This document is the result of a search for potential Y2K bugs in the sources of ITG 2, a product of AccuMeasure Inc.

Contents

1 Introduction

The information given here was obtained studying the CD-Rom "ITG-19970627" created 27 June 1997.

As all the file modification dates are June 27 1997, there is no information to obtain from them.

1.1 method

At first I began by looking at the ITG CD-Rom, trying to understand what was in the different directories.

All went well until I tried to understand “edge/src”, with it’s multitude of subdirectories. I had a look in the 3852 & analfunc subdirectories. The makefiles where much more complicated. So I had to find the compiler help file “msvc/help/mscopts.hlp”.

After that I decided to change my strategy, I tried to understand what was in the ITG2 distribution disks. It gave me some insight the way ITG was done.

And then, as time was running away, I changed directions again, I went hunting after the Y2K bugs, see § 4.1.

1.2 conclusion

It seems that ITG uses dates only for the following usages :

- waveform datation, during acquisition or anlysis
- some operationsduring the initial installation :
  - maintaining the original modification file times while copying files
  - finding the directory where is the “good version of qcl.exe”
  - generating date information in the INSTALL.OUT file

I found 3 potential bugs while looking at the code, but they are of no importance and not Y2K related :

- the method to find the path of the “qcl.exe” file during installation is dirty, see § 4.3.1
- the waveform original date is replaced by the current system date if the waveform is saved in HPW_DADISP format, see § 4.5.1
- the date of a waveform date can be modified by the user, and ITG generates the code to replay that exact modification, see § 4.5.2. I call that a bug, some call it a feature.

There is another small potential problem in the fact that the user can get a waveform date by calling the “Hpw_get_attr_st” function. But then it is his code, so I don’t think you are concerned.

And as usual, all bets are off if the system or the C run-time behaves badly. So it would be interesting to do a test by changing the system date. I haven’t done that as ITG is not installed on my PC.

But after looking at the code, I am pretty sure that ITG is going to work exactly as before after Y2K.

1.3 spent time

date	hours	usage
31 may	2	CD file tree, root
8 june	4	CD file tree, \edge\tools
18 june	4	looking in one makefile : compiler options, defined symbols
22 june	3	makefiles : generalization
23 june	3	targets
25 june	4	searching some info about Y2K on the web
29 june	4	searching date patterns
1 july	8	searching date patterns
2 july	4	searching date patterns
2 july	2	trying to make my notes somewhat readable
total	38	

2 Contents of the CD ITG-19970627

It contains the following directories :

- C5, C6, QC45: C compilers
- QB45 and : Quick Basic
- Win3 : a version of Windows
- MKS : Unix like utilities that can be used on a Windows 3.1 or DOS ystem
- Edge : contains ITG

The only part that is described in this document is "edge".

The file “edge\tools\compress_” cannot be read/opened/copied. As it’s size is 0 bytes I don’t think it’s important.

2.1 Files left over

There is a "trans.tbl " file in nearly directory. They seem to have been used while transforming file names (uppercase / lowercase).

2.2 Directory "\edge"

It contains the following directories :

name	content	interesting
compress	compressed files, probably version of files from "dist/*"	NO
demo	demonstration version of ITG	
dist	ITG distribution	
ispy	Public Domain Windows Spy Program	
makelibs	used to Make "hpitg.lib hpitgbas.lib, hpitgl.lib" that are copied afterwards to dist/lib	
newlibs	other versions of "hpitg.lib hpitgbas.lib, hpitgl.lib", with batch file to compress them, no Makefile,	NO
qa	contains some subdirectories with various test programs, QuickBasic, C	
qctest	sample/test program to check that it is possible to compile/use a Quick C program	
src	ITG	YES
temp	temporary files, last use = list xxxx.cid internals with hpidstat	NO
tools	utilities	YES

2.3 Directory "\edge\tools"

name	content	interesting
distrib	data used by makeitg.bat to generate ITG distribution disks	YES
rcs	archive version of the batch command files	NO
*.bat	command files to generate ITG distribution disks	YES
src	source for many ITG utilities	YES

batch file name	content	interesting
makeitg.bat	used to make the distribution floppies	?
makeitg.bku	old version of makeitg.bat	NO
cp2net?.bat	used to make the distribution disks, probably on a network disk cp2net.bat => I:\edge\bits (cp2net1.bat == cp2net.bat) cp2net2.bat => I:\edge\b0001	YES
enc_*.bat	used to encode (compress) the distribution files in "\edge\compress" encodem.bat == enc_itg.bat + some files (xxx.cid , xxx.ih)	NO
vernet.bat	verification that files on ITG disks in i:/edge/bits are identical to sources	NO
verb01.bat	idem with files in i:/edge/b001	NO
same.bat	idem with files on floppies (beep.exe is used to ask for next one)	NO
beep.exe	asks for next floppy insertion	NO

2.4 Directory "\edge\tools\src"

name	content	interesting
comp_id	comp.exe : utility that compiles a list of drivers, by starting hpidc.exe N times, with a command line : hpidc -W c:\edge\dist\drivers\xxxx	NO
compress	encode.exe, decode.exe : utilities that compress/decompress the distribution files the directory contains also compress.c (probably obsolete source) and various files that were used to do some tests	
convert	convert.exe : utility which reads in SETUP.DAT and creates: SETUP.INF - SETUP program information file COMPRESS.BAT - Batch file for compressing install files CUTDISKS.BAT - Batch file for cutting install disks	
getc60	extracts object files from d:\c600\lib\mlibce and repackages them in \edge\src\libextra\ hpmlibce.lib (stdlib, ...)	
getclib	idem but objects extracted from from c:\lib\mlibce to	
hidden	hidden.exe : utility that sets the hidden attribute of a file specified	
hpcheck	hpcheck.exe : HP ITG II System Integrity Check Utility, memory, IEEE 488, coprocessor, path, compilers, ...	
hpchkarg	qbmake.bat, hpchkarg.exe : utility that make some checks and help to generate a Quickbasic program. Presence of compiler & linker, source file existence, disk space, ..	
hpchkc hpchklc hpchkqc	utilities to make some checks before C compiling, called from the generated batch files (qcmake.bat, ...)	
hpdetect	hpdetect.exe : utility that checks, among other things, which version of ITG is installed	
hpemstat	hpemstat.exe : utility that gets information about EMS memory, manager, frames , ...	
hpinit	hpinit.exe : utility called from qcstart.bat, creates a makefile, make some checks	
hpwinit	hpwinit.exe : idem but called from wcmake.bat, for Windows C compiler	
icons	icons.exe : pseudo-utility, in fact a container for icons	
idstat	idstat.exe : utility program which prints the stats of a compiled ID	
indent	indent.exe : utility that copy files, converting '@' to ASCII(171), and prepending 6 spaces to each line	
qcwinit	qcwinit.exe : utility that creartes a QuickC makefile xxxxx.mak depending on the current HPITG environment variable	
t&mgroup	t&mgroup.exe : utility that creates the HP ITG T&M Windows Applications Group	
wfstat	wfstat.exe : utility that displays the stats of a workfile	

2.5 Directory "\edge\src"

name	content	interesting
3852	makefile and source files for the 3852	
analfunc	makefiles and source to generate analfunc.lib	
install	makefile and sources to generate setup.exe subdirectory A_2_00 seems to duplicate many files	

3 ITG Distribution

ITG 2 distribution contains a lot of files. I considered that those files were potential targets, and tried to understant how they were “made”. Some information was found in “edge\src\install\itgfiles”
I listed here the targets I found most interesting. The other files seem much less interesting : various C or driver sample sources.

3.1 Potential problems

One of the problems with ITG sources is that the same source files are used to build different targets :

- ITG's executables
- the three static libraries that are used by clients to build DOS executables
  hpitg.lib, hpitgl.lib, hpitgbas.lib
- the DLL that is used by clients to build Windows executables
  hpitgw.dll and the associated import library hpitgw.lib

That kind of problem should have been resolved by :

- defining the compiler options in one or two "option files"
- creating a makefile in each source directory, that defines the target list and includes the "option files". That makefile is used to make the intermediate targets, usually static library files.
- creating some command files that can be used to "make" all the final targets, by linking/concatenating the intermediate libraries, to obtain the final EXE, DLL and LIB files

Alas the options are defined (sometimes differently) in every makefile.
What's more, there should be other targets :

- clean : that is used to delete all the generated files
- debug versions

3.2 Debug version

.Nowadays a "debug version" filled with assertions is considered as absolutely necessary.
Nearly every software house now delivers two versions of each library/DLL : the "normal one" and the "debug one". The "debug version" is used by the software house to test its products, and by the client whenever there is a difficult bug to resolve.
In the case of ITG, it seems that when it was considered necessary, the makefile was modified by hand to create the debug version. After some tests, the code was considered correct and the makefiles modified again.
That is a very bad practice. One should be able to produce the two versions simultaneously, by starting two command files, and going out to eat something while the computer works.
What's more, one should never modify lightly the makefiles.

3.3 Makefiles

Most of the sources directories contain two makefiles. I have described those that are in "\edge\src_analfunc".

name	used to build versions
makefile	standard/debug medium memory, Quickbasic, Windows
lmakefile	large memory and DLL

Alas, it seems that the makefiles of other edge/src/subdirectories are slightly different. The naming conventions and the compile options are not allways the same

Some times the C compiler is called with the option "/Fa" that Create assembly listing, that option explains in most of the cases the "xxx.asm" files. They are compilation left-overs.

In most of the cases the same C source file is used to generate 4 objects :

xxx.o	medium memory
xxx.o	medium memory debug
Wxxx.o	windows medium memory
Qxxx.o	quickbasic
Lxxx.o	large memory
Dxxx.o	DLL

Compiler options are :

option	target type	role
-c	all	compile only (no link)
-W2	all	warning level 2
-AM	all - (Lxxx.o & Dxxx.o)	medium memory model
-AL	Lxxx.o	large memory model
-ALw	Dxxx.o	large memory model function entry : DS !=SS (for DLLs)
-Gs	all	disable stack checking
-Gw	Wxxx.o, Dxxx.o	create Window prolog/epilog for functions
-Od	all - xxx.o	optimization disabled
-Ox	xxx.o	maximum optimization
-Zpe	all	pack code enable microsoft language extensions
-nologo	all	suppress copyright message
/NT HPTWAV	all - Wxxx.o	name default code segment

Symbols defined by compiler options are :

name	target type	generates	my opinion	use counts
DEBUG	all - Wxxx.o	debug code (assertions, print, ..)	common good practice	
PANEL_MODE	Wxxx.o	Windows executables specific code	seems badly misused	198
NOMINMAX	Wxxx.o	do not define min & max macros in windows.h		no meaning
C_RUNTIME	xxx.o	stubs functions hpa_xxx() that encapsulate calls to hpa_ixxx() and aborts if errors are returned	bad idea, the stubs functions should have been in other files. There is no valid reason for this conditional compilation	81
QBASIC	Qxxx.o	modifies the error handling code of the internal functions. It is sometimes used by #ifdef, & sometimes by #ifndef	bad idea, the internal code should be clean. There should be qbasic stubs that encapsulate the specifics to a language, particularly the error handling code	26
DLL	Dxxx.o	DLL specific code : direct window handling, encapsulate calls to large memories functions, ..	quick and dirty code modifications when somebody asked for a DLL version	57

There are very few valid reasons for conditional code :

- assertions (DEBUG symbol)
- quick and dirty modifications, if you are sure that nobody is going to modify that code again

3.4 Targets

When not precised otherwise, the source files are in the same directory as the makefile.

3.4.1 HPITG2/BIN : EXEs and DLLs

name	itgfiles description	makefile	comment
HPITGIBW.CSB	IBASIC CSUB file		
HPIDC.EXE	Instrument Driver Compiler	not found	
HPDWT.EXE	Driver Writer Tool	not found	
HPDWT.HLP	Driver Writer Tool Windows Help	not found	
HPIDSTAT.EXE	Instrument Driver Status Utility	tools/src/idstat/makefile	= copy of idstat.exe leftovers in src/device : device/makefil2

			device/makefile.old
HPIOSTAT.EXE	I/O Interface Status Utility	src/io/hpiostat/makefile	
HPEMSTAT.EXE	Expanded Memory Status Utility	tools/src/hpemstat/makefile	
HPWFSTAT.EXE	Workfile Status Utility	tools/src/wfstat/makefile	= copy of wfstat.exe
HPCHKARG.EXE	Ensures QBMAKE.BAT filename is okay	tools/src/hpchkarg/makefile	utility that make some checks and help to generate a Quickbasic client program. Presence of compiler & linker, source file existence, disk space, ..
HPCHKQC.EXE	Ensures QCMAKE.BAT filename is okay	tools/src/hpchkqc/makefile	idem for QuickC client program
HPCHKC.EXE	Ensures CMAKE.BAT filename is okay	tools/src/hpchkc/makefile	idem for standard C, medium memory
HPCHKLC.EXE	Ensures LCMAKE.BAT filename is okay	tools/src/hpchklc/makefile	idem for tandard C, large memory
HPINIT.EXE	Creates a QuickC .MAK file	tools/src/hpinit/makefile	= copy of init.exe
HPCHECK.EXE	Checks the correctness of HPITG	tools/src/hpcheck/makefile	
ITGEXEC.EXE	Execute DOS runtime compile/link	src/itgexec/makefile	
HPWINIT.EXE	Aids the WCMAKE (Windows-make) file	tools/src/hpwinit/makefile	
HPITG2.EXE	ITG main executable	src/fh/makefile	= copy of fh.exe linked with libraries of src/libpan & src/libextra
LOADPROG.EXE	Load a program into IBASIC	src/loadprog/makefile	
T&MGROUP.EXE	Creates the T&M group	tools/src/t&mgroup/makefile	
DDELIB.DLL	SID dynamic-link library for DDE	not found	DDE communication is difficult, instable & obsolete
HPIB.DLL	HPIB dynamic-link library	not found	
EPCDICW.DLL	Radisys VXI dynamic-link library	not found	
HPITGW.DLL	Windows dynamic-link library	src/libdll/makefile	some sources in src/libdll, linked with libraries of src/libdll & src/libextra
HPITGIBW.DLL	IBASIC/Windows dynamic-link library	src/pebbles/makefile	there is also a src/pebbles2/makefile that seems to be a leftover temporary

3.4.2 HPITG2/BIN : other files

HPITG.HLP	ITG Windows Help		
DOS.PIF	DOS Window in T&M Group		
RUN.PIF	Run C or QB Program DOS environment		
MAKE.PIF	Make C or QB Program DOS Environment		

3.4.3 HPITG2/LIB :libraries

name	itgfiles description	makefile	comment

MLIBCE.LIB	Bogus QuickBASIC library support	not found	
HPITGBAS.LIB	QuickBASIC runtime library	makelibs/makefile	concatenation of libraries of src/libqb, src/libextra, & some objects of src/librt
HPITG.LIB	C medium-model runtime library	makelibs/makefile, makelibs/medium is probably an obsolete makefile	concatenation of libraries & some objects of src/librt
HPITGL.LIB	C large-model runtime library	makelibs/makefile	concatenation of libraries & some objects of src/llibrt,
HPITGW.LIB	C/Windows static "implib" library	src/libdll/makefile	import library (symbols of hpitgw.dll)

3.4.4 HPITG2/BIN : files that should have been in HPITG2/LIB

STUB.OBJ	Stub file for building a windows app	
WSTUB.OBJ	Stub file for building a QC/Win app	

3.4.5 other potentially interesting files

HPITG2/INCLUDE

name	original description	
HPITG.H	C include file	
HPITG.BI	QuickBASIC include file	
HPITGW.H	C/Windows include file	

HPITG2/SYSTEM

HPITG.ERR	Runtime Error File	
STUB.C	Stub source file	
WSTUB.C	Stub source file	
STOCK.ICO	Icon for an ITG-generated windows app	
RUN.ICO	IBASIC Get and RUN icon	
DOS.ICO	Dow Window Icon	
RUNPROG.ICO	Run Program Icon	
ITGSETUP.ICO	ITG SETUP program Icon	

4 Y2k potential problems

4.1 search methods

I began by searching some info on the web. I was very disappointed, as most info I found was either hysteric, trivial or good common sense.

There seems to be a lot of companies making bags of money oout of that thing. It is a pity I am not really in that business.

Most of the bug hunters begin by searching the sources for some patterns.

So I did like the others : I searched ITG sources for patterns.

I found information about the C functions in the Microsoft file “msvc/help/mscxx.hlp”.

4.2 patterns used in search

items of structures

name	structure	contents	hit count
tm_year	tm	number of years since 1900, so will be 100 in 2000	5

year	dosdate_t	year as a value from 1980 to 2099	3
date	waveform_type	creation/modification date as a string “%u/%u/%u” (month,day,year) or “%u-%u-%u”	14
nYear	DATE tagDATE LPDATE	the year (structure apparently declared but never used)	0

functions :

name	role	hit count	comment
time	returnc current UTC time	1	all usage was already surveyed fileio/fileio.c(183)
localtime	converts UTC time to local time	1	all usage was already surveyed fileio/fileio.c(184)
_dos_getdate	asks DOS for the current date	3	all usage was already surveyed waveform/peek_wf.c(180), waveform/wcreate.c(85), install/status.c(88)
hpw_iget_attr_str	asks for the Waveform date attribute	1	in fact called by Hpw_get_attr_st(wf_handle, HPW_DATE,string)
_dos_getftime	asks DOS for file modification date/time	4	install/file.c(52,82,535), encode (532)
_dos_setftime	asks DOS to set file modification date/time	4	install/file.c(53,103,536), encode (533)
_dos_setdate		0	
FileSetDateTime		0	
FileGetDateTime		0	
SysGetDate		0	
SysSetDate		0	

other names

pattern		hit count	
year	all other uses of the “year”	5	1 comment, 2 variables used in usages already surveyed install/path.c(259), hckqc/path.c(403)
yy	could have been a good pattern	12	9 symbols of libdll/toolbox.h never used 2 comments 1 "yyyyyyyyy" not related to a date
date	whole word only	87	18 RCS files date 18 variables for dossetftime/dosgetftime 14 waveform.date 13 comments 10 variable name of type dosdate_t 06 unused definitions of toolbox.h 02 strings for comparison in fileio.c (already surveyed) 02 definitions of symbol HPW_DATE 01 char* pointer unused 01 string “date” for waveform attribute 01 format string for a fprintf 01 item definition of waveform.date
datelen		1	unused symbol of toolbox.h

4.3 sources containg references to "tm.tm_year"

file	line number(s)	function	exe or dll concerned	usage
src/install/path.c	270	get_quickc_path()	setup.exe	find where is QCL.EXE
tools/src/hpchkqc/path.c	409	check_quickc_version()	hpchkqc.exe	find where is QCL.EXE
src/fileio/fileio.c	185	hpw_iwrite()	all libraries, hpitgw.dll & hpith2.exe	Internal function to transfer waveform data to a file

there were two other files containg tm_year :

file	line number(s)	function	comment
src/fileio/fileio.new	142	hpw_iwrite()	old version of "fileio/fileio.c"
src/hpchkqc/x	16	get_quickc_path()	old version of "install/path.c"

4.3.1 get_quickc_path()

source :

```
void    get_quickc_path(void)
{
int     i, fd, year, month, mark;
char    qcl_exe[128], *date;
struct stat     stats;
struct tm*time;

qc_path[0] = 0;             // initialize to NO path
for (i = 0; i < num_env_dirs; i++) {
        sprintf(qcl_exe, "%s\\QCL.EXE", path[i]);
        if ((fd = open(qcl_exe, O_TEXT | O_RDONLY)) != -1) {
                if (fstat(fd, &stats) != -1) {
                        time  = gmtime(&stats.st_atime);
                        year  = time->tm_year;
                        month = time->tm_mon + 1;
                        mark = year * 100 + month;
                        if (mark >= 8811) {
                                strcpy(qc_path, path[i]);
                                dos_languages_detected++;
                                return;
                                }
                        }
                }
        }
}
```

interpretation :

- the programmer wants to find the correct path for "QCL.EXE", the quick C compiler
- the programmer is searching all directories whose names are in the array path[ ] for a file named "QCL.EXE" that was created after november 1988
- he made many errors :
  - st_atime is defined as "last accessed time" that does not exist on FAT16 file systems, and is really the "last modified time"
  - he used a year number with two digits
  - that's not the proper way to determine the version of an executable

what is going to happen after Y2K : nothing

- if suppose that the *fstat()* C runtime library is correct
- the modification date of "/qc25/bin/qcl.exe" has no reasons to be changed
- even if somebody modifies it, the comparison will give the good result

4.3.2 check_quickc_version()

source :

```
int   check_quickc_version(fd)
int   fd;
{
int   i, year, month, mark;
struct stat     stats;
struct tm       *time;

     if (fstat(fd, &stats) != -1) {
               time  = gmtime(&stats.st_atime);
               year  = time->tm_year;
               month = time->tm_mon + 1;
               mark = year * 100 + month;
               if (mark >= 8811)
                         return 1;
     }
     return 0;
}
```

interpretation :

- it's a variation of the same thing that the *get_quickc_path()* already described.
- the programmer is trying "very dirtily" to find if QCL.EXE is of a good version

what is going to happen after Y2K : nothing

for the same reason that for *get_quickc_path()*

4.3.3 hpw_iwrite()

source :

- it was a bit long to copy everything here
- the code is executed when trying to write a waveform in HPW_DADISP format

```
.....
time(&aclock);                    /* Get time in seconds */
 newtime = localtime(&aclock);          /* Convert time to struct tm: */
if((nwrote = fprintf(fp, "DATE %d-%d-%d\n",
      newtime->tm_mon+1, newtime->tm_mday, newtime->tm_year+1900)) <= 0)
.....
```

what is going to happen after Y2K : nothing

the code is **CORRECT**, because in 20xx :

- tm_year = 1xx
- tm_year + 1900 = 20xx

4.4 sources containg references to "dosdate_t.year"

file	line number(s)	function	exe or dll concerned	function usage	date usage
src/waveform/ peek_wf.c	183	hpt_ipeek_wf()	all	transfers info from a component in the specified device, into the waveform specified by "wf"	initializes the date field with current date

src/waveform/ create.c	87	hpw_icreate()	all	allocates a new WaveForm	initializes the date field with current date
src/install/ status.c	99	write_install_out ()	setup.exe	creates installation log file, INSTALL.OUT	writes installation completion date

4.4.1 hpt_ipeek_wf ()

source :

```
_dos_getdate(&date);
sprintf(buff, "%u/%u/%u", date.month, date.day, date.year);
hpt_lstrcpy(wf->date, buff);
```

interpretation :

- the programmer is keeping current date in a waveform.

what is going to happen after Y2K : nothing

- I suppose that *dos_getdate()* is performing as specified
- then everything goes well at least until 2099

4.4.2 hpw_icreate ()

source :

```
_dos_getdate(&date);
sprintf(buff, "%u-%u-%u", date.month, date.day, date.year);
hpt_lstrcpy(wp->date, buff);
```

interpretation :

- the programmer is initializing a waveform with current date

what is going to happen after Y2K : nothing

- same as for hpt_ipeek_wf(), the separator character is different

4.4.3 write_install_out ()

source :

```
sprintf(buff, "%s %d, %d at %d:%02d %s", mon[date.month-1], date.day, date.year,
        time.hour, time.minute, am_pm);
sprintf(filename, "%s\\INSTALL.OUT", hpitg_system);
fp = fopen(filename, "wt");
fprintf(fp, "HP ITG installation --- Completed %s\n\n", buff);
```

interpretation :

- the programmer is writing the current date in the installation log file

what is going to happen after Y2K : nothing

4.5 sources containg references to "waveform_type.date"

file	line number(s)	function	exe or dll concerned	function usage	date usage
src/waveform/ debug.c	32	print_wf ()	all	dump Waveform on stdout	printf("Date = %Fs\n", w->date);
src/waveform/ peek_wf.c	184	hpt_ipeek_wf()	all	already described for dosdate_t.year	see peek_wf.c/183

src/waveform/ create.c	88	hpw_icreate ()	all	already described for dosdate_t.year	see create.c/87
src/waveform/ wgetstr.c	43	hpw_iget_attr_str()	all	returns an attribute as string	if called with “attribute ==HPW_DATE”
src/waveform/ wsetstr.c	46	hpw_iset_attr_str()	all	sets an attribute	idem
src/fileio/fileio.c	435	hpw_iwrite()	all	write Waveform data to file	if (format == HPW_ITG) fprintf(fp, "date %Fs\n", w1->date)
=	965	hpw_iread()	all	read Waveform data from a file	if (format == HPW_DADISP) hpt_lstrcpy(w1->date, ++cp)
=	1443	hpw_iread()	all	read Waveform data from a file	if (format == HPW_ITG) hpt_lstrcpy(w1->date, ++cp);
src/analysis/ modvar.c	457, 459, 464	wavemod()	hpitg2.exe	calls Waveform modification dialog	generate the code to redo the date modification at runtime
=	592	offer_attribute()	=	Waveform attribute modification dialog	see description of analysis/modvar.c
=	701	display_attribute_value()	=	=	=
=	937	set_attribute_value()	=	=	=

what is going to happen after Y2K : nothing

In fact a waveform date is an user information. ITG is only creating/modifying it but does not use it.

4.5.1 fileio/fileio.c

There is a bug about Waveform Dates but it is not Y2K related, and not important :

- when writing in HPW_DADISP format, the current system date is written in the file, (not the modification date of the waveform)
- when reading the file in HPW_DADISP format, what was the current sytem date is read as the modification date of the waveform.
- that explains why there is 1 only usage of “date” in hpt_iwrite() although there are 2 in hpt_iread()

4.5.2 analysis/modvar.c

I think there is something that should be called a feature bug. It is also non Y2K related :

- that code is used when analysis data is modified in the “Analysis module”
- I see no good reason to replace the acquisition date of the data by another date, as is implied by ITG documentation, appendix C, “analysis waveform, how they are stored” :
  date is the date the original data was created
- anyway the date & time are among the attributes that can be modified by the user (ITG Users’s guide p 129)
- when the dialog is validated by OK, “wavemod()” tests if the date was modified by the user. If he did that, then it generates code to do replay that modification.
- so any time the user is going to replay that, he is going to have a waveform with the same date