NOTE: This Technical Note has been retired. Please see the Technical Notes page for current documentation.

Technical Note TE09

Script Manager 2.0 Date & Time Problems

CONTENTS

This Technical Note describes known bugs and features in and solutions to the date and time routines introduced in Script Manager 2.0.

[Feb 01 1990]

Introduction

From the beginning, the Macintosh's ability to handle dates was limited to a rather small range--slightly more than a century. Enhancements to the Script Manager, introduced with System Software 6.0, extended this range to +/-35,000 years. Unfortunately, there is a minor bug in one of the crucial calls and a "feature" that looks like a bug in another.

Back to top

You Said It Would Be A Long Time

`_LongSecs2Date`, the routine that translates a `LongDateTime` to a `LongDateRec`, has a bug caused by using a variable that has not been properly initialized. This bug rears its ugly head when negative values are passed to the routine. System Software 6.0.4 and later fix this bug, and there is a simple solution for earlier systems.

If using System Software 6.0.3 and earlier, if you call `_LongSecs2Date` once before you really want to use it, the variable is cleared. After the initial call, `_LongSecs2Date` works correctly.

For example:

MPW Pascal

```
PROCEDURE DoDateStuff;
VAR
  lsecs: LongDateTime;
  ldr:   LongDateRec;
BEGIN
  InitDateCache(dcr);
  lsecs := 0;
  LongSecs2Date(lsecs,ldr);
  {now you can call LongSecs2Date for real}
```

MPW C

```
void DoDateStuff()
{
  LongDateTime lsecs;
  LongDateRec  ldr;


/* work around the bug */
  lsecs = 0;
  LongSecs2Date(&lsecs,&ldr);
/* now call LongSecs2Date for real */
```

Back to top

Any String To Date

The routine `_String2Date` was originally designed to be as forgiving as possible. It is so forgiving that it accepts any non-alphabetic character as a separator and accepts a single number as a valid date. For instance, if you pass `_String2Date` a string like "<20" it generously assumes that the less than sign (<) is intended as a divider and that "20" must be intended as a day, since there are only 12 months in a year. It returns a result of `noErr` and a date which is the twentieth of the current month in the current year. The string "<3*3" produces March 3 of the current year, while "4>1" politely gives the date April 1 of the current year.

This forgiveness really is not a bug, but a feature. Unfortunately it isn't a feature that has been greatly appreciated in the developer community. For that reason, the rules for date and time dividers are tighter in System 7.0. Current thinking is that all list separators now used in `'itl0'` resources will be allowed with a few common date separators used in the U.S. (e.g., colon (:) and hyphen (-)). For now, it is important to be aware of, shall we say, the flexibility of `_String2Date` and avoid thinking of it as an intelligent date parser. If you want to parse something, you can use `_IntlTokenize`.

Back to top

References

Inside Macintosh , Volume V, The Script Manager

The Script Manager 2.0 , Interim Chapter (DTS)

Back to top

Downloadables

Acrobat version of this Note (K). Download
