

Lab 7 – 共有パラメータ

Created by A. Nagy, May 2011

Updated by DevTech AEC WG

Last modified: 6/9/2017

<C#>C#バージョン</C#>

目的:この実習では、共有パラメータを作成する方法を学習します。学習する項目は次のとおりです。

- 共有パラメータ グループを作成する
- 共有パラメータ定義を作成する
- それを特定のカテゴリにバインドする 例 > ドア

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定義する
2. ヘルパー関数を作成する
3. ドア用の共有パラメータを作成する
4. ドキュメント毎のパラメータを作成する
5. サマリ

1. 新しい外部コマンドを定義する

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: **7_SharedParameter.cs**
- コマンド クラス名: **SharedParameter**

要求される名前空間:

いままで使用した名前空間に加えて、次の名前空間を加えてください:

- System.IO

さらに、作成するパラメータのプロパティを保持するいくつかの定数を宣言します。

```
<C#>
const string kSharedParamsGroupAPI = "API Parameters";
const string kSharedParamsDefFireRating = "API 耐火等級";
const string kSharedParamsPath = "C:\\temp\\SharedParams.txt";

public Result Execute(
    ExternalCommandData commandData,
    ref string message,
    ElementSet elements)
{
    UIDocument uidoc = commandData.Application.ActiveUIDocument;
    Application app = commandData.Application.Application;
    Document doc = uidoc.Document;
}
</C#>
```

2. ヘルパー関数を作成する

最初に、共有パラメータ ファイルを返す関数を定義する必要があります。この関数は共有パラメータ ファイルが存在しない際に、新しい共有パラメータ ファイルを作成する機能を持つものとしします。

```
<C#>
public static DefinitionFile GetSharedParamsFile(Application app)
{
    // Get current shared params file name
    string sharedParamsFileName;
    try
    {
        sharedParamsFileName = app.SharedParametersFilename;
    }
    catch (Exception ex)
    {
        TaskDialog.Show(
            "Get shared params file", "No shared params file set:" +
            ex.Message);
        return null;
    }

    if (0 == sharedParamsFileName.Length
```

```

        || !System.IO.File.Exists(sharedParamsFileName))
    {
        StreamWriter stream;
        stream = new StreamWriter(kSharedParamsPath);
        stream.Close();
        app.SharedParametersFilename = kSharedParamsPath;
        sharedParamsFileName = app.SharedParametersFilename;
    }

    // Get the current file object and return it
    DefinitionFile sharedParametersFile;
    try
    {
        sharedParametersFile = app.OpenSharedParameterFile();
    }
    catch (Exception ex)
    {
        TaskDialog.Show(
            "Get shared params file", "Cannot open shared params file:"
            + ex.Message);
        sharedParametersFile = null;
    }
    return sharedParametersFile;
}
</C#>

```

次に、パラメータグループが既に存在するかチェックして、それを返す関数を定義する必要があります。もし、パラメータグループが存在しない場合には、新しくパラメータを作成して返すものとします。

```

<C#>
public static DefinitionGroup GetOrCreateSharedParamsGroup(
    DefinitionFile sharedParametersFile,
    string groupName)
{
    DefinitionGroup g = sharedParametersFile.Groups.get_Item(groupName);
    if (null == g)
    {
        try
        {
            g = sharedParametersFile.Groups.Create(groupName);
        }
        catch (Exception)
        {
            g = null;
        }
    }
    return g;
}
</C#>

```

最後に、与えられたパラメータ定義が既に存在するかチェックして、それを返す関数を定義します。もし、パラメータ定義が存在しない場合には、新しくパラメータを作成して返すものとします。

```

<C#>
    public static Definition GetOrCreateSharedParamsDefinition(
        DefinitionGroup defGroup,
        ParameterType defType,
        string defName,
        bool visible)
    {
        Definition definition = defGroup.Definitions.get_Item(defName);
        if (null == definition)
        {
            try
            {
                //definition = defGroup.Definitions.Create(defName, defType,
visible);

                // 'Autodesk.Revit.DB.Definitions.Create(string,
Autodesk.Revit.DB.ParameterType, bool)' is obsolete:
                // 'This method is deprecated in Revit 2015. Use
Create(Autodesk.Revit.DB.ExternalDefinitonCreationOptions) instead'

                // Modified code for Revit 2015
                // and fixed typo in class name in Revit 2016

                ExternalDefinitionCreationOptions opt
                    = new ExternalDefinitionCreationOptions(
                        defName, defType);
                opt.Visible = true;
                definition = defGroup.Definitions.Create(opt);
            }
            catch (Exception)
            {
                definition = null;
            }
        }
        return definition;
    }
}

</C#>

```

3. ドア用の共有パラメータを作成する

The screenshot shows the 'プロパティ' (Properties) dialog box for a door. The 'API FireRating' field is highlighted with a red rectangle. The dialog box contains various sections for configuring the door, including '拘束' (Constraints), '建設' (Construction), 'マテリアルと仕上げ' (Materials and Finishes), '識別情報' (Identification Information), 'フェーズ' (Phases), and 'その他' (Other). The 'API FireRating' field is located under the 'その他' section.

プロパティ	
片側フラッシュ 0915 x 2134mm	
ドア (1) タイプを編集	
拘束	
レベル	レベル 1
敷居の高さ	0.0
建設	
フレーム タイプ	
マテリアルと仕上げ	
フレームのマテリアル	
仕上	
識別情報	
イメージ	
コメント	
マーク	2
フェーズ	
作成されたフェーズ	フェーズ 1
解体されたフェーズ	なし
その他	
上端高さ	2134.0
API FireRating	
プロパティ ヘルプ 適用	

ドア用の共有パラメータを作成するために必要となるのは、共有パラメータ ファイル、パラメータ定義グループ、パラメータ定義と、それをドア カテゴリに結び付ける作業です。

最初に、共有パラメータ ファイルを取得して、パラメータ グループが存在することを確認めます。

```
<C#>
// Get the current shared params definition file
DefinitionFile sharedParamsFile = GetSharedParamsFile(app);
if (null == sharedParamsFile)
{
    message = "Error getting the shared params file.";
    return Result.Failed;
}

// Get or create the shared params group
DefinitionGroup sharedParamsGroup = GetOrCreateSharedParamsGroup(
    sharedParamsFile, kSharedParamsGroupAPI);
if (null == sharedParamsGroup)
{
    message = "Error getting the shared params group.";
    return Result.Failed;
}
</C#>
```

次に、必要とする(ドア)カテゴリを取得して、共有パラメータ定義を作成します。

```
<C#>
    Category cat =
        doc.Settings.Categories.get_Item(BuiltInCategory.OST_Doors);

    // Visibility of the new parameter:
    // Category.AllowsBoundParameters property indicates if a category can
    // have shared or project parameters. If it is false,
    // it may not be bound
    // to shared parameters using the BindingMap.
    // Please note that non-user-visible
    // parameters can still be bound to these categories.

    bool visible = cat.AllowsBoundParameters;

    // Get or create the shared params definition

    Definition fireRatingParamDef = GetOrCreateSharedParamsDefinition(
        sharedParamsGroup, ParameterType.Number,
        kSharedParamsDefFireRating, visible);

    if (null == fireRatingParamDef)
    {
        message = "Error in creating shared parameter.";
        return Result.Failed;
    }
</C#>
```

共有パラメータ定義が作成できたら、それをドア カテゴリにバインドすることができます。

```
<C#>
    CategorySet catSet = app.Create.NewCategorySet();
    try
    {
        catSet.Insert(cat);
    }
    catch (Exception)
    {
        message = string.Format(
            "Error adding '{0}' category to parameters binding set.",
            cat.Name);
        return Result.Failed;
    }

    // Bind the param
    try
    {
        Binding binding = app.Create.NewInstanceBinding(catSet);

        // We could check if already bound, but looks like Insert will
        // just ignore it in such case
    }
    catch { }
```

```

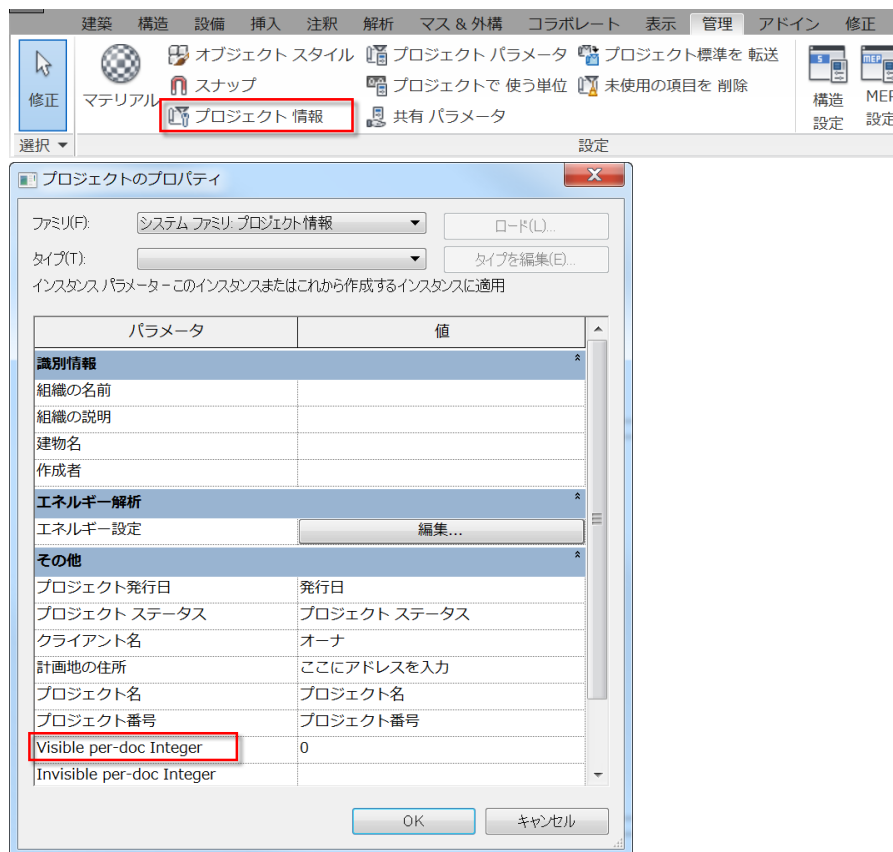
doc.ParameterBindings.Insert(fireRatingParamDef, binding);
}
catch (Exception ex)
{
    message = ex.Message;
    return Result.Failed;
}
}
</C#>

```

さて、アドインを実行して確認してみましょう。ユーザ インターフェイスでドアを選択すると、最初に示したプロパティパレットのイメージと同じように、追加したパラメータがプロパティパレットに表示されることを確認できるはずです。

もしアドインの実行時にエラーが発生する場合は、編集しようとしている共有パラメータファイルが読み取り専用となっている可能性があります。その場合は、GetSharedParamsFile メソッドの、sharedParamsFileName = app.SharedParametersFilename を囲む try - catch をコメントアウトして、再度実行してみてください。

4. ドキュメント毎にパラメータを作成する



ドキュメント毎のパラメータに表示/非表示を作成する新しいコマンドを追加します。ドキュメント毎のパラメータが意味するのは、プロジェクト内に単一のインスタンスを持つ要素に追加されるパラメータ、すなわち、プロジェクト

情報の要素のことです。前の実習と比較して、唯一の違いは、今回、作成したパラメータ定義を BuiltInCategory.OST_ProjectInformation カテゴリにバインドする点です。

さらに、FilteredElementCollector を使用してプロジェクト内でプロジェクト情報要素を検索、もしくは直接 Document.ProjectInformation プロパティから要素を取得し、新しく作成したパラメータ値を設定してください。

5. サマリ

この実習では、共有パラメータの作成方法を学習しました。学習した内容は次のとおりです。

- 共有パラメータ グループを作成する
- 共有パラメータ定義を作成する
- それを特定のカテゴリにバインドする 例 > ドア