

Lab 3 – TaskDialog

May, 2011 by A. Nagy
Updated by DevTech AEC WG
Last modified: 8/4/2015

<C#>C#バージョン</C#>

目的:この実習では、TaskDialog の機能を使用する方法を学習します。学習する項目は次のとおりです。

- TaskDialog の静的メソッドである Show() 関数を使用する
- TaskDialog のインスタンスを使用する

この実習の実装と確認の手順は、下記のとおりです:

1. 新しい外部コマンドを作成する
2. TaskDialog.Show()を使用してメッセージをポップアップする
3. TaskDialog インスタンスを使用する
4. 家の作成ダイアログ
5. サマリ

1. 新しい外部コマンドを作成する

現在のプロジェクトに新しい外部コマンドを追加します。

1.1 新しいファイルを追加して、プロジェクトに新しい外部コマンドを定義します。ファイル名とクラス名は、下記のようにしてください:

- ファイル名: **3_TaskDialog.cs**
- コマンド クラス名: **UITaskDialog**

(繰り返しになりますが、ここで希望する名前を使用しても構いません。ただし、その場合、プロジェクト名など、このドキュメント内では記述されている名称は、自分でつけた名称で代替して参照してください)

要求される名前空間:

この実習に必要な名前空間は次のとおりです:

- Autodesk.Revit.DB
- Autodesk.Revit.UI
- Autodesk.Revit.ApplicationServices
- Autodesk.Revit.Attributes
- Autodesk.Revit.UI.Selection(選択用)
-

注意: (VB.NET のみ):VB.NET でコードを記述している場合、プロジェクト・レベルで名前空間をインポートします(つまり、プロジェクトプロパティでは、各ファイル内で明示的にインポートする必要はありません)。

クラス内で UIApplication とアクティブな UIDocument を参照する変数を宣言します。

```
<C#>
[Transaction(TransactionMode.ReadOnly)]
public class UITaskDialog : IExternalCommand
{
    UIApplication _uiApp;
    UIDocument _uiDoc;

    public Result Execute(
        ExternalCommandData commandData,
        ref string message,
        ElementSet elements )
    {
        _uiApp = commandData.Application;
        _uiDoc = _uiApp.ActiveUIDocument;

        // ...
    }
}
</C#>
```

2. TaskDialog の静的メソッド Show()関数を使用する

ユーザに情報を通知する必要がある際には、一般に、ダイアログ ボックスを使用します。違和感なくアドインを Revit に統合するために、TaskDialog クラスを使用して、Revit 内蔵のポップアップ ダイアログと同じ見た目のダイアログをユーザに提示することができます。

TaskDialog は、多くのオーバーロードを持つ静的メソッド Show() を提供します。このメソッドは静的メソッドであり、呼び出す前に TaskDialog のインスタンスを作成する必要はありません。

Show() メソッドのすべてのオーバーロードメソッドは、ダイアログのヘッダー(タイトル)になるヘッダー文字列と、ユーザに提示するメインの文字列を必要とします。

これとは別に、ダイアログにボタン([Ok]/[はい]/[いいえ]/[キャンセル]/[再試行]/[閉じる])を追加したり、既定のボタンを設定したりすることができます。

```
<C#>
    // (1) simplest of all. title and main instruction.
    // has default [Close] button at lower right corner.

    TaskDialog.Show( "Task Dialog Static 1", "Main message" );

    // (2) this version accepts command buttons in addition to above.
    // Here we add [Yes] [No] [Cancel]

    TaskDialogResult res2 = default( TaskDialogResult );
    res2 = TaskDialog.Show( "Task Dialog Static 2", "Main message",
    ( TaskDialogCommonButtons.Yes | TaskDialogCommonButtons.No |
    TaskDialogCommonButtons.Cancel ) );

    // What did the user pressed?
    TaskDialog.Show( "Show task dialog", "You pressed: " +
res2.ToString() );

    // (3) this version accepts default button in addition to above.
    // Here we set [No] as a default (just for testing purposes).

    TaskDialogResult res3 = default( TaskDialogResult );
    TaskDialogResult defaultButton = TaskDialogResult.No;
    res3 = TaskDialog.Show( "Task Dialog Static 3", "Main message",
    ( TaskDialogCommonButtons.Yes | TaskDialogCommonButtons.No |
    TaskDialogCommonButtons.Cancel ), defaultButton );

    // What did the user press?

    TaskDialog.Show("Show task dialog", "You pressed: " + res3.ToString());
</C#>
```

3. TaskDialog のインスタンスを使用する



TaskDialog の機能を十分に活用するには、そのインスタンスを作成してプロパティを設定し、次に、インスタンスのShow() 関数を呼び出す必要があります。

この方法では、表示されるアイコンを指定したり、ユーザが選択できるカスタム オプションを追加したり、ユーザが詳細な情報にアクセスするためのWebサイトへのハイパーリンクを提示することが可能です。

```
</C#>
// (0) create an instance of task dialog to set more options.
TaskDialog myDialog = new TaskDialog( "Revit UI Labs - Task Dialog
Options" );
// Just declare stepByStep as bool and initialize to true.
if( stepByStep ) myDialog.Show(); // Just declare stepByStep

// (1) set the main area. These appear at the upper portion
// of the dialog.

myDialog.MainIcon = TaskDialogIcon.TaskDialogIconWarning;
// or TaskDialogIcon.TaskDialogIconNone.
if( stepByStep ) myDialog.Show();

myDialog.MainInstruction =
    "Main instruction: This is Revit UI Lab 3 Task Dialog";
if( stepByStep ) myDialog.Show();

myDialog.MainContent =
    "Main content: You can add detailed description here.";
if( stepByStep ) myDialog.Show();

// (2) set the bottom area
```

```

        myDialog.CommonButtons = TaskDialogCommonButtons.Yes |
TaskDialogCommonButtons.No | TaskDialogCommonButtons.Cancel;
        myDialog.DefaultButton = TaskDialogResult.Yes;
        if( stepByStep ) myDialog.Show();

        myDialog.ExpandedContent = "Expanded content: the visibility of this
portion is controled by Show/Hide button.";
        if( stepByStep ) myDialog.Show();

        myDialog.VerificationText =
            "Verification: Do not show this message again comes here";
        if( stepByStep ) myDialog.Show();

        myDialog.FooterText = "Footer: <a
href=\"http://www.autodesk.com/developrevit\">Revit Developer Center</a>";
        if( stepByStep ) myDialog.Show();

        // (4) add command links. You can add up to four links

        myDialog.AddCommandLink( TaskDialogCommandLinkId.CommandLink1, "Command
Link 1", "description 1" );
        if( stepByStep ) myDialog.Show();
        myDialog.AddCommandLink( TaskDialogCommandLinkId.CommandLink2, "Command
Link 2", "description 2" );
        if( stepByStep ) myDialog.Show();
        myDialog.AddCommandLink( TaskDialogCommandLinkId.CommandLink3, "Command
Link 3", "you can add up to four command links" );
        if( stepByStep ) myDialog.Show();
        myDialog.AddCommandLink( TaskDialogCommandLinkId.CommandLink4, "Command
Link 4", "Can also have URLs e.g. Revit Product Online Help" );
        //if (stepByStep) myDialog.Show();

        // Show it.
        TaskDialogResult res = myDialog.Show();
        if( TaskDialogResult.CommandLink4 == res )
        {
            System.Diagnostics.Process process =
                new System.Diagnostics.Process();

            process.StartInfo.FileName =
                "http://wikihelp.autodesk.com/Revit/enu/2014";
            process.Start();
        }

        TaskDialog.Show("Show task dialog", "The last action was: " +
            res.ToString());
</C#>

```

4. 家の作成ダイアログ

対話的に家を作成するか単に既定の家を作成するかユーザに選択を促したり、ダイアログをキャンセルさせたい場合には、TaskDialog を使用します。選択されたオプションによって、以前の実習で作成した対話型の家の

生成と既定の家の作成のどちらかを実行してください。ダイアログが取り消された場合には、何もしないようにします。

5. サマリ

この実習では、TaskDialog の機能を使用する方法を学習しました。学習した項目は次のとおりです。

- TaskDialog の静的メソッド Show() 関数を使用する
- TaskDialog のインスタンスを使用する