

Dave's Development Blog

Software Development using Borland / Codegear /

Embarcadero RAD Studio



Check if the IDE is in a Modal state

By David | August 26, 2019

0 Comment

Yesterday I fixed a long standing bug in my IDE Auto Save (https://www.davidghoyle.co.uk/WordPress/?page_id=918) plug-in which would either lock up the IDE or as it did yesterday when working on another plug-in, crash the IDE.

What was happening was that I had created a new unit for some code and was in the middle of manually saving the code when my plug-in decided to auto save and displayed another Save As dialogue. Once this happens, and I've seen it many times now, you either end up not being able to save the file with the IDE locking up or as per yesterday, it crashed the IDE for the first time (I'm using 10.3.2).

So I needed a way to test whether the IDE was in a modal state before attempting to save any files. A quick web search shows the following item from Stack Overflow (<https://stackoverflow.com/questions/3306260/delphi-detecting-if-my-app-has-a-modal-dialog-open>) which looked promising.

The article refers to a property of the `TApplication` class called `Modal Level` which should indicated whether the application is in a modal state if the level returned is greater than zero. Since the IDE is a Delphi application this should be simple to check this property however under testing I found that this did not work all the time. For instance, in my primary case of saving a file the `Modal Level` returned zero but if the Open Unit dialogue is on the screen it returned 1. So that wasn't going to work 😞

However further, down the article someone suggested using the windows API call `IsWindowEnabled()` and I found that that worked all the time.

So I updated by code as follows to fix the issue:

```
Procedure TDGHAutoSaveWizard.TimerEvent(Sender: TObject);
```

```
{ $IFDEF DEBUG }
```

```
Const
```

```
    strIDEInModalState = 'IDE is in a modal state (%d)!';
```

```
{ $ENDIF DEBUG }
```

```
Begin
```

```
    FTimer.Enabled := False;
```

```
    Try
```

```
Inc(FCounter);
If FCounter >= FSettings.Interval Then
  Begin
    If FSettings.Enabled Then
      If IsWindowEnabled(Application.MainForm.Handle) Then // New Test
        Begin
          FCounter := 0; // Moved from earlier to ensure autosave engages
                        // immediately after the modal state finishes
          TDGHI DEAutoSaveTool sAPI Functions.SaveAllModifiedFiles(FSettings);
        End
        {$IFDEF DEBUG}
      Else // Debugging code where I would the issue
        CodeSite.SendFmtMsg(strIDEInModalState, [Application.ModalLevel])
        {$ENDIF DEBUG}
      ;
    End;
  Finally
    FTimer.Enabled := True;
  End;
End;
```

Regards

Dave

Category: AutoSave Delphi Open Tools API RAD Studio