# Dave’s Expression Evaluator Add-in for Visual Studio 2008, 2010, 2012, 2013, 2015, 2017

This is an expanded version of the Microsoft sample expression evaluator EEAddin. It currently supports the following Windows types:

* FILETIME
* SYSTEMTIME
* PROPERTYKEY

## Background

Visual Studio has a useful sample debugging expression evaluator for the FILETIME and SYSTEMTIME structures <http://msdn.microsoft.com/en-us/library/8fwk67y3(v=vs.90).aspx> . Unfortunately, the original supplied sample would crash because the exported functions don’t have the correct calling convention (\_stdcall) defined. This version corrects this and extends the display of the FILETIME type to show the value interpreted as both UTC and local time as I’ve often found it useful to see both at the same time. It also displays PROPERTYKEY values in their canonical and display name forms.

If you want to build the project yourself, the source code and a Visual Studio 2015 project and solution are included.

## What you should see

Once installed, when you debug code that has the supported types variables, here’s an example of what you’ll see. The time and dates will display in your own locale format and time zone name, the examples are what I see with British UK settings:

|  |  |
| --- | --- |
| FILETIME | |
| Invalid | Invalid [0xcccccccc:0xcccccccc] 1 |
| Valid | [utc] 16/06/2016 13:14:15 [GMT Summer Time] 16/06/2016 14:14:15 |
| SYSTEMTIME | |
| Invalid | Invalid |
| Valid | 06/03/2014 17:03:022 |
| PROPERTYKEY | |
| Invalid | Invalid |
| Valid | Canonical Name "Display Name" e.g. System.ApplicationName "Program name" |

1 The two DWORD values making up the FILETIME are in high/low order. I’ve included them because I’ve found it’s sometimes useful to see what their underlying values are when the value is an invalid FILETIME value.

2 There’s no UTC/local interpretation for SYSTEMTIME - it’s unambiguous.

## Installation for VS2008/2010

I think VS2005 will also work, but haven’t tried it.

Modify your autoexp.dat file. You’ll need administrator access to modify it as it’s under the %ProgramFiles% directory. For example, on a 64-bit OS installation, it’s under here for VS2010:

C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\Packages\Debugger

The file already has commented entries for the sample so uncomment them and modify them to the location of the add-in DLL. For example:

\_SYSTEMTIME=$ADDIN(C:\MyPath\DavesEEAddin.dll,AddIn\_SystemTime)  
\_FILETIME=$ADDIN(C:\MyPath\DavesEEAddin.dll,AddIn\_FileTime)  
\_tagpropertykey=$ADDIN(C:\MyPath\DavesEEAddin.dll, AddIn\_PropertyKey

## Installation for VS 2012…2017 (and hopefully later versions)

VS2012 dropped the use of autoexp.dat; however, you can still use existing add-ins through the new mechanism - an XML “natvis” file, like this:

<?xml version="1.0" encoding="utf-8"?>

<AutoVisualizer xmlns="http://schemas.microsoft.com/vstudio/debugger/natvis/2010">

<Type Name="\_FILETIME">

<DisplayString LegacyAddin="C:\YourPath\YourAddin.dll"

Export="AddIn\_FileTime">

</DisplayString>

</Type>

<Type Name="\_SYSTEMTIME">

<DisplayString LegacyAddin="C:\YourPath\Youraddin.dll"

Export="AddIn\_SystemTime">

</DisplayString>

</Type>

<Type Name="\_tagpropertykey">

<DisplayString LegacyAddin="C:\YourPath\Youraddin.dll"

Export="AddIn\_PropertyKey">

</DisplayString>

</Type>

</AutoVisualizer>

The supplied “DavesEEAddin.natvis” and “DavesEEAddin.dll” can be copied together to either of these locations:

* %VSINSTALLDIR%\Common7\Packages\Debugger\Visualizers (requires admin access)
* %USERPROFILE%\My Documents\Visual Studio 201x\Visualizers\

There’s more information on natvis here “Create custom views of native objects in the debugger”: <http://msdn.microsoft.com/en-us/library/jj620914.aspx>