (or the shell prompt under LINUX):

```
delta -?
```

- Going to the other side

When you are looking at a comparison side by side, you can go to the other side by hitting the `Tab` key (the side you are in is shown by the cursor). When the display has a top and a bottom, `Tab` takes you from top to bottom or from bottom to top.

- Beep

The computer will beep whenever you type a key which has no meaning for the screen you are in.

- Esc

Throughout your `delta` work, `Esc` will let you go back to the previous screen or cancel what you have done; it is also taken as a “no” answer to questions like *Are you sure you want to delete abc?*. However, you cannot

undelete files which you have deleted or undo the overwriting done by a file copy¹.

- Browsing

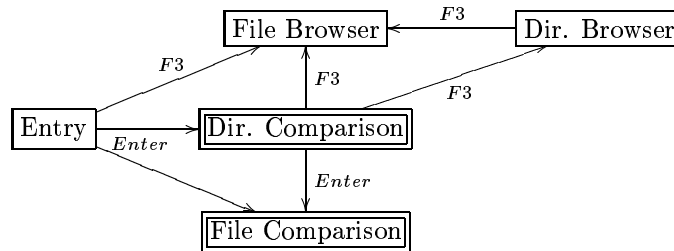
When looking at a directory, F3 will always let you browse what your cursor is on. If it is

- a directory name, F3 changes the current directory to that directory (this is often called navigating).
- a file name, you will see what is in the file. This is often called a “pager” or “more” because you can *page* up and down or see *more* of the file.

- OK

F10 always means yes, ok, go ahead, that’s what I want. It specifically means to start the comparison in the entry screen and save for the current session in the options menu. Also, frequently but not always, **Enter** will have the same meaning (sometimes **Enter** will rather enter a subdirectory).

2.2 The screens of delta



Map of main screens

Besides its main screens, `delta` has a few side screens:

- The Help Screens
- The Print Screen

Finally from most main screens, you can shell out to MS-DOS with F9 (or under LINUX go to a new shell), do a few things, then type `exit` to return to `delta`.

¹In MS-DOS versions 5.0 to 6.22, there is an `undelete` command which can enable you to recover from an accidental `delete` if you catch it early enough; then again, in Windows, Microsoft took it back.

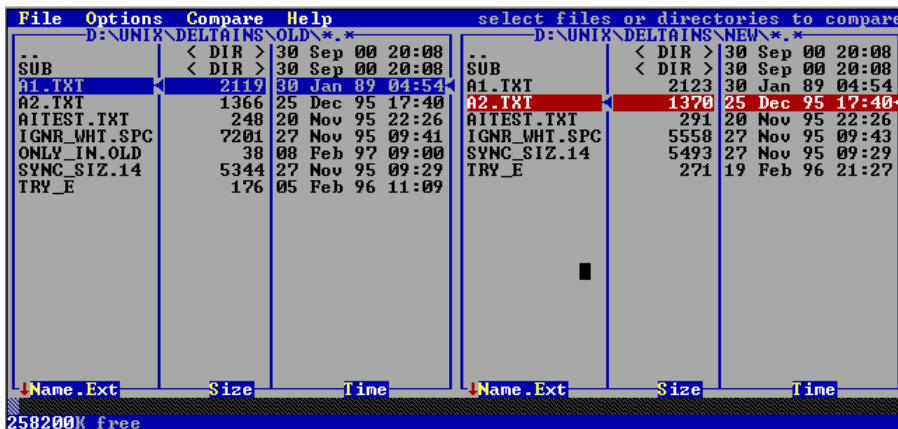


Figure 2.1: The entry screen

2.3 The entry screen

This was described briefly in section 1.1, but is elaborated on here. In particular, we will look at how you can type in file or directory names or navigate to them. When you are in the entry screen, you can navigate by using the cursor keys, **Enter** to enter a subdirectory, or **Tab** to go to the other panel. When you are satisfied with your choice, hit **F10** to start a comparison. You can also edit directly the file name above the current panel by hitting **AltP**. You can enter a wildcard there, and only the files which match will be shown. You can also use this to change drives, by typing a name starting with a drive specification (like **D:**). Browsing (key **F3**) a directory means go there (attach to it, change the default directory to it), and display the directory again (navigating is another term for it). Browsing a file means call the built-in browser (also called paging, more, etc...).

Changing the display

- You can sort the entries by Name, Extension, Size or Time by typing respectively **Alt-N**, **Alt-E**, **Alt-S** or **Alt-T** or by using your mouse to click on the word name, ext, size or date at the bottom of the directory panel. If you click a second time you will reverse the direction of the sort. You can achieve a sort by Extension together with a secondary sort by Size within Extension by typing **Alt-S**, then **Alt-E**. Other combinations are also possible.

Moving around

- You can scroll up or down, one line at a time with the arrow keys, half a page at a time with **^U** and **^D**, or a page at a time with **PgUp** or **PgDn**.

- You can go directly to the first file with `^PgUp` or to the last file with `^PgDn` or `$`.

Actions

The following table is critical in understanding the selector.

	<i>When the cursor is on a directory</i>	<i>When the cursor is on a file</i>
F3	browse	browse
Enter	browse	select
F10	select	select

If you stick to `F3` for browsing and `F10` for selecting, you will not risk any confusion.

You leave the browsing of a file with `q` (for quit) or `Esc`, but when you leave the browsing of directory, it is because you either selected an object to be compared or abandoned your selecting (`Esc`).

2.4 What's in that file? (the file browser)

The file browser lets you conveniently page up or down and search for patterns. If you have previously used either the program `more` of The Berkeley Utilities or Mi-Shell's built-in pager, you are already familiar with our powerful browser. Its many features will be displayed with `F1` or `h`, and are broken down by function:

Moving around

- You can scroll up or down, one line at a time with the arrow keys, half a page at a time with `^U` and `^D`, or a page at a time with `PgUp` or `PgDn` (you can estimate how far you have paged down by looking at the handle bar on the right side of the display).
- You can go directly to the top of the file with `^PgUp`, to the bottom of the file with `^PgDn` or `$`, or to line `n` simply by typing `nG`.
- You can scroll right or left one column at a time with the arrow keys, 10 columns at a time by holding down the control key while you are pushing the arrow keys, or go directly to the first (last) column by hitting `Home` (`End`).

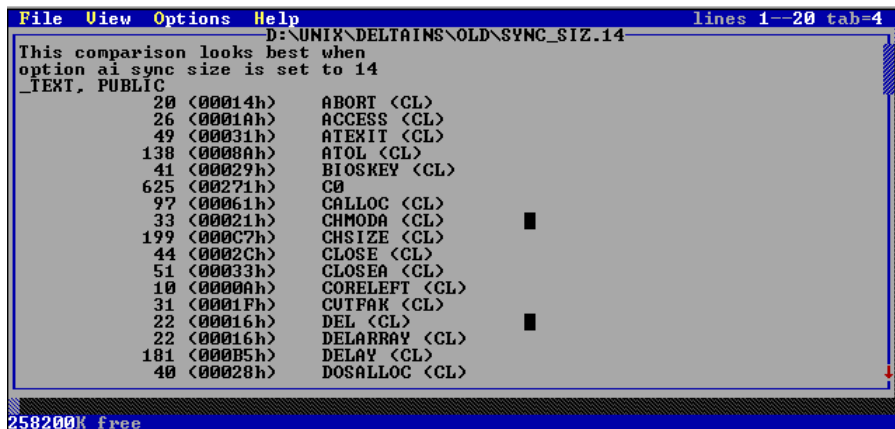


Figure 2.2: The File Browser

- You can go to the first line which matches a chosen pattern. For instance, /abc will go to the next line containing the string abc and using ?abc will search backward. Then, typing n will look for the next occurrence in the same direction, while N will look for the next occurrence in the opposite direction.

Changing the display

It is useful to fully understand the status line at the top of the screen:

- “lines 21-44 ” means you are looking at lines 21 to 44.
- “Col 20” means that your leftmost column is column 20. If no Col is shown, Col 1 is implied.
- [modes: oETa] tab=4 means that you have turned on all the display toggles, and tab size is 4 (i.e., tabs are expanded to 4 spaces; typing *nt* will change it to *n* spaces). Each letter in the following list is the indicator of whether the toggle is on or off, as well as the key to hit to toggle it.
 - E displays in a different color the space past the end of line.
 - T displays Tabs in a different color.
 - a interprets ANSI sequences as such (take a look at the files `menu.ans` in the `old` and `new` subdirectories).
 - B interprets backspaces as you would like (this is useful for downloaded files).
 - o wraps lines longer than the screen’s width.

The second line displays the name of the file you are browsing.



Figure 2.3: The Directory Comparison

2.5 Comparing directories

As soon as you have selected two directories to be compared, the directory comparison screen will appear and the cursor (highlighted line) will go progressively down the display indicating which 2 files are being compared. When the cursor jumps back to the top, you get control back and you can do one of three things: move around, change the display or act.

Moving around

This is very similar to moving around in the file browser; in addition to the keys to move up, down, page up or page down, you can:

- move the cursor from one panel to the other with the Tab key.
- move the cursor to the next (previous) different file in the current panel with the + (-) key.

Changing the display

Sorting directories

When you are looking at a directory screen, you can sort the entries by Name, Extension, Size or Time by typing Alt-N, Alt-E, Alt-S or Alt-T, respectively or by using your mouse to click on the word name, ext, size or date at the bottom of the directory panel. The sort will apply to the side where the cursor is located. You can achieve a primary sort by Extension together with a secondary sort by Size within Extension by typing Alt-S, then Alt-E.

Display toggles

The view menu shows a certain number of items ticked or not, which correspond to the marker to the right side of the entry names. They are associated to the keys = < > l r. A checkmark on the menu line indicates that the corresponding kind of files (described just below) are currently being shown; by choosing the menu entry or pressing the corresponding key, you also toggle whether you want these kind of files are shown:

- = refers to equal files. They may have different timestamps, but their contents are equal (according to the currently chosen comparison options).
- < (>) refers to files that are different and have a more recent timestamp on the right (left)
- l (r) refers to files that exist only in the directory on the left (right)

Pressing F4 will toggle between displaying all the files (i.e. =<>l r) and displaying only unequal files (i.e. <>).

Actions

- move the cursor to a pair of files deemed different from each other, i.e. separated by a < or >, and hit **Enter**: you will get a file comparison of these two files. From there, **Esc** will bring you back to the directory comparison.
- move the cursor to a pair of directories with the same name and separated by
 - a ? (we don't know if they are different),
 - an = (we know they are identical)
 - a < or > (we know they are different).

Hitting **Enter** will enter the directory, either computing the directory comparison or just checking it is still valid; as usual **Esc** will bring you right back.

- move the cursor to a file and browse it with F3. **Enter** will also browse a file if it is absent from the other directory (the word "absent" is written there in pale gray instead of a size and date).
- move the cursor to a subdirectory and browse it with F3. **Enter** will also browse a subdirectory if it is absent from the other directory.
- move the cursor to a file and type d, c or e to delete it, copy it to the other side or edit it.
- move the cursor to a directory and type d or c to delete it or copy it to the other side (recurse subdirectory option should be on).



Figure 2.4: The File Comparison Screen

2.6 Comparing files

Changing the display

In the default configuration shipped to you, black on white refers to lines identical on both sides, while color points out differences. Please take some time to understand these colors, or the ones you eventually choose as your own.

Similarly, it is useful to master the status line at the bottom of the screen, which we repeat here even though it is almost the same as the one at the top of the file browser screen:

- 35000K free means that the program still has that amount of free RAM. Watching this may help you understand why certain comparisons fail – you might then be able to alleviate the problem by choosing different options in the option menu (when the program was a 16-bit dos program this was critical but now the problem is very rare).
- 12% col.1 means that you are 12% down in the file comparison and that your leftmost column is column 1.
- modes[ETaB] tab= 4 means that you have turned on all the display toggles, and tab size is 4 (i.e., that tabs are expanded as 4 spaces; typing *nt* will change it to *n* spaces). Each letter in the following list is the indicator of whether the toggle is on or off, as well as the key to hit to toggle it.
 - E displays the space past the end of lines in a different color, as long as your color choices are not the same background for “normal text” and for “empty space.”

- T means “display Tabs in a different color.” Just as before, the color will appear different as long as your color choices are not the same background for “normal text” and for “Tabs.”
 - a interprets ANSI sequences as they should (take a look at the files `menu.ans` in the old and new subdirectories).
 - B interprets backspaces as you would like (this is useful for downloaded files).
- The F4 key will toggle between the side by side and the top-bottom displays.

Moving around

- the + (-) key will take you to the next (previous) difference.
- You can scroll up or down, one line at a time with the arrow keys, half a page at a time with `^U` or `^D`, and a page at a time with `PgUp` or `PgDn` (you can estimate how far you have paged down by looking at the handle bar on the right side of the display).
- You can go directly to the top of the file with `^PgUp`, to the bottom of the file with `^PgDn` or `$`, or to line n simply by typing `nG`.
- You can scroll right or left one column at a time with the arrow keys, 10 columns at a time by holding down the control key while you are pushing the arrow keys, or go directly to the first (last) column by hitting `Home` (`End`).

Actions

You can:

- Do some limited editing directly on the screen
 - move the cursor to a line and type `d` or `c` to delete the line or copy it to the other side.
- Edit with an external editor
 - by typing `e` to edit the file the cursor is on or by typing `v` to edit both files if your editor allows it.
- print
 - by typing `p` or `F7` or menu `File, print` (see next section).
- miscellaneous
 - press the `F9` key to shell out to MS-DOS (or a new shell under LINUX), or `Esc` to exit this screen.

2.7 Printing with delta

There are two ways to create a written record of the differences between two files:

1. From the file comparison screen, F7 or menu **File**, **print** calls the print menu. The printed comparison will always look like a side by side display. The print menu has 3 items:
 - file you wish to print to: the default is `prn` which sends the output directly to the printer. We recommend you type a name and browse through your output before printing. It will save a few trees and be less frustrating, as you discover you actually wanted something slightly different.
 - number of lines per page: change it if you use a special paper size or font, or can print in landscape mode.
 - number of characters per line: this should be changed if you can print in landscape mode.

2.8 Changing the options

If the comparison you get of your 2 files does not seem entirely satisfactory, it is time to roll up your sleeves and make `delta` work better for these files. Sometimes, it is obvious that changing the ignore case option (F6 or **Options** menu, submenu **Comparison Options**) will solve the problem, sometimes it is much harder to find the answer. We recommend you first read the tutorial in section 1.8.

- ignore whitespace

The basic idea is that several whitespaces will not be considered as different from just one (whitespace consists of blanks and tabs). Please note that one or several blanks will still be considered different from zero. You will also note that, when ignore whitespace is selected, the display still shows you the lines as they really are.

This option is useful if you download a file from a unix system and edit it a bit. The downloaded file has its lines ending in carriage returns, while the edited one has its lines ending in carriage return/line feeds, a subtle and sometimes baffling difference.

- ignore blank lines

When this option is turned on, `delta` will not resynchronize on blank lines. This is achieved by treating blank lines as different from each other.

- ignore case

This option causes `delta` to ignore upper-case lower-case. differences However, the lines are displayed as they really are .

- recurse subdirs

This means that when directories will be compared, they will immediately be recursively explored (instead of when you go there). This is best when you want to do an unattended comparison of two large filesystems (like a desktop computer and a portable across a direct cable connection). You can leave the comparison running when you are away and when you come back do instant browsing.

- 'h' algorithm

The usual algorithm finds optimal differences by using the Stone algorithm. It is very accurate but not very fast and uses some fixed amount of memory for each difference found. The 'h' algorithm uses little memory, is faster, but is unable to deal with very different files or to find the optimal set of differences. The name "h" comes from its being a "half-hearted" attempt. If `delta` runs out of memory, it calls the automatically the 'h' algorithm. If the 'h' option is checked, the 'h' algorithm is always used. This may be useful if you find that some comparisons take forever.

A second option is the length of the longest difference after which the 'h' algorithm can resynchronize (the Stone algorithm has no resynchronize limit). A larger value makes 'h' slower.

A third option is how many lines must be equal for the 'h' algorithm to resynchronize. Contrary to the Stone algorithm which finds optimal differences, 'h' by default resynchronizes on 3 equal lines. When 'h' gives mediocre output, increasing this value may help.

- Best alignment

If this option is checked, `delta` searches the best possible alignment for blocks which differ. Not checking this option gives the fastest possible operating speed for `delta`.

- Artificial intelligence

If this option is checked, what was called `ai` in `delta 1.xx` is done: different blocks are analyzed and the best match of individual lines within them is sought (this option supersedes 'best alignment'). Checking this option may make operation much slower.

A second option when doing `ai` is how many consecutive characters must be equal for two positions in text to be considered for a match. The value by default is 4. For some files a value of 3 or even 2 may be better. However, more often a larger value like 10 or even 15 may be appropriate.

Other options

- (changing the) colors

When you look at 2 files, the differences should jump at you. If you are not happy with the default colors, you can change them. First try

the few color schemes in the color option, then run the `trycolor.bat` file included with the distribution disk. Load the color scheme closest to what you want, by typing on the command line something like:

```
delta -C d3.cfg
```

then modify it by going to the `Options` menu and choosing colors. You will be asked to choose each of the dozen or so colors which `delta` uses:

- normal text, for instance identical sections
- lines, boxes everywhere
- Highlighted text in titles, etc.
- differing areas in different but matched lines
- menu text
- selection bar or cursor
- empty space, after the end of a line (if `E` is active)
- Tabs (if `T` is active)
- Unmatched text
- Equal areas in different but matched lines
- menu keys at the top
- absent lines, non-existent in the file
- grey-out items
- menu bar info text, like "Microsoft Mouse Detected"

- editor

Here you indicate your preferred editor. If you can start it at a given line number, so much the better. For instance, if you are using OPENetwork's freeware "mwe" (multi-windows editor), you can start at the proper line by setting editor as

```
mwe +%f %d
```

- edit both

Some people are lucky enough to be able to specify that they want to edit 2 files from the command line. Modern versions of `vi` are among them, so, for instance, you can invoke the freeware `vim` (written by Bran Moolenaar) by setting "edit both" as

```
vim "+:e +%d2 %f2|new +%d1 %f1"
```

`brief` is another and gets invoked by creating a batch file called `dobrief.bat` which consists of

```
set d_edit = L1 %2, L2 %4
brief %1 %3
```

and by setting “edit both” as:

```
dobrief %f1 %d1 %f2 %d2
```

- save options

When you are done with the comparison options submenu, you can hit F10 to save for the current session only or hit Esc to abandon your changes.

In the option menu you can chose the save options submenu and save your changes in a permanent `cfg` file. The program suggests a file name which we recommend for normal operations. If you start having different preferences for different directories, you can store a different configuration file in each of them (see next section).

2.9 The configuration file

`delta` can work with different configuration files:

- It will first use the one specified on the command line if you used the `-Cfilename` option,
- else use the `delta.cfg` in the current directory if there is one,
- else use the `delta.cfg` in the directory where `delta.exe` is located if there is one.
- else use the values inside `delta.exe`.

Similarly, when you wish to save the options you have modified, we recommend you use the name suggested by the prompt, i.e. `.cfg` in the directory where `.exe` is located, but you can change this name at will (you may, for instance, want to have certain options saved for your “C” work and some others saved for your database work).

Under MS-DOS, you can save the options while the configuration name is blank. That will cause the default settings of `delta` to be changed permanently. Yes, the executable will be changed! which may toggle a virus warning or drive your network administrator crazy, so please use carefully and don’t try it if you don’t understand the above remarks.

“One must imagine Sisyphus happy!”
Albert Camus, in The Myth of Sisyphus

Chapter 3

Smart Backups

You have heard “**Backup!**” many times, but now that you have `delta`, you can do it intelligently. The strategy we propose is nothing more than common sense.

3.1 Many full backups?

Let us assume that you can afford to backup everything frequently. Should you do it? Please think about it before looking at the answer below. On the upside, you will never lose anything; the downside is lots of backups to manage. Well, the answer is no, you should not! What you must do instead is backup your “DATA” and make sure that your “PROGRAMS” have not changed! We’re going to explain these two words in detail.

3.2 Two kinds of files:

Your hard disk contains 2 kinds of files. The first kind (“PROGRAMS”) are the programs and their setup. Since you spent time and effort making the programs work well, you want to prevent them from changing without your permission and understanding. But if you lose these files, it is not catastrophic because you can always reinstall the programs. We call the other kind of files the “DATA.” They are the result of your working with the programs: your writings, your spreadsheets, your source code and even your compiled programs if you are a programmer, etc... It is critical to preserve these files because they are not replaceable. If your computer is stolen, and you have no backup, no amount of money in the world can ever recover what is lost.

Separate the two kinds of files:

So, an important priority is keeping “DATA” in subdirectories separate from the subdirectories where the “PROGRAMS” are installed. We even recommend keeping the two kinds of files on different physical drives because besides simplifying backup issues, it will also increase the performance of many large programs! If you tend to work at the MS-DOS prompt and want a way to make that task easier, we recommend you try our shareware program called Mi-Shell (it resembles Norton Commander).

The initial full backup:

There are many kinds of backups, but we’ll simplify here and concern ourselves only with full backups (also called complete backups) and incremental backups which only save the changes since the last backup. There was a time when a full backup meant 4 to 10 hours and more than 100 floppies. Now, a tape backup is inexpensive (less than \$200) and will probably do a great job while you go out to lunch. But a SyQuest or Bernoulli drive is even better because it looks and behaves like a hard disk drive (random access rather than linear access). They are fast and cheap: between 7 and 20 cents a megabyte. Finally, a frequently overlooked backup method consists of keeping several PCs always in sync., an extremely desirable strategy which is described below. For simplicity, we will assume that your backup is a simple and complete copy of your hard disk.

3.3 “PROGRAM” changes:

How do you know that a program has changed? MS-DOS provides part of the answer, called the archive bit, but it is not enough for many reasons (some viruses and some software will change files without changing the archive bit). First, let us assume that you have followed our advice and that, for instance, the WordPerfect program is under `\wp`, that its backup is under `\wp.bak` and that the data is under `\writing`. Now, type on the command line:

```
delta -r \wp \wp.bak
```

The `-r` means turn on the recursive option, i.e. compare not just these two subdirectories, but also all the ones below them, and below those, etc... If there has been no change, you are all set. If there is a change, `delta` will show you which directories and which files were changed. It may be serious: a virus attached to `wp.com`, or innocuous: you installed a fax, but either way, two things are now urgent:

- **Understanding what happened is the key to our strategy.** Making lots of backups without understanding is better than nothing, but it only delays the decision of what must be done. Saving every piece of paper in a (large) shoe box is an easy and user-friendly storage method but not

a particularly efficient retrieval method: when you need something, you know it's in the box but you can't get it. Similarly, if you have 167 backups of WordPerfect, some will be good for some things, but which ones? Even though `delta` is written primarily as a comparison tool for programmers, it is ideally suited for this job!

- Even if you do understand, you have to make another backup (incremental or full), because there are interactions that even the smartest wizard cannot anticipate. For instance, you added a fax to your PC; you changed the setup of WordPerfect so you can “print” to your fax. So you are not surprised to see changes in `\wp`. Furthermore WordPerfect seems to work fine. But your rarely used CD-ROM player will refuse to play music two months from now and you will not know why. Three more packages will have been installed by then, making your task not a trivial one. Therefore, the proper time to backup your “PROGRAMS” is **Every Time They Change**, no more, no less.

3.4 “DATA” changes:

When you work on your PC, you fully expect your “DATA” to change. It is nice to have a recent backup while you work. Two months ago, I deleted everything on my home directory because I thought I was on `a:` but had typed `a`; instead of `a:` failing to get there. `&@!!&@@!!` No problem, I thought, I'll `undelete`. Alas, my drive was networked and refused. In order to recover, I had to stop the network, `undelete`, and then reconnect the network: it was a pain and a waste of time. Anyway, backing up your “DATA” should be done more or less frequently depending on the importance of said “DATA.” We keep our list of clients and financial records on a database. We back it up every day because we can't afford to lose it. But we promised you details, so we'll give you some: we keep a mirror image of the directory on a server; we have 5 diskettes for the last 5 days; every day, we backup the database over the oldest backup of the 5. In addition, we keep a permanent diskette at the end of each month. This is not perfect, but better than the usual **no backup at all**.

3.5 How I keep several PCs in sync.

Rather than tell you what to do, I'll describe what I do and you can adapt it to your situation.

When I am not consulting:

I do all my work at home. Since my children come every week-end to spend time with me and play their favorite computer games, I do a complete backup before they show up. In my basement is hidden a backup system which contains

last week's backup. The PCs are linked via ethernet cards and a LANTASTIC network. I type on the command line:

```
delta -r d:/july96.jc c:/
```

then methodically explore all the differences (all my week's work and mistakes), and eliminate them, hence fixing my mistakes and making the backup current.

When my children leave on Sunday night, I type the same command and I can see if they have done any damage.

When I am consulting at a client's office:

I go to work every day, spend all day at my work PC, then come home and frequently spend many hours working at my home PC. Finally, I frequently spend the night at my girlfriend's where I sometimes end up working on her PC.

It is not imperative nor even desirable for these PCs to be identical, what I need is to always have my latest work with me. So, I try hard to keep all my "DATA" in various subdirectories below c:\home and I carry a ZIP drive with a copy of c:\home on a ZIP disk.

Whenever I sit down at a different PC, I compare c:\home with the ZIP disk version D:\home and bring them into sync. Whenever I am about to leave a PC, I compare them and again bring them into sync so that I always travel with the latest version.

This process may seem a bit time consuming, but I will now never wonder how old my backup was when my PC crashed (or worse, was stolen). Maybe even more important for me is that I will no longer have the frequent frustration "I wanted to work on xxx, but forgot it at home."

Technical Details:

A JAZ drive or a SyQuest EZ-135 or EZ-230, or an optical drive, or one of the new gigabyte removable media will do fine too. It is possible to operate with a tape backup, but it is so time consuming that we strongly advise you to get one of these new removable hard disks instead.

3.6 Comparing computers

If the objective is to keep two PCs completely in sync., there are two cases: when they are close to each other, they should be networked (read the first section of the next chapter if they are not); then use `delta` to compare the two hard disks. When they are in different locations, use a removable hard disk and the technique outlined above.

A harder situation is when the two PCs are similar (but not identical) and you want to solve a mystery. We once wondered in our office why “ghostscript” worked well on one PC, but not on the other.

- The first step is to use the method described in the preceding section to compare the root directory, the Windows directory and the ghostscript directory.
- The second step is to run diagnostics on each machine, to write the output to a file, and then to compare the 2 files. For instance, type

```
mem/c > memc
```

on each computer, then compare the 2 memc files. There are many programs we use and recommend like `checkit` (from Touchstone Software Corporation) and `mft` (Manifest from Quarterdeck).

“You can tell a professional by the sharpness of his tools.” *Ancient saying*

Chapter 4

Advanced use of delta

4.1 Connecting computers inexpensively

If **all** your computers are already networked, you can skip this section. Otherwise, the hardware needed to connect two of them is a bi-directional parallel cable worth around \$ 10 (in a pinch, a serial cable will also work, although not as fast). The software needed is DOS 6.xx or Windows 95.

The method consists in choosing one computer as the one you want to use (we'll call it the client or guest) and in considering the other one as simply a set of distant drives (we'll call that one the server or host).

Two MS-DOS computers

Under MS-DOS 6.xx, on the server, you type at the MS-DOS prompt

```
intersvr
```

then leave the server alone until you wish to break the connection.

On the client, you add to your `config.sys` a line like

```
device = c:\dos\interlnk.exe
```

When you reboot the client computer, the server's drives become available to the client computer with different names, for instance the server's C drive may have become the client's G drive.

Two Windows 95 computers

For Windows 95 computers, this is called direct cable connection. Here, the 2 computers are called host and guest.

- Setup

On the host, select Start, Programs, Accessories, Direct Cable Connection, and choose host. At this point, you may need to install the Dial-Up Adapter and restart. Choose the port, then setup file and print sharing and again restart. Eventually you reach a screen that says “You have successfully set up the host computer.” Click on Finish.

On the guest computer, do the same except that you choose guest. You must also choose a network protocol.

- Establishing the connection

Again, select Start, Programs, Accessories, Direct Cable Connection and make the host PC Listen and the guest PC connect.

A MS-DOS computer and a Windows 95 computer

To connect a MS-DOS 6.xx computer to a Windows 95 computer, you have only one choice: make the client the MS-DOS 6.xx computer as described above. You can then start `intersvr` on the Windows 95 computer. However you will then be unable to task switch out of it and so use concurrently your Window 95 computer in another way (this is a limitation of Windows 95).

4.2 Command line options...

If you type `delta -?`, you will get help screens with a summary of the information in this section.

... for `delta`'s non visual mode

The general syntax for `delta` is

```
delta [options] [f1/dir1 [f2/dir2]]
```

If one of the `fi`'s is '-', then “stdin,” the standard input (or console) is used: this means that you type the file at the keyboard, ending with a `^Z`.

- `-ln` Set the number of screen lines to `n` (`n=25, 43` or `50`).
- `-Vn` Set display to visual mode `n` (see section 1.5).
- `-Ecommand` Specify which editor is used — see section 2.8.
- `-E2command` Specify which editor is used to edit both files at the same time — see section 2.8.
- `-vtn` Set tabsize to `n`.
- `-vbn` `n` is the sum of the browser flags coded as follows:

- 1 = Fold long lines (i.e. set toggle “o” on).
 - 2 = Show empty space in different color (i.e. set toggle “E” on).
 - 4 = Show tabs in a different color (i.e. set toggle “T” on).
 - 8 = Interpret ANSI sequences as such (i.e. set toggle “a” on).
 - 16 = Treat backspaces properly (i.e. set toggle “b” on). This is very useful if the file is the log of a telecommunication.
- **-va[n]** Use ai and optionally set ai sync size to n.
 - **-vn n** is the sum of the following flags:
 - 1 = use best alignment
 - 2 = Side by side display
 - 4 = Sort by extension
 - 8 = Sort by size
 - 16 = Sort by time
 - 32 = the cursor starts in the second window in comparison screens.
 - 64 = use artificial intelligence (as done in `delta 1.xx`)
 - 128 = use blinking colors rather than highlighted colors.
 - **-a** Force all files to be compared as text
 - **-b** Ignore leading and trailing whitespace and squeeze other whitespace when comparing lines.
 - **-i** Ignore differences between upper-case and lower-case when comparing lines.
 - **-I** Ignore blank lines when comparing files.
 - **-h[n[:s]]** Use the 'h' algorithm which does not run out of memory on large files but may fail to resynchronize (“h” stands for half-hearted attempt). If n is given, it specifies the largest resynchronizable difference (default 200) and if s is given, it specifies how many identical lines are needed to resynchronize (default 3).
 - **-Cname** Take configuration from file `name.cfg`. By default, a configuration file `delta.cfg` is sought first in current directory, then in the directory where the executable is located. If there is an option **-C**, the corresponding configuration file is executed after all the command line options have been read.

4.3 Running delta in Windows

We recommend that you use your PIF editor to load the MS-DOS prompt and replace `command.com` by `delta.exe`, save as `delta.pif` and exit, then invoke the new pif file. If you have chosen full screen, you are done. If you have chosen a window (i.e. you are in a MS-DOS box), we recommend you maximize the window and change the box's font size to suit your needs. The next time you go in, the settings will be remembered. If you are uncomfortable with the smaller fonts, Ctrl-Enter will toggle you back to a full screen display.

4.4 Wildcard expansion

This is a section out of The Berkeley Utilities Manual. As in MS-DOS (and unix),

- `*` stands for anything (including possibly nothing)¹
- `?` stands for exactly one character. Please note that `*.???` is not the same as `*.*`.
- `.` stands for the current directory
- `..` stands for the parent directory

As in unix,

- `[a-e]` stands for any letter between a and e
- `[abp-z]` stands for a or b or any letter between p and z
- `[!p-x]` stands for any character except a letter between p and x

Finally, the following extension is unique to OPENetwork's products.

- `//` stands for any amount of successive directories

Here are a few examples:

- `*.doc` All the files with a "doc" extension in the current directory
- `manual.*` All the files with a filename "manual" in the current directory
- `*.d?` All the files in the current directory with a 2 character extension where the first character is a "d"
- `//*.doc` All the files with a "doc" extension in the current drive
- `.//*.doc` All the files with a "doc" extension in or below the current directory

¹This is poorly implemented in DOS: don't try to `del *A.*`, all your files would disappear!

- `//temp/*.doc` All the files with a “doc” extension in a “temp” subdirectory of the current drive
- `[c-g]*.*` All the files in the current directory with a filename starting with a letter between “c” and “g”
- `[!c-g]*.*` All the files in the current directory with a filename not starting with a letter between “c” and “g”
- `*[c-g]*.*` All the files in the current directory with a filename containing a letter between “c” and “g”
- `*[!c-g]*.*` All the files in the current directory with a filename not containing a letter between “c” and “g”

“Comparaison n’est pas raison!”
“*An analogy does not prove much!*”

Appendix A

OPENetwork

A.1 Contacting OPENetwork

Our customers have always been our biggest asset, so we always offer

- a 30 day money back guarantee (in the U.S.)
- free tech support for registered users
- free upgrades for 12 months and low cost annual upgrades thereafter

Tech Support

OPENetwork’s technical support is provided primarily via e-mail:

`open@dti.net`

and secondarily via its free 24-hour BBS (718) 638-2239

You can also use our voice telephone (718) 398-3838, but it should mostly be used to make sure we’re aware there is a problem which you have documented on our BBS or via e-mail.

Troubleshooting

If you experience trouble, please try to reproduce the problem in different environments (i.e. from a different PC, when you just boot, when you do a clean boot, from a Windows MS-DOS box, with different options, without any configuration file, etc...) then upload your observations, the `delta.cfg`, and maybe the offending files. If you can compress your files with PKZIP, ARJ or RAR, we prefer it to a bunch of files. Then upload these to our BBS or send them as an attachment to an e-mail to us.

This gives us a chance to think about your problem, reproduce it (particularly if you have included all the relevant problem files) and hopefully resolve it.

Feedback

We want to know how you feel about our software. Of course, we will fix bugs as quickly as we can. But tell us also what you have difficulty with, what is unclear or what you would like to see in the next version. Version 2 of `delta` will undergo many revisions. Since we don't sell many copies, you have a very important opportunity to influence the future of our product.

Web Page

We maintain a Web Page on the Internet where you can monitor new announcements about `delta`, bugs found and fixed, new versions, etc... our address there is:

`http://home.dti.net/open/delta.html`

A.2 More about delta

delta's history

- Stone Algorithm 1976 (Harold Stone)
- Independently, Wayne Hunt and Tom Szymanski come up with a similar algorithm: "A fast algorithm for computing longest common subsequences," CACM, May, 1977.
- `unix diff` 1976 (Doug McIlroy)
- The Berkeley Utilities `diff` 1989 (Jean Michel, Patrick Mérisser-Coffinières)
- `delta 1.xx` 1990–1993 (John Lowenthal)
- `delta 2.xx` 1995–present (Jean Michel)

What's new in version 2

- new features:
 - We have simplified the user interface.
 - You can compare much faster
 - You can compare much bigger files
 - You can compare binary files
 - You can now browse files (typically with F3)
 - You can copy or delete files (with `c` or `d`)
 - You can copy or delete lines on the fly (with `c` or `d`)
 - You can ignore upper-case lower-case differences

- You can change all the colors
- You can display long filenames in Windows 95
- You can use your favorite editor (another way of saying that we dropped the built-in editor)
- Design considerations
 - The main idea we have in mind is that of bringing two computers, two hard disks, two subdirectories or two files in sync. While simply overwriting the older version is a common practice, we suggest that using `delta` enables you to be sure that you are not losing something valuable in the process.
- Future enhancements
 - Hexadecimal display toggle (mostly for binary files).
 - Ignore specified columns. Ignore comments specific to a given language (maybe more generally ignore a given regular expression).
 - A very friendly interface for creating a merged document.
 - Please let us know which features (including things we never thought about) you want to see implemented first.

Upgrading from version 1.xx

If you have been using `delta` all these years, we have tried to preserve many key assignments for your convenience:

- F1 for help
- F4 to toggle “stereo,” which we now call top/bottom or side by side
- F5 to edit the file the cursor is on
- F6 for options
- F7 to print
- F9 to shell out to MS-DOS
- Esc to abandon almost anything

However, with added power comes the need to learn a few new tricks and to unlearn a few old ones:

- F2 no longer means select files.
- F3 no longer means “recalc” but browse. Recalc is now done automatically when needed.
- F8 for “common unequal” has now become F4.
- F10 has been replaced by Esc when you want to Quit `delta`. Instead, F10 stands for yes, ok.

Glossary of ambiguous terms

- Mode is used with 4 different meanings in this manual:
 1. the mode your VGA card puts your monitor in (e.g. 130 columns and 50 rows, see section 1.5)
 2. your printer mode (e.g. landscape mode, see section 2.7)
 3. `delta` overall mode, visual or non-visual (see section 4.2)
 4. display modes controlled by the toggles like `E` in the file browser (see section 2.4) or file comparison screen (see section 2.6)
- Options is used with 3 slightly different meanings in this manual:
 1. command line options, invoked at the same time `delta` is invoked (see section 4.2)
 2. the options menu where options are modified and possibly saved in a file, usually `delta.cfg` (see section 2.8)
 3. display options (which we've tried to call display toggles), like `E` to display space past the carriage return (see sections 2.4, 2.5 and 2.6)

A.3 OPENetwork's Other Products

Demo versions of our software, together with the best of our shareware and freeware can be found on the demo disk included with this manual or on our Home Page.

- The Berkeley Utilities are a set of 40 unix commands ported to MS-DOS. The commands are `awk`, `basename`, `cal`, `cat`, `cb`, `cmp`, `comm`, `cp`, `cut`, `df`, `diff`, `dtree`, `du`, `ech`, `ed`, `expand`, `find`, `grep`, `head`, `join`, `ls`, `make`, `more`, `mv`, `od`, `paste`, `rederr`, `rm`, `sed`, `sort`, `split`, `tail`, `tee`, `touch`, `tr`, `unexpand`, `uniq`, `wc`, `which` and `xstr`. They follow rigorously the unix System V syntax and include all the options found on any unix system plus a few carefully chosen ones (for an example, see section 4.4). They sell for \$200.

Why do we think you need The Berkeley Utilities if you are a power user or a programmer? Here are a few examples:

You wonder if all the "doc" files on your hard disk would fit on your somewhat filled diskette in the `A:` drive, so you just type

```
ls //*.doc -Ua
```

You want to create a file with the first 200 lines, the last 40 lines, the first 7 columns of file `abc.dat`, so you just type:

```
head -200 abc.dat > new
tail -40 abc.dat > new
cut -c1-7 abc.dat > new
```

You want to take all the “C” files on your hard disk which are less than seven days old and copy them to a diskette:

```
find -name *.c -mtime -7 -exec copy {} a: ;
```

You want to delete your Windows subdirectory or all the files with a filename ending with the letter a, so you type:

```
rm -r windows
rm *a.*
```

In order to achieve the second goal just mentioned, please don’t try MS-DOS’s

```
del *a.*
```

Where does the string xyz343 occur in my “C” code? Where does Smith, smith, Smyth or smyth occur in a “doc” file located in a demo subdirectory?

```
grep xyz343 /*.c
grep [Ss]m[iy]th //demo/*.doc
```

I have almost finished writing this manual (in LaTeX) and I want to know how many times I have used each English word:

```
tr -cs A-Za-z \n < manual.tex | sort | uniq -c
```

so that I now know that “You” appears 48 times while “you” appears 201 times!

This list is hardly comprehensive; it just tries to convey the flavor of The Berkeley Utilities.

- Mi-Shell is a programmable shell whose default look and feel resembles that of Norton Commander. It lets you totally control your MS-DOS environment, leading to significant time savings. Complete with a built-in browser, Mi-Shell script language and debugger, it is distributed as Shareware and can be registered for \$50.
- Star of Poland is a go-playing program which was among the top competing programs a few years ago. It plays around the 12 kyu level. \$79.
- BOREL, a research tool for serious Bridge players, explores by Monte Carlo the probabilities to see certain hands in situations you specify. \$200.

- OPEN Minesweeper, our version of Minesweeper, is free. It has many features not found on its Microsoft counterpart: saving games, scoring based on skill rather than luck, different playing modes, etc. Its author, John Lowenthal, has also written a very thorough book on how to play Minesweeper like a pro. The book can be purchased from OPENetwork.
- Life, our version of John Conway's mathematical concept, is free. It is not only beautiful and fast, it also includes scholarly references to this new field of mathematics.

A.4 OPENetwork's Team

Jean-Claude Chetrit, our President, consults in applied statistics for Fortune 500 Companies, runs the Brooklyn Go Club, writes the manuals and keeps learning from the masters he is lucky enough to run into.

Stephan Lugert, our emerging technology expert, works on an empirical study of the selfish gene hypothesis (via artificial life) when he is not fine-tuning all our PCs.

John Lowenthal, the author of `delta` version 1, does contract programming in C++/Windows and is a world class bridge player. He is working on a Windows version of DELTA.



Jean Michel, the author of `delta version 2`, is based in Paris. He is a research mathematician in group theory by profession, a world class go player and a butterfly collector. He programs in his spare time for relaxation.

Bill Buckley, of Antokas Graphics in the Village, is the talented artist responsible for the professional look of our covers, logos and ads.

Spencer Portée, our youngest recruit, is going to Polytech in Brooklyn. Of course, he is majoring in computer science.