Dave's Development Blog

Software Development using Borland / Codegear / Embarcadero RAD Studio

# CodeSite and TCP/IP

By David | September 29, 2019 0 Comment

There are some circumstances when debugging an application where the default dispatch mechanism for CodeSite does not work. The default mechanism is to use the `WM_COPYDATA` windows message. In my case this is an Excel COM add-in and the `WM_COPYDATA` doesn't work so I had to switch to using TCP/IP.

There are a number of issues related to this. First, when using TCP/IP messages from your application, the application cannot automatically start the CodeSite Dispatcher and the second, which is the point of the post, if the CodeSite Dispatcher is not running all CodeSite messages will raise an exception as below:

```
Project CodeSiteTest.exe raised an exception class ECSIdSocketError with message
'Socket Error # 10061
Connection refused.'
```

A further consequence is that those message will never get to a log file either. If you are using CodeSite just purely for debugging and wrapping all the calls in `{$IFDEF DEBUG}` sections then your release code will be okay, however, if you start to use CodeSite in your production application for logging issues and exceptions (because lets face it, how many time have users said, "Oh yeah, I had an error this morning." to which you ask ,"What was the message?", and get the reply, "Don't know, I just cancelled the message". Doh!

So I needed to find out whether the Dispatcher is running on the same machine and in the same session as the application (this is just to ensure the exception aren't raised – they are seen by the debugger but don't seem to be propogated). I created a test application to enumerate all the top level windows running and after trawling through hundreds of the damn things on my new Windows 10 machine, I found a window with the class name `TfrmDispatcher`. I checked that it was associated with the dispatcher by closing and re-opening the dispatcher. So I found that all I needed to do was conditionally check that the dispatcher was running before changing the data transport mechanism to TCP/IP.

The below code is from the aforementioned application.

```
Const
  strLogExt = '.csl';
  strAppDataEnviroVar = 'AppData';
  strSeasonFallDir = '\Season''s Fall\Eidolon\';
  strFileDateFmt = ' yyyy-mm-dd';
  strTFrmDispatcherClassName = 'TFrmDispatcher';

Var
  FileDest: TCodeSiteDestination;
Initialization
  FileDest := TCodeSiteDestination.Create(Nil);
  FileDest.LogFile.Active := True;
  FileDest.LogFile.FilePath := GetEnvironmentVariable(
    strAppDataEnviroVar) + strSeasonFallDir;
  FileDest.LogFile.FileName := ChangeFileExt(
    ExtractFileName(ParamStr(0)),
    FormatDateTime(strFileDateFmt, Now()
  ) + strLogExt);
  FileDest.Viewer.Active := True;
  CodeSite.Destination := FileDest;
  If FindWindow(strTFrmDispatcherClassName, Nil) > 0 Then
    CodeSiteLogging.CodeSiteManager.ConnectUsingTcp();
Finalization
  FileDest.Free;
End.
```

This might help some but it is a corner case.

regards
Dave.

Category: Applications Eidolon Miscellaneous Tags: CodeSite

Iconic One Theme | Powered by Wordpress