CSE 560 **Winter 2005**
Prof. Paul Sivilotti

Writing A User's Guide

Distributed: Friday, January 7th.

The user's guide should explain *what your program does*, and how a user can get the program to do it. Our hypothetical user does not need to know why you wrote the program. They do not care about the internal implementation aspects of your program. They simply want to do X and want to find out if your program can do it. *Based on this part of your writeup, a user must be able to install your program, make it run, and be able to understand its output, error messages and all.*

As a specific example, consider error messages or warning issued by your program. Understanding the meaning and cause of such a message should *not* depend on understanding the program's algorithm or implementation. The message (and associated documentation in the user's guide) should be informative enough to the user for the problem to be fixed without examining (or even thinking about too carefully) *how* the program came to generate this message.

When describing what your program does, it is not necessary, or even desirable, to copy the original requirements verbatim into your documentation. You will, presumably, find that the lab assignments themselves are not good user's guides. They're not meant to be! It would be surprising, then, to find the prose or figures in these handouts appearing verbatim in a user's guide. Keep your intended audience in mind and present the information in a manner most appropriate to that audience.

Also, there is considerable latitude in what features to include in your programs and what interactions with the user should be supported. You will all, therefore, have many original things to say about what your program does. Note that when you are working in a group (as you are in this course), it is important to have this part of your documentation done very early so that everyone in the group is working from the same requirements.

The following sections give some suggestions for content of your user's guide. Whether or not you incorporate these ideas and others depends on the nature of your program and how you decide to structure your user's guide.

Front Matter

This document should be viewed as a stand-alone reference (like the programmer's guide). Therefore, it has similar requirements for front matter: a descriptive title page, a useful table of contents (and list of figures/tables as appropriate), and a concise introduction.

Getting Started

The opening sections should describe the basic information for installing the software. This basic information includes requirements on the environment, platform, compiler, memory, disk space, etc. It also includes specific instructions for installing the software properly and how to configure one's system appropriately (e.g., environment variables that must be set, directories that should be created, and paths to establish).

Some user's guides have a "Quick Guide" section for people who are too impatient to read the documentation! Some user's guides give a brief and simple example to introduce, in a general way, the operation of the program. Some user's guides give background information that is needed to understand the operation of the program.

Running the Program

The main part of any user's guide is its description of how to run the program. This description will include the basic commands and command syntax, including, for example, command-line arguments and their interpretation.

Sometimes, it is useful to distinguish between basic and advanced features. The former might be used quite often while the later are used only infrequently. This may affect how their description is presented in the user's guide. Some features are optional, their presence depends on decisions made at installation. These kinds of relationships should be made clear to the reader.

Descriptions of the inputs to and outputs from your program, including their formats, are essential in a user's guide. That is, the required syntax of input files must be carefully documented. Furthermore, the relationship between correct input and the generated output should also be clear from the user's guide.

For programs with user interaction, command syntax and function must also be carefully documented.

The user's guide should be clear, complete, and concise. Pictures and diagrams can be very helpful. Including example sessions, with input, output, and user interaction can also convey considerable information. Some user's guides include screen shots.

If there are limitations to the program that the typical user might find significant or surprising, these should be described as well.

Errors and Error Messages

All error messages that the program can generate should be listed in a way that they can be quickly found. The user's guide should describe the cause of the error condition, or give a list of possible causes if there is more than one. It may also be appropriate for the user's guide to suggest ways to correct the problem.