

# Offsetting The Year 2000 'Demons

**By Peter King and Bob Neiht**

**Editor's note:** *The Year 2000 is a concern for everyone in the software industry. In order to provide a unique Business Basic perspective on possible Year 2000 solutions, The BASIS Advantage has invited a variety of Year 2000 solution providers to submit articles about their solutions. This article kicks off the series with a look at the Date Offset solution provided by QDEAL of Australia. Other Year 2000 solution providers interested in writing an article for this series should contact BASIS at [sales@basis.com](mailto:sales@basis.com).*

In the early days of software development, programmers often succumbed to the temptation to save storage space by designing software that dealt only with dates falling within the current century. By assuming the first two digits of a four-digit date--in other words, by not storing the 19 at the beginning of a date like 1972--thousands of bytes in the program's data files were saved but at a price developers are realizing could be very high.

As the next century approaches, making sure that a program's essential date functions work properly in 2000 and beyond has become much more important than saving a few bytes of storage space. Any program that needs to calculate dates across a century boundary will be unable to function properly starting on January 1, 2000--if not sooner. The effects will range from minor inconvenience to complete loss of mission-critical services for many companies. Software developers owe it to their customers and to themselves to minimize the effects that Year 2000 will have on their applications.

## ***The Date Offset Solution***

Any application with the Year 2000 problem deals, by definition, strictly with dates that fall within the 1900s--the range of dates between January 1, 1900, and December 31, 1999. But many of these applications can be modified to work on January 1, 2000, and beyond by changing a single assumption about the year values stored in their data files. Instead of treating those values as fixed references to particular years (00 is 1900, 72 is 1972, and so forth), developers can treat these values as year *offsets* and convert them to and from the actual dates with wraparound functions that work in the parts of the program dealing with data entry and output.

The commonly used Julian calendar operates on a repeating cycle that helps solve the problem. Every twenty-eight years, this calendar repeats itself with all

dates (including leap days) falling on the same days of the week, so that a printed calendar for 1970 is also valid for 1998 and 2026. By using a fixed date offset of twenty-eight years, a program which formerly worked with dates from 1900 to 1999 can transparently work with dates from 1928 to 2027. Around the year 2027, the offset can be shifted again, to 56 years and later to 84. The cycle only breaks down in 2100, which violates the usual four-year rule for leap years.



As soon as a date is entered in a software package which uses this method, a function is used to convert the actual year to the offset value which represents it. Another function is used to reverse the process anytime the date is output. Dates held in memory or on disk can be compared using the program's existing date calculation functions because the values representing the dates still fall between 00 and 99.

The disadvantage of this approach is that the twenty-eight year offset makes it necessary to store dates falling before 1928 using a different method. But for software packages with a maximum date span of one hundred years, which includes most software, the Date Offset Solution is an easy and cost-effective solution that can quickly make many applications Year 2000-compliant.

## ***Implementing Date Offsets***

The core of this solution is a pair of functions that do the actual work of converting a date to and from the offset value that represents it within the program's code. The software must also store the offset value that the functions should use. This value can be encoded in several different places:

- In the functions themselves, to be altered in the future using the BBx® utilities.
- In a variable set for each program screen on startup of BBx, protected by software `BeginS`.
- In the BBx global string table.

Because the offset value can be set to zero while the offset functions are being integrated into a software package, it is possible to make these modifications with no disruption during the upgrade process. A typical conversion would work as follows:

1. Merge the date conversion functions into each program in the software package that deals with date input or output.
2. Set the date offset for these functions to zero.
3. Wrap the code dealing with input and output of dates in the Date Offset functions, so that all entered dates are converted to the offset format and all output dates are converted back to the appropriate calendar dates.
4. Write a program that will search your program's data files and convert the dates inside them to the new offset format.

Once this process is completed, it is a simple matter to run the new date conversion program on any relevant data files, set the date offset used in your programs to its new value--probably 28--and return to transparently doing business as usual.

## ***Date-Offset Software Operating In The Field***

QDEAL owns and maintains the DEAL accounting software, which is designed for the parts and service industries. DEAL is a fully integrated accounting system with modules for unit inventory, parts inventory, workshop and service, rental management, general ledger, job costing, payroll, job streaming, spooling, and overseas purchase-order handling--all of which, as the end of the century approaches, have to be fully Year 2000-compliant.

As part of the process of becoming compliant, QDEAL has implemented the Date Offset Solution with a set of five functions that can be integrated into any software package. This



includes the two base Date Offset functions and three additional functions which convert dates between a number of human readable formats (such as the MMDDYY format used in the United States and the DDMMYY format used elsewhere) and convert dates in these formats to and from their offset representations. These additional functions have helped QDEAL to achieve its goal of making DEAL run anywhere in the world without software changes.

DEAL is made up of approximately 1800 programs consisting of nearly fourteen megabytes of source code, but the Date Offset solution was incorporated throughout the entire package by a single programmer in only six months. At this writing, all twenty-five Australian DEAL customer sites have installed the Date Offset version of the package, with nine of those customers currently operating with the 28-year offset enabled in their programs. Over two hundred seats are successfully running the offset-enabled version of DEAL, leading QDEAL and its clients to believe that this is one of the

safest Year 2000 solutions available. After seeing the success of the QDEAL solution in the field, the company now offers these functions as a separate package available to any Business Basic developer.

The Year 2000 challenge is a significant one that will only be conquered if it is dealt with as soon as possible. The solution implemented by QDEAL and offered in its separate Date Offset function package has the potential to help many developers bring their software up to speed in the few months remaining before January 1, 2000. 📧

QDEAL is located in Brisbane, Australia, and may be contacted by fax at (International Access Code +) 61.7.32646211.

***Peter King*** is the Managing Director of QDEAL. Peter has nearly thirty years of software development experience, with qualifications in Pure Mathematics.

***Bob Neihl*** is the Marketing Manager for QDEAL. Bob has more than twenty years of experience in the business and academic sectors.