4th Dimension and the Year 2000

Introduction

As we approach the year 2000, the computing industry is frantically scrutinizing all current hardware and software systems for compatibility with dates after the year 1999. In fact, the issue of the year 2000 is already a problem in using databases that manage Credit Cards and Insurance contracts. Both of these examples already deal with expiration dates that go beyond the "fatal" date of December 31, 1999.

4th Dimension has, from the very beginning, had the ability to handle dates from 1/1/100 AD through 12/31/32,000 AD. Therefore, 4th Dimension users and developers may be wondering: "Why all the concern and publicity about the year 2000? It is just another year." The purpose of this white paper is to acquaint you with the issues of year 2000 compliance, and to re-affirm that 4th Dimension is, and has always been, year 2000 compliant.

Note: all dates in this white paper are expressed in the US-style format: Month/Day/Year.

The Year 2000 - What are the Problems?

If a computer system does not support dates with four-digit years, this limitation will result in three problems: inability to store dates after the year 1999, inability to perform date arithmetic on dates after the year 1999, and inability to accept data entry of dates after the year 1999. Below is a brief discussion of these issues:

Inability to Store Dates After the Year 1999

For a computing system, the first issue with the year 2000 is the number of digits with which the year of a Date value is stored. If a system stores years in two digits (e.g. 12/27/97), the system must be rewritten, because it has no means to store years after 1999. For example, such a system assumes that the date 12/27/15 is in the year 1915; there is no way to store the year 2015.

Inability to Perform Date Arithmetic on dates after the year 1999

Most computer systems support "date arithmetic." The system can:

* Subtract one date from another and return a number of days:

  12/27/97 - 11/27/97 = 30 days

* Subtract a number of days from a date and return an earlier date:

  12/27/97 - 30 days = 11/27/97

* Add a number of days to a date and return a later date:

  11/27/97 + 30 days = 12/27/97

If the system does not support 4-digit years in dates, it has a problem performing the above simple operations when the calculations involve dates beyond the year 1999. The system will return results as follows:

12/27/01 - 11/27/97 = -35,034 (negative 35,034 days)

12/27/01 - 30 days = 11/27/01 (the year 1901)

12/27/99 + 30 days = 1/26/00 (the year 1900)

Inability to Accept Data Entry of Dates After the Year 1999

A common requirement of computer systems is to accept data entry of dates. The systems that have a problem with the year 2000 are those that only allow data entry of years with two digits: the user types the year and the system assumes that the year is a 19XX. If a system cannot accept dates with a 4-digit year, then the system is limited to accepting data entry of dates between 1/1/00 (January 1, 1900) and 12/31/99 (December 31, 1999). There is no ability for the user to enter a date such as January 1, 2000. That date will appear in the system as 1/1/00 and be interpreted by the system as January 1, 1900.

Clearly, companies using systems that do not recognize 4-digit dates must budget thousands or millions of dollars for software engineering to deal with this issue.

How 4th Dimension Handles Dates

4th Dimension can store, display, and perform date arithmetic on, all dates within the range 1/1/100 and 12/31/32,000 (from the year 100 AD through the year 32,000 AD). 4th Dimension has had this ability since the release of Version 1.0 of the product. This means that 4D does not have a problem with the year 2,000, or even the year 20,000. With that in mind, let's see how 4D will handle the three requirements of year 2000 compliance:

Ability to Store Dates After the Year 1999

As we mentioned earlier, the first issue with the year 2000 is the number of digits with which the year of a Date value is stored. If a system stores years in two digits (e.g. 12/27/97), the system must be rewritten, because it has no means to identify years after 1999. For example, such a system assumes that the date 12/27/15 is in the year 1915; there is no way to express the year 2015.

4th Dimension expresses dates as 3, 4, or 5 digits as appropriate. It can recognize and store all of the following dates: 12/27/115, 12/27/2015, and 12/27/20,115.

Ability to Perform Date Arithmetic on dates after the year 1999

4th Dimension supports "date arithmetic" on all dates within the range 1/1/100 through 12/31/32,000. For all dates within this range, 4D can:

* Subtract one date from another and return a number of days:

  12/27/2001 - 11/27/1997 = 1,491 days

* Subtract a number of days from a date and return an earlier date:

  12/27/2001 - 30 days = 11/27/2001

* Add a number of days to a date and return a later date:

  12/27/99 + 30 days = 1/26/2000

Ability to Accept Data Entry of Dates After the Year 1999

4th Dimension *allows* the entry of a two-digit year, but it does not *require* that the year be two digits. Depending on how the developer designed the data entry screen, the year can be three, four, or five digits. If the developer did not restrict data entry with an "entry filter," the user can enter any date between 1/1/100 and 12/31/32000.

The Developer's Responsibility

Although 4th Dimension itself is completely "year 2000 compliant," it is the responsibility of 4D developers to make sure that their applications written in 4D are also year 2000 compliant. This means that the developer must consider the following:

1) Data Entry Filters: Although 4D gives the developer the ability to create data entry filters to enforce business rules, the developer must not use those filters to prohibit four-digit dates in data entry screens.

2) Date Display Formats: Although 4D gives the developer the ability to create date display formats so that dates are always displayed in a certain way, the developer must not use the formatting ability of 4D to restrict date displays to two-digit years.

3) Default Century: If the developer wants to give the user the ability to enter a two-digit year (for speed and convenience), the developer must write logic into the program that makes an 'educated guess' as to which century the user intends to enter. However, the developer must also give the user a way to override the system if it makes an incorrect assumption.

Examples and Instructions for Writing Year 2000-Compliant Code

Please refer to ACI US Technical Note 96-37, "4th Dimension, The Year 2000" for instructions and examples of writing year 2000-compliant code. If your database is in Version 6.X of 4th Dimension or Version 6.X of 4D Server, please see also the V6 documentation for explanation and examples of the command SET DEFAULT CENTURY.

Summary

As we approach the year 2000, the computing industry is frantically scrutinizing all current hardware and software systems for compatibility with dates beyond the year 1999.

All versions of 4th Dimension are, and always have been, year 2000 compliant. If 4D developers follow the instructions in Technical Note 96-37 and the examples of the Version 6 command SET DEFAULT CENTURY, their applications will be ready for the year 2000 and beyond.

About ACI US

ACI US, Inc. was founded in 1987 as the US publisher of 4th Dimension and the 4D product line which has set the relational database standard for over ten years. In 1995, ACI extended 4th Dimension technology to the Windows platform.

ACI US, Inc.
20883 Stevens Creek Boulevard
Cupertino, CA 95014
http://www.acius.com

Tel 800-881-3466
408-252-4444
Fax 408-252-4829

Contacts

ACI US Marketing
Karen Purtich
kep@acius.com
408-252-4444 x285

ACI US Engineering
Gordon Muirhead
gordonm@acius.com
408-252-4444 x223