Wonderful: A Terrific Application and Fascinating Paper

Your N. Here
Your Department
Your Institution
Your City, State, ZIP
yourname@host.dom
http://host.dom/yoururl

Name Two
Two's Institution
two@host.dom

Submitted to Cool Stuff Conference 2002

Abstract

Your abstract text goes here. Just a few facts. Whet our appetites.

1 Introduction

A paragraph of text goes here. Lots of text. Plenty of interesting text. Lots of text. Lots of text. Lots.

More fascinating text. Features galore, plethora of promises.

2 This is Another Section

Some embedded literal typeset code is shown below. Note that line or page breaks can occur in the middle of code typeset this way. To avoid such line or page breaks, put the code inside a figure environment instead.

```
int wrap_fact(ClientData clientData,
              Tcl_Interp *interp,
              int argc, char *argv[]) {
    int result;
    int arg0;
    if (argc != 2) {
        interp->result = "wrong # args";
        return TCL_ERROR;
    }
    arg0 = atoi(argv[1]);
    result = fact(arg0);
    sprintf(interp->result,"%d",result);
    return TCL_OK;
}
```

Now we're going to cite somebody. Watch for the cite tag. Here it comes [5, 7, 8]. And a bit later we will cite another one. Stay tuned [6].

3 This Section has Sub-Sections

This text is the introduction to Section 3.

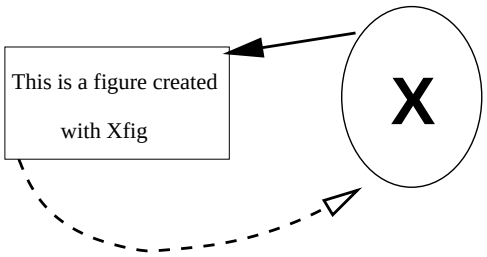

Figure 1: This figure was created with `xfig`. If you want it to span two columns, use `figure*` in the LaTeX source file.

```
<insert figure here>
```

Figure 2: Wonderful flowchart

3.1 First Sub-Section

Here's a typical figure reference. Figure 2 is centered at the top of the column. It may be scaled. If so, you may have to tweak the numbers to get the size you want. It may be hard to do this.

This text came after the figure, so we'll casually refer to Figure 2 as we go on our merry way.

3.2 Footnotes

For the Usenix style, footnotes are not allowed: endnotes are, although they are deprecated.[1] Try to avoid both footnotes and endnotes in technical writing. It is best to use parenthetical or subordinate clauses instead. If you want endnotes anyhow, use the "endnotes" documentstyle option and include a `\makeendnotes` command at the end of your document. You will still be whined at.

3.3 Tables and Code

It can get tricky typesetting Tcl and C code in LaTeX because they share a lot of mystical feelings about certain magic characters. You will have to do a lot of escaping to typeset curly braces and percent signs, for example,

Cloak	User `ezk`				User `joe`						Meaning
Mask	J1	J2	J3	J4	E5	E6	E7	E8	E9	E10	for files J1–E10
+000											Show files to owners only
+007			A				A	A			Show files to owners and others
+070		A	A		A		A	A			Show files to owners and group members

Table 1: Here is a complex table that spans two columns. It shows how also to straddle the table cells.

like this: "The `%module` directive sets the name of the initialization function. This is optional, but is recommended if building a Tcl 7.5 module. Everything inside the `%{`, `%}` block is copied directly into the output. allowing the inclusion of header files and additional C code."

Sometimes you want to really call attention to a piece of text. You can center it in the column like this:

```
_1008e614_Vector_p
```

and people will really notice it.

Now this is an ingenious way to get a forced space. `Real *` and `double *` are equivalent.

3.4 Lists

You can make lists using LaTeX's listing environments (`itemize`, `enumerate`, and `description`). These environments can be nested (e.g. an itemized list can be an element of an enumerated list).

An `itemize` list looks like this:

- The map structure defines an address space.
- The page structure manages a page of physical memory.

An `enumerate` list is like an itemized list, except that it is numbered:

1. The map structure defines an address space.
2. The page structure manages a page of physical memory.

A `description` list uses words rather bullets or numbers:

map structure: defines an address space.
page structure: manages a page of physical memory.

3.5 Last Sub-Section

Well, it's getting boring isn't it. This is the last subsection before we wrap it up.

4 Acknowledgments

A polite author always includes acknowledgments. You should thank everyone, especially those who funded the work.

5 Availability

It is great news if this section can say that your app, WonderfulApp is free software, available via anonymous FTP from `ftp://ftp.dom/pub/myname/Wonderful`. Also, it's even better when you can write that information is also available on the Wonderful homepage at `http://www.dom/~myname/SWIG`.

Now we get serious and fill in those references. Remember you will have to run latex twice on the document in order to resolve those cite tags you met earlier. This is where they get resolved. We've preserved some real ones in addition to the template-speak. After the bibliography you are DONE.

Notes

1. Thus, this is not a footnote

References

[1] D. M. Beazley and P. S. Lomdahl, *Message-Passing Multi-Cell Molecular Dynamics on the Connection Machine 5*, Parall. Comp. 20 (1994) p. 173-195.

[2] A. N. Author and A. N. Other, *Title of Riveting Article*, JournalName VolNum (Year) p. Start-End

[3] Embedded Tk, `ftp://ftp.vnet.net/pub/users/drh/ET.html`

[4] Don Libes, *Exploring Expect*, O'Reilly & Associates, Inc. (1995).

[5] Wolfgang Heidrich and Philipp Slusallek, *Automatic Generation of Tcl Bindings for C and C++ Libraries.*, USENIX 3rd Annual Tcl/Tk Workshop (1995).

[6] John K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley Publishers (1994).

[7] Perl5 Programmers reference, `http://www.metronet.com/perlinfo/doc`, (1996).

[8] D. Wetherall, C. J. Lindblad, "Extending Tcl for Dynamic Object-Oriented Programming", Proceedings of the USENIX 3rd Annual Tcl/Tk Workshop (1995).