

How do I parse a string into a FILETIME?

 devblogs.microsoft.com/oldnewthing/20121102-00

November 2, 2012



Raymond Chen

Public Service Announcement: Daylight Saving Time ends in most parts of the United States this weekend. Other parts of the world may change on a different day from the United States.

The NLS functions in Win32 provide functions to convert a SYSTEMTIME into a string, but it does not provide any functions to perform the reverse conversion. Here are few things you can try:

The OLE automation VarDateFromStr conversion function converts a string into a **DATE** . From there, you can convert it to some other format.

```
BOOL SystemTimeFromStr(__in LPCWSTR psz, LCID lcid, __out LPSYSTEMTIME pst)
{
    DATE date;
    return SUCCEEDED(VarDateFromStr(psz, lcid, 0, &date)) &&
        VariantTimeToSystemTime(date, pst);
}
BOOL FileTimeFromStr(__in LPCWSTR psz, LCID lcid, __out LPFILETIME pft)
{
    SYSTEMTIME st;
    return SystemTimeFromStr(psz, lcid, &st) &&
        SystemTimeToFileTime(&st, pft);
}
```

If you have something in which parses CIM datetime format (which The Scripting Guys liken to Klingon) you can use the SWbemDateTime object. Since this is a scripting object, using it from C++ is rather cumbersome.

```

BOOL FileTimeFromCIMDateTime(__in LPCWSTR psz, __out LPFILETIME pft)
{
    BOOL fSuccess = FALSE;
    ISwbemDateTime *pDateTime;
    HRESULT hr = CoCreateInstance(__uuidof(SwbemDateTime), 0,
        CLSCTX_INPROC_SERVER, IID_PPV_ARGS(&pDateTime));
    if (SUCCEEDED(hr)) {
        BSTR bstr = SysAllocString(psz);
        if (bstr) {
            hr = pDateTime->put_Value(bstr);
            if (SUCCEEDED(hr)) {
                BSTR bstrFT;
                hr = pDateTime->GetFileTime(VARIANT_FALSE, &bstrFT);
                if (SUCCEEDED(hr)) {
                    __int64 i64FT = _wtoi64(bstrFT);
                    pft->dwLowDateTime = LODWORD(i64FT);
                    pft->dwHighDateTime = HIDWORD(i64FT);
                    fSuccess = TRUE;
                    SysFreeString(bstrFT);
                }
            }
            SysFreeString(bstr);
        }
        pDateTime->Release();
    }
    return fSuccess;
}

```

From the managed side, you have [DateTime.TryParse](#) and [DateTime.ParseExact](#) methods.

I leave you to investigate the time zone and locale issues associated with these techniques.
(Because I can't be bothered.)

[Raymond Chen](#)

Follow

