BIM二开系列

- 基础篇____
- 模型篇____

基础篇

外部调用内部插件

```
Component component = new Component();
component.Name = teklaPLuginList.Name;
component.Number = teklaPLuginList.Number;
ComponentInput componentInput = new ComponentInput();
switch (teklaPLuginList.Name)
{
   case "DKQGGQCoQ螺栓连接      Q螺栓公共节点          ":
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      componentInput.AddOneInputPosition(new Picker().PickPoint());
      componentInput.AddOneInputPosition(new Picker().PickPoint());
      break;
   case "DKQGGQDoQ边柱与梁      Q端板竖放一        ":
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      break;
   case "DKQGGQDoQ边柱与梁      Q端板竖放二        ":
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      break;
   case "DKQGGQDoQ边柱与梁      Q端板平方      ":
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      break;
   default:
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
      break;

}
component.SetComponentInput(componentInput);
component.Insert();
_model.CommitChanges();
```

获取 Tekla文件夹目录

```
 if (_model.GetConnectionStatus())
 {
    marcofile = "";
    TeklaBMP = "";
```

```
    Teklaexe = "";
    string xsbin = "";
    TeklaStructuresSettings.GetAdvancedOption("XSBIN", ref xsbin);
    DirectoryInfo path = new DirectoryInfo(xsbin);
    string tt = path.Parent.Parent.FullName;//上  2层目录
    TeklaBMP = tt + "\\Bitmaps\\";
    Teklaexe = tt + "\\nt\\bin";
    tt += "\\nt\\bin\\plugins\\Tekla\\Model\\ModelTemplates\\KDplugin\\";
    marcofile = tt;
    List<string> vs = new List<string>();
    vs.Add(TeklaBMP);
    vs.Add(tt);
    vs.Add(Teklaexe);
    RememberKey.remeberKeyMethod(vs);
    return true;
}
```

获取插件列表

```
CatalogHandler catalogHandler = new CatalogHandler();
//组件管理器
//ComponentItemEnumerator componentItemEnumerator1 = catalogHandler.GetComponentItems();//
获取管理器下的组件项

if (catalogHandler.GetConnectionStatus())//有没有链接
{
    ComponentItemEnumerator componentItemEnumerator = catalogHandler.GetComponentItems();

    int t = catalogHandler.GetComponentItems().GetSize();

    while (componentItemEnumerator.MoveNext())//遍历当前组件管理器下的每个插件项
    {
        string pluginName = componentItemEnumerator.Current.Name;
        int pluginNumber = componentItemEnumerator.Current.Number;
        string[] type = pluginName.Split(new string[] { "Q" }, StringSplitOptions.RemoveEmptyEntries);
        if (type.Length > 0 && type[0] == "DK")
        {
            switch (type[1])
            {
                case "GG":
                    switch (type[2]) //简单的     switch(){case :beak;....default :break;}//获取到的组件的类型
                    {
                        case "Do":
                            addpluin(pluginName, pluginNumber, teklaPLuginListDo);
                            break;
                        case "Fr":
                            addpluin(pluginName, pluginNumber, teklaPLuginListFr);
                            break;
                        case "Tr":
                            addpluin(pluginName, pluginNumber, teklaPLuginListTr);
```

```
            break;
        case "Gr":
            addpluin(pluginName, pluginNumber, teklaPLuginListGr);
            break;
        case "Co":
            addpluin(pluginName, pluginNumber, teklaPLuginListCo);
            break;
        default:
            break;
    }
    break;
case "GJ":
    //□□□    switch(){case :beak;....default :break;}
    //□□□□□□□□□
    switch (type[2])
    {
        case "Pu":
            addpluin(pluginName, pluginNumber, teklaPLuginListPu);
            break;
        case "Pi":
            addpluin(pluginName, pluginNumber, teklaPLuginListPi);
            break;
        case "St":
            addpluin(pluginName, pluginNumber, teklaPLuginListSt);
            break;
        case "Wa":
            addpluin(pluginName, pluginNumber, teklaPLuginListWa);
            break;
        case "Pl":
            addpluin(pluginName, pluginNumber, teklaPLuginListPl);
            break;
        case "Ba":
            addpluin(pluginName, pluginNumber, teklaPLuginListBa);
            break;
        case "Go":
            addpluin(pluginName, pluginNumber, teklaPLuginListGo);
            break;
        default:
            break;
    }
    break;
case "TZ":
    switch (type[2]) //□□□    switch(){case :beak;....default :break;}//□□□□□□□□□
    {
        case "Re":
            addpluin(pluginName, pluginNumber, teklaPLuginListRe);
            break;
        case "As":
            addpluin(pluginName, pluginNumber, teklaPLuginListAs);
            break;
        case "Pa":
            addpluin(pluginName, pluginNumber, teklaPLuginListPa);
            break;
```

```
                case "Ga":
                    addpluin(pluginName, pluginNumber, teklaPLuginListGa);
                    break;
                case "To":
                    addpluin(pluginName, pluginNumber, teklaPLuginListTo);
                    break;
                default:
                    break;
            }
            break;
        case "BB":
            switch (type[2]) //简单的     switch(){case :beak;....default :break;}//获取到的组件的类型
            {
                case "Sl":
                    addpluin(pluginName, pluginNumber, teklaPLuginListSl);
                    break;
                case "Rb":
                    addpluin(pluginName, pluginNumber, teklaPLuginListRb);
                    break;
                case "Bo":
                    addpluin(pluginName, pluginNumber, teklaPLuginListBo);
                    break;
                default:
                    break;
            }
            break;
    }

  }

 }

}
```

碰撞检查初始化

```
        ClashCheckOptions clashCheckOptions = new ClashCheckOptions();//碰撞检查初始化:                d,t
        TeklaStructuresSettings.GetOptions(ref clashCheckOptions);//tekla设置或者得到碰撞将车的值
        MessageBox.Show("d=" + clashCheckOptions.BoltHeadDiameter.ToString()
            + "\r\n" + "t=" + clashCheckOptions.NutThickness.ToString(), "Tekla碰撞检查获取          ");
        clashCheckOptions.BoltHeadDiameter = 10.0;//d
        clashCheckOptions.NutThickness = 5.0;//t
        TeklaStructuresSettings.SetOptions(clashCheckOptions);//设置碰撞检查初始化值的
        MessageBox.Show("d=" + clashCheckOptions.BoltHeadDiameter.ToString()
            + "\r\n" + "t=" + clashCheckOptions.NutThickness.ToString(), "碰撞检查修改          ");
```

组件初始化

```
ComponentOptions componentOptions = new ComponentOptions();//组件
TeklaStructuresSettings.GetOptions(ref componentOptions);//获取
MessageBox.Show(
    "板名称     " + componentOptions.PlateProfileName + "\r\n" +
    "折叠板名称         " + componentOptions.FoldedPlateProfileName + "\r\n" +
     "螺栓边距系数           " + componentOptions.BoltEdgeDistanceFactor + "\r\n" +
     "比较距离       " + componentOptions.BoltEdgeDistanceReference + "\r\n" +
      "螺栓标准       " + componentOptions.BoltStandard + "\r\n" +
      "螺栓尺寸       " + componentOptions.BoltSize + "\r\n" +
      "零件材料       " + componentOptions.PartMaterial + "\r\n" +
       "焊接主构件的零件前缀                 " + componentOptions.PartWeldedToPrimaryPositionPrefix + "\r\n" +
    "焊接主构件的零件起始编号                   " + componentOptions.PartWeldedToPrimaryStartNumber + "\r\n" +
    "焊接次构件的零件前缀                " + componentOptions.PartWeldedToSecondaryPositionPrefix + "\r\n" +
    "焊接次构件的零件起始编号                   " + componentOptions.PartWeldedToSecondaryStartNumber + "\r\n" +
    "松散零件前缀           " + componentOptions.LoosePartPositionPrefix + "\r\n" +
    "松散零件其实编号             " + componentOptions.LoosePartStartNumber + "\r\n" +
    "构件松散零件编号前缀                 " + componentOptions.AssemblyLoosePartPositionPrefix + "\r\n" +
    "构件松散零件编号             " + componentOptions.AssemblyLoosePartStartNumber + "\r\n"
   , "Tekla组件默认值设置修改               ");
componentOptions.PlateProfileName = "测试呀名       ";
componentOptions.FoldedPlateProfileName = "折叠版名称         ";
componentOptions.BoltEdgeDistanceReference = ComponentOptions.BoltEdgeDistanceReferenceEnum.HOLE_DI
TeklaStructuresSettings.SetOptions(componentOptions);//修改
```

模块配置

```
  MessageBox.Show(
     "分析和设计管理           =" + ModuleManager.AnalysisAndDesign.ToString() + "\r\n" +
     "混凝土     =" + ModuleManager.ConcreteDetailing.ToString() + "\r\n" +
     "负载建模       =" + ModuleManager.LoadModeling.ToString() + "\r\n" +
     "多材质建模         =" + ModuleManager.MultimaterialModeling.ToString() + "\r\n" +
     "钢筋建模       =" + ModuleManager.RebarModeling.ToString() + "\r\n" +
     "钢结构详图         =" + ModuleManager.SteelDetailing.ToString() + "\r\n" +
     "任务管理       =" + ModuleManager.TaskManagement.ToString() + "\r\n" +
     "司晨提醒       " + "\r\n" +
     "当前程序运行的配置               =" + ModuleManager.Configuration, "Tekl软件配置信息          "


     );
```

Tekla路径文件相关

```
TeklaStructuresFiles teklaStructuresFiles = new TeklaStructuresFiles();//寻找   Tekla相关文件
List<string> filesList = new List<string>();
int i = 0;
string files = "";
while (i < teklaStructuresFiles.PropertyFileDirectories.Count)
{
   files += teklaStructuresFiles.PropertyFileDirectories[i] + "\r\n";
```

```
    i++;
}//遍历的    Tekla相关文件夹
MessageBox.Show(files, "查询路径列表          ");
string file = teklaStructuresFiles.GetAttributeFile("rebar_with_couplers.tpl").ToString();
//寻找相应文件的文件路径
MessageBox.Show(file);
bool vs = false;
filesList = teklaStructuresFiles.GetMultiDirectoryFileList("rpt", vs);
files = "";
foreach (string tt in filesList)
{
    files += tt + "\r\n";
}
MessageBox.Show(files, "查询文件后缀名的列表                ");
filesList.Clear();
FileInfo tttt = teklaStructuresFiles.GetAttributeFile(teklaStructuresFiles.PropertyFileDirectories, "rebar_with_coupl
MessageBox.Show(tttt.ToString(), "查询文件路径          ");
```

版本相关

```
 string infoS = "";
 infoS +=
 "内部版本      =" + TeklaStructuresInfo.GetBuildNumber() + "\r\n" +
 "环境安装文件夹           =" + TeklaStructuresInfo.GetCommonAppDataFolder() + "\r\n" +
 "Tekla结构版权      =" + TeklaStructuresInfo.GetCopyRightText() + "\r\n" +
 "Tekla结构版本      =" + TeklaStructuresInfo.GetCurrentProgramVersion() + "\r\n" +
 "Tekla结构用户      =" + TeklaStructuresInfo.GetCurrentUser() + "\r\n" +
 "Tekla结构的完整注册表项              =" + TeklaStructuresInfo.GetFullTSRegistryKeyText() + "\r\n" +
 "用户本地      appdata文件夹     =" + TeklaStructuresInfo.GetLocalAppDataFolder() + "\r\n" +
  "Tekla结构的修订日期           =" + TeklaStructuresInfo.GetRevisionDate();
 MessageBox.Show(infoS, "Tekla结构版本信息          ");
```

浇筑体相关

```
bool ispourendabled = TeklaStructuresSettings.IsPourEnabled();
MessageBox.Show(ispourendabled.ToString(), "是否启动浇筑体管理               ");
string istoolOR = "";
istoolOR += "全部工具选项         =" + TeklaStructuresSettings.IsToolOptionOn("sel_all") + "\r\n";
istoolOR += "分析节点链接工具选项              =" + TeklaStructuresSettings.IsToolOptionOn("sel_analysisnodelinks") + "\r\n";
istoolOR += "分析节点工具选项           =" + TeklaStructuresSettings.IsToolOptionOn("sel_analysisnodes") + "\r\n";
istoolOR += "分析部分工具选项           =" + TeklaStructuresSettings.IsToolOptionOn("sel_analysisparts") + "\r\n";
istoolOR += "单根钢筋工具选项           =" + TeklaStructuresSettings.IsToolOptionOn("sel_single_rebars") + "\r\n\r\n";

istoolOR += "施工线工具选项          =" + TeklaStructuresSettings.ToolOptionNames.ConstructionLines + "\r\n";
istoolOR += "自定义对象工具选项             =" + TeklaStructuresSettings.ToolOptionNames.CustomObjects + "\r\n";
istoolOR += "切割工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Cuts + "\r\n";
istoolOR += "直接操作工具选项           =" + TeklaStructuresSettings.ToolOptionNames.DirectManipulation + "\r\n";
istoolOR += "过滤器对话框工具选项
```

```
=" + TeklaStructuresSettings.ToolOptionNames.DisplaySelectionFilterDialog + "\r\n";
 istoolOR += "距离工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Distances + "\r\n";
 istoolOR += "轴网工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Grid + "\r\n";
 istoolOR += "轴网线工具选项          =" + TeklaStructuresSettings.ToolOptionNames.GridLine + "\r\n";
 istoolOR += "节点工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Joints + "\r\n";
 istoolOR += "负载工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Loads + "\r\n";
 istoolOR += "节点中对象工具选项              =" + TeklaStructuresSettings.ToolOptionNames.ObjectsInJoints + "\r\n";
 istoolOR += "零件工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Parts + "\r\n";
 istoolOR += "平面工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Planes + "\r\n";
 istoolOR += "点工具选项        =" + TeklaStructuresSettings.ToolOptionNames.Points + "\r\n";
 istoolOR += "终止工具选项         =" + TeklaStructuresSettings.ToolOptionNames.PourBreaks + "\r\n";
 istoolOR += "钢筋组工具选项          =" + TeklaStructuresSettings.ToolOptionNames.RebarGroup + "\r\n";
 istoolOR += "钢筋工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Rebars + "\r\n";
 istoolOR += "参考模型工具选项            =" + TeklaStructuresSettings.ToolOptionNames.ReferenceModels + "\r\n";
 istoolOR += "螺丝钉工具选项          =" + TeklaStructuresSettings.ToolOptionNames.Screws + "\r\n";
 istoolOR += "选择构件工具选项            =" + TeklaStructuresSettings.ToolOptionNames.SelectAssemblies + "\r\n";
 istoolOR += "选择构建中的对象工具选项
=" + TeklaStructuresSettings.ToolOptionNames.SelectObjectsInAssemblies + "\r\n";
 istoolOR += "选择任务工具选项            =" + TeklaStructuresSettings.ToolOptionNames.SelectTasks + "\r\n";
 istoolOR += "单根栓钉工具选项            =" + TeklaStructuresSettings.ToolOptionNames.SingleScrews + "\r\n";
 istoolOR += "表面工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Surfaces + "\r\n";
 istoolOR += "视图工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Views + "\r\n";
 istoolOR += "焊缝工具选项         =" + TeklaStructuresSettings.ToolOptionNames.Weldings + "\r\n";
 MessageBox.Show(istoolOR, "工具选项       ");
 //TeklaStructuresSettings.GetAdvancedOption("XS_ENABLE_POUR_MANAGEMENT",ref istoolOR);
 TeklaStructuresSettings.GetAdvancedOption("XS_POUR_BREAK_COLOR", ref istoolOR);
 MessageBox.Show(istoolOR, "高级选项的名称              ");
```

轮廓点相关

```
 Beam beam = (Beam)new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART);//获取对象
 Solid solid = beam.GetSolid();
 FaceEnumerator faceEnumerator = solid.GetFaceEnumerator();//获取到的面           集合
 int i = 0;
 while (faceEnumerator.MoveNext()) //每个面
 {
    LoopEnumerator loopEnumerator = faceEnumerator.Current.GetLoopEnumerator();//获取当前面上的边                集合
    while (loopEnumerator.MoveNext())  //每条当前面上的边
    {
       VertexEnumerator vertexEnumerator = loopEnumerator.Current.GetVertexEnumerator();//壳体上的         点的
集合
       while (vertexEnumerator.MoveNext())// 壳体上的点（单独点）
       {
          dataGridView1.Rows.Add();//添加表中一行
          dataGridView1.Rows[i].Cells[0].Value = string.Format("坐标点     {0}", i);
          dataGridView1.Rows[i].Cells[1].Value = vertexEnumerator.Current as Point;//
当前面上的当前闭合线上形成壳体上的一点
          i++;
       }
    }
```

```
}
```

□□□□

```
string sourceFile = @"albl_up_Developer--main_menu.xml";
string destinationFile = @"C:\Users\Administrator\AppData\Local\Trimble\Tekla Structures\2020.0\UI\Ribbons\albl_
bool isrewrite = true; // true=□□□□□□□□□□                ,false□□□
System.IO.File.Copy(sourceFile, destinationFile, isrewrite);
```

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Ribbon Format="2016i">
 <QuickAccessToolbar>
  <SimpleButton X="0" Y="0" Width="1" Height="1" Command="Common.Save" Icon="command:ScalableIcon"
  <Separator X="1" Y="0" Width="1" Height="1" Orientation="Vertical" Thickness="1" />
  <SimpleButton X="2" Y="0" Width="1" Height="1" Command="Common.Undo" Icon="command:ScalableIcon"
  <SimpleButton X="3" Y="0" Width="1" Height="1" Command="Edit.Redo" Icon="command:ScalableIcon" Sho
 </QuickAccessToolbar>
 <StaticTab>
  <SplitButton X="0" Y="0" Width="2" Height="2" Command="Common.Interrupt" Icon="command:ScalableIcor
   <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Common.Interrupt" Text="command:FullText
   <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Edit.SelectAll" Text="command:FullText" Icon
   <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Edit.SelectPrevious" Text="command:FullTex
   <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Edit.SelectByIdentifier" Text="command:FullT
  </SplitButton>
  <SplitButton X="0" Y="2" Width="2" Height="2" Command="Inquiry.Object" Icon="command:ScalableIcon" Sh
   <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Inquiry.Object" Text="command:ShortText" Ic
   <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Inquiry.PointCoordinates" Text="command:Sh
   <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Inquiry.CenterOfGravity" Text="command:Sh
   <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Inquiry.Custom" Text="command:FullText" Ic
   <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Inquiry.WeldedParts" Text="command:ShortT
   <SimpleButton X="0" Y="10" Width="8" Height="2" Command="Inquiry.PrimaryWeldedPart" Text="comman
   <SimpleButton X="0" Y="12" Width="8" Height="2" Command="Inquiry.AssemblyObjects" Text="command:S
   <SimpleButton X="0" Y="14" Width="8" Height="2" Command="Inquiry.ComponentObjects" Text="comman
   <SimpleButton X="0" Y="16" Width="8" Height="2" Command="Inquiry.Phases" Text="command:ShortText"
   <SimpleButton X="0" Y="18" Width="8" Height="2" Command="Inquiry.ModelSize" Text="command:ShortTe
  </SplitButton>
 </StaticTab>
 <Tab Header="translation:ribbon_Steel" IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="3" Height="4" Command="Modeling.CreateSteelColumn" Text="comman
  <SplitButton X="3" Y="0" Width="3" Height="4" Command="Modeling.CreateSteelBeam" Text="command:Sh
   <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateSteelBeam" Text="command
   <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.CreateSteelPolybeam" Text="comm
   <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Modeling.CreateSteelCurvedBeam" Text="co
   <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Modeling.CreateSteelTwinProfile" Text="com
   <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Modeling.CreateSteelOrthogonalBeam" Text=
   <SimpleButton X="0" Y="10" Width="8" Height="2" Command="Modeling.CreateSteelSpiralBeam" Text="co
  </SplitButton>
  <SplitButton X="6" Y="0" Width="3" Height="4" Command="Modeling.CreateSteelContourPlate" Text="comm
   <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateSteelContourPlate" Text="co
```

```
    <SimpleButton X="0" Y="2" Width="8" Height="3" Command="Modeling.CreateCylindricalBentPlate" Text="c
    <SimpleButton X="0" Y="5" Width="8" Height="3" Command="Modeling.CreateConicalBentPlate" Text="com
    <SimpleButton X="0" Y="8" Width="8" Height="3" Command="Modeling.CreateStandaloneBend" Text="com
    <SimpleButton X="0" Y="11" Width="8" Height="3" Command="Modeling.CreateLoftedPlate" Text="commar
  </SplitButton>
  <SimpleButton X="9" Y="0" Width="3" Height="4" Command="Bolts.CreateBolts" Text="command:ShortText"
  <SplitButton X="12" Y="0" Width="4" Height="4" Command="Weld.CreateBetweenParts" Text="command:Sh
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Weld.CreateBetweenParts" Text="command:
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Weld.CreateToPart" Text="command:FullText
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Weld.CreatePolygon" Text="command:FullTe
    <Separator X="0" Y="6" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <SimpleButton X="0" Y="7" Width="8" Height="3" Command="Weld.PreparePart.WithPolygon" Text="comma
    <SimpleButton X="0" Y="10" Width="8" Height="3" Command="Weld.PreparePart.WithAnotherPart" Text="c
    <SimpleButton X="0" Y="13" Width="8" Height="2" Command="Weld.ConvertToPolygonWeld" Text="comma
  </SplitButton>
  <SimpleButton X="16" Y="0" Width="8" Height="1" Command="Modeling.CreateSteelItem" Text="command:S
  <SimpleButton X="16" Y="1" Width="8" Height="2" Command="Bolts.EditBoltedParts" Text="command:Short
  <DropdownButton X="16" Y="3" Width="8" Height="1" Text="translation:albl_Assembly" Icon="resource:Brush
    <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Assembly.MakeIntoAssembly" Text="commar
    <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Assembly.AddAsSubAssembly" Text="comma
    <SimpleButton X="0" Y="6" Width="8" Height="3" Command="Assembly.SetAsNewMainObjectOfAssembly" T
    <SimpleButton X="0" Y="9" Width="8" Height="3" Command="Assembly.RemoveFromAssembly" Text="com
    <SimpleButton X="0" Y="12" Width="8" Height="3" Command="Assembly.Explode" Text="command:ShortTe
    <SimpleButton X="0" Y="15" Width="8" Height="3" Command="Assembly.ExplodeSubAssembly" Text="com
  </DropdownButton>
</Tab>
<Tab Header="translation:ribbon_Concrete" IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="3" Height="4" Command="Modeling.CreateConcreteColumn" Text="comn
  <SplitButton X="3" Y="0" Width="3" Height="4" Command="Modeling.CreateConcreteBeam" Text="command
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateConcreteBeam" Text="comm
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.CreateConcretePolybeam" Text="co
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Modeling.CreateConcreteSpiralBeam" Text="
  </SplitButton>
  <SplitButton X="6" Y="0" Width="3" Height="4" Command="Modeling.CreateConcretePanel" Text="command
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateConcretePanel" Text="comm
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.CreateWallLayout" Text="command
  </SplitButton>
  <SplitButton X="9" Y="0" Width="3" Height="4" Command="Modeling.CreateConcreteSlab" Text="command:S
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateConcreteSlab" Text="comma
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.CreateLoftedSlab" Text="command
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Modeling.CreateFloorLayout" Text="comman
  </SplitButton>
  <SplitButton X="12" Y="0" Width="4" Height="4" Command="Modeling.CreateConcretePadFooting" Text="tra
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateConcretePadFooting" Text="c
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.CreateConcreteStripFooting" Text="
  </SplitButton>
  <SimpleButton X="16" Y="0" Width="6" Height="2" Command="Modeling.CreateConcreteItem" Text="comma
  <DropdownButton X="16" Y="2" Width="6" Height="2" Text="translation:albl_Cast_unit" Icon="resource:Brush
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="CastUnit.Create" Text="command:FullText" Ic
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="CastUnit.AddToCastUnit" Text="command:Fu
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="CastUnit.RemoveFromCastUnit" Text="comm
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="CastUnit.Explode" Text="command:ShortText
    <Separator X="0" Y="8" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
```

```
    <SimpleButton X="0" Y="9" Width="8" Height="2" Command="CastUnit.SetTopInFormFace" Text="command
    <SimpleButton X="0" Y="11" Width="8" Height="2" Command="CastUnit.ShowTopInFormFace" Text="comm
  </DropdownButton>
  <SplitButton X="22" Y="0" Width="4" Height="4" Command="Reinforcement.CreateReinforcingBarGroup" Tex
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Reinforcement.CreateReinforcingBarGroup" T
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Reinforcement.CreateRebar" Text="command
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Reinforcement.CreateCurvedReinforcingBarG
    <SimpleButton X="0" Y="6" Width="8" Height="3" Command="Reinforcement.CreateCircularRebarGroup" Te
    <SimpleButton X="0" Y="9" Width="8" Height="2" Command="Reinforcement.CreateReinforcementMesh" Te
    <SimpleButton X="0" Y="11" Width="8" Height="2" Command="Reinforcement.CreateStrandPattern" Text="
    <SimpleButton X="0" Y="13" Width="8" Height="3" Command="Reinforcement.CreateReinforcementSplice"
    <Separator X="0" Y="16" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <SimpleButton X="0" Y="17" Width="8" Height="3" Command="Reinforcements.RebarShapeCatalog" Text="
    <SimpleButton X="0" Y="20" Width="8" Height="2" Command="Reinforcement.AttachToPart" Text="comma
    <SimpleButton X="0" Y="22" Width="8" Height="2" Command="Reinforcement.DetachFromPart" Text="com
    <Separator X="0" Y="24" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <SimpleButton X="0" Y="25" Width="8" Height="2" Command="Reinforcement.GroupRebars" Text="comma
    <SimpleButton X="0" Y="27" Width="8" Height="2" Command="Reinforcement.UngroupRebars" Text="comm
  </SplitButton>
  <DropdownButton X="26" Y="0" Width="8" Height="2" Text="translation:ribbon_Rebar_set" Icon="resource:Bi
    <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Reinforcement.CreateCrossingBeamReinforce
    <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Reinforcement.CreateLongitudinalBeamReinf
    <SimpleButton X="0" Y="6" Width="8" Height="3" Command="Reinforcement.CreateSlabOrWallReinforceme
    <SimpleButton X="0" Y="9" Width="8" Height="3" Command="Reinforcement.CreateRebarsOnSurface" Text
    <SimpleButton X="0" Y="12" Width="8" Height="3" Command="Reinforcement.CreateReinforcementSetViaP
    <SimpleButton X="0" Y="15" Width="8" Height="3" Command="Catalogs.RebarSetShapeCatalog" Text="con
    <SimpleButton X="0" Y="18" Width="8" Height="3" Command="Reinforcement.RegenerateSelectedReinforce
  </DropdownButton>
  <DropdownButton X="26" Y="2" Width="8" Height="2" Text="translation:ribbon_Rebar_display_options" Icon=
    <CheckButton X="0" Y="0" Width="8" Height="2" Command="Reinforcement.RebarSetLegFaceVisibility" Tex
    <CheckButton X="0" Y="2" Width="8" Height="2" Command="Reinforcement.RebarSetGuidelineVisibility" Te
    <CheckButton X="0" Y="4" Width="8" Height="2" Command="Reinforcement.RebarSetPropertyStripVisibility
    <CheckButton X="0" Y="6" Width="8" Height="2" Command="Reinforcement.RebarSetSplitterVisibility" Text
    <CheckButton X="0" Y="8" Width="8" Height="2" Command="Reinforcement.RebarSetEndDetailStripVisibilit
    <CheckButton X="0" Y="10" Width="8" Height="2" Command="Reinforcement.RebarDimensionVisibility" Tex
    <Separator X="0" Y="12" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <CheckButton X="0" Y="13" Width="8" Height="2" Command="Reinforcement.RebarSetGroupColoring" Text
  </DropdownButton>
  <SimpleButton X="34" Y="0" Width="8" Height="1" Command="Representation.ShowPours" Text="command:
  <DropdownButton X="34" Y="1" Width="8" Height="1" Text="translation:Commands.Pours.PourBreak.FullText
    <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Modeling.CreatePourBreak.UsingOnePoint" Te
    <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Modeling.CreatePourBreak.UsingTwoPoints" T
    <SimpleButton X="0" Y="6" Width="8" Height="3" Command="Modeling.CreatePourBreak.UsingMultiplePoint
  </DropdownButton>
  <SimpleButton X="34" Y="2" Width="8" Height="2" Command="Pours.CalculatePourUnits" Text="command:Fu
</Tab>
<Tab Header="translation:ribbon_Edit" IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Detailing.CutPart.WithPolygon" Text="commar
  <SimpleButton X="0" Y="2" Width="8" Height="1" Command="Detailing.CutPart.WithLine" Text="command:S
  <SimpleButton X="0" Y="3" Width="8" Height="1" Command="Detailing.CutPart.WithAnotherPart" Text="com
  <SimpleButton X="8" Y="0" Width="8" Height="2" Command="Detailing.FitPartEnd" Text="command:FullText
  <SimpleButton X="8" Y="2" Width="8" Height="1" Command="Edit.Split" Text="command:FullText" Icon="co
  <SimpleButton X="8" Y="3" Width="8" Height="1" Command="Edit.Combine" Text="command:FullText" Icon=
```

```
<SimpleButton X="16" Y="0" Width="8" Height="2" Command="Chamfer.CreateForPartEdge" Text="comman
<DropdownButton X="16" Y="2" Width="8" Height="2" Text="translation:albl_Inquire_added_material" Icon="
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Material.AttachToPart" Text="command:Shor
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Material.DetachFromPart" Text="command:S
  <SimpleButton X="0" Y="6" Width="8" Height="3" Command="Material.ExplodePart" Text="command:FullTe
</DropdownButton>
<DropdownButton X="24" Y="0" Width="8" Height="2" Text="translation:albl_Components" Icon="resource:Br
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Components.ApplicationsAndComponents" Te
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Components.CreateCurrentConnection" Text=
  <SimpleButton X="0" Y="6" Width="8" Height="3" Command="AutoConnection.Create" Text="command:Full
</DropdownButton>
<DropdownButton X="24" Y="2" Width="8" Height="1" Text="translation:ribbon_Surfaces" Icon="resource:Bru
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="SurfaceTreatment.CreateToPartFace" Text="c
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="SurfaceTreatment.CreateToSelectedAreaOnPa
  <SimpleButton X="0" Y="6" Width="8" Height="3" Command="SurfaceTreatment.CreateToAllFacesOfPart" Te
  <Separator X="0" Y="9" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
  <SimpleButton X="0" Y="10" Width="8" Height="3" Command="SurfaceTreatment.CreateSurfaceObject" Tex
</DropdownButton>
<DropdownButton X="24" Y="3" Width="8" Height="1" Text="translation:albl_Compare" Icon="resource:Brush
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Tools.CompareParts" Text="command:FullTex
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Tools.CompareAssemblies" Text="command:
</DropdownButton>
<SplitButton X="32" Y="0" Width="4" Height="4" Command="Measure.Distance" Text="translation:ribbon_Me
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Measure.Distance" Text="command:ShortTex
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Measure.HorizontalDistance" Text="comman
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Measure.VerticalDistance" Text="command:S
  <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Measure.Angle" Text="command:ShortText" I
  <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Measure.Arc" Text="command:ShortText" Ico
  <SimpleButton X="0" Y="10" Width="8" Height="2" Command="Measure.BoltSpacing" Text="command:Shor
</SplitButton>
<SimpleButton X="36" Y="0" Width="6" Height="2" Command="Edit.Copy" Text="command:FullText" Icon="c
<DropdownButton X="36" Y="2" Width="6" Height="2" Text="translation:lbl_Copy_special" Icon="resource:Bru
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Edit.CopySpecial.Linear" Text="command:Sh
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Edit.CopySpecial.Mirror" Text="command:Sho
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Edit.CopySpecial.Rotate" Text="command:Sh
  <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Edit.CopySpecial.ToAnotherObject" Text="cor
  <SimpleButton X="0" Y="8" Width="8" Height="3" Command="Edit.CopySpecial.AllContentToAnotherObject"
  <SimpleButton X="0" Y="11" Width="8" Height="2" Command="Edit.CopySpecial.ToAnotherPlane" Text="co
  <SimpleButton X="0" Y="13" Width="8" Height="2" Command="Edit.CopySpecial.FromAnotherModel" Text="
</DropdownButton>
<SimpleButton X="42" Y="0" Width="6" Height="2" Command="Edit.Move" Text="command:FullText" Icon="c
<DropdownButton X="42" Y="2" Width="6" Height="2" Text="translation:lbl_Move_special" Icon="resource:Br
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Edit.MoveSpecial.Linear" Text="command:Sh
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Edit.MoveSpecial.Rotate" Text="command:Sh
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Edit.MoveSpecial.Mirror" Text="command:Sho
  <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Edit.MoveSpecial.ToAnotherPlane" Text="com
  <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Edit.MoveSpecial.ToAnotherObject" Text="co
</DropdownButton>
<SplitButton X="48" Y="0" Width="3" Height="4" Command="Modeling.CreateRectangularGrid" Text="transla
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateRectangularGrid" Text="com
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.CreateRadialGrid" Text="command:
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Modeling.AddGridLine" Text="command:FullT
</SplitButton>
```

```
  <DropdownButton X="51" Y="0" Width="3" Height="4" Text="translation:albl_Points" Icon="resource:Brush.Co
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Points.AddOnLine" Text="command:ShortTex
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Points.AddOnPlane" Text="command:ShortTe
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Points.AddParallelToTwoPoints" Text="comma
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Points.AddAlongExtensionOfTwoPoints" Text=
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Points.AddProjectedPointsOnLine" Text="com
    <SimpleButton X="0" Y="10" Width="8" Height="3" Command="Points.AddUsingCenterAndArcPoints" Text="
    <SimpleButton X="0" Y="13" Width="8" Height="3" Command="Points.AddUsingThreeArcPoints" Text="comm
    <SimpleButton X="0" Y="16" Width="8" Height="2" Command="Points.AddTangentToCircle" Text="comman
    <SimpleButton X="0" Y="18" Width="8" Height="2" Command="Points.AddAtAnyPosition" Text="command:S
    <SimpleButton X="0" Y="20" Width="8" Height="2" Command="Points.AddBoltPoints" Text="command:Shor
    <SimpleButton X="0" Y="22" Width="8" Height="2" Command="Points.AddAtIntersectionOfTwoLines" Text="
    <SimpleButton X="0" Y="24" Width="8" Height="2" Command="Points.AddAtIntersectionOfPlaneAndLine" Te
    <SimpleButton X="0" Y="26" Width="8" Height="2" Command="Points.AddAtIntersectionOfPartAndLine" Text
    <SimpleButton X="0" Y="28" Width="8" Height="2" Command="Points.AddAtIntersectionOfCircleAndLine" Te
    <SimpleButton X="0" Y="30" Width="8" Height="2" Command="Points.AddAtIntersectionOfTwoPartAxes" Tex
  </DropdownButton>
  <DropdownButton X="54" Y="0" Width="8" Height="2" Text="translation:ribbon_Construction_object" Icon="r
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.AddConstructionLine" Text="comma
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.AddConstructionPlane" Text="comn
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Modeling.AddConstructionCircle" Text="comm
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Modeling.AddConstructionArc" Text="comma
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Modeling.AddConstructionPolycurve" Text="c
    <SimpleButton X="0" Y="10" Width="8" Height="2" Command="Modeling.AddConstructionObjectWithOffset"
  </DropdownButton>
  <DropdownButton X="54" Y="2" Width="8" Height="2" Text="translation:ribbon_Parametric_modeling" Icon="
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.AddFixedDistance" Text="command
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.AddReferenceDistance" Text="comm
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Modeling.Variables" Text="command:FullText
  </DropdownButton>
</Tab>
<Tab Header="translation:ribbon_View" IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="4" Height="4" Command="Views.ViewList" Text="command:ShortText" Ic
  <DropdownButton X="4" Y="0" Width="4" Height="4" Text="translation:ribbon_New_view" Icon="resource:Bru
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="View.CreateModelBasicView" Text="comman
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="View.CreateModelViewUsingTwoPoints" Text=
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="View.CreateModelViewUsingThreePoints" Text
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="View.CreateModelViewOnWorkPlane" Text="c
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="View.CreateModelViewAlongGridLines" Text=
    <Separator X="0" Y="10" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <SimpleButton X="0" Y="11" Width="8" Height="2" Command="View.CreateModelViewOnPlane" Text="comm
    <SimpleButton X="0" Y="13" Width="8" Height="2" Command="View.CreateModelViewOnPartFrontPlane" Te
    <SimpleButton X="0" Y="15" Width="8" Height="2" Command="View.CreateModelViewOnPartTopPlane" Text
    <SimpleButton X="0" Y="17" Width="8" Height="2" Command="View.CreateModelViewOnPartBackPlane" Tex
    <SimpleButton X="0" Y="19" Width="8" Height="2" Command="View.CreateModelViewOnPartBottomPlane"
    <Separator X="0" Y="21" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <SimpleButton X="0" Y="22" Width="8" Height="2" Command="View.CreatePart3DView" Text="command:Sh
    <SimpleButton X="0" Y="24" Width="8" Height="2" Command="View.CreatePartDefaultViews" Text="comma
    <SimpleButton X="0" Y="26" Width="8" Height="2" Command="View.CreatePartUndeformedView" Text="co
    <Separator X="0" Y="28" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <SimpleButton X="0" Y="29" Width="8" Height="2" Command="View.CreateComponent3DView" Text="comm
    <SimpleButton X="0" Y="31" Width="8" Height="2" Command="View.CreateComponentDefaultViews" Text=
  </DropdownButton>
```

```
<SplitButton X="8" Y="0" Width="4" Height="4" Command="View.CreateClipPlane" Text="command:ShortTex
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="View.CreateClipPlane" Text="command:Short
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Macro.CatalogMacroModelingItem?Delete All
</SplitButton>
<DropdownButton X="12" Y="0" Width="5" Height="4" Text="translation:ribbon_Work_area" Icon="resource:B
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="View.FitToEntireModel.AllViews" Text="comm
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="View.FitToEntireModel" Text="command:Shor
  <SimpleButton X="0" Y="6" Width="8" Height="3" Command="View.FitToSelectedParts.AllViews" Text="com
  <SimpleButton X="0" Y="9" Width="8" Height="3" Command="View.FitToSelectedParts" Text="command:Sh
  <SimpleButton X="0" Y="12" Width="8" Height="3" Command="View.FitUsingTwoPoints" Text="command:Sl
</DropdownButton>
<SplitButton X="17" Y="0" Width="4" Height="4" Command="Views.RedrawAll" Text="command:ShortText" Ic
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Views.RedrawAll" Text="command:FullText" I
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Views.UpdateAll" Text="command:ShortText"
</SplitButton>
<SplitButton X="21" Y="0" Width="5" Height="4" Command="View.SetWorkPlane.UsingWorkplaneTool" Text=
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="View.SetWorkPlane.UsingWorkplaneTool" Tex
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="View.SetWorkPlane.ParallelToXYZ" Text="con
  <SimpleButton X="0" Y="6" Width="8" Height="3" Command="View.SetWorkPlane.UsingOnePoint" Text="co
  <SimpleButton X="0" Y="9" Width="8" Height="3" Command="View.SetWorkPlane.UsingTwoPoints" Text="c
  <SimpleButton X="0" Y="12" Width="8" Height="3" Command="View.SetWorkPlane.UsingThreePoints" Text=
  <SimpleButton X="0" Y="15" Width="8" Height="3" Command="View.SetWorkPlane.ParallelToViewPlane" Te
</SplitButton>
<SplitButton X="26" Y="0" Width="4" Height="4" Command="Representation.Parts.ShadedWireframe" Text="
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Representation.Parts.Wireframe" Text="comm
  <SimpleButton X="0" Y="2" Width="8" Height="3" Command="Representation.Parts.ShadedWireframe" Text
  <SimpleButton X="0" Y="5" Width="8" Height="2" Command="Representation.Parts.HiddenLines" Text="con
  <SimpleButton X="0" Y="7" Width="8" Height="2" Command="Representation.Parts.Rendered" Text="comm
  <SimpleButton X="0" Y="9" Width="8" Height="2" Command="Representation.Parts.ShowOnlySelected" Text
  <Separator X="0" Y="11" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
  <SimpleButton X="0" Y="12" Width="8" Height="2" Command="Representation.Components.Wireframe" Tex
  <SimpleButton X="0" Y="14" Width="8" Height="3" Command="Representation.Components.ShadedWirefra
  <SimpleButton X="0" Y="17" Width="8" Height="2" Command="Representation.Components.HiddenLines" T
  <SimpleButton X="0" Y="19" Width="8" Height="2" Command="Representation.Components.Rendered" Text
  <SimpleButton X="0" Y="21" Width="8" Height="3" Command="Representation.Components.ShowOnlySelec
  <SimpleButton X="0" Y="24" Width="8" Height="3" Command="Representation.ShowComponentContent" Te
</SplitButton>
<SplitButton X="30" Y="0" Width="4" Height="4" Command="Visualizer.VisualizeAll" Text="translation:lbl_Vis
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Visualizer.VisualizeSelected" Text="command
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Visualizer.VisualizeAll" Text="command:FullT
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Visualizer.MaterialTypeMapping" Text="comm
</SplitButton>
<SimpleButton X="34" Y="0" Width="8" Height="2" Command="View.TogglePerspective" Text="command:Ful
<DropdownButton X="34" Y="2" Width="8" Height="1" Text="translation:ribbon_Navigate" Icon="resource:Bru
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="View.Rotate" Text="command:FullText" Icon=
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="View.Rotate.SetViewPoint" Text="command:F
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="View.Pan" Text="command:FullText" Icon="c
</DropdownButton>
<DropdownButton X="34" Y="3" Width="8" Height="1" Text="translation:albl_Zoom" Icon="resource:Brush.Co
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="View.ZoomInSpecial" Text="command:FullTe
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="View.ZoomOutSpecial" Text="command:FullT
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="View.ZoomOriginal" Text="command:FullText
  <SimpleButton X="0" Y="6" Width="8" Height="2" Command="View.ZoomPrevious" Text="command:FullTex
```

```
    <SimpleButton X="0" Y="8" Width="8" Height="3" Command="View.ZoomToSelected" Text="command:FullT
  </DropdownButton>
  <SimpleButton X="42" Y="0" Width="8" Height="1" Command="View.Fly" Text="command:FullText" Icon="co
  <SimpleButton X="42" Y="1" Width="8" Height="1" Command="View.Properties" Text="command:FullText" Ic
  <SimpleButton X="42" Y="2" Width="8" Height="1" Command="Representation.ObjectRepresentation" Text="
  <DropdownButton X="42" Y="3" Width="8" Height="1" Text="translation:albl_Snapshot" Icon="resource:Brush
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Tools.Screenshot" Text="command:FullText"
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Tools.CreatePreviewImage" Text="command:
  </DropdownButton>
</Tab>
<Tab Header="translation:ribbon_Drawings_reports" IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="4" Height="4" Command="Drawing.DrawingList" Text="command:ShortTe
  <DropdownButton X="4" Y="0" Width="5" Height="4" Text="translation:ribbon_Drawing_properties" Icon="res
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Drawing.SinglePartDrawingProperties" Text="
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Drawing.AssemblyDrawingProperties" Text="
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Drawing.GeneralArrangementDrawingPropert
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Drawing.CastUnitDrawingProperties" Text="c
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Drawing.MultiDrawingProperties" Text="comm
    <SimpleButton X="0" Y="10" Width="8" Height="2" Command="Drawing.DrawingLayout" Text="command:F
  </DropdownButton>
  <SplitButton X="9" Y="0" Width="4" Height="4" Command="Drawing.CreateDrawing" Text="command:ShortT
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Drawing.CreateDrawing" Text="command:Fu
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Drawing.CreateSinglePartDrawing" Text="cor
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Drawing.CreateAssemblyDrawing" Text="com
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Drawing.CreateCastUnitDrawing" Text="comm
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Drawing.CreateGeneralArrangementDrawing"
  </SplitButton>
  <DropdownButton X="13" Y="0" Width="8" Height="2" Text="translation:albl_Perform_numbering" Icon="reso
    <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Numbering.NumberSeriesOfSelectedObjects"
    <SimpleButton X="0" Y="3" Width="8" Height="2" Command="Numbering.NumberWelds" Text="command:F
    <SimpleButton X="0" Y="5" Width="8" Height="2" Command="Numbering.NumberModifiedObjects" Text="c
  </DropdownButton>
  <DropdownButton X="13" Y="2" Width="8" Height="2" Text="translation:j_d_j_Multi_drawing" Icon="resource:
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Drawing.CreateEmptyMultiDrawing" Text="co
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Drawing.CreateMultiDrawing.FromSelectedDr
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Drawing.CreateMultiDrawing.FromSelectedDr
    <SimpleButton X="0" Y="6" Width="8" Height="3" Command="Drawing.CreateMultiDrawing.FromSinglePartD
    <SimpleButton X="0" Y="9" Width="8" Height="3" Command="Drawing.CreateMultiDrawing.FromSinglePartD
    <SimpleButton X="0" Y="12" Width="8" Height="3" Command="Drawing.CreateMultiDrawing.FromAssembly
    <SimpleButton X="0" Y="15" Width="8" Height="4" Command="Drawing.CreateMultiDrawing.FromAssembly
  </DropdownButton>
  <DropdownButton X="21" Y="0" Width="8" Height="2" Text="translation:albl_Numbering_settings" Icon="reso
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Numbering.Settings" Text="command:FullTex
    <SimpleButton X="0" Y="2" Width="8" Height="3" Command="Numbering.SavePreliminaryNumbers" Text="
    <SimpleButton X="0" Y="5" Width="8" Height="2" Command="Numbering.AssignControlNumbers" Text="co
    <SimpleButton X="0" Y="7" Width="8" Height="3" Command="Numbering.ToggleLockControlNumbers" Text
  </DropdownButton>
  <DropdownButton X="21" Y="2" Width="8" Height="2" Text="translation:lbl_Change_Number" Icon="resource
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Numbering.ChangePartNumber" Text="comm
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Numbering.ChangeAssemblyNumber" Text="
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Numbering.ChangePartMultiNumber" Text="c
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Numbering.ChangeAssemblyMultiNumber" Te
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Numbering.ChangeFamilyNumber" Text="cor
```

```
    <Separator X="0" Y="10" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <SimpleButton X="0" Y="11" Width="8" Height="2" Command="Numbering.ClearPartAndAssemblyNumbers"
    <SimpleButton X="0" Y="13" Width="8" Height="2" Command="Numbering.ClearPartNumbers" Text="comm
    <SimpleButton X="0" Y="15" Width="8" Height="2" Command="Numbering.ClearAssemblyNumbers" Text="
    <SimpleButton X="0" Y="17" Width="8" Height="2" Command="Numbering.ClearRebarNumbers" Text="com
  </DropdownButton>
  <SimpleButton X="29" Y="0" Width="3" Height="4" Command="Reports.CreateReport" Text="command:Shor
</Tab>
<Tab Header="translation:ribbon_Manage" IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="5" Height="4" Command="Tools.Organizer" Text="command:FullText" Ico
  <SimpleButton X="5" Y="0" Width="8" Height="2" Command="Tools.PhaseManager" Text="command:ShortTe
  <SimpleButton X="5" Y="2" Width="8" Height="2" Command="Tools.ClashCheckManager" Text="command:Sh
  <SimpleButton X="13" Y="0" Width="8" Height="2" Command="Tools.ConvertIFCObjects" Text="command:Fu
  <SimpleButton X="13" Y="2" Width="8" Height="2" Command="Tools.LayoutManager" Text="command:FullTe
  <SimpleButton X="21" Y="0" Width="3" Height="4" Command="LockManager.ObjectLocks" Text="command:S
  <SimpleButton X="24" Y="0" Width="8" Height="1" Command="Tools.TaskManager" Text="command:ShortTe
  <SimpleButton X="24" Y="1" Width="8" Height="1" Command="Tools.Sequencer" Text="command:FullText" I
  <SimpleButton X="24" Y="2" Width="8" Height="1" Command="Tools.Lotting" Text="command:FullText" Icon
  <SimpleButton X="24" Y="3" Width="8" Height="1" Command="Tools.ProjectStatusVisualization" Text="comm
</Tab>
<Tab Header="translation:ribbon_Analysis_design" IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="6" Height="4" Command="Analysis.AnalysisAndModels" Text="command:
  <SimpleButton X="6" Y="0" Width="4" Height="4" Command="Loads.LoadGroups" Text="command:ShortText
  <SplitButton X="10" Y="0" Width="4" Height="4" Command="Loads.CreateLineLoad" Text="translation:albl_L
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Loads.CreateLineLoad" Text="command:Shor
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Loads.CreatePointLoad" Text="command:Sho
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Loads.CreateAreaLoad" Text="command:Sho
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Loads.CreateUniformLoad" Text="command:S
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Loads.CreateWindLoad" Text="command:Sho
    <SimpleButton X="0" Y="10" Width="8" Height="2" Command="Loads.CreateTemperatureLoad" Text="comm
  </SplitButton>
  <SimpleButton X="14" Y="0" Width="4" Height="4" Command="Analysis.CreateNode" Text="command:Short
  <SimpleButton X="18" Y="0" Width="8" Height="2" Command="Analysis.CreateRigidLink" Text="command:Sh
  <SimpleButton X="18" Y="2" Width="8" Height="2" Command="Analysis.MergeSelectedNodes" Text="comma
  <DropdownButton X="26" Y="0" Width="8" Height="2" Text="translation:ribbon_Part_analysis_properties" Icon
    <SimpleButton X="0" Y="0" Width="8" Height="1" Command="Analysis.SteelBeamProperties" Text="comma
    <SimpleButton X="0" Y="1" Width="8" Height="1" Command="Analysis.SteelOrthogonalBeamProperties" Tex
    <SimpleButton X="0" Y="2" Width="8" Height="1" Command="Analysis.SteelColumnProperties" Text="comm
    <SimpleButton X="0" Y="3" Width="8" Height="1" Command="Analysis.SteelTwinProfileProperties" Text="co
    <SimpleButton X="0" Y="4" Width="8" Height="1" Command="Analysis.SteelContourPlateProperties" Text="
    <Separator X="0" Y="5" Width="8" Height="1" Orientation="Horizontal" Thickness="1" />
    <SimpleButton X="0" Y="6" Width="8" Height="1" Command="Analysis.ConcretePadFootingProperties" Text=
    <SimpleButton X="0" Y="7" Width="8" Height="1" Command="Analysis.ConcreteStripFootingProperties" Text
    <SimpleButton X="0" Y="8" Width="8" Height="1" Command="Analysis.ConcreteColumnProperties" Text="c
    <SimpleButton X="0" Y="9" Width="8" Height="1" Command="Analysis.ConcreteBeamProperties" Text="con
    <SimpleButton X="0" Y="10" Width="8" Height="1" Command="Analysis.ConcreteSlabProperties" Text="con
    <SimpleButton X="0" Y="11" Width="8" Height="1" Command="Analysis.ConcretePanelProperties" Text="co
  </DropdownButton>
  <DropdownButton X="26" Y="2" Width="8" Height="2" Text="translation:ribbon_Load_properties" Icon="resou
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Loads.LineLoadProperties" Text="command:S
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Loads.PointLoadProperties" Text="command:
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Loads.AreaLoadProperties" Text="command:S
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Loads.UniformLoadProperties" Text="comma
```

```
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Loads.WindLoadProperties" Text="command:
    <SimpleButton X="0" Y="10" Width="8" Height="2" Command="Loads.TemperatureLoadProperties" Text="c
  </DropdownButton>
  <SimpleButton X="34" Y="0" Width="8" Height="2" Command="Analysis.ResetEditingOfSelectedParts" Text="
  <SimpleButton X="34" Y="2" Width="8" Height="2" Command="Analysis.PartProperties" Text="command:Full
  <CheckButton X="42" Y="0" Width="8" Height="1" Command="Analysis.ToggleShowNodeNumbers" Text="co
  <CheckButton X="42" Y="1" Width="8" Height="1" Command="Analysis.ToggleShowMemberNumbers" Text="
 </Tab>
 <Tab Header="translation:Commands.Export.TrimbleConnector.ShortText" IsCollapsed="false" IsUserDefined="f
  <SplitButton X="0" Y="0" Width="4" Height="4" Command="TrimbleConnect.Web" Text="command:FullText"
    <SimpleButton X="0" Y="0" Width="8" Height="2" Command="TrimbleConnect.ProjectData" Text="comman
    <SimpleButton X="0" Y="2" Width="8" Height="2" Command="TrimbleConnect.ProjectModel" Text="comma
    <SimpleButton X="0" Y="4" Width="8" Height="2" Command="TrimbleConnect.ProjectTeam" Text="commar
  </SplitButton>
  <SimpleButton X="4" Y="0" Width="4" Height="4" Command="TrimbleConnect.ProjectPublish" Text="commar
  <SimpleButton X="8" Y="0" Width="4" Height="4" Command="TrimbleConnect.ConnectorModels" Text="com
  <SimpleButton X="12" Y="0" Width="4" Height="4" Command="TrimbleConnect.ConnectorTodos" Text="com
  <SimpleButton X="16" Y="0" Width="4" Height="4" Command="TrimbleConnect.Desktop" Text="command:Sh
  <SimpleButton X="20" Y="0" Width="2" Height="2" Command="TrimbleConnect.AdjustTsView" Text="comma
  <SimpleButton X="20" Y="2" Width="2" Height="2" Command="TrimbleConnect.SelectTsObjects" Text="com
  <SimpleButton X="22" Y="0" Width="2" Height="2" Command="TrimbleConnect.AdjustTcdView" Text="comm
  <SimpleButton X="22" Y="2" Width="2" Height="2" Command="TrimbleConnect.SelectTcdObjects" Text="con
 </Tab>
 <Tab Header="晨缘迅宫      " IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="3" Height="4" Command="Plugin.CatalogPluginComponentItem?SC
门刚边柱梁竖连接节点                " Text="门钢边柱      " Icon="D:\北宫集团      \BIM系列插件      \TSEP工具   \钢结构工具组          \图标
\favicon.ico" ShowText="true" ShowIcon="true" />
  <CheckButton X="3" Y="0" Width="3" Height="4" Command="Plugin.CatalogPluginComponentItem?SC
船舶梯子节点          32节课程     " Text="船舶梯子      " Icon="D:\北宫集团      \BIM系列插件      \TSEP工具   \钢结构工具组          \图标
\favicon (3).ico" ShowText="true" ShowIcon="true" />
  <CheckButton X="6" Y="0" Width="3" Height="4" Command="Plugin.CatalogPluginComponentItem?SC
螺丝套筒节点          29节 " Text="螺丝套筒      " Icon="D:\北宫集团      \BIM系列插件      \TSEP工具   \钢结构工具组          \图标
\favicon (2).ico" ShowText="true" ShowIcon="true" />
  <CheckButton X="9" Y="0" Width="3" Height="4" Command="Plugin.CatalogPluginComponentItem?SC
旋转楼梯      38节课程     " Text="旋转楼梯      " Icon="D:\北宫集团      \BIM系列插件      \TSEP工具   \钢结构工具组          \图标
\favicon (1).ico" ShowText="true" ShowIcon="true" />
 </Tab>
</Ribbon>
```

图纸管理器

```
string str = "基础属性      \r\n\r\n";
DrawingHandler drawingHandler = new DrawingHandler();//图纸列表管理器
var  drawings= drawingHandler.GetDrawings();//获取图纸列表
while (drawings.MoveNext())
{
   str += "图纸名称:        " + drawings.Current.Name + "\r\n\r\n";
   str += "图纸类型:        " + drawings.Current.GetType() + "\r\n\r\n";
   str += "图纸编号:        " + drawings.Current.Mark + "\r\n\r\n";
   str += "冻结状态:        " + ((drawings.Current as Drawing).IsFrozen == true ? "冻结   " : "没冻结     ") + "\r\n\r\n";
```

```
    str += "□□□□□        " + ((drawings.Current as Drawing).IsLocked == true ? "□□   " : "□□□    ") + "\r\n\r\n";
    str += "□□    1□ " + drawings.Current.Title1 + "\r\n\r\n";
  }
  return str;
```

Tekla□□□□□

Tekla.Structures.Dialog.ApplicationFormBase

□□□□□□□

```
   public partial class Form1 : ApplicationFormBase
   {
           public Form1()
      {
         InitializeComponent();
         InitializeForm();
         if (GetConnectionStatus())
         {
            string messageFolder = null;
            TeklaStructuresSettings.GetAdvancedOption("XS_MESSAGES", ref messageFolder);
            messageFolder = Path.Combine(messageFolder, @"DotAppsStrings");
            Dialogs.SetSettings(string.Empty);
            Localization.Language = (string)Settings.GetValue("language");
            Localization.LoadFile(Path.Combine(messageFolder, Application.ProductName + ".xml"));
            Localization.Localize(this);
         }
         else
         {
            MessageBox.Show("Tekla Structures is NOT running...");
         }
      }
//□ Tekla□□□□□
 class Program : WindowsFormsApplicationBase
 {
    public Program()
    {
       IsSingleInstance = true;
       EnableVisualStyles = true;
    }
    protected override void OnCreateMainForm()
    {
       MainForm = new Form1();
       MainForm.Show(TeklaStructures.MainWindow);

    }
    protected override void OnStartupNextInstance(StartupNextInstanceEventArgs eventArgs)
    {
       eventArgs.BringToForeground = true;
       base.OnStartupNextInstance(eventArgs);
    }
```

```
    /// <summary>
    /// 应用程序的主入口点。
    /// </summary>
    [STAThread]
    static void Main(string[] args)
    {
        if (TeklaStructures.Connect())
        {
            TeklaStructures.Closed += delegate
            {
                Application.Exit();
            };
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            try
            {
                Program program = new Program();
                program.Run(args);
            }
            finally
            {
                TeklaStructures.Disconnect();
            }
        }
    }
}
```

属性大全

梁柱

```
Part beam = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART) as Part;//选择一根梁
string property = "基础属性       \r\n\r\n";
//基础属性
property += "类型    : " + beam.GetType().Name + "\r\n\r\n";//名称
property += "名称    : " + beam.Name + "\r\n\r\n";//名称
property += "截面    : " + beam.Profile.ProfileString + "\r\n\r\n";//截面
property += "材料    : " + beam.Material.MaterialString + "\r\n\r\n";//材料
property += "完成    : " + beam.Finish + "\r\n\r\n";//完成
property += "等级    : " + beam.Class + "\r\n\r\n";//等级
property += "零件前缀       : " + beam.PartNumber.Prefix + "\r\n\r\n";//零件前缀
property += "零件编号       : " + beam.PartNumber.StartNumber + "\r\n\r\n";//零件编号
property += "构件前缀       : " + beam.AssemblyNumber.Prefix + "\r\n\r\n";//构件前缀
property += "构件编号       : " + beam.AssemblyNumber.StartNumber + "\r\n\r\n";//构件编号

property += "构件垂直方向位置              : " + beam.Position.Depth + "\r\n\r\n";//构件垂直方向位置
property += "位置偏移       : " + beam.Position.DepthOffset + "\r\n\r\n";//位置偏移

if (beam.GetType() == typeof(Beam))
{
```

```
    property += "旋转位置       : " + beam.Position.Rotation + "\r\n\r\n";//旋转位置
    property += "位置偏移       : " + beam.Position.RotationOffset + "\r\n\r\n";//位置偏移
    property += "水平方向位置          : " + beam.Position.Plane + "\r\n\r\n";//水平方向位置
    property += "位置偏移       : " + beam.Position.PlaneOffset + "\r\n\r\n";//位置偏移

    property += "起点    X偏移   : " + ((Beam)beam).StartPointOffset.Dx + "\r\n\r\n";//
    property += "起点    y偏移   : " + ((Beam)beam).StartPointOffset.Dy + "\r\n\r\n";//
    property += "起点    z偏移   : " + ((Beam)beam).StartPointOffset.Dz + "\r\n\r\n";//
    property += "终点    X偏移   : " + ((Beam)beam).EndPointOffset.Dx + "\r\n\r\n";//
    property += "终点    y偏移   : " + ((Beam)beam).EndPointOffset.Dy + "\r\n\r\n";//
    property += "终点    z偏移   : " + ((Beam)beam).EndPointOffset.Dz + "\r\n\r\n";//
    property += "扭曲起点       : " + beam.DeformingData.Angle + "\r\n\r\n";//扭曲起点
    property += "扭曲终点       : " + beam.DeformingData.Angle2 + "\r\n\r\n";//扭曲终点
    property += "起拱    : " + beam.DeformingData.Cambering + "\r\n\r\n";//起拱
    property += "减短    : " + beam.DeformingData.Shortening + "\r\n\r\n";//减短
}

textBox1.Text = property;
//进阶属性
property = "";
string pos = "";
int pos2 = 0;
double pos1 = 0.0;
Phase phase;
property += "进阶属性       " + "\r\n\r\n";
property += "零件编号:          " + beam.GetReportProperty("PART_POS", ref pos) + ":   " + pos + "\r\n\r\n";
property += "状态:      " + beam.GetPhase(out phase) + ":   " + phase.PhaseName + "\r\n\r\n";
property += "重量:      " + beam.GetReportProperty("WEIGHT", ref pos1) + ":   " + pos1 + "\r\n\r\n";
property += "体积:      " + beam.GetReportProperty("VOLUME", ref pos1) + ":   " + pos1 + "\r\n\r\n";
property += "长度:      " + beam.GetReportProperty("LENGTH", ref pos1) + ":   " + pos1 + "\r\n\r\n";
property += "数量:      " + beam.GetAssembly().GetReportProperty("NUMBER", ref pos2) + ":   " + pos2 + "\r\n\r\n";
property += "顶面标高:          " + beam.GetReportProperty("TOP_LEVEL", ref pos) + ":   " + pos + "\r\n\r\n";
property += "底面标高:          " + beam.GetReportProperty("BOTTOM_LEVEL", ref pos) + ":   " + pos + "\r\n\r\n";
property += "平面位置:          " + beam.GetAssembly().GetReportProperty("POSITION_CODE", ref pos) + ":
" + pos + "\r\n\r\n";
textBox2.Text = property;
```

焊缝

```
Weld weld = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_WELD) as Weld;//选择一根梁
string basicproperty = "基础属性       \r\n\r\n";
basicproperty += "类型:      " + weld.GetType().Name + "\r\n\r\n";
basicproperty += "边缘    -四周:       " + (weld.AroundWeld == true ? "四周    " : "边缘    ") + "\r\n\r\n";
basicproperty += "工厂    -车间:       " + (weld.ShopWeld == true ? "工地    " : "车间    ") + "\r\n\r\n";
basicproperty += "位置:      " + weld.Position + "\r\n\r\n";
basicproperty += "形状:      " + weld.IntermittentType + "\r\n\r\n";

basicproperty += "连接为:        " + (weld.ConnectAssemblies == true ? "次零件      " : "次构件      ") + "\r\n\r\n";//
basicproperty += "布置:      " + weld.Placement + "\r\n\r\n";
//basicproperty += "预加工:        " + weld.Preparation + "\r\n\r\n";//预加工为主零件

basicproperty += "上部的线前缀:             " + weld.PrefixAboveLine + "\r\n\r\n";//上部的线前缀
```

```
basicproperty += "上部焊缝的类型:            " + weld.TypeAbove + "\r\n\r\n";//上部焊缝的类型
basicproperty += "上部的尺寸:          " + weld.SizeAbove + "\r\n\r\n";//上部的尺寸
basicproperty += "上部的角度:          " + weld.AngleAbove + "\r\n\r\n";//上部的角度
basicproperty += "上部轮廓类型:            " + weld.ContourAbove + "\r\n\r\n";//上部轮廓类型
basicproperty += "上部完成的类型:            " + weld.FinishAbove + "\r\n\r\n";//完成的类型
basicproperty += "上部焊角面:          " + weld.RootFaceAbove + "\r\n\r\n";
basicproperty += "上部有效喉高:            " + weld.EffectiveThroatAbove + "\r\n\r\n";
basicproperty += "上部根部开孔:            " + weld.RootOpeningAbove + "\r\n\r\n";
basicproperty += "上部增量编号:            " + weld.IncrementAmountAbove + "\r\n\r\n";
basicproperty += "上部长度:          " + weld.LengthAbove + "\r\n\r\n";
basicproperty += "上部斜度:          " + weld.PitchAbove + "\r\n\r\n";

basicproperty += "下部的线前缀:            " + weld.PrefixBelowLine + "\r\n\r\n";//下部的线前缀
basicproperty += "下部焊缝的类型:            " + weld.TypeBelow + "\r\n\r\n";//下部焊缝的类型
basicproperty += "下部的尺寸:          " + weld.SizeBelow + "\r\n\r\n";//下部的尺寸
basicproperty += "下部的角度:          " + weld.AngleBelow + "\r\n\r\n";//下部的角度
basicproperty += "下部轮廓类型:            " + weld.ContourBelow + "\r\n\r\n";//下部轮廓类型
basicproperty += "下部完成的类型:            " + weld.FinishBelow + "\r\n\r\n";//完成的类型
basicproperty += "下部焊角面:          " + weld.RootFaceBelow + "\r\n\r\n";
basicproperty += "下部有效喉高:            " + weld.EffectiveThroatBelow + "\r\n\r\n";
basicproperty += "下部根部开孔:            " + weld.RootOpeningBelow + "\r\n\r\n";
basicproperty += "下部增量编号:            " + weld.IncrementAmountBelow + "\r\n\r\n";
basicproperty += "下部长度:          " + weld.LengthBelow + "\r\n\r\n";
basicproperty += "下部斜度:          " + weld.PitchBelow + "\r\n\r\n";

basicproperty += "NDT检验类型:          " + weld.NDTInspection + "\r\n\r\n";//NDT检验类型
basicproperty += "焊条的分类:          " + weld.ElectrodeClassification + "\r\n\r\n";//焊条的分类
basicproperty += "焊条的强度:          " + weld.ElectrodeStrength + "\r\n\r\n";//焊条的强度
basicproperty += "焊条的系数:          " + weld.ElectrodeCoefficient + "\r\n\r\n";//焊条的系数
basicproperty += "过程类型:          " + weld.ProcessType + "\r\n\r\n";//过程类型
basicproperty += "评注:      " + weld.ReferenceText + "\r\n\r\n";
textBox3.Text = basicproperty;
//进阶类型
double ww1 = 0.0;
string ww = "";
basicproperty = "进阶属性       \r\n\r\n";

basicproperty += "焊缝编号:          " + weld.GetReportProperty("WELD_NUMBER", ref ww) + ":   " + ww + "\r\n\r\n";
basicproperty += "父编号:       " + weld.GetReportProperty("WELD_FATHER_NUMBER", ref ww) + ":
" + ww + "\r\n\r\n";
basicproperty += "长度:      " + weld.GetReportProperty("LENGTH", ref ww1) + ":   " + ww1 + "\r\n\r\n";
textBox4.Text = basicproperty;
```

构件

```
double num = 0.0;
string number = "";
Assembly assembly = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_OBJECT) as Assembly;//
选择一根梁
string basicproperty = "基础属性       \r\n\r\n";
basicproperty += "类型    " + assembly.GetAssemblyType() + "\r\n\r\n";
basicproperty += "重量    " + assembly.GetReportProperty("ASSEMBLY_WEIGHT", ref num) + ":
```

```
" + num + "\r\n\r\n";
 basicproperty += "体积   " + assembly.GetReportProperty("ASSEMBLY_VOLUME", ref num) + ":
" + num + "\r\n\r\n";
 basicproperty += "长度   " + assembly.GetReportProperty("LENGTH", ref num) + ":  " + num + "\r\n\r\n";
 basicproperty += "数量   " + assembly.GetReportProperty("NUMBER", ref num) + ":  " + num + "\r\n\r\n";
 basicproperty += "顶标高     " + assembly.GetReportProperty("ASSEMBLY_TOP_LEVEL", ref number) + ":
" + number + "\r\n\r\n";
 basicproperty += "底标高     " + assembly.GetReportProperty("ASSEMBLY_BOTTOM_LEVEL", ref number) + ":
" + number + "\r\n\r\n";
 basicproperty += "构件位置       " + assembly.GetReportProperty("ASSEMBLY_POSITION_CODE", ref number) + ":
" + number + "\r\n\r\n";
 textBox6.Text = basicproperty;
 basicproperty = "进阶属性       \r\n\r\n";
 basicproperty += "构件编号       " + assembly.GetReportProperty("ASSEMBLY_POS", ref number) + ":
" + number + "\r\n\r\n";
 basicproperty += "主零件     " + assembly.GetMainPart().GetReportProperty("PART_POS", ref number) + ":
" + number + "\r\n\r\n";
 var secondes = assembly.GetSecondaries();
 int i = 1;
 foreach (Part part in secondes)
 {
    if (part.GetType() == typeof(Beam))
    {
       basicproperty += string.Format("次零件     {0}梁编号
", i) + part.GetReportProperty("PART_POS", ref number) + ":  " + number + "\r\n\r\n";
       basicproperty += string.Format("次零件     {0}梁截面     ", i) + part.Profile.ProfileString + "\r\n\r\n";
       if (part.GetWelds().GetSize() > 0)
       {
          var welds = part.GetWelds();
          while (welds.MoveNext())
          {
             basicproperty += string.Format("次零件     {0}焊缝编号
", i) + (welds.Current as Weld).GetReportProperty("WELD_NUMBER", ref number) + ":  " + number + "\r\n\r\n";
          }

       }
    }

    if (part.GetType() == typeof(ContourPlate))
    {
       basicproperty += string.Format("次零件     {0}板编号
", i) + part.GetReportProperty("PART_POS", ref number) + ":  " + number + "\r\n\r\n";
       basicproperty += string.Format("次零件     {0}板截面     ", i) + part.Profile.ProfileString + "\r\n\r\n";
       if (part.GetWelds().GetSize() > 0)
       {
          var welds = part.GetWelds();
          while (welds.MoveNext())
          {
             basicproperty += string.Format("次零件     {0}焊缝编号
", i) + (welds.Current as Weld).GetReportProperty("WELD_NUMBER", ref number) + ":  " + number + "\r\n\r\n";
          }
       }
```

```
    }
    i++;
}

textBox5.Text = basicproperty;
```

螺栓

```
BoltArray boltArray = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_BOLTGROUP) as BoltArray;//选择螺栓

string basicproperty = "基础属性       \r\n\r\n";
basicproperty += "螺栓的主零件          : " + boltArray.PartToBeBolted.GetPartMark() + "\r\n\r\n"
    + "螺栓的次零件          : " + boltArray.PartToBoltTo.GetPartMark() + "\r\n\r\n"
    + "螺栓的尺寸        : " + boltArray.BoltSize + "\r\n\r\n"
    + "螺栓孔尺寸        : " + (boltArray.BoltSize + boltArray.Tolerance) + "\r\n\r\n"
    + "螺栓类型       : " + boltArray.BoltType + "\r\n\r\n"
    + "螺栓标准        : " + boltArray.BoltStandard + "\r\n\r\n"
     + "螺栓标准        : " + boltArray.BoltStandard + "\r\n\r\n"
     + "螺栓杆件长度          : " + boltArray.Length + "\r\n\r\n"
     + "螺栓附加长度          : " + boltArray.ExtraLength + "\r\n\r\n"
     + "螺栓切割长度          : " + boltArray.CutLength + "\r\n\r\n"
     + "剪切面是否有螺纹              : " + boltArray.ThreadInMaterial + "\r\n\r\n"
      + "在平面上         : " + boltArray.Position.Plane + "\r\n\r\n"
       + "在旋转上         : " + boltArray.Position.Rotation + "-" + boltArray.Position.RotationOffset + "\r\n\r\n"
        + "螺栓主杆件          : " + (boltArray.Bolt == true ? "有 " : "无 ") + "\r\n\r\n"
         + "螺帽侧垫圈        1有无  :" + (boltArray.Washer1 == true ? "有 " : "无 ") + "\r\n\r\n"
         + "螺母侧垫圈        1有无  :" + (boltArray.Washer2 == true ? "有 " : "无 ") + "\r\n\r\n"
          + "螺母侧垫圈        2有无  :" + (boltArray.Washer3 == true ? "有 " : "无 ") + "\r\n\r\n"
           + "螺母   1有无  :" + (boltArray.Nut1 == true ? "有 " : "无 ") + "\r\n\r\n"
           + "螺母   2有无  :" + (boltArray.Nut2 == true ? "有 " : "无 ") + "\r\n\r\n"
           + "扩大孔    5个 :" + (boltArray.Hole1 == true ? "有 " : "无 ") + "\r\n\r\n"
           + "有无扩大孔        :" + (boltArray.HoleType == BoltGroup.BoltHoleTypeEnum.HOLE_TYPE_OVERSIZED ? "有 " : "无 ") + "\r\n\r\n"
           + "有扩大孔则孔直径为              :" + (boltArray.BoltSize + boltArray.Tolerance + boltArray.SlottedHoleX) + "\r\n\r\n"
            + "旋转槽    :" + boltArray.RotateSlots + "\r\n\r\n"
            + "x方向上的间距          :" + boltArray.GetBoltDistX(0).ToString() + "\r\n\r\n"
             + "y方向上的间距          :" + boltArray.GetBoltDistY(0).ToString() + "\r\n\r\n"
              + "起点   Dx偏移   :" + boltArray.StartPointOffset.Dx + "\r\n\r\n"
              + "终点   Dx偏移   :" + boltArray.EndPointOffset.Dx + "\r\n\r\n";
textBox8.Text = basicproperty;
int t = 0;
string str = "";
basicproperty = "进阶属性       \r\n\r\n";

basicproperty += "本身构件编号:
" + boltArray.GetReportProperty("SECONDARY_1.ASSEMBLY_POS", ref str) + "-" + str + "\r\n\r\n";
basicproperty += "连接构件:
" + boltArray.GetReportProperty("CONNECTED_ASSEMBLIES", ref str) + "-" + str + "\r\n\r\n";
basicproperty += "连接零件:
" + boltArray.GetReportProperty("CONNECTED_PARTS", ref str) + "-" + str + "\r\n\r\n";
basicproperty += "螺栓数量:          " + boltArray.GetReportProperty("NUMBER", ref t) + "-" + t + "\r\n\r\n";
```

```
textBox7.Text = basicproperty;
```

插件或组件

```
string t1 = "", t2 = "", t3 = "", t4 = "", t5 = "";
CatalogHandler catalogHandler = new CatalogHandler();//组件管理器

if (catalogHandler.GetConnectionStatus())//有没有链接
{
    ComponentItemEnumerator componentItemEnumerator = catalogHandler.GetComponentItems();//
获取管理器下的组件项
    //t1 = componentItemEnumerator.GetSize().ToString() + "\r\n\r\n";//获取插件的个数
    while (componentItemEnumerator.MoveNext())//遍历当前组件管理器下的每个插件项
    {
        if (componentItemEnumerator.Current.Name == "DkBoltCommon")
        {
            t1 += "插件编号      :" + (componentItemEnumerator.Current as ComponentItem).Name + "\r\n\r\n"
                + "插件名称      :" + componentItemEnumerator.Current.Name + "\r\n\r\n"
                + "插件   UI:" + componentItemEnumerator.Current.UIName + "\r\n\r\n"
                 + "插件类型      :" + componentItemEnumerator.Current.Type;
        }
        componentItemEnumerator.Current.Select(componentItemEnumerator.Current.Name, componentItemEnum
        t2 += "插件编号      :" + componentItemEnumerator.Current.Number + "\r\n\r\n";//插件编号
        t3 += "UI名称   :" + componentItemEnumerator.Current.UIName + "\r\n\r\n";//插件   UI界面名称
        switch (componentItemEnumerator.Current.Type.ToString()) //简单的
switch(){case :beak;....default :break;}//获取到的组件的类型
        {
            case "UNKNOWN":
                t4 += "未知类型      " + "\r\n\r\n";
                break;
            case "CONNECTION":
                t4 += "连接类型      " + "\r\n\r\n";
                break;
            case "COMPONENT":
                t4 += "自定义组件类型          " + "\r\n\r\n";
                break;
            case "SEAM":
                t4 += "接缝类型      " + "\r\n\r\n";
                break;
            case "DETAIL":
                t4 += "详细类型      " + "\r\n\r\n";
                break;
            case "CUSTOM_PART":
                t4 += "自定义零件对象          " + "\r\n\r\n";
                break;
            case "DRAWING_PLUGIN":
                t4 += "绘图插件      " + "\r\n\r\n";
                break;
            default:
                t4 += "那我就不知道了          " + "\r\n\r\n";
                break;
        }
```

```
    }
  }
  textBox9.Text = t1;
  textBox10.Text = t2;
  textBox15.Text = t3;
  textBox16.Text = t4;
```

浇筑体

```
  string str = "";
  int t = 0;
  Part castA = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART) as Part;//选择零件对象
  Assembly cast = castA.GetAssembly();//获取土建浇筑体，获取钢结构构件
  string castproperty = "零件属性       \r\n\r\n";
  castproperty += "对象类型:          " + castA.GetAssembly().GetType().Name + "\r\n\r\n";
  castproperty += "零件    ID:   " + castA.Identifier.ToString() + "\r\n\r\n";//重点
  castproperty += "零件编号:          " + castA.GetReportProperty("PART_POS", ref str) + ":   " + str + "\r\n\r\n";
  castproperty += "零件数量:          " + castA.GetReportProperty("NUMBER", ref t) + ":   " + t + "\r\n\r\n";
  castproperty += "零件状态:          " + castA.GetReportProperty("PHASE.NAME", ref str) + ":   " + str + "\r\n\r\n";
  castproperty += "零件名称:          " + castA.Name + "\r\n\r\n";
  castproperty += "零件平面位置:            " + castA.GetAssembly().GetReportProperty("POSITION_CODE", ref str) + ":
" + str + "\r\n\r\n";
  castproperty += "零件顶部标高:            " + castA.GetReportProperty("TOP_LEVEL", ref str) + ":   " + str + "\r\n\r\n";
  castproperty += "零件底部标高:            " + castA.GetReportProperty("BOTTOM_LEVEL", ref str) + ":   " + str + "\r\n\r\n";
  textBox12.Text = castproperty;
  castproperty = "浇筑体属性          \r\n\r\n";
  castproperty += "对象类型:          " + cast.GetAssemblyType().ToString() + "\r\n\r\n";
  castproperty += "浇筑体      ID:   " + cast.Identifier.ToString() + "\r\n\r\n";//重点
  castproperty += "浇筑体编号:            " + cast.GetReportProperty("CAST_UNIT_POS", ref str) + ":   " + str + "\r\n\r\n";
  castproperty += "浇筑体数量:            " + cast.GetReportProperty("NUMBER", ref t) + ":   " + t + "\r\n\r\n";
  castproperty += "浇筑体状态:            " + cast.GetReportProperty("PHASE.NAME", ref str) + ":   " + str + "\r\n\r\n";
  castproperty += "浇筑体名称:            " + cast.GetReportProperty("MAINPART.NAME", ref str) + ":   " + str + "\r\n\r\n";
  castproperty += "浇筑体平面位置:              " + cast.GetReportProperty("CAST_UNIT_POSITION_CODE", ref str) + ":
" + str + "\r\n\r\n";
  castproperty += "浇筑体顶部标高:              " + cast.GetReportProperty("TOP_LEVEL", ref str) + ":   " + str + "\r\n\r\n";
  castproperty += "浇筑体底部标高:              " + cast.GetReportProperty("BOTTOM_LEVEL", ref str) + ":   " + str + "\r\n\r\n";
  textBox11.Text = castproperty;
```

钢筋

```
  ArrayList arrayList = new ArrayList();//集合
  Part castA = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART) as Part;//选择零件对象
  int t = 0;
  double st = 0.0;
  string str = "";
  string castproperty = "基本属性       \r\n\r\n";
  string castproperty2 = "特殊属性       \r\n\r\n";
  var rebars = castA.GetChildren();//获取浇筑体          \土建的对象上的钢筋时候，用到的方法。
  //while (rebars.MoveNext())
  //{
```

```
//    if (rebars.Current.GetType() == typeof(RebarGroup)
//        || rebars.Current.GetType() == typeof(SingleRebar)
//        || rebars.Current.GetType() == typeof(RebarSet))//
只有当浇筑体的子集里面是钢筋、单根钢筋、钢筋设置才进行获取属性。
//    {
//        arrayList.Add(rebars.Current);
//        castproperty += "对象类型:         " + rebars.Current.GetType() + "\r\n\r\n";
//        castproperty += "钢筋编号:         " + rebars.Current.GetReportProperty("REBAR_POS", ref str) + ":
" + str + "\r\n\r\n";
//        castproperty += "钢筋数量:         " + rebars.Current.GetReportProperty("NUMBER", ref t) + ":  " + t + "\r\n\r\n";
//        castproperty += "钢筋等级:         " + rebars.Current.GetReportProperty("GRADE", ref str) + ":
" + str + "\r\n\r\n";
//        castproperty += "钢筋尺寸:         " + rebars.Current.GetReportProperty("SIZE", ref str) + ":  " + str + "\r\n\r\n";
//        castproperty += "钢筋长度:         " + rebars.Current.GetReportProperty("LENGTH", ref st) + ":
" + st + "\r\n\r\n";
//        castproperty += "钢筋单重:         " + rebars.Current.GetReportProperty("WEIGHT", ref st) + ":
" + st + "\r\n\r\n";
//        castproperty += "钢筋总重:         " + st * t + "\r\n\r\n";

//        //重点，钢筋形状的大样图（示意图）
//        castproperty2 += "钢筋形状:          " + rebars.Current.GetReportProperty("SHAPE", ref str) + ":
" + str + "\r\n\r\n";
//    }

//}
textBox14.Text = castproperty;
textBox13.Text = castproperty2;
```

图纸管理器属性

```
string str = "基础属性       \r\n\r\n";
DrawingHandler drawingHandler = new DrawingHandler();//图纸列表管理器
var  drawings= drawingHandler.GetDrawings();//获取图纸列表
while (drawings.MoveNext())
{
    str += "图纸名称:         " + drawings.Current.Name + "\r\n\r\n";
    str += "图纸类型:         " + drawings.Current.GetType() + "\r\n\r\n";
    str += "图纸编号:         " + drawings.Current.Mark + "\r\n\r\n";
    str += "冻结状态:         " + ((drawings.Current as Drawing).IsFrozen == true ? "冻结    " : "没冻结     ") + "\r\n\r\n";
    str += "锁定状态:         " + ((drawings.Current as Drawing).IsLocked == true ? "锁定    " : "没锁定     ") + "\r\n\r\n";
    str += "标题    1:  " + drawings.Current.Title1 + "\r\n\r\n";
}
return str;
```

零件翼缘板等属性

```
double thickness = 0.0, length = 0.0, weight = 0.0, width = 0.0;
Part part = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART) as Part;//选择零件
string str = "基础属性       \r\n\r\n";//定义字符串
str += "截面    :" + part.Profile.ProfileString + "\r\n\r\n";
str += "材料    :" + part.Material.MaterialString + "\r\n\r\n";
```

```
str += "长度   :" + part.GetReportProperty("LENGTH", ref length) + "-" + length + "\r\n\r\n";
str += "重量   :" + part.GetReportProperty("WEIGHT", ref weight) + "-" + weight + "\r\n\r\n";
textBox20.Text = str;
double weights = 0.0;
part.GetReportProperty("FLANGE_THICKNESS_U", ref thickness);
part.GetReportProperty("FLANGE_WIDTH_U", ref width);
part.GetReportProperty("FLANGE_LENGTH_U", ref length);
str = "进阶属性       \r\n\r\n";//定义字符串
str += "上翼缘板截面:             " + "PL" + thickness + "*" + width + "\r\n\r\n";
str += "上翼缘板长度:             " + length + "\r\n\r\n";
//weights += (7.85 * thickness * width * length / 1000000);
//str += "上翼缘板重量:             " +weights+ "\r\n\r\n";
//腹板
part.GetReportProperty("PROFILE.WEB_THICKNESS", ref thickness);
part.GetReportProperty("WEB_WIDTH", ref width);
part.GetReportProperty("WEB_LENGTH", ref length);
str += "腹板截面:         " + "PL" + thickness + "*" + width + "\r\n\r\n";
str += "腹板长度:         " + length + "\r\n\r\n";
//weights += (7.85 * thickness * width * length / 1000000);
//str += "腹板重量:         " + (7.85 * thickness * width * length / 1000000) + "\r\n\r\n";
//下翼缘板
part.GetReportProperty("FLANGE_THICKNESS_B", ref thickness);
part.GetReportProperty("FLANGE_WIDTH_B", ref width);
part.GetReportProperty("FLANGE_LENGTH_B", ref length);
str += "下翼缘板截面:             " + "PL" + thickness + "*" + width + "\r\n\r\n";
str += "下翼缘板长度:             " + length + "\r\n\r\n";
//weights += (7.85 * thickness * width * length / 1000000);
//str += "下翼缘板重量:             " + (7.85 * thickness * width * length / 1000000) + "\r\n\r\n";
//str += "总重   "+weights;
textBox19.Text = str;
```

打开插件列表选择并返回数据

```
Tekla.Structures.Dialog.UIControls.ComponentCatalog componentCatalog1;
private void Component_Click(object sender, EventArgs e)
{
   componentCatalog1.SelectedName = textBox9.Text;
   componentCatalog1.SelectedNumber =
      string.IsNullOrEmpty(textBox10.Text) ? Constants.XS_DEFAULT : int.Parse(textBox10.Text);

}

private void Component_selectionDone(object sender, EventArgs e)
{
   SetAttributeValue(textBox9, componentCatalog1.SelectedName);
   SetAttributeValue(textBox10, componentCatalog1.SelectedNumber);
}
```

```
//□□□
namespace Tekla.Technology.Akit.UserScript
{
    public class Script
    {
        public static void Run(Tekla.Technology.Akit.IScript akit)
        {
            // akit.Callback("acmd_partnumbers_all", "", "main_frame");
            akit.Callback("acmd_partnumbers_selected", "", "main_frame");
        }
    }
}
```