Date2Secs Convert a DateTimeRec into a "raw" seconds value

#include <OSUtils.h> **Operating System Utilities**

```
void           Date2Secs(dtrp, secs );
DateTimeRec    *dtrp ;          contains day, month, year, hour, etc.
unsigned long  *secs ;          receives "raw" seconds since 1/1/1904
```

Date2Secs converts the fields of a DateTimeRec into a "raw" seconds value. The result could be used in comparing two dates or setting the clock.

dtrp is the address of a 14-byte DateTimeRec structure. The dayOfWeek field is ignored. The other fields are used (in a flexible manner) to calculate the value for *secs*.

secs is the address of a 4-byte unsigned long int. Upon return, it will contain a "raw" seconds value corresponding to the values of the fields in the DateTimeRec addressed by *dtrp*.

Returns: none

Notes: **Date2Secs** is a handy intermediate step for generating a textual representation of a date (see **IUDateString**). It is also useful for adding, subtracting, or comparing two dates/times. For instance:

```
#define          SECS_PER_DAY (60*60*24)

DateTimeRec      ivcDate, shipDate;
unsigned long    ivcSecs, shipSecs;
short            elapsedDays;

Date2Secs( &ivcDate, &ivcSecs);
Date2Secs( &shipDate, &shipSecs);
elapsedDays = (shipSecs-ivcSecs)/SECS_PER_DAY;
```

Although the year and month fields of the DateTimeRec should be valid, this call is flexible about other fields. Thus, you can convert an "invalid" date (such as January 200 or today+30) to "raw" seconds, and back to date/time to make it valid. For instance:

```
DateTimeRec   theDate = {1989, 3,15 };        /* March 15 */

theDate.day +=30;                             /* March 45 */
Date2Secs( &theDate, &secs);
Secs2Date( secs, &theDate);                   /* April 14 */
```

Since the base date for the any "raw seconds" value is 1/1/1904 and since *secs* is a 32-bit value, you won't be able to calculate with dates beyond Feb. 6, 2040.