

Software

TeX and the Year 2000

Barbara Beeton

During the past year or so, perhaps the most high-profile question in computer circles has been, Will this system continue to work as usual after the clocks strike midnight on 31 December 1999? Although the millennium doesn't really arrive until 2001, the year 2000 (Y2K) is just as important to anyone who depends on a computer. Many so-called "legacy systems" assumed that two digits would be sufficient to accurately represent a year, and standard mainframe tape labels (to cite but one example) designated 99366 as the date (in Julian form) on which data on the tapes could be considered "scratch". To be fair to the designers and programmers of those systems, when this convention was devised, computer memory and storage were very expensive, and the saving in space permitted by this shorthand made it possible to create programs to do things that would otherwise have been prohibitively expensive.

However, the chronometer is about to cycle, and, for two-digit-based systems, how is the year "00" to be interpreted? As 2000 or as 1900?

Some specialties have probably already had to take "real" years into consideration—books were published long before 1900, so libraries might well have holdings from the 1800s or even earlier, if only in special collections. But the bulk of commerce, and governments, are much less farsighted, and system managers and CIOs are carefully auditing their programs and procedures for residual code and assumptions that could raise havoc when the year 2000 arrives.

Is TeX one of the programs that could have Y2K problems? The short answer is, "No, the TeX program will not have Y2K problems." However, since this question will be asked at many user sites, it would be helpful to have some guidelines for answering it.

Year handling in TeX

Let's start by examining the code in `tex.web`. This is the canonical source, from which all implementations are ultimately derived. The extracts on the next two pages, where "year" or "date" occurs in either code or commentary, are from version 3.14159, dated 1998/08/08, at `CTAN:systems/knuth/tex/tex.web`.

From these extracts and other code not shown, it can be determined that the primitive `\year` is

defined as a 4-byte signed integer. An integer of this size is more than adequate to hold a 4-digit date, or any other value likely to be passed to it from the system or input by the user. It also turns out that the modulo function¹ applied in line 24125 has no effect on any internal value; this is used only to write out information about the preloaded format in the log file. For example, a TeX implementation on a local Unix machine reports the following:

```
This is TeX, Version 3.14159 (C version 6.1)
(format=tex 97.10.7)
```

Although one can safely conclude that TeX itself is not subject to the kind of problems that cause Y2K worries, there are two caveats.

- The format message might raise a red flag to someone who is concerned more with form than with substance.
- An application *can* set `\year` to an unconventional value and use it for some operation that might cause a problem.

The second case is a site-dependent problem, not one of TeX; it is up to whoever is performing the local audit to check applications for such use.

The first case, while harmless, is nonetheless amenable to being fixed.

Expanding the TeX format date to report a 4-digit year

The possible ramifications of leaving the format year with two digits was discussed at length on the `tex-implementors` list. The consensus was that, although it is harmless, it would cause questions to be asked and occupy time that would be better spent in tending to real problems. Therefore, after verifying the facts with Peter Breitenlohner, I took advantage of my status as Knuth's "TeX entomologist" and sent him a message asking him to consider another change to TeX and METAFONT (which treats the year in a similar manner).

Within a few days, I received the following answer.

-- * --

Date: Tue, 24 Nov 1998 00:02:14 -0800 (PST)

Subject: note from Don Knuth

Dear Barbara,

Thanks for your note. It spurred me to [...] update my macro files for errata in TAOCP; now they are Y2K compliant!

¹ The modulo function yields the remainder after dividing the operand by a specified value; in the present example, 1999 modulo 100 becomes simply 99.

Extracts from tex.web 3.14159 containing “year” or “date”

```

2137:  (Actually there are three places where \TeX\ uses |div| with a possibly
        negative numerator. These are harmless; see |div| in the index. Also if the
        user sets the \.{\time} or the \.{\year} to a negative value, some diagnostic
        information will involve negative-numerator division. The same remarks
        apply for |mod| as well as for |div|. )

4918:  @d time_code=20 {current time of day}
        @d day_code=21 {current day of the month}
        @d month_code=22 {current month of the year}
        @d year_code=23 {current year of our Lord}

4981:  @d time==int_par(time_code)
        @d day==int_par(day_code)
        @d month==int_par(month_code)
        @d year==int_par(year_code)

5045:  time_code:print_esc("time");
        day_code:print_esc("day");
        month_code:print_esc("month");
        year_code:print_esc("year");

5130:  primitive("time",assign_int,int_base+time_code);@/
        @!@:time_{\.{\time} primitive@>
        primitive("day",assign_int,int_base+day_code);@/
        @!@:day_{\.{\day} primitive@>
        primitive("month",assign_int,int_base+month_code);@/
        @!@:month_{\.{\month} primitive@>
        primitive("year",assign_int,int_base+year_code);@/
        @!@:year_{\.{\year} primitive@>

5218:  @ The following procedure, which is called just before \TeX\ initializes its
        input and output, establishes the initial values of the date and time.
        @^system dependencies@>
        Since standard \PASCAL\ cannot provide such information, something special
        is needed. The program here simply specifies July 4, 1776, at noon; but
        users probably want a better approximation to the truth.

        @p procedure fix_date_and_time;
        begin time:=12*60; {minutes since midnight}
        day:=4; {fourth day of the month}
        month:=7; {seventh month of the year}
        year:=1776; {Anno Domini}
        end;

10292: @<Print the banner line, including the date and time@>;

10322: @ @<Print the banner...@>=
        begin wlog(banner);
        slow_print(format_ident); print(" ");
        print_int(day); print_char(" ");
        months:='JANFEBMARAPR MAYJUNJUL AUGSEPOCTNOVDEC';
        for k:=3*month-2 to 3*month do wlog(months[k]);
        print_char(" "); print_int(year); print_char(" ");
        print_two(time div 60); print_char(":"); print_two(time mod 60);
        end

```

- 11630: @ The preamble contains basic information about the file as a whole. As stated above, there are six parameters:
 $\$\$ \hbox{\@!i[1] | \@!num[4] | \@!den[4] | \@!mag[4] | \@!k[1] | \@!x[k] | .} \$\$$
 The |i| byte identifies \.{DVI} format; currently this byte is always set to 2. (The value |i=3| is currently used for an extended format that allows a mixture of right-to-left and left-to-right typesetting. Some day we will set |i=4|, when \.{DVI} format makes another incompatible change---perhaps in the year 2048.)
- 12252: @ @<Initialize variables as |ship_out| begins@>=
`dvi_h:=0; dvi_v:=0; cur_h:=h_offset; dvi_f:=null_font;
 ensure_dvi_open;
 if total_pages=0 then
 begin dvi_out(pre); dvi_out(id_byte); {output the preamble}
 @^preamble of \.{DVI} file@>
 dvi_four(25400000); dvi_four(473628672); {conversion ratio for sp}
 prepare_mag; dvi_four(mag); {magnification factor is frozen}
 old_setting:=selector; selector:=new_string;
 print(" TeX output "); print_int(year); print_char(".");
 print_two(month); print_char("."); print_two(day);
 print_char(":"); print_two(time div 60);
 print_two(time mod 60);
 selector:=old_setting; dvi_out(cur_length);
 for s:=str_start[str_ptr] to pool_ptr-1 do dvi_out(so(str_pool[s]));
 pool_ptr:=str_start[str_ptr]; {flush the current string}
 end`
- 23668: The global variable |format_ident| is a string that is printed right after the |banner| line when \TeX\ is ready to start. For \.{INITEX} this string says simply '\.{(INITEX)}'; for other versions of \TeX\ it says, for example, '\.{(preloaded format=plain 82.11.19)}', showing the year, month, and day that the format file was created. We have |format_ident=0| before \TeX's tables are loaded.
- 24122: @ @<Create the |format_ident|...@>=
`selector:=new_string;
 print(" (preloaded format="); print(job_name); print_char(" ");
 print_int(year mod 100); print_char(".");
 print_int(month); print_char("."); print_int(day); print_char(")");`
- 24214: `initialize; {set global variables to their starting values}
 @!init if not get_strings_started then goto final_end;
 init_prim; {call |primitive| for each primitive}
 init_str_ptr:=str_ptr; init_pool_ptr:=pool_ptr; fix_date_and_time;
 tini@/`
- 24358: @<Get the first line...@>=
`begin @<Initialize the input routines@>;
 if (format_ident=0)or(buffer[loc]="&") then
 begin if format_ident<>0 then initialize; {erase preloaded format}
 if not open_fmt_file then goto final_end;
 if not load_fmt_file then
 begin w_close(fmt_file); goto final_end;
 end;
 w_close(fmt_file);
 while (loc<limit)and(buffer[loc]=" ") do incr(loc);
 end;
 if end_line_char_inactive then decr(limit)
 else buffer[limit]:=end_line_char;
 fix_date_and_time;@/`

Peter Breitenlohner’s comments are 100% correct. Also, I might note that the METAFONT documentation already mentions that METAFONT (as it stands) cannot be run after the year 32767; the latter “problem” is not applicable however to T_EX.

I agree that it would now be best to remove the “mod 100” from T_EX module 1328 and from MF module 1200 (and from METAPOST in the corresponding place). I hereby give permission to implementors to make such changes in their change files. No change to the version numbers are needed.

I’ve actually made the changes in my personal copies of `tex.web` and `mf.web` and `tex82.bug` and `mf85.bug` and `errata.eight` and `errata.tex`; but those sources won’t be updated at labrea until 2002.

I’m not changing page 23 of *The T_EXbook* — where the example was probably my original motivation for cutting to two digits, since the example wouldn’t fit on a single line otherwise — nor the corresponding page of *The METAFONTbook*. The format-identifier details are not an essential part of T_EX’s actions.

Incidentally, I’m not considering this to be the “final bug” in T_EX. But it may well turn out to be the final change ever made.

Cordially, Don

— * —

Thus all implementors now have permission to make this change — it is official.

Status of T_EX implementations

While most implementors have probably made this change by now, I can confirm the changes for only those who have announced it publicly.

Most commercial implementors have done so, and have posted confirmation on their Web sites.

— * —

TrueT_EX. At <http://truetex.com/y2k.htm>, there is a general statement of the relevant issues.

T_EX Year 2000 (Y2K) Issues in Summary

This page summarizes the year 2000 issues surrounding T_EX and METAFONT, based on discussions in the newsgroup `comp.text.tex` and among the members of the mailing list `tex-implementors@ams.org`. We attempt herein to merely set forth the chief matters at hand, without engaging the controversial aspects of what solutions should be taken.

1. Crashing: The programs T_EX and METAFONT themselves will not crash due to dates.

(However, each executable implementation depends on a run-time library and an operating system, which should be evaluated in this regard.)

2. Timestamps: A 2-digit year is (a) printed in logfiles, and (b) stored in format file and base file time stamps. These items should not be of general concern, because they are intended for human readers and not as input to other programs.

On November 24, 1998, Donald Knuth granted an unusual permission to modify T_EX and METAFONT to use 4-digit timestamps (nearly all implementations, such as web2c, had already been doing so), saying:

I agree that it would now be best to remove the “mod 100” from T_EX module 1328 and from MF module 1200 (and from METAPOST in the corresponding place). I hereby give permission to implementors to make such changes in their change files. No change to the version numbers are needed. *[As reported by Barbara Beeton on the tex-implementors e-mail list.]*

This permission means that 4-digit timestamps, while changing the output of T_EX and METAFONT slightly from the current autographs, still meet Knuth’s authoritative standards required of software calling itself T_EX or METAFONT.

3. The `\year` primitive: T_EX TRIP certification, in the strictest sense, does not require that `\year` return a meaningful value (T_EX may be certifiably implemented on platforms that do not even supply date reporting, such as standard Pascal). *The T_EXbook* does define `\year` as “the current year of our Lord”, which is the only correct meaning of `\year` for those implementations which can supply a meaningful value, which is to say nearly all of them. In short, T_EX implementations should provide a value in `\year` giving a 4-digit year Anno Domini, or the value 1776 if the platform does not support a date function. T_EX does not provide any state variables to indicate whether `\year` contains a meaningful value, and while 1776 could have been considered a signal value for a lack of meaning to `\year`, this is not a standardized requirement.
4. External software: The T_EX corpus embodies many accessory programs, such as macro packages and DVI translators, which may compute dates from the value of `\year` (or rarely, from timestamps). Such accessories should be checked individually for correct behavior when `\year` is assumed to return a correct 4-digit `\year` value before and after

2000. Accessories with an additional “defensive level” of correctness will behave reasonably when `\year` contains a two-digit value or a meaningless value such as 1776.

See the TrueTeX home page at <http://truotex.com>.

— * —

Other commercial TeX Web sites are more product-specific.

Blue Sky Research. A statement about Macintosh and Textures appears at <http://www.bluesky.com/y2knote.html>:

Year 2000 Compliance

As you may or may not know, the Macintosh in general has been Y2K compliant since it was first introduced in 1984. Contrast this with Microsoft’s Windows software — even the Windows 95 version is not Y2K compliant, nor is their Windows NT 4.0! See [Apple’s year 2000 page](#) on the web.

Likewise, most Macintosh software is built to handle the year 2000, and Textures is no exception. Don Knuth, the author of the TeX typesetting system and the multi-volume set “The Art of Computer Programming” was well ahead of the curve, and TeX has been Y2K compliant [sic.] since the 70’s!

In short: Yes, Textures is Y2K compliant.
Apple Macintosh: the choice for the year 2000 and beyond ;-)

— * —

PCTeX. At <http://www.pctex.com/iwhatnew.htm>, there is a posting that says

Is PCTeX Year 2000 Compatible
Because PCTeX and accompanying macros etc. are operating system/date dependant, as long as you have taken proper precautions to ensure your operating system is year 2000 compliant, you will experience no change in operation, come January 1, 2000. If you need a verification letter for your files, click [HERE](#).

The link connects to <ftp://ftp.crl.com/users/rw/pti/98765/yr2000.dvi>.

— * —

Y&Y. This statement appears at <http://www.yandy.com/y2k.htm>:

Year 2000 (Y2K)

Relative to the Year 2000, there are no date issues with Y&Y TeX System release 2.1

Y&Y TeX uses only standard calls to the Microsoft C++ library and WIN32 API to manage the dates and times for files. The operating system sets the

date and time arithmetic values based on what the PC’s realtime clock provides. It is the operating system, the underlying BIOS, and the realtime clock upon which Y&Y software depends for Year 2000 compliance.

We do not embed or implement any algorithm of our own or any third party’s to compute dates in any of our products.

TeX itself does not do any computation based on dates. It simply provides informational access to the dates and times obtained from the underlying operating system. The year is not truncated to two digits when offered to the user from inside TeX source code via the controlsequence `\year`, or in information output on screen or in the log file.

In addition, in release 2.1, the date recorded in the format file uses four digits for the year.

This information is provided “AS IS” without warranty of any kind, either express or implied. Any further question may be addressed to the attention of support@YandY.com.

— * —

MacKichan Software. From the developers of Scientific Word and Scientific WorkPlace (which incorporates TrueTeX), at <http://www.mackichan.com/products/y2k.htm>:

Dear Sir or Madam:

In response to your inquiry about the MacKichan Software products and the Year 2000 problem, I can give you the following assurances:

Scientific Notebook:

Dates are not used in computations. We have run Scientific Notebook on a system that has had the date advanced beyond the year 2000 and have not experienced any problems. We do not anticipate any Year 2000 issues with this product.

Scientific Word and WorkPlace:

Version 2.5: Scientific Word and Scientific WorkPlace 2.5 include as a component TrueTeX from Kinch Computer Company. Unfortunately, the version of TrueTeX included with Scientific Word 2.5 and Scientific WorkPlace 2.5 does not operate if you advance the date beyond the year 2000 (see Note below). Version 2.5 is **not** Year 2000 compliant. We expect to post on our web site new versions of the non-Y2K compliant portions of Version 2.5

Version 3.0: We received a new version of TrueTeX from Kinch Computer Company and it does not exhibit a problem if the date is advanced beyond the year 2000. Dates are not used in computations. We have run Scientific Word 3.0 and Scientific WorkPlace 3.0 on a system that has had the date advanced beyond the year 2000 and have not experienced any problems. We do not anticipate any Year 2000 issues with these products.

Note: For the technically inquisitive, Kinch Computer Company reports that the failure of True \TeX to run properly on a system with the date advanced beyond the year 2000 was corrected after updating to a newer version of the Microsoft C run-time library.

Sincerely,
Barry MacKichan
President

-- * --

The bottom line

It appears that anyone using the current version of a commercial implementation of \TeX can be confident that no untoward results will come from \TeX itself.

An article in the latest issue of *NTG MAPS* (Number 22, Voorjaar 1999, pages 136–141) con-

tains some additional information on freeware and shareware implementations of \TeX .

Once again, please remember that it would be wise to observe the caution about possible side effects resulting from arithmetic manipulation of year values in *application code* written for use with \TeX . Or, as someone wittily observed, they would not like to have to attest to the Y2K implications of David Carlisle's `xii.tex` as published in the last *TUGboat* (19(4), page 348).

Let us go forward confidently into the new year 00, and thence into the new millennium.

◇ Barbara Beeton
American Mathematical Society
P. O. Box 6248
Providence, RI 02940-6248
bnb@ams.org

---*--- ---*--- ---*--- ---*---