Updates for *Practical Programming in Tcl and Tk: 4th edition*

Preface

pli 3rd bullet point in the Why Tcl? section should say "available on the Internet" instead of "available on the network".

plx 4th Edition Thanks section, 2nd paragraph. Third word from end of paragraph should be "and" not "an".

Chapter 3

p37 Example 3-3 botches the first line of output. It should be

```
puts "Content-Type: text/html"
```

p42 The first paragraph should say "loop variable name" instead of "loop variable x"

Chapter 4

p50 The string first description should say that -1 is returned if subString is not found (not string)

p52 The text next to the Hot Tip may be confusing. Kevin suggests this, which reads pretty well:

> "Despite the quotes, the expression operators that work on numbers and strings first try converting items to numbers if possible, and then the operators convert them back if they detect a case of string comparison."

I would add that it helps to know that Tcl represents values internally as *both* a string and some other native value, like a double-precision floating point, or 32-bit integer, or Tcl list, or byte code, or binary blob. Tcl Math expressions compute values that are native double-precision values, and are only converted to strings for printing. The Hot Tip is reminding us that there are some automatic conversions between native representations and strings, and in some subtle cases those conversions might suprise you. Tcl_Obj values are described in Chapter 47 on C programming.

p55 The caption for Example 4-4 says "Microsoft World" instead of "Microsoft Word".

p57 Near the bottom of the page, the format %#08x should result in:

```
0x00000a
```

because the field width of 8 includes the leading 0x.

Chapter 5

p65 Example 5-1. The last line of the example has an extra }. It should be:

```
 => {{1 2} {a b} {$foo}} {1 2}
```

Chapter Preface

pli 3rd bullet point in the Why Tcl? section should say "available on the Internet" instead of "available on the network".

plx Thanks section, 2nd paragraph. Third word from end of paragraph should be "and" not "an".

Chapter 3

p37 Example 3-3 botches the first line of output. It should be

```
puts "Content-Type: text/html"
```

p42 The first paragraph should say "loop variable name" instead of "loop variable x"

Chapter 4

p50 The string first description should say that -1 is returned if subString is not found (not string)

p52 The text next to the Hot Tip may be confusing. Kevin suggests this, which reads pretty well:

> "Despite the quotes, the expression operators that work on numbers and strings first try converting items to numbers if possible, and then the operators convert them back if they detect a case of string comparison."

I would add that it helps to know that Tcl represents values internally as *both* a string and some other native value, like a double-precision floating point, or 32-bit integer, or Tcl list, or byte code, or binary blob. Tcl Math expressions compute values that are native double-precision values, and are only converted to strings for printing. The Hot Tip is reminding us that there are some automatic conversions between native representations and strings, and in some subtle cases those conversions might suprise you. Tcl_Obj values are described in Chapter 47 on C programming.

p55 The caption for Example 4-4 says "Microsoft World" instead of "Microsoft Word".

p57 Near the bottom of the page, the format %#08x should result in:

```
0x00000a
```

because the field width of 8 includes the leading 0x.

Chapter 5

p65 Example 5-1. The last line of the example has an extra }. It should be:

```
 => {{1 2} {a b} {$foo}} {1 2}
```

Chapter 6

p78 There is a reference to regular expressions and Chapter 11 with a stray "page" in it. The closest page reference would be to page 145, Regular Expression Syntax.

p84 Example 6-15. The paragraph after this example says that the catch command *is not* grouped with curly braces, but it is. (It wasn't in earlier editions.) It turns out you can get away without the curly braces, as in:

```
if [catch { blah } result ] { # handle error }
```

But it is better style and slightly more efficient to always enclose the if expression in braces.

Chapter 7

p89 The cross reference for namespaces should be Chapter 14, not Chapter 7.

Chapter 8

p99 The second-to-last line says "Example 5-8 uses *on* array", which should be "Example 5-8 uses *one* array."

p100 In the last paragraph, the word implement should be implemented

p102 The first paragraph says "Example 8-10 defines RecordInsert" but it should say it defines RecordAppend.

Chapter 9

p109 Table 9-2. The file volumes command does not take a *name* argument.

p123 Example 9-11. The second foreach erroneously has curly braces around its [glob -nocomplain *] list value argument.

Chapter 10

p133 Example 10-2 has a typo in the rename command, which should be

```
rename $procName $procName-orig
```

p136 In the middle of the page, after

```
PackedButton .save "Save" { Save $file } {-side left}
```

The next two in-line examples should reference .save instead of .new. E.g., they should be:

```
button .save -text Save -command { Save $file }
pack .save -side left
```

Chapter 11

p150 In the last paragraph, the in-line reference to the pattern [[:<:]] is missing a right bracket.

p162 In Table 11-7 the pattern for hexadecimal numbers is incorrect. It includes the character ranges a-hA-H, which of course should be a-fA-F. This same bug also appears on **p164**.

Chapter 12

p174 In the first paragraph of the **Packages Implemented in C Code** section, there is an incorrect reference to Chapter 37. It should refer to Chapter 47.

Chapter 13

p191 In the paragraph before Example 13-6, the last sentence is missing an "and":
Chapter 10 describes callbacks *and* why the curly braces are used with eval as they are in this example:

Chapter 14

p211-212 The in-line examples for namespace code return a new command that begins with **::**namespace inscope. The book omits the leading ::, which is present to ensure the command is not ambiguous. If you evaluated that command in a namespace that (foolishly) had its own namespace command, you would not want to use that redefined namespace command.

Chapter 15

p221 In Conversions Between Encodings, the gb12345 encoding is incorrectly referred to as the Big5 encoding. Those are in fact two different encodings. See www.ascc.net

Chapter 17

p240 In Example 17-1 it is safer to cancel the after event when the socket becomes writeable, otherwise it could be possible for the after event to run after the fileevent and incorrectly change the value of the connected variable.

```
set ::afterid [after $timeout {set connected timeout}]
set sock [socket -async $host $port]
fileevent $sock w {after cancel $::afterid ; set connected ok}
vwait connected
```

Chapter 18

p265 In Example 18-5 the call to Mtype_Add has its arguments reversed. It should read:

```
Mtype_Add .junk application/myjunk
```

p275 In Example 18-16 the two uses of html::varEmpty should really be ncgi::empty

p276 In Example 18-17 there is an extra double-quote (") after addr2 in the arguments to Form_Simple

p278 In Table 18-2 the procedure name Url_PrefixInstall is misspelled.

Chapter 22

p355 Example 22-5 is missing the $ in the second reference to $starkit::topdir

p355 Example 22-6 doesn't create the "data" file. One way to do that would be

```
echo "Hello, World!" > write.vfs/data
```

Chapter 24

p378 Example 24-1 does not work as written on Windows because of limitations in pipes on that platform. Here is a version that works on Windows by using the cmd command processor. Note, however, that output is still batched up and so the fileevent is not really necessary.

p385 The CD-ROM version of Example 24-3 has an out of date cross reference to the Scrolled_Text procedure in a comment. The cross reference in the printed example is correct.

p393 The screen shots in examples 24-5, 24-6, 24-7 are not updated, so they incorrectly refer to Example 31-3, which is now really Example 37-3.

Chapter 25

p401 The second sentence in the first paragraph should read:
"Due to other constraints the request might not be honored, ..."

Chapter 26

p412-414 The screen shots in this chapter were taken on Windows where wish has a white default background. If you are on a Unix platform you will need to preface the examples with

```
. config -bg white
```

p415 Example 26-7 could usefully cross reference the numeric color specifications shown on page 622.

p416 Example 26-9 is supposed to be tried interactively so you can see the effect of each of the three grid commands. If you run the whole example you'll end up with the appearance in the lower right figure. In addition, the text of the labels should be "left side" and "right side".

p418 Example 26-11 is missing from the CD-ROM archive.

Chapter 33

p500 The CD-ROM version of Example 33-1 loads the "/etc/passwd" file, while the printed example loads the sample Tk file "colors.tcl".

p502 The CD-ROM version of Example 33-3 It is missing the three toplevel commands at the end of the example.

Chapter 34

p509 Table 34-1 does not list %S, which is "The text string being inserted/deleted, if any, {} otherwise."

p510 The paragraph before Example 34-3 should say "Example **34-3** demonstrates..."

Chapter 37

p558 At the top of the page, the tk scale command should be tk scaling

Chapter 39

p604 The bindings for Tab and Shift-Tab are not correct because tk_focusNext and tk_focusPrev don't actually set the focus (they did in an early Tk 4.0 beta...). The current bindings are now:

```
bind all <Tab> {tkTabToWindow [tk_focusNext %W]}
bind all <Shift-Tab> {tkTabToWindow [tk_focusPrev %W]}
```

p606 Using both Example 39-1 and 39-2 together reveals an inconsistency in the value of the variable used to wait. In 39-1 it is set to "cancel", but in 39-2 it expect it to be either "0" or "1". The simplest fix is to change Dialog_Wait in Example 39-1 to contain:

```
bind $top  [list set $varName 0]
```

Chapter 44

p660 In the first paragraph under the Icons section, the state operation returns iconic, not "iconified"

Chapter 41

p632 On Windows, the no cursor *should* result in no cursor being displayed. However, due to a bug in Tk 8.3 and 8.4, the cursor is a circle with a slash through it.

Chapter 46

p689 Example 46-7: The Bind_Save procedure takes args but is never passed anything. Instead, it should use the global bind(class) value.

```
proc Bind_Save { dotfile } {
  global bind
  set out [open $dotfile.new w]
  foreach w $bind(class) {
    # etc
  }
}
```

Chapter 47

p701 In Example 47-2, the sprintf call specifies buf as its first argument. That should be buffer.

Chapter 49

p765

In Example 49-15 the ClockEventProc should cancel the timer event associated with the TICKING flag in the case where it schedules the ClockDisplay. This avoids queuing up lots and lots of extra calls to the Display procedure.

redrawPreface

pli 3rd bullet point in the Why Tcl? section should say "available on the Internet" instead of "available on the network".

p**lx** 4th Edition Thanks section, 2nd paragraph. Third word from end of paragraph should be "and" not "an".

Chapter 3

p37 Example 3-3 botches the first line of output. It should be

```
puts "Content-Type: text/html"
```

p42 The first paragraph should say "loop variable name" instead of "loop variable x"

Chapter 4

p50 The string first description should say that -1 is returned if subString is not found (not string)

p52 The text next to the Hot Tip may be confusing. Kevin suggests this, which reads pretty well:

> "Despite the quotes, the expression operators that work on numbers and strings first try converting items to numbers if possible, and then the operators convert them back if they detect a case of string comparison."

I would add that it helps to know that Tcl represents values internally as *both* a string and some other native value, like a double-precision floating point, or 32-bit integer, or Tcl list, or byte code, or binary blob. Tcl Math expressions compute values that are native double-precision values, and are only converted to strings for printing. The Hot Tip is reminding us that there are some automatic conversions between native representations and strings, and in some subtle cases those conversions might suprise you. Tcl_Obj values are described in Chapter 47 on C programming.
p55 The caption for Example 4-4 says "Microsoft World" instead of "Microsoft Word".
p57 Near the bottom of the page, the format %#08x should result in:

```
0x00000a
```

because the field width of 8 includes the leading 0x.

Chapter 5

p65 Example 5-1. The last line of the example has an extra }. It should be:

```
 => {{1 2} {a b} {$foo}} {1 2}
```

Chapter 6

p78 There is a reference to regular expressions and Chapter 11 with a stray "page" in it. The closest page reference would be to page 145, Regular Expression Syntax.
p84 Example 6-15. The paragraph after this example says that the catch command *is not* grouped with curly braces, but it is. (It wasn't in earlier editions.) It turns out you can get away without the curly braces, as in:

```
if [catch { blah } result ] { # handle error }
```

But it is better style and slightly more efficient to always enclose the if expression in braces.

Chapter 7

p89 The cross reference for namespaces should be Chapter 14, not Chapter 7.

Chapter 8

p99 The second-to-last line says "Example 5-8 uses *on* array", which should be "Example 5-8 uses *one* array."
p100 In the last paragraph, the word implement should be implemented
p102 The first paragraph says "Example 8-10 defines RecordInsert" but it should say it defines RecordAppend.

Chapter 9

p109 Table 9-2. The file volumes command does not take a *name* argument.
p123 Example 9-11. The second foreach erroneously has curly braces around its [glob -nocomplain *] list value argument.

Chapter 10

p133 Example 10-2 has a typo in the rename command, which should be

```
rename $procName $procName-orig
```

p136 In the middle of the page, after

```
PackedButton .save "Save" { Save $file } {-side left}
```

The next two in-line examples should reference .save instead of .new. E.g., they should be:

```
button .save -text Save -command { Save $file }
pack .save -side left
```

Chapter 11

p150 In the last paragraph, the in-line reference to the pattern [[:<:]] is missing a right bracket.
p162 In Table 11-7 the pattern for hexadecimal numbers is incorrect. It includes the character ranges a-hA-H, which of course should be a-fA-F. This same bug also appears on **p164**.

Chapter 12

p174 In the first paragraph of the **Packages Implemented in C Code** section, there is an incorrect reference to Chapter 37. It should refer to Chapter 47.

Chapter 13

p191 In the paragraph before Example 13-6, the last sentence is missing an "and":
Chapter 10 describes callbacks *and* why the curly braces are used with eval as they are in this example:

Chapter 14

p211-212 The in-line examples for namespace code return a new command that begins with **::**namespace inscope. The book omits the leading ::, which is present to ensure the command is not ambiguous. If you evaluated that command in a namespace that (foolishly) had its own namespace command, you would not want to use that redefined namespace command.

Chapter 15

p221 In Conversions Between Encodings, the gb12345 encoding is incorrectly referred to as the Big5 encoding. Those are in fact two different encodings. See www.ascc.net

Chapter 17

p240 In Example 17-1 it is safer to cancel the after event when the socket becomes writeable, otherwise it could be possible for the after event to run after the fileevent and incorrectly change the value of the connected variable.

```
set ::afterid [after $timeout {set connected timeout}]
set sock [socket -async $host $port]
fileevent $sock w {after cancel $::afterid ; set connected ok}
vwait connected
```

Chapter 18

p265 In Example 18-5 the call to Mtype_Add has its arguments reversed. It should read:

```
Mtype_Add .junk application/myjunk
```

p275 In Example 18-16 the two uses of html::varEmpty should really be ncgi::empty
p276 In Example 18-17 there is an extra double-quote (") after addr2 in the arguments to Form_Simple
p278 In Table 18-2 the procedure name Url_PrefixInstall is misspelled.

Chapter 22

p355 Example 22-5 is missing the $ in the second reference to $starkit::topdir
p355 Example 22-6 doesn't create the "data" file. One way to do that would be

```
echo "Hello, World!" > write.vfs/data
```

Chapter 24

p378 Example 24-1 does not work as written on Windows because of limitations in pipes on that platform. Here is a version that works on Windows by using the cmd command processor. Note, however, that output is still batched up and so the fileevent is not really necessary.

p385 The CD-ROM version of Example 24-3 has an out of date cross reference to the Scrolled_Text procedure in a comment. The cross reference in the printed example is correct.

p393 The screen shots in examples 24-5, 24-6, 24-7 are not updated, so they incorrectly refer to Example 31-3, which is now really Example 37-3.

Chapter 25

p401 The second sentence in the first paragraph should read:
"Due to other constraints the request might not be honored, ..."

Chapter 26

p412-414 The screen shots in this chapter were taken on Windows where wish has a white default background. If you are on a Unix platform you will need to preface the examples with

```
. config -bg white
```

p415 Example 26-7 could usefully cross reference the numeric color specifications shown on page 622.
p416 Example 26-9 is supposed to be tried interactively so you can see the effect of each of the three grid commands. If you run the whole example you'll end up with the appearance in the lower right figure. In addition, the text of the labels should be "left side" and "right side".

p**418** Example 26-11 is missing from the CD-ROM archive.

Chapter 33

p500 The CD-ROM version of Example 33-1 loads the "/etc/passwd" file, while the printed example loads the sample Tk file "colors.tcl".
p502 The CD-ROM version of Example 33-3 It is missing the three toplevel commands at the end of the example.

Chapter 34

p509 Table 34-1 does not list %S, which is "The text string being inserted/deleted, if any, {} otherwise."
p510 The paragraph before Example 34-3 should say "Example **34-3** demonstrates..."

Chapter 37

p558 At the top of the page, the tk scale command should be tk scaling

Chapter 39

p604 The bindings for Tab and Shift-Tab are not correct because tk_focusNext and tk_focusPrev don't actually set the focus (they did in an early Tk 4.0 beta...). The current bindings are now:

```
bind all <Tab> {tkTabToWindow [tk_focusNext %W]}
bind all <Shift-Tab> {tkTabToWindow [tk_focusPrev %W]}
```

p606 Using both Example 39-1 and 39-2 together reveals an inconsistency in the value of the variable used to wait. In 39-1 it is set to "cancel", but in 39-2 it expect it to be either "0" or "1". The simplest fix is to change Dialog_Wait in Example 39-1 to contain:

```
bind $top  [list set $varName 0]
```

Chapter 44

p660 In the first paragraph under the Icons section, the state operation returns iconic, not "iconified"

Chapter 41

p632 On Windows, the no cursor *should* result in no cursor being displayed. However, due to a bug in Tk 8.3 and 8.4, the cursor is a circle with a slash through it.

Chapter 46

p689 Example 46-7: The Bind_Save procedure takes args but is never passed anything. Instead, it should use the global bind(class) value.

```
proc Bind_Save { dotfile } {
  global bind
  set out [open $dotfile.new w]
  foreach w $bind(class) {
    # etc
  }
}
```

Chapter 47

p701 In Example 47-2, the sprintf call specifies buf as its first argument. That should be buffer.

Chapter 49

p765 In Example 49-15 the ClockEventProc should cancel the timer event associated with the TICKING flag in the case where it schedules the ClockDisplay. This avoids queuing up lots and lots of extra calls to the Display procedure.

```
redraw:
    if ((clockPtr->tkwin != NULL) &&
            !(clockPtr->flags & REDRAW_PENDING)) {

        if (clockPtr->flags & TICKING) {
            Tk_DeleteTimerHandler(clockPtr->token);
            clockPtr->flags &= ~TICKING;
        }

        Tk_DoWhenIdle(ClockDisplay, (ClientData) clockPtr);
        clockPtr->flags |= REDRAW_PENDING;
    }
```

Index

p837 There are no entries for < or > under Symbols. These are used as event delimiters on page 439.

p843 The clock command is missing from the index, nor is it listed as a sub-entry under the Tcl Command index entry. However, clock is listed in the command index on page 22, and is described starting on page 183.

p838 Entry for "arrow on canvas" should reference page 572.

p849 Entry for "file: is plain" is really about "file: isfile. This confusion shows up for the "plain file" cross reference, which should really be a "see also" reference.

p856 Entry for "lsearch, Tcl command" should reference page 69.


```
    if ((clockPtr->tkwin != NULL) &&
            !(clockPtr->flags & REDRAW_PENDING)) {

        if (clockPtr->flags & TICKING) {
            Tk_DeleteTimerHandler(clockPtr->token);
            clockPtr->flags &= ~TICKING;
        }

        Tk_DoWhenIdle(ClockDisplay, (ClientData) clockPtr);
        clockPtr->flags |= REDRAW_PENDING;
    }
```

Index

p837 There are no entries for < or > under Symbols. These are used as event delimiters on page 439.

p843 The clock command is missing from the index, nor is it listed as a sub-entry under the Tcl Command index entry. However, clock is listed in the command index on page 22, and is described starting on page 183.

p838 Entry for "arrow on canvas" should reference page 572.

p849 Entry for "file: is plain" is really about "file: isfile. This confusion shows up for the "plain file" cross reference, which should really be a "see also" reference.

p856 Entry for "lsearch, Tcl command" should reference page 69.

6

p78 There is a reference to regular expressions and Chapter 11 with a stray "page" in it. The closest page reference would be to page 145, Regular Expression Syntax.

p84 Example 6-15. The paragraph after this example says that the catch command *is not* grouped with curly braces, but it is. (It wasn't in earlier editions.) It turns out you can get away without the curly braces, as in:

```
if [catch { blah } result ] { # handle error }
```

But it is better style and slightly more efficient to always enclose the if expression in braces.

Chapter 7

p89 The cross reference for namespaces should be Chapter 14, not Chapter 7.

Chapter 8

p99 The second-to-last line says "Example 5-8 uses *on* array", which should be "Example 5-8 uses *one* array."

p100 In the last paragraph, the word implement should be implemented

p102 The first paragraph says "Example 8-10 defines RecordInsert" but it should say it defines RecordAppend.

Chapter 9

p109 Table 9-2. The file volumes command does not take a *name* argument.

p123 Example 9-11. The second foreach erroneously has curly braces around its [glob -nocomplain *] list value argument.

Chapter 10

p133 Example 10-2 has a typo in the rename command, which should be

```
rename $procName $procName-orig
```

p136 In the middle of the page, after

```
PackedButton .save "Save" { Save $file } {-side left}
```

The next two in-line examples should reference .save instead of .new. E.g., they should be:

```
button .save -text Save -command { Save $file }
pack .save -side left
```

Chapter 11

p150 In the last paragraph, the in-line reference to the pattern [[:<:]] is missing a right bracket.

p162 In Table 11-7 the pattern for hexadecimal numbers is incorrect. It includes the character ranges a-hA-H, which of course should be a-fA-F. This same bug also appears on **p164**.

Chapter 12

p174 In the first paragraph of the **Packages Implemented in C Code** section, there is an incorrect reference to Chapter 37. It should refer to Chapter 47.

Chapter 13

p191 In the paragraph before Example 13-6, the last sentence is missing an "and":
Chapter 10 describes callbacks *and* why the curly braces are used with eval as they are in this example:

Chapter 14

p211-212 The in-line examples for namespace code return a new command that begins with **::**namespace inscope. The book omits the leading ::, which is present to ensure the command is not ambiguous. If you evaluated that command in a namespace that (foolishly) had its own namespace command, you would not want to use that redefined namespace command.

Chapter 15

p221 In Conversions Between Encodings, the gb12345 encoding is incorrectly referred to as the Big5 encoding. Those are in fact two different encodings. See www.ascc.net

Chapter 17

p240 In Example 17-1 it is safer to cancel the after event when the socket becomes writeable, otherwise it could be possible for the after event to run after the fileevent and incorrectly change the value of the connected variable.

```
set ::afterid [after $timeout {set connected timeout}]
set sock [socket -async $host $port]
fileevent $sock w {after cancel $::afterid ; set connected ok}
vwait connected
```

Chapter 18

p265 In Example 18-5 the call to Mtype_Add has its arguments reversed. It should read:

```
Mtype_Add .junk application/myjunk
```

p275 In Example 18-16 the two uses of html::varEmpty should really be ncgi::empty

p276 In Example 18-17 there is an extra double-quote (") after addr2 in the arguments to Form_Simple

p278 In Table 18-2 the procedure name Url_PrefixInstall is misspelled.

Chapter 22

p355 Example 22-5 is missing the $ in the second reference to $starkit::topdir
Rainer Schwarzinger

p355 Example 22-6 doesn't create the "data" file. One way to do that would be

```
echo "Hello, World!" > write.vfs/data
```

Chapter 24

p378 Example 24-1 does not work as written on Windows because of limitations in pipes on that platform. Here is a version that works on Windows by using the cmd command processor. Note, however, that output is still batched up and so the fileevent is not really necessary.

p385 The CD-ROM version of Example 24-3 has an out of date cross reference to the Scrolled_Text procedure in a comment. The cross reference in the printed example is correct.

p393 The screen shots in examples 24-5, 24-6, 24-7 are not updated, so they incorrectly refer to Example 31-3, which is now really Example 37-3.

Chapter 25

p401 The second sentence in the first paragraph should read:
"Due to other constraints the request might not be honored, ..."

Chapter 26

p412-414 The screen shots in this chapter were taken on Windows where wish has a white default background. If you are on a Unix platform you will need to preface the examples with

```
. config -bg white
```

p415 Example 26-7 could usefully cross reference the numeric color specifications shown on page 622.

p416 Example 26-9 is supposed to be tried interactively so you can see the effect of each of the three grid commands. If you run the whole example you'll end up with the appearance in the lower right figure. In addition, the text of the labels should be "left side" and "right side".

p418 Example 26-11 is missing from the CD-ROM archive.

Chapter 33

p500 The CD-ROM version of Example 33-1 loads the "/etc/passwd" file, while the printed example loads the sample Tk file "colors.tcl".

p502 The CD-ROM version of Example 33-3 It is missing the three toplevel commands at the end of the example.

Chapter 34

p509 Table 34-1 does not list %S, which is "The text string being inserted/deleted, if any, {} otherwise."

p510 The paragraph before Example 34-3 should say "Example **34-3** demonstrates..."

Chapter 37

p558 At the top of the page, the tk scale command should be tk scaling

Chapter 39

p604 The bindings for Tab and Shift-Tab are not correct because tk_focusNext and tk_focusPrev don't actually set the focus (they did in an early Tk 4.0 beta...). The current bindings are now:

```
bind all <Tab> {tkTabToWindow [tk_focusNext %W]}
bind all <Shift-Tab> {tkTabToWindow [tk_focusPrev %W]}
```

p606 Using both Example 39-1 and 39-2 together reveals an inconsistency in the value of the variable used to wait. In 39-1 it is set to "cancel", but in 39-2 it expect it to be either "0" or "1". The simplest fix is to change Dialog_Wait in Example 39-1 to contain:

```
bind $top  [list set $varName 0]
```

Chapter 44

p660 In the first paragraph under the Icons section, the state operation returns iconic, not "iconified"

Chapter 41

p632 On Windows, the no cursor *should* result in no cursor being displayed. However, due to a bug in Tk 8.3 and 8.4, the cursor is a circle with a slash through it.

Chapter 46

p689 Example 46-7: The Bind_Save procedure takes args but is never passed anything. Instead, it should use the global bind(class) value.

```
proc Bind_Save { dotfile } {
  global bind
  set out [open $dotfile.new w]
  foreach w $bind(class) {
    # etc
  }
}
```

Chapter 47

p701 In Example 47-2, the sprintf call specifies buf as its first argument. That should be buffer.

Chapter 49

p765 In Example 49-15 the ClockEventProc should cancel the timer event associated with the TICKING flag in the case where it schedules the ClockDisplay. This avoids queuing up lots and lots of extra calls to the Display procedure.

```
redraw:
    if ((clockPtr->tkwin != NULL) &&
            !(clockPtr->flags & REDRAW_PENDING)) {

        if (clockPtr->flags & TICKING) {
            Tk_DeleteTimerHandler(clockPtr->token);
            clockPtr->flags &= ~TICKING;
        }

        Tk_DoWhenIdle(ClockDisplay, (ClientData) clockPtr);
        clockPtr->flags |= REDRAW_PENDING;
    }
```

Index

p837 There are no entries for < or > under Symbols. These are used as event delimiters on page 439.

p843 The clock command is missing from the index, nor is it listed as a sub-entry under the Tcl Command index entry. However, clock is listed in the command index on page 22, and is described starting on page 183.

p838 Entry for "arrow on canvas" should reference page 572.

p849 Entry for "file: is plain" is really about "file: isfile. This confusion shows up for the "plain file" cross reference, which should really be a "see also" reference.

p856 Entry for "lsearch, Tcl command" should reference page 69.