Excel Custom Task Pane not showing

Asked 8 years, 5 months ago Modified 1 year, 5 months ago Viewed 7k times

7

I'm showing a custom task pane in an excel VSTO add-in, I'm building it and showing it as thus:

```
var ctrl = new CellTaskPane();
var pane = CustomTaskPanes.Add(ctrl, "Custom Sheet");
pane.DockPosition = Office.MsoCTPDockPosition.msoCTPDockPositionRight;
pane.DockPositionRestrict =
Office.MsoCTPDockPositionRestrict.msoCTPDockPositionRestrictNoChange;
pane.Visible = true;
```

This is being done in the `ThisAddin.cs` file and it working just fine on my machine, both under a Debug session and with the add-in installed via the click-once installer.

However, installing the add-in on a colleague's machine is proving troublesome.

The add-in is functioning and the context menu / ribbon is working perfectly, but the pane just refuses to show.

I have a toggle button on the ribbon which toggles the `Visible` property on the pane and even clicking that isn't forcing the pane to show.

Any help on this would be greatly appreciated, Google is proving useless for this.

Thanks.

I should mention that `CellTaskPane` is just a `UserControl` as per the docs on MSDN: http://msdn.microsoft.com/en-us/library/aa942846.aspx

excel vsto excel-2013 excel-addins customtaskpane

Share Follow edited Jul 7, 2014 at 13:00 asked Jul 4, 2014 at 15:50

Clint
6,070 2 24 48

I think it might be helpful to show how you define CellTaskPane? I mocked this up quickly with an example from MSDN and it worked fine with the following definition: public class CellTaskPane : System.Windows.Forms.UserControl { ....etc... – Richard Hansell Jul 7, 2014 at 10:42

1 One thing I have read in a few places is that if you have Excel configured to load up files automatically at start up then this can "interfere" with custom task panes. For example, if you have a "Personal.xls" or anything lurking in "XLStartup" then it might be worth disabling these temporarily? – Richard Hansell Jul 7, 2014 at 14:28

@RichardHansell Unfortunately there's nothing in the XLStart folder or many of the other locations he found by looking around on the internet... – Clint Jul 7, 2014 at 14:56

10 Answers

Sorted by:

Highest score (default) ⬍

答えが見つからない？日本語で聞いてみましょう。 ✕

4

Turns out it wasn't anything we were doing directly!

There was another add-in installed (third party) which for some bizarre reason was interfering with the pane being shown (no idea why or how).

Shame that Excel doesn't show any sort of error or at least throw an exception.

Ah well.

Share Follow

answered Jul 7, 2014 at 16:27

Clint

6,070 2 24 48

1 Hi Clint -- any idea which 3rd party add-in it was? We're seeing the same problem with our add in and Oracle SmartView. When SmartView is enabled, our CustomTaskPane windows are only visible when there is no worksheet open (they seem to be hidden behind the worksheet). – Eric Hirst Dec 5, 2014 at 0:17

Hi Clint - I have disabled all Add-Ins (COM+ and Excel), copied the code below as-is, and still cannot get the custom task pane to display. – Doug Kimzey Jul 26, 2020 at 16:53

3

+100

I suggest you try a very, very simple custom task pane first to see if this works. I put together the most simple example I could think of, basically a single text box that gets a value pushed into it and this is returned to the ribbon when a button is pushed.

If you were to try this then I would do it as a new solution. Create a new VSTO project with a "Designer mode" ribbon. Add a toggle button and a normal button below it. Then copy in this code:

ThisAddIn.cs

```
using System;
using Office = Microsoft.Office.Core;
```

```
    public partial class ThisAddIn
    {
        private Microsoft.Office.Tools.CustomTaskPane pane;
        private CellTaskPane ctrl = new CellTaskPane();

        private void ThisAddIn_Startup(object sender, System.EventArgs e)
        {
            pane = CustomTaskPanes.Add(ctrl, "Custom Sheet");
            pane.DockPosition = Office.MsoCTPDockPosition.msoCTPDockPositionRight;
            pane.DockPositionRestrict =
Office.MsoCTPDockPositionRestrict.msoCTPDockPositionRestrictNoChange;
            pane.Visible = true;
            pane.VisibleChanged += new EventHandler(taskPaneValue_VisibleChanged);
            ctrl.SetName("test");
        }

        private void taskPaneValue_VisibleChanged(object sender, System.EventArgs e)
        {
            Globals.Ribbons.Ribbon1.toggleButton1.Checked = pane.Visible;
        }

        public Microsoft.Office.Tools.CustomTaskPane TaskPane
        {
            get
            {
                return pane;
            }
        }

        public CellTaskPane MyContainer
        {
            get
```

Add a new class called CellTaskPane.cs:

```
using System;
using System.Windows.Forms;
using System.Drawing;
using System.ComponentModel;

namespace ExcelAddIn1
{
    public class CellTaskPane : System.Windows.Forms.UserControl
    {
        public System.Windows.Forms.TextBox test;

        public CellTaskPane()
        {
            InitializeComponent();
        }

        public void InitializeComponent()
        {
            test = new System.Windows.Forms.TextBox();
            test.Location = new System.Drawing.Point(120, 8);
            test.Size = new System.Drawing.Size(232, 20);
            test.TabIndex = 0;
            Controls.AddRange(new System.Windows.Forms.Control[] { test });
            Size = new System.Drawing.Size(375, 150);
        }
        public string GetName()
        {
            return test.Text;
        }
    }
}
```

Add the following code to Ribbon1.cs:

```
using System;
using Microsoft.Office.Tools.Ribbon;

namespace ExcelAddIn1
{
    public partial class Ribbon1
    {
        private void toggleButton1_Click(object sender, RibbonControlEventArgs e)
        {
            Globals.ThisAddIn.TaskPane.Visible =
((RibbonToggleButton)sender).Checked;
        }

        private void button1_Click(object sender, RibbonControlEventArgs e)
        {
            button1.Label = Globals.ThisAddIn.MyContainer.GetName();
        }
    }
}
```

Obviously you will need to do a little tweaking to get this to work, I tried to take the default names for a new project and the buttons.

When you run this you should get a custom task pane. When you switch to the "TabAddIn" and click the toggle button it should show/ hide the task pane. When you click the normal button the contents of the only field in the task pane should be copied as the button name. I defaulted this to "test", so even if the task pane isn't visible you can see if it is in memory or not?

I tested this and it appears to work fine. Basically this is just a hacked up version of the examples on MSDN. If you wanted you could probably do this yourself anyway? If nothing else this will enable you to see if there is anything in the more complex ribbon you are working on that causes issues... or if this is a fundamental problem with your colleague's machine.

Share Follow

answered Jul 7, 2014 at 15:16

Richard Hansell
5,210 1 18 35

Unfortunately the task pane I have is already very simple, it's just a few labels that I'm setting values to, nothing particularly special :/ – Clint Jul 7, 2014 at 15:18

Okay, can you communicate with the invisible task pane, i.e. set or return a value to one of the fields on it even though you can't actually see it? – Richard Hansell Jul 7, 2014 at 15:35

the pane is there. – Clint Jul 7, 2014 at 15:54

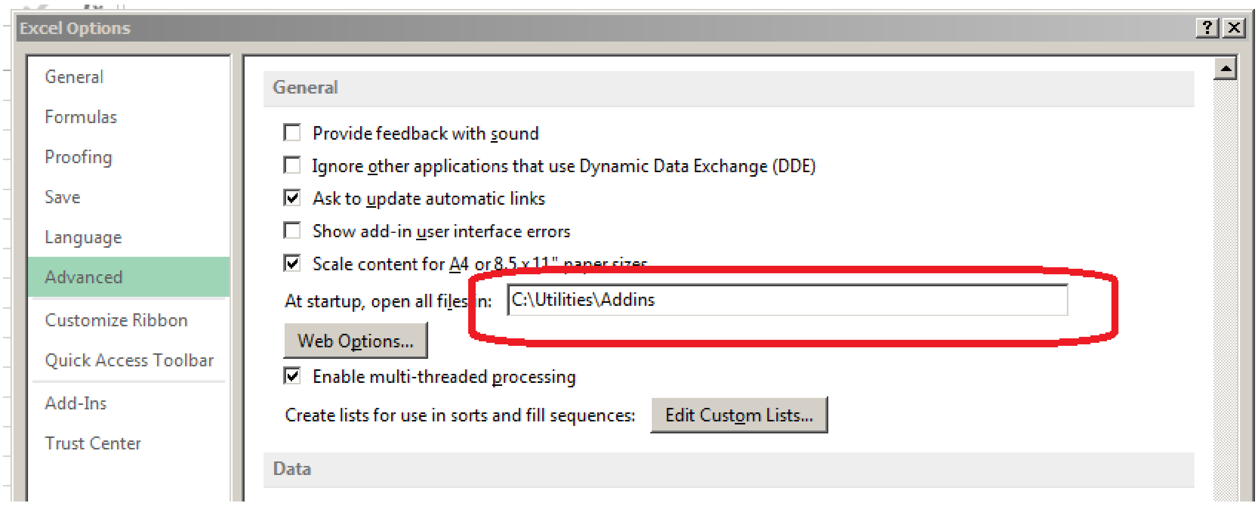
Have you tried making the custom task pane floating instead of docked? – Richard Hansell Jul 7, 2014 at 16:00

I've found out what was wrong, I've upvoted this answer so that you get the bounty when it expires though. Thanks for your help! – Clint Jul 7, 2014 at 16:28

3

I had the same problem, but it was not any addin I could disable (COM+ or Excel).

I had my excel configured to open files at startup (Excel Options -> Advanced -> General)



General

☐ Provide feedback with sound
☐ Ignore other applications that use Dynamic Data Exchange (DDE)
☑ Ask to update automatic links
☐ Show add-in user interface errors
☑ Scale content for A4 or 8.5 x 11" paper sizes
At startup, open all files in: C:\Utilities\Addins
☑ Enable multi-threaded processing
Create lists for use in sorts and fill sequences:

There, I had an .XLAM that customized the ribbon. When I cleared this configuration, my addin started working.

Share Follow

answered Mar 16, 2015 at 18:33

Fabio Marreco
2,056 1 22 22

This is a common thing we see as well, we have a corporate provided startup workbook that prevents our custom addin running entirely, so yes, other addins and indeed startup workbooks can both cause problems! – Clint Mar 17, 2015 at 11:29

2

I ran into exactly this problem while trying to get the Microsoft sample code for "Walkthrough: Synchronizing a Custom Task Pane with a Ribbon Button" working. Here's a link to the page:

http://msdn.microsoft.com/en-us/library/bb608590.aspx

After starting from scratch about three times and scouring the Internet for a clue as to what I might have been doing wrong, I came across this question and Clint's answer that an add-in was causing his problem. I had a few add-ins enabled, but with some trial and error I found the culprit: Microsoft's own "Analysis Toolpack"!

So, as Clint discovered, the first thing you should probably try if you run into this issue is to disable all add-ins and see if that does the trick. If so, then you can go back and begin turning them on until you find the one that is interfering with your custom pane visibility.

Share Follow

answered Dec 19, 2014 at 5:41
GaryP
91 1 6

Analysis Toolpack for me too. Had to restart Excel. – ThunderFrame Sep 7, 2016 at 6:39

1

Well, after following @GaryP's advice, disabling my other add-ins, and *thinking* that I'd solved the problem (albeit without access to my other add-ins), I discovered that the add-in would disappear whenever I opened more than one workbook.

But at that point, I didn't just get a missing taskpane or a silent fail, I actually got an error:

> The taskpane has been deleted or is otherwise no longer valid

So it seems that disabling add-ins isn't solving the problem in itself, but rather disabling add-ins is reducing the number of open workbooks (even if add-ins aren't visible, they can still have a Ribbon handle)...

The underlying cause is the use of SDI in 2013 and later.

So, now I can have all of my add-ins loaded.

Share Follow

edited May 23, 2017 at 11:46
Community Bot
1 1

answered Sep 8, 2016 at 3:19
ThunderFrame
9,191 2 27 57

1

Create a new instance of the task pane for each workbook. Make the following changes to your code and the task pane works even with addins enabled.

```
private void Application_WorkbookActivate(Microsoft.Office.Interop.Excel.Workbook wb)
{
    pane = CustomTaskPanes.Add(ctrl, "Custom Sheet");
    pane.DockPosition = Office.MsoCTPDockPosition.msoCTPDockPositionRight;
    pane.DockPositionRestrict =
Office.MsoCTPDockPositionRestrict.msoCTPDockPositionRestrictNoChange;
    pane.Visible = true;
    pane.VisibleChanged += new EventHandler(taskPaneValue_VisibleChanged);
    ctrl.SetName("test");
}

private void ThisAddIn_Startup(object sender, System.EventArgs e)
{
```

```
            Application_WorkbookActivate);
    }
```

Share Follow

edited Apr 18, 2018 at 2:21
NightOwl888
54.6k 24 133 207

answered Apr 18, 2018 at 2:13
kurainashe
11 1

1

If you find that even after closing all other add-ins, the TaskPane still doesn't show, it may be because it is loaded in your "Personal.xlsb" workbook. After closing it, I tried making the pane visible again and I received an error that it had been closed.

Share Follow

answered Aug 13, 2019 at 16:06
seadoggie01
491 1 4 17

0

I had the same problem and didn't fix it by disableing the analysis toolpack, but, rather i had to move the XLAM out of its installed folder (Break the reference to it, since you couldn't remove it through Excel) and it started working.

I've sense added the files back and it continues to work. Activating the addin does cause my custom taskbar to break. Not sure what this long term fix is here.

Share Follow

answered Dec 18, 2015 at 16:38
tyler
1

0

I know this is very old, but it can be useful for anyone who may look up for an answer, but here we go: if you are adding the new taskpane under ThisAddIn_Startup, it will only add it once at the start of the excel, and it will not be present for any other Excel session, so based on the following link that shows how to handle multiple sessions:

https://learn.microsoft.com/en-us/previous-versions/office/developer/office-2007/bb264456(v=office.12)?redirectedfrom=MSDN#Anchor_2

I came up with the conclusion that I should create a new taskpane under the other events that can fire when I needed the taskpane, then validate if the current window has the taskpane or not, and create a new one if not and show it. the event can be any trigger like ribbon button, open document, etc.

```
Dim CurrentTaskPane As Microsoft.Office.Tools.CustomTaskPane = Nothing
Globals.ThisAddIn.RemoveOrphanedTaskPanes() 'to remove any unused taskpane
For Each ctp As Microsoft.Office.Tools.CustomTaskPane In
Globals.ThisAddIn.CustomTaskPanes
    If ctp.Window.Hwnd = Excel.Application.ActiveWindow.Hwnd Then
```

```
Next
If CurrentTaskPane Is Nothing Then
    CurrentTaskPane = Globals.ThisAddIn.CustomTaskPanes.Add(New ControlName, "My
TaskPane", Excel.Application.ActiveWindow)
End If
CurrentTaskPane.Visible = True
```

You can find the 'RemoveOrphanedTaskPanes' code from the link provided.

Share Follow

answered Jul 14, 2021 at 23:41
Ash
1

-1

Summarizing other answers: It appears that this is due to loading other Add-ins, .XLAM files, etc. These can be loaded from many different places, and you need to check them all, and remove them. You may be able to reenable them later, so back everything up. Here is a checklist:

1. File -> Options -> Advanced -> General -> "At startup, open all files in..." Remove those files and disable the option.
2. File -> Options -> Add-ins -> Check all the Active application add-ins. Near the bottom of the dialog, use the Mange: Excel/Com Add-ins to view the Add-ins and disable or remove them. Even the Microsoft included ones could be causing it, so disable them as well. They do not allow you to remove them, which is fine, as long as they are disabled.
3. `C:\Users\$USERNAME\AppData\Roaming\Microsoft\Excel\XLSTART` Remove all files from this directory.
4. `C:\Users\$USERNAME\AppData\Roaming\Microsoft\AddIns` Remove all files from this directory.

After this, try again to load the Add-In. Then, add back the things you need one by one and keep testing. They may break it again, or not, there does not appear to be consensus on what kind of Add-In does and does not break the task pane.

If anyone discovers more places to look for Add-Ins, I will add them to the list.

Share Follow

answered Apr 12, 2018 at 15:51
Isaiah Shiner
381 7 15