

Intelligent File Comparison

DateWise File Compare Tool

by Vic Fanberg

***The file comparison tool that automatically compares dates...
...without being told where the dates are nor their format***

Executive Summary

This article presents a new kind of patent pending technology for attacking the single most difficult part of the Year 2000 problem (testing). The patent pending process is simple and powerful...use a file comparison tool that automatically recognizes if dates cause two files not to match. If both files contain possible dates, check every possible combination of the two dates to see if they have been reformatted and/or aged.

There is no need to tell the comparison tool where the dates are and what format they are in. This is a tremendous time savings over traditional approaches.

This tool completely automates 30% - 40% of the testing effort (which is 40% - 60% of the entire Year 2000 problem). The net effect is a reduction in excess of 10% in the total effort in Year 2000 remediation, regardless of the technique chosen to solve the problem. The *DateWise File Comparison* tool works the same for windowing, expansion, encoding or whatever other technique was chosen.

In addition, this tool is capable of comparing files which were not previously comparable using automated tools.

Two required file comparison techniques

A typical regression test

The goal of regression testing is to ensure that any program modifications have not disturbed the other functions of the program. Its goal is to see that everything which worked before, continues to work the same way after program modification.

This is no different for testing after year 2000 remediation. In fact, one of the strange criteria for Year 2000 program remediation is that the programs must operate *exactly* the same way before remediation as they do after testing *until* it begins processing data that contains Year 2000 dates. This makes testing a real challenge, because testing for equivalency of two programs is a hard problem. The closest one can reasonably get is an approximation.

If the Year 2000 strategy is procedural (modify the programs and leave the data unchanged), there will be no difference in the data outputted from the program before and after remediation. Any file comparison tool should function as well as the next in determining that each pair of files match from the programs before and after remediation.

Intelligent File Comparison

If a data approach has been used in the Year 2000 modifications, there will be changes in the format of dates in the files, matched by changes in the program to handle those changes. For an example of changes required to a program required because of expansion of the year to four digits, see figure 1. The change in file formats means that files no longer match and many available comparison tools no longer work for this type of comparison.

```
If year < 90
  Then perform old-method-of-calculation
  Else perform new-method-of-calculation.
```

Figure 1a. Typical program change required when expanding dates.

```
If year < 1990
  Then perform old-method-of-calculation
  Else perform new-method-of-calculation.
```

Figure 1b. Corrected version of Figure 1a.

A few attempts have been made to solve this problem. So far, the attempts have centered on identifying where this change in date format is at in the file. Then some action has been taken to mask or removed the added columns from the comparison process.

In the new technology introduced by the *DateWise File Comparison* tool, two files are compared until a mismatch is found, then the difference is checked to see if it could have been caused by a date being reformatted (such as adding the century information, reversing the order of the month and day, etc.). In this case, the dates have to match exactly, so “05/30/98” will never be ever equal “05/29/98” in a regression test, but “30/05/1998” would match. If they match after being reformatted, the identified dates are skipped over and the comparison process continues.

A typical regression test is shown in Figure 2 A single set of data which exercises the code well is run through the program before and after conversion for Year 2000. If the file structure did not change, the two files will match exactly as shown by the “A compare” method. If the file structure did change, as in the “B compare”, some process is applied to remove or mask the change and the files can be compared using ordinary comparison tools. At the minimum, this requires manually telling the computer which positions in each record to mask. In many data files, it would also involve manually telling the computer how to distinguish one record type from another.

Intelligent File Comparison

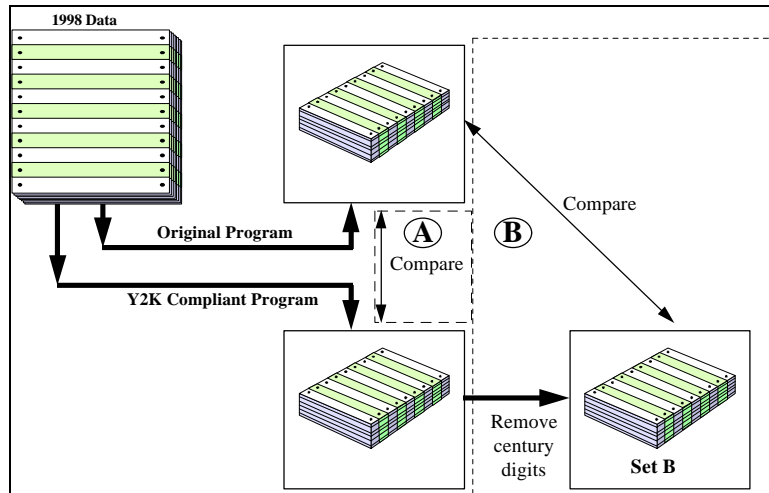


Figure 2. A typical regression test. Option A is the process that happens when the date formats have not changed. Option B is the process when the formats did change.

It is interesting to note that the new technology introduced by the *DateWise File Comparison* tool always looks like 1A, regardless of whether there was a change in the structure of the file or not. It will always be a direct comparison of two files, without any need for preparation of the files and without the need to specify exactly where the dates are in the file. In practice, if it is known that a certain file contains only dates of a particular format (such as all the dates with two digit years are in the format mm/dd/yy) or range (such as all the dates with two digit years fall in the range 1900-1999) it will improve the preciseness of the comparison process to tell the computer that information, but it is never required for a successful comparison process. In fact, specifying both the range and formats used will generally yield the same preciseness as specifying the position and format of each specific date and telling the computer how to distinguish one record type from another, but with much less manual work.

What is tested in Year 2000 testing?

Year 2000 testing seeks to verify that an application which works in the in the range 1900-1999, continues to work the same way after the century rolls over to the next year. This is usually verified by checking a few basic facts including:

Intelligent File Comparison

Examples of what to test	Why test this
The difference between any date in the last half of 1999 and the first half of 2000 is at most a year, not 99 years.	When the year is expressed as a two digit number, the difference may appear to be ± 99
Any date in the year 2000 is after any date in the year 1999	When the year is expressed as two digit number, it would appear that 99 is greater than 00
January 1, 2000 falls on a Saturday.	To insure the year 00 is 2000, not 1900
February 29, 2000 is a valid date, the number of days between February 28, 2000 and March 1, 2000 is two, the year 2000 is a leap year.	To check leap year logic was coded correctly
The difference between January 1, 2000 and January 1, 2001 is 366 days, the Julian date for December 31, 2000 is 00366, there are 366 days in the year 2000	Verify an effect of leap year

Table 1. Year 2000 tests.

A typical Year 2000 test

The goal of Year 2000 testing is to verify that sufficient changes have been made to cause the program to function properly after it begins processing Year 2000 data. Even though there is a seeming discontinuity between the years “99” and “00”, ensure dates crossing this range are processed the same way as a four digit year would have been processed.

In a Year 2000 test, input regression data is run through a process to age the data¹ by a predetermined quantity of days. It is expected that the output from of running the Y2K compliant program with this aged data will have all the dates differ from the regression output by the same predetermined number of days².

¹ The process of aging the dates is beyond the scope of this paper. It is just assumed that somehow the input data was aged. Some of the commercial tools available for aging data include:

- Compuware FileAid with Ager option
- Specialized Software International with TransFile/2000
- Princeton SoftTech with DataAger™

² It is possible for the “days” to be something other than calendar days, for example, they could be work days and exclude weekends and holidays. The *DateWise File Comparison* tool will allow for this possibility.

Intelligent File Comparison

Regression input data is run through an aging tool to age the dates by a predetermined amount. Two runs are made. The first is the regression test already performed. The second uses the data aged and the system date aged by the same amount. The output of the regression run and the aged run should differ by exactly the aged amount. If they do match that amount, the Year 2000 test was successful.

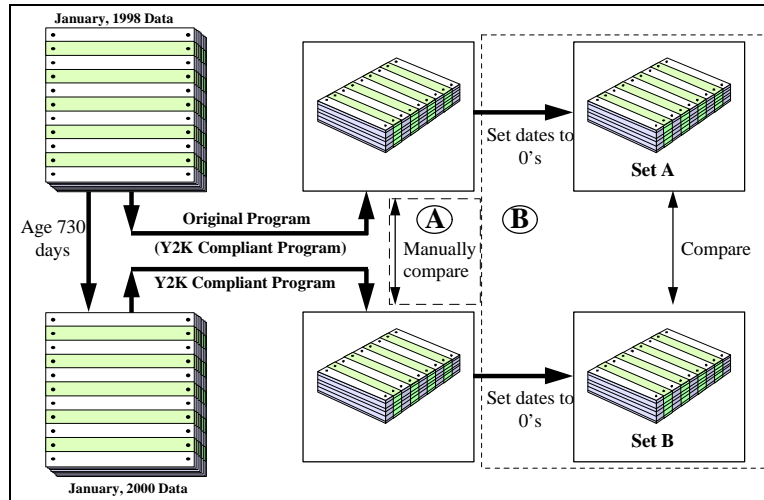


Figure 3. Typical Year 2000 tests.

As shown in Figure 3, the usual strategies for comparison of these two files have involved the inaccuracy of bypassing the dates completely or manually comparing the files.

In the past few months, another option has appeared. That additional option consists of manually identifying everywhere dates appear and the format of those dates. For files with multiple record types, it is also necessary to tell the comparison tool how to identify record types. This option has the advantage of allowing exact comparison of dates for most types of computable readable files.

Unfortunately, this option is a manually intensive procedure. Also, the new option will not generally work for reports with more than one format. This method is only intended for comparison of files where exact record format can be identified from information in the record. This approach is illustrated in figure 4.

Another alternative for year 2000 testing is to embed aging commands within the computer program and perform all the aging dynamically within the program. Immediately after reading a record, dates are aged to the appropriate date. Immediately prior to writing a record, the dates are un-aged. Immediately after writing, the aged image replaces the un-aged image in memory. This modifying the computer program, may require extensive modifications to track whether each date field actually contains a date or some other value that just looks like a date. For example, seeing the number "000101", can you determine this is a actually a date or is it some other redefined field? In general, this can not be determined. This option also has additional run-time performance overhead. If programs are modified to remove the aging statements when they enter

Intelligent File Comparison

production, they need to be tested again after the aging statements are removed, eliminating most of their advantage. If aging statements are not removed, is there ever a chance of the aging feature to be accidentally activated in production?

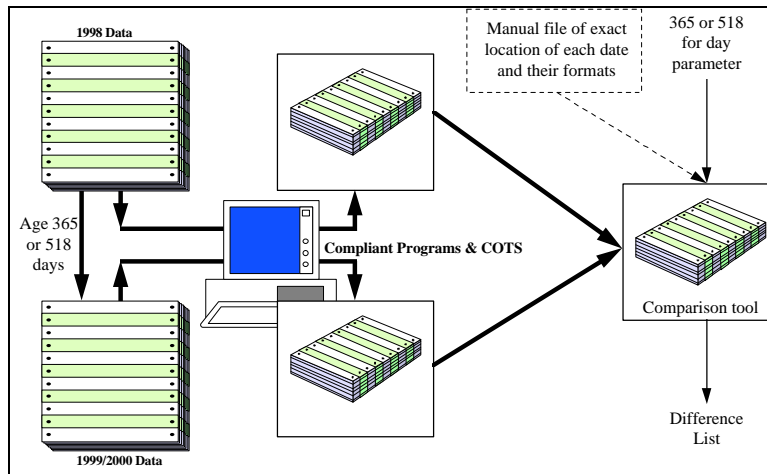


Figure 4. Modern Year 2000 tool with manual identification of date locations and formats. This has the disadvantage of having to manually identify the date locations and formats. Furthermore, it does not work for general reports.

The new patent pending idea

The ideal solution is to eliminate the need for manually entering the date location and format. That is exactly what the *DateWise File Comparison* tool does.

Intelligent File Comparison

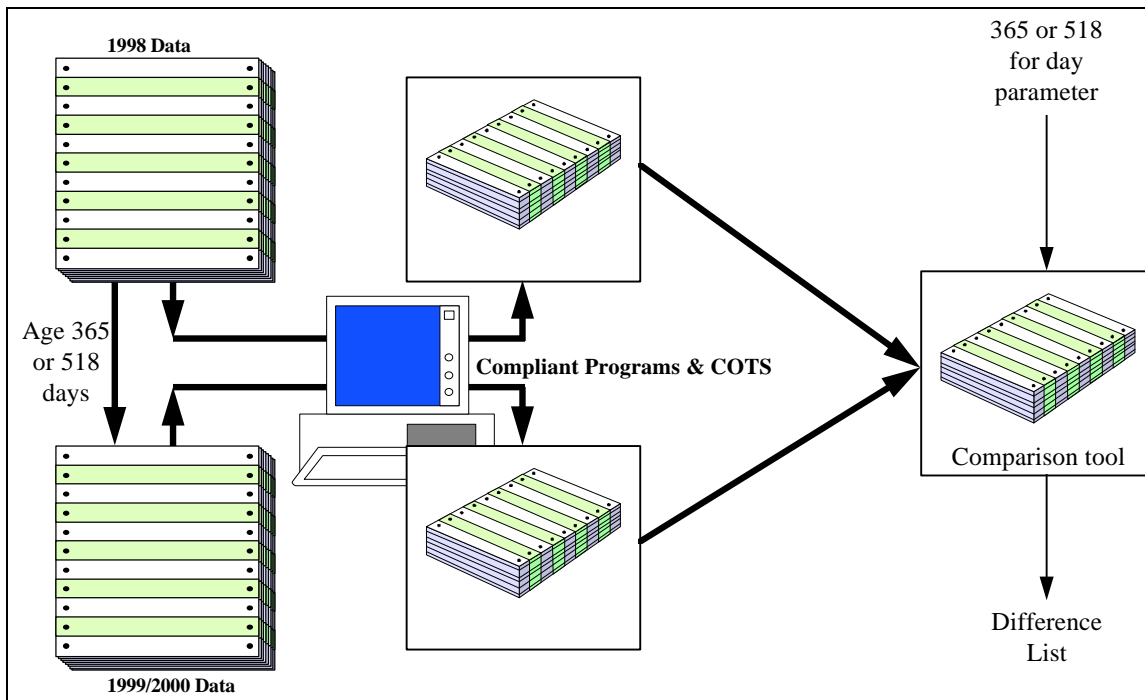


Figure 5. *DateWise File Comparison tool.*

Figure 5 shows the *DateWise File Comparison* tool. The computer calculates all possible interpretations of any mismatch to see if any of them could be explained by being a file date. If so, the field is tentatively flagged as a match until proven otherwise.

The tool is a file comparison tool designed specifically for Year 2000 or other remediation work. It automatically recognizes formatting changes of dates and dates which differ only by a single constant (for the entire file, i.e. it can identify aged files). This includes such elements as

- Dates which were expanded to include a four digit year
- Files which were expanded to accommodate the date changes
- A Julian date with a two digit year that was changed to a Gregorian date with a four digit year
- Where the month and day fields were reversed for consistency with other fields or standards
- etc.

Intelligent File Comparison

What the tool requires and can provide:

The minimum requirements of the tool are to provide the names of two files. Given the minimum information, the tool will provide a ternary answer:

- The files compare exactly
- The files compare, but contain information in which dates are either reformatted or aged
- The files are different. (With the minimum parameters, it will recognize that dates have apparently been aged different amounts, but not know which one was right, therefore, no additional printout is desired. Such a printout would list all dates which were aged, not just the ones which were aged incorrectly.)

In addition, if a third parameter is specified, the program would know which dates have been aged properly and which have not. It could then provide a dump of the records where the dates are not aged properly and identify where there were mismatches.

What the tool does not require:

It requires no preconceived idea of the file content or structure (i.e. no copy books, no manual identification of where date are, nor the structure of each dates).

Common questions:

Q1: How can this tool know where the dates are, if one does not tell the tool where they are?

A1: The *DateWise File Comparison* tool does not know where the dates are without additional input. That additional input is the two files being compared. It does not look for dates unless the files do not match at any particular position. Only after a mismatch is detected does the tool search for an explanation of why they do not match.

Q2: How many date formats will the *DateWise File Comparison* tool recognize?

A2: This tool uses a parsing technique for recognizing dates. It will recognize literally thousands of date formats. This includes those with characters, number, separators and/or optional elements.

Q3: I use a proprietary format for my dates of the number of days since January 1, 1850, can the *DateWise File Comparison* tool work for me?

A3: Yes it will. The base date doesn't matter since the tool is only interested in the quantity of days between two dates (which is the same for any base date).

Q4: I have packed four digits of the year in the same space as the original two digit year, will the *DateWise File Comparison* tool work for me?

A4: Yes, but it will require some fine tuning by the manufacturer from the standard version. Contact the manufacturer with the specifics of your methodology.

Intelligent File Comparison

Q5: What if there are two possible interpretations of a date from each file that match.

How can the *DateWise File Comparison* tool accurately identify which one is correct?

A5: When the dates really do match, this situation actually happens quite frequently. The algorithm handles the problem in the most sure way. There is a slim chance of a single pair of mismatched dates being declared as matches by the program³. For this reason, in any testing effort, some diversity of data is expected and required.

Q6: Where can I get additional information on this tool?

A6: Visit DateWise, Ltd. at one of the following locations: <http://www.dateWise.com>; PO Box 14321, Columbus, OH 43214; vic@dateWise.com; or call (614) 799-2521 (phone or fax).

³ There is a corresponding possibility of bad data passing as good in any testing, just based on the data passed to the program. For example, consider a function to calculate squares of any value. If the program is mis-typed slightly, using a single "*" (for multiplication) rather than the proper symbol "**" (for exponentiation) and tested with the data values "0" and "2", it would pass testing. This is no different with the *DateWise File Comparison* tool. Certain combinations of data will pass testing, others will not.