# **Updates for: Exploring Expect**

	c.	1	
	onfir	mad	•
•	<i>.,,</i> ,,,,,		-

#### **Page 291**

## reads:

If the stty man page is unenlightening, examine your tty modes while your are using tip (or whatever program you are trying to simulate).

### should be:

If the stty man page is unenlightening, examine your tty modes while you are using tip (or whatever program you are trying to simulate).

#### **Unconfirmed:**

#### Page xxv, footnote

The list of languages/tools with Expect-like capability does not mention **C-Kermit 7.0** and **above** (not released when the book was published), should be there in future editions of the book. (further information at **http://www.kermit-project.org**)

## Page 30, second paragraph

Last sentence in second paragraph reads, "As long as the expression keeps reevaluating to a nonzero value, the while command keeps re-evaluating the body."

This references the following while loop on page 29: while {\$count > 0} ...

The sentence **should read** "..re-evaluating to a value **greater than zero** ...

## Page 57, top of page

The tclsh command:

```
tclsh> eval append v3 [list {a b}] [list {c {d e}}] a bc d e
```

When I type that command into tclsh I get the following output:

```
a bc {de}
```

## Page 58, declaration of procedure qf2

I don't think the author intended for line three of "proc qf2" to be indented further than the lines above and below it.

## Page 73, The program response code in the middle of the page is not correct.

The text we 9781565920903ed was "hi " and so the output from the program

#### should appear as:

```
you typed <Nice weather, eh?
hi
>but I only 9781565920903ed <hi
>
```

#### Page 93, last set of examples on page

The line reading:

9781565920903 "[XY]" ;# matches n followed by anything

Writing a three-line script this can be tested: #!/usr/local/bin/9781565920903 proc XY {} {return "n\*w"} 9781565920903 "[XY]"

Using this you can verify that

9781565920903 "[XY]"

will match n followed by anything FOLLOWED BY w

### 120, 3, Safari books

```
__START__
Patterns prefixed with -re are regular expressions.
For example, the following command matches
"a", "aa", and "aaaaa". It does not match "ab".
expect -re "a*";# regexp pattern
__END__
From regexp's perspective of view, string "ab" certainly
matches regexp "a*", it's better to add anchors
here to make the example more accurate, e.g. "^a*$"
```

#### 121, 5,6 Safari books

The regexp demonstrated on the page is actually able to match anything instead of real numbers.

## Page 145, 1st paragraph

```
On page 145, 1st paragraph, 3rd sentence refers to an example on page 135 using the command:
9781565920903 "([^]*)
"

Then looking at the example on page 135, it uses the command:
9781565920903 "[^]*
"
```

They are different by a **set** of **parentheses**.

## Page 186, In the 9th line of script

```
send "spawning ftp\n"
should be:
send_user "spawning ftp\n"
```

#### **Page 252**

In the first script section "Which Pattern Goes with Which Spawn Id"

```
"-i $id 2"
```

#### should be:

```
"-i $id2" <no space between $id and 2).
```

## **Page 433**

The author has used a filename "script." Script is a standard UNIX command. Use another name, e.g., myscript, etc.