



Center for Clinical Trials

*Department of Biostatistics
Department of Epidemiology
Department of International Health*

*Department of Medicine
Department of Ophthalmology
Oncology Center*

Wednesday, 20 July 2005

Memorandum

To: Center for Clinical Trials Students, Staff, and Faculty

Fr: Curtis Meinert

Re: Tables 101: The Table Maker's Oath

I attach "The Table Maker's Oath". For a document of mysterious origin, I am dismayed at the extent to which it reflects my own views on "table making".

The first two items in the "whereas" preamble are obvious. Just look around. Find me a magazine or book printed in landscape orientation. Find me a "turn" in Time Magazine. Show me a "turn" in US Today and I will treat you and your loved one to lunch at Mama Mia's. Imagine the trouble you would have reading a newspaper in the sardine section en route to Chicago if newspapers required turns. The world is portrait-oriented. No doubt about it!

The world is also one-sided when it comes to working documents. Show me a court brief printed on both sides of a page. Show me a set of blueprints and I will show you a one-sided copy job. Look at a draft manuscripts and see if you can find one printed on both sides.

You want confusion? Have people on a treatment effects monitoring committee deal with a two-sided report. Give a two-sided document to someone to copy and more likely than not what comes back will be half a report.

Save a nickel and spend a dollar. That is what you are doing when you "save" paper by printing working documents on both sides.

The eye is indeed an exquisite graphic instrument. The trained eye of an artisan often is a better "measuring tool" than the fancy instruments the artisan has. A good carpenter can match an angled cut simply by looking. But the eye, to work, has to have clues. In table construction that means paying attention to spacing.

The goal in table construction is to make the use of written text unnecessary and superfluous because all the information needed is conveyed in the table by its title, its headers, its stubs, and its explanatory notes.

Now, as to the "NEVER" list: I am inclined to encourage readers to pay close attention to the first several entries in the list. I abhor documents requiring "turns" for reasons already made clear.

I expect heaven for table makers is a place where paper is on a roll, where they can produce tables as wide as they like, and where monitoring reports come in rolls. There is no doubt that the TP design would be emancipating but here on earth we are limited by the 8.5"x11" defining a standard page. So down here forget turning the page on its long axis to "buy" more space. If it does not fit portrait style, redesign to make it fit!

My SWAG estimate of the median age of eyes in treatment effects monitoring committees is 59.8 and the SWAG for table producers is 27.4 years of age. Hence, just because the producer can read all the fine print used in a table to "save" lines and space does not mean those who need to read the table will be able to do so. Nothing smaller than 11 point in tables save, perhaps, footer information!

An ever increasing frustration as one's eyes age, is the "diminishing" size of fonts with time. I hate reports where the most important information – the explanatory notes to a table – are printed in 8 pt font. If we continue that practice we should be providing magnifying glasses with every report we produce.

A definite No No is managing page breaks by use of the hard return key or hard page breaks.

Why? Because the practice is counter to robust design. The table, as produced, will look fine, but add or delete lines and the break is likely to be "bad". Let the system manage breaks by use of block protect and conditional end of page codes.

If I had my way I would disable the space bar on PCs as soon as the table maker enters the body of a table. Use of the space bar in the body of a table is usually a sign that the maker did not set tabs appropriately.

Similarly, except for the stub of a table, there should not be the need to ever press the tab key more than once in the body of a table. Pressing the key more than once generally means that the maker is using the tab key to move the cursor. It means that the tab settings are inadequate. Go back and reset!

So who cares so long as the document looks OK when printed? Because the extra code (rogue code) is an accident waiting to happen. If you do not believe me just produce a table with extra tab settings. Then change the font size and print the document. The likelihood is that it will be "messed up".

We read from left to right but work from right to left when adding and subtracting numbers. Think back to your days in grade school. Suppose your teacher asked you to add 14, 7, 123, and 22. What is the first thing you did? You wrote the numbers in a column. First, you might have written:

```
14
 7
123
22
```

but you would have soon discovered or been told by your teacher that you should have right aligned as below:

```
14
 7
123
22
```

Similarly, had you been given the task of adding decimal numbers like 22.4, 17.8, and 83.4 you would have right aligned to yield:

```
22.4
17.8
83.4
```

and gone merrily on your way.

So now suppose the numbers are 12.4, 77.98, 1.444, 33, and 1056.6. You right align:

```
12.4
77.98
1.444
 33
1056.6
```

Obviously right alignment is not enough. To add, you have to right align around the decimal number:

```
12.4
77.98
 1.444
 33
1056.6
```

So, OK. If this is what we learned way back before we had remote controls, why do we forget it when producing number tables?

How do I know it is forgotten? By the tables I see with arithmetic numbers in columns left aligned or centered. Anybody here who produces a table like that will have to take me to Mama Mia's for lunch and will automatically be a front runner for the Center watch.

When things do not quite "fit" across a page the "fix" is to reduce the right and left margin. If you need an extra line or two you change the top and bottom margin and move on happily. But tell me. What would books look like if contributors were free to set their own margins?

Like it or not, we are in the "book" business to the extent that our documents for meetings and monitoring committees take the form of "bound" books, even if only "O ringed". People producing tables here have to have a "book mentality". That is to say they have to assume that what they are producing will ultimately find its way into a "bound" report. So do not "cheat" by changing the margins. Live within the space allocated as dictated by overall settings for the master document.

If I had my way, I would get Mike to connect the two parentheses keys on keyboards of

table makers to a klaxon so that every time they pressed either key in the body of a table the klaxon would sound.

Parentheses are used to label material as superfluous or as "explanatory". In text they usually mean that we can read around the parentheses and miss nothing. Why clutter tables with superfluous material? If the numbers being displayed are not important why are they taking up space in the table? If they are important why are they entombed in parentheses? The added clutter makes them harder to read.

A trialist is a relativist and, hence, prefers relative to absolute numbers. Percentages are wonderful relative numbers – the best! By and large, percentages in trials (eg, as percentages of persons across clinics or treatment groups completing a given visit) are more informative and easier to understand than absolute numbers. So why then do we insist on producing tables with percentages in parentheses? Stop that!

Again, if I had my way I would institute a system in which anyone producing a table, absent proper footers, would be required to write "I produced a table without proper footer information" 100 times. The second time the person would be required to wear a dunce cap and the third time around would be required to parade in front of the School wearing a board placard announcing the failure to have produced proper footers.

Everyone producing tables needs to follow the H&G model for finding home. Tables are living breathing documents. There is virtually no table that does not get modified countless times on its way to use, to say nothing about modification after use. If the footer date is not changed after each revision there will be no way to distinguish it from the previous version (save by "eye balling" the tables).

The other big problem is finding the electronic version of the table without proper footer information. An essential "crumb" is the file name and path. It must appear on the print version to find back home!

Mark knows well my disdain for "hot wired" dates in footers (a date that is automatically changed when the document is retrieved to the screen and when printed). The automatic dating feature should NEVER, NEVER, EVER be used to "manage" footer dates denoting creation or revision dates. A use that persists (to my dislike) is for "print date" but who cares when a table is printed? The only date that matters is when it was created or last revised.

If I was smart enough I would invent a RCD (rogue code detector) and require table makers to run it after creation or any revision of a table. Basically, rogue code is code having no function in a document; generally debris left over from previous versions of the document. Ideally, one would strip away all unused codes when "finishing" a document, but being too fastidious in this regard can cause iatrogenic disease. Some old code can give clues to finding back home or the route taken from home, ala the H&G model. Further, by leaving some "setting code" you can walk backwards by simply deleting the existing setting to fall back to the previous setting, eg, as in deleting the existing tab setting to fall back to the previous setting.

In theory, a good housekeeping practice is to strip "finished" documents of unused styles to produce a "lean" document. The more styles the more "sluggish" the retrieval processing becomes.

However, generally the only time one starts to notice "sluggishness" is when assembling master documents.

Styles in WP are "crossed". That is to say if one document has five styles and another document has five different styles (regardless of whether or not used in the documents), the associated style library in the master file consisting of the two documents will contain ten styles. At some point, generally near the "final" assembly process, it may be useful to strip the master file of unused styles and then to condense, thereby saving the individual files in "style lean" mode.

But a word of caution! Make certain before you start down this road that you have scattered sufficient crumbs to allow you to get back to Schneider should bad things happen. One other word of caution. Once the base document is "style stripped" and saved you will not have access to unused styles unless you go back to the original style library.

The last four items in the "NEVER" list are elementary. All you have to have is a single instance in which "math" malfunctions (invariably because you failed to connect the math feature with the appropriate columns) before you check, check, check.

As far as I am concerned Word is a disaster. The fact that you have to work in the dark (no access to underlying codes) makes it an electronic form of Russian Roulette. If I were in charge of the world I would rid it of Word, but alas I am not in charge. In reality it is a Word world. So increasingly we feel compelled to convert WP documents to Word for the convenience of the world when electronically transmitted. All I can say is "Be careful"! The conversion can change the data!

Most items in the "ALWAYS" list are obvious. An essential element in the H&G approach to table building is "saving" early and often. But that said, who among us has not "walked" on ourselves by modifying a table and saving it to its original name and only then to discover we need the old version!

The biggest deterrent to good table production to my way of thinking is the table feature in WP. It can be made to produce decent tables, but not if the maker just takes the default settings. Indeed, it is those settings that leads to left aligned arithmetical numbers in tables.

If one is to take advantage of the graphic nature of the eye, you have to forego uniform spacing between columns. You will automatically get that kind of spacing with the table feature, unless you change the setting. So pay me now or pay me later. If you want to build decent robust tables you should build them yourself – the old fashioned way – by setting tabs. Forget the "cell" method of construction.

In regard to spacing: The "Total" column (generally the last or first column in the collection of columns in the body of a table) should be separated by more space than that separating the other columns. Similarly, in tables having two or more columns under a treatment group header, the space separating those columns should be less than the separation for the adjoining sets of columns relating to other treatment groups.

Likewise, lines in tables intended to convey different information than lines above should be set off from the lines above by added space or by bolding, eg, as with "Total" lines.

I swear that merely looking at a document introduces spelling errors. How else can one explain mis-spelled words in documents repeatedly spell checked?

I can tell you how. You introduced the errors when you modified the document and left them undetected by failure to spell check before saving. Get in the habit of spell checking whenever you save or re-save the document and follow that practice, regardless of how minor the change.

It is said that "When you assume you make an ass out of u and me". There is a good chance of doing that whenever you hand off a table without first printing it to make certain it "looks OK". But printing just for a "look see" is not enough. You have to "FOTD read". You cannot reliably read through the knot hole we call screens. To read with a critical eye you have to move away from the computer, put your feet on the desk, and read (and even then you will have errors that escape you!).

VTables.101\TblOath.WPD

Enclosure

The Table Maker's Oath

Distribution

Debbie Amend-Libercci
Ming-Wen An
Jeannette Beasley
Pat Belt
Elena Blasco-Colemanares
Cathy Bosley
Rob Casper
Hui-Ming Chung
Betty Collison
Ryan Colvin
Kay Dickersin
John Dodge
Michele Donithan
Lea Drye
Ann Ervin
Ingrid Friberg
Julia Gage
Judy Harle
Janet Holbrook
Rosemary Hollick

Milana Isaacson
Rosetta Jackson
Jennifer Jones
Charlene Levine
Simon Liu
Hope Livingston
Nancy Maldeis
Barbara Martin
Reena Masih
Curtis Meinert
Jill Meinert
Wai Ping Ng
Deborah Nowakowski
Kapreena Owens
Bonnie Piantadosi
Steven Piantadosi
Nancy Prusakowski
Linda Roberts
Karen Robinson
Dave Shade

Anne Shanklin
Jackie Smith
Michael Smith
Paul Smith
Alice Sternberg
Christine Szekely
Jennifer Thorne
Andrea Tibbs
Ada Tieman
James Tonascia
Susan Tonascia
Aynur Ünalp-Arida
Mark Van Natta
Margie Wild
Laura Wilson
Robert Wise
Claudine Woo Shinoff
Kathy Yates
Tables Notebook
Chronologic file

[Tables] J:\TableOath.wpd / bjc

The Table Maker's Oath

WHEREAS,

The world is portrait-oriented,
"Turns" are abhorrent,
Paper was created to have a blank side,
The eye as an exquisite graphic instrument,
God did not create our eyes for 8 point,
Tables were invented to save words.

Therefore, I, as a table maker, shall at all times strive to ensure that I

NEVER:

- 1 Produce landscape tables
- 2 Use a font size smaller than 11 point
- 3 Use hard returns or hard page code to manage page breaks in tables
- 4 Use the space bar to move the cursor in the body of a table
- 5 Press the tab key more than once in the body of a table before keying
- 6 Enter numbers in columns that are left aligned or centered
- 7 Expand the margin settings or reduce the font size to make things "fit"
- 8 Disguise numbers by placing them in parentheses in the body of the table
- 9 Create a table without a footer indicating date of creation and file location
- 10 Use "hot wired" dates in table footers
- 11 Revise a table, no matter how small the revision, without updating the footer date
- 12 Leave rogue code lurking about in the electronic version of document
- 13 Rely on the math feature in tables without checking the calculations
- 14 Assume that a table will print as it appears on the screen
- 15 Transmit a table electronically without first checking as to whether the document will open
- 16 Assume tables produced in one format will convert seamlessly to another format without loss of code or format structure

At the same time I shall

ALWAYS:

- 1 Save old versions of tables for the life of a project
- 2 Produce tables with proper headers and footers
- 3 Use block protect or conditional end of page codes to avoid bad page breaks
- 4 Eliminate rogue code
- 5 Use headers and spacing to identify related columns in tables
- 6 Economize on words in titles and footnotes
- 7 Spell check early and often
- 8 Spell check after any revision of a table
- 9 Ensure different versions of the same table are distinguishable from one another by footer dates
- 10 Print and read tables by the FOTD method of reading before "handing off"
- 11 Periodically back-up table files
- 12 Produce a printout of tables before handing them off to others