



```
Component component = new Component();
component.Name = teklaPLuginList.Name;
component.Number = teklaPLuginList.Number;
ComponentInput componentInput = new ComponentInput();
switch (teklaPLuginList.Name)
{
    case "DKQGGQCoQ□□□□ Q□□□□□":
        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□□□□ Q□□□□":
        componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
        componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
        break;
    case "DKQGGQDoQ□□□□ Q□□□□":
        componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
        componentInput.AddInputObject(new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART));
        break;
    case "DKQGGQDoQ□□□□ Q□□□□":
        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();
```

□□ Tekla□□□□□

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

```

TeklaBMP = "";
Teklaexe = "";
string xsbin = "";
TeklaStructuresSettings.GetAdvancedOption("XSBIN", ref xsbin);
DirectoryInfo path = new DirectoryInfo(xsbin);
string tt = path.Parent.Parent.FullName; // 2
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;
}

```



```

CatalogHandler catalogHandler = new CatalogHandler();
//
//ComponentItemEnumerator componentItemEnumerator1 = catalogHandler.GetComponentItems();//
//
if (catalogHandler.GetConnectionStatus())//
{
    ComponentItemEnumerator componentItemEnumerator = catalogHandler.GetComponentItems();

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

    while (componentItemEnumerator.MoveNext())//
    {
        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]) // switch(){case :beak;....default :break;}//
                    {
                        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 "PI":
            addpluin(pluginName, pluginNumber, teklaPLuginListPI);
            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);

```



```

ComponentOptions componentOptions = new ComponentOptions();//[]
TeklaStructuresSettings.GetOptions(ref componentOptions);//[]
MessageBox.Show(
    "[] " + componentOptions.PlateProfileName + "\r\n" +
    "[][] " + componentOptions.FoldedPlateProfileName + "\r\n" +
    "[][][] " + componentOptions.BoltEdgeDistanceFactor + "\r\n" +
    "[][] " + componentOptions.BoltEdgeDistanceReference + "\r\n" +
    "[][] " + componentOptions.BoltStandard + "\r\n" +
    "[][] " + componentOptions.BoltSize + "\r\n" +
    "[][] " + componentOptions.PartMaterial + "\r\n" +
    "[][][][] " + componentOptions.PartWeldedToPrimaryPositionPrefix + "\r\n" +
    "[][][][] " + componentOptions.PartWeldedToPrimaryStartNumber + "\r\n" +
    "[][][][] " + componentOptions.PartWeldedToSecondaryPositionPrefix + "\r\n" +
    "[][][][] " + componentOptions.PartWeldedToSecondaryStartNumber + "\r\n" +
    "[][][] " + componentOptions.LoosePartPositionPrefix + "\r\n" +
    "[][][] " + componentOptions.LoosePartStartNumber + "\r\n" +
    "[][][][] " + componentOptions.AssemblyLoosePartPositionPrefix + "\r\n" +
    "[][][] " + componentOptions.AssemblyLoosePartStartNumber + "\r\n"
    , "Tekla[][]");
componentOptions.PlateProfileName = "[] ";
componentOptions.FoldedPlateProfileName = "[] ";
componentOptions.BoltEdgeDistanceReference = ComponentOptions.BoltEdgeDistanceReferenceEnum.HOLE_DI/
TeklaStructuresSettings.SetOptions(componentOptions);//[]

```



```

MessageBox.Show(
    "[][][] =" + ModuleManager.AnalysisAndDesign.ToString() + "\r\n" +
    "[] =" + ModuleManager.ConcreteDetailing.ToString() + "\r\n" +
    "[] =" + ModuleManager.LoadModeling.ToString() + "\r\n" +
    "[][] =" + ModuleManager.MultimaterialModeling.ToString() + "\r\n" +
    "[] =" + ModuleManager.RebarModeling.ToString() + "\r\n" +
    "[][] =" + ModuleManager.SteelDetailing.ToString() + "\r\n" +
    "[] =" + ModuleManager.TaskManagement.ToString() + "\r\n" +
    "[] " + "\r\n" +
    "[][][][] =" + ModuleManager.Configuration, "Tekl[][] "
);

```

Tekla[][][]

```

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

```

```

    i++;
} // Tekla
MessageBox.Show(files, " ");
string file = teklaStructuresFiles.GetAttributeFile("rebar_with_couplers.tpl").ToString();
//
MessageBox.Show(file);
bool vs = false;
filesList = teklaStructuresFiles.GetMultiDirectoryFileList("rpt", vs);
files = "";
foreach (string tt in filesList)
{
    files += tt + "\r\n";
}
MessageBox.Show(files, " ");
filesList.Clear();
FileInfo tttt = teklaStructuresFiles.GetAttributeFile(teklaStructuresFiles.PropertyFileDirectories, "rebar_with_coupl
MessageBox.Show(tttt.ToString(), " ");

```



```

string infoS = "";
infoS +=
"    =" + TeklaStructuresInfo.GetBuildNumber() + "\r\n" +
"    =" + TeklaStructuresInfo.GetCommonAppDataFolder() + "\r\n" +
"Tekla    =" + TeklaStructuresInfo.GetCopyRightText() + "\r\n" +
"Tekla    =" + TeklaStructuresInfo.GetCurrentProgramVersion() + "\r\n" +
"Tekla    =" + TeklaStructuresInfo.GetCurrentUser() + "\r\n" +
"Tekla    =" + TeklaStructuresInfo.GetFullTSRegistryKeyText() + "\r\n" +
"    appdata    =" + TeklaStructuresInfo.GetLocalAppDataFolder() + "\r\n" +
"Tekla    =" + TeklaStructuresInfo.GetRevisionDate();
MessageBox.Show(infoS, "Tekla ");

```



```

bool ispourendabled = TeklaStructuresSettings.IsPourEnabled();
MessageBox.Show(ispourendabled.ToString(), " ");
string istoolOR = "";
istoolOR += "    =" + TeklaStructuresSettings.IsToolOptionOn("sel_all") + "\r\n";
istoolOR += "    =" + TeklaStructuresSettings.IsToolOptionOn("sel_analysisnodelinks") + "\r\n";
istoolOR += "    =" + TeklaStructuresSettings.IsToolOptionOn("sel_analysisnodes") + "\r\n";
istoolOR += "    =" + TeklaStructuresSettings.IsToolOptionOn("sel_analysisparts") + "\r\n";
istoolOR += "    =" + TeklaStructuresSettings.IsToolOptionOn("sel_single_rebars") + "\r\n\r\n";

istoolOR += "    =" + TeklaStructuresSettings.ToolOptionNames.ConstructionLines + "\r\n";
istoolOR += "    =" + TeklaStructuresSettings.ToolOptionNames.CustomObjects + "\r\n";
istoolOR += "    =" + TeklaStructuresSettings.ToolOptionNames.Cuts + "\r\n";
istoolOR += "    =" + TeklaStructuresSettings.ToolOptionNames.DirectManipulation + "\r\n";
istoolOR += "    ";

```

```

=" + TeklaStructuresSettings.ToolOptionNames.DisplaySelectionFilterDialog + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Distances + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Grid + "\r\n";
istoolOR += "███████" = " + TeklaStructuresSettings.ToolOptionNames.GridLine + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Joints + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Loads + "\r\n";
istoolOR += "██████████" = " + TeklaStructuresSettings.ToolOptionNames.ObjectsInJoints + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Parts + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Planes + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Points + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.PourBreaks + "\r\n";
istoolOR += "███████" = " + TeklaStructuresSettings.ToolOptionNames.RebarGroup + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Rebars + "\r\n";
istoolOR += "██████████" = " + TeklaStructuresSettings.ToolOptionNames.ReferenceModels + "\r\n";
istoolOR += "███████" = " + TeklaStructuresSettings.ToolOptionNames.Screws + "\r\n";
istoolOR += "██████████" = " + TeklaStructuresSettings.ToolOptionNames.SelectAssemblies + "\r\n";
istoolOR += "██████████████"
=" + TeklaStructuresSettings.ToolOptionNames.SelectObjectsInAssemblies + "\r\n";
istoolOR += "██████████" = " + TeklaStructuresSettings.ToolOptionNames.SelectTasks + "\r\n";
istoolOR += "██████████" = " + TeklaStructuresSettings.ToolOptionNames.SingleScrews + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Surfaces + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Views + "\r\n";
istoolOR += "██████" = " + TeklaStructuresSettings.ToolOptionNames.Weldings + "\r\n";
MessageBox.Show(istoolOR, "██████ ");
//TeklaStructuresSettings.GetAdvancedOption("XS_ENABLE_POUR_MANAGEMENT",ref istoolOR);
TeklaStructuresSettings.GetAdvancedOption("XS_POUR_BREAK_COLOR", ref istoolOR);
MessageBox.Show(istoolOR, "██████ ");

```



```

Beam beam = (Beam)new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART);//██████
Solid solid = beam.GetSolid();
FaceEnumerator faceEnumerator = solid.GetFaceEnumerator();//██████ ████████
int i = 0;
while (faceEnumerator.MoveNext()) //██████
{
    LoopEnumerator loopEnumerator = faceEnumerator.Current.GetLoopEnumerator();//██████████ ████████
    while (loopEnumerator.MoveNext()) //██████████
    {
        VertexEnumerator vertexEnumerator = loopEnumerator.Current.GetVertexEnumerator();//██████ ████████
        while (vertexEnumerator.MoveNext())// ████████████
        {
            dataGridView1.Rows.Add();//██████████
            dataGridView1.Rows[i].Cells[0].Value = string.Format("██████ {0}", i);
            dataGridView1.Rows[i].Cells[1].Value = vertexEnumerator.Current as Point;//
            i++;
        }
    }
}

```

```
}
```



```
string sourceFile = @"abl_up_Developer--main_menu.xml";  
string destinationFile = @"C:\Users\Administrator\AppData\Local\Trimble\Tekla Structures\2020.0\UI\Ribbons\abl_  
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" Show  
</QuickAccessToolbar>  
<StaticTab>  
  <SplitButton X="0" Y="0" Width="2" Height="2" Command="Common.Interrupt" Icon="command:ScalableIcon" Sh  
  <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:FullText  
  <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="command  
  <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="command  
  <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="command  
  <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="cor  
  <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="cor  
</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:l
<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="commi
<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="commar
</SplitButton>
<SimpleButton X="16" Y="0" Width="8" Height="1" Command="Modeling.CreateSteellItem" Text="command:9
<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="commar
<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="cc
<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:9
<SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateConcreteSlab" Text="commar
<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="command
</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.CreateConcreteltem" Text="commar
<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="comr
</SplitButton>
<DropDownButton X="26" Y="0" Width="8" Height="2" Text="translation:ribbon_Rebar_set" Icon="resource:Bl
<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.CreateSlabOrWallReinforce
<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=
<CheckBox X="0" Y="0" Width="8" Height="2" Command="Reinforcement.RebarSetLegFaceVisibility" Tex
<CheckBox X="0" Y="2" Width="8" Height="2" Command="Reinforcement.RebarSetGuidelineVisibility" Te
<CheckBox X="0" Y="4" Width="8" Height="2" Command="Reinforcement.RebarSetPropertyStripVisibility
<CheckBox X="0" Y="6" Width="8" Height="2" Command="Reinforcement.RebarSetSplitterVisibility" Text
<CheckBox X="0" Y="8" Width="8" Height="2" Command="Reinforcement.RebarSetEndDetailStripVisibilit
<CheckBox 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" />
<CheckBox 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:F
</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="command:FullText" Icon="resource:Brush" />
<DropDownButton X="16" Y="2" Width="8" Height="2" Text="translation:lbl_Inquire_added_material" Icon="resource:Brush" />
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Material.AttachToPart" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Material.DetachFromPart" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="6" Width="8" Height="3" Command="Material.ExplodePart" Text="command:FullText" Icon="resource:Brush" />
</DropDownButton>
<DropDownButton X="24" Y="0" Width="8" Height="2" Text="translation:lbl_Components" Icon="resource:Brush" />
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="Components.ApplicationsAndComponents" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="Components.CreateCurrentConnection" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="6" Width="8" Height="3" Command="AutoConnection.Create" Text="command:FullText" Icon="resource:Brush" />
</DropDownButton>
<DropDownButton X="24" Y="2" Width="8" Height="1" Text="translation:ribbon_Surfaces" Icon="resource:Brush" />
  <SimpleButton X="0" Y="0" Width="8" Height="3" Command="SurfaceTreatment.CreateToPartFace" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="3" Width="8" Height="3" Command="SurfaceTreatment.CreateToSelectedAreaOnPart" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="6" Width="8" Height="3" Command="SurfaceTreatment.CreateToAllFacesOfPart" Text="command:ShortText" Icon="resource:Brush" />
  <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" Text="command:ShortText" Icon="resource:Brush" />
</DropDownButton>
<DropDownButton X="24" Y="3" Width="8" Height="1" Text="translation:lbl_Compare" Icon="resource:Brush" />
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Tools.CompareParts" Text="command:FullText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Tools.CompareAssemblies" Text="command:FullText" Icon="resource:Brush" />
</DropDownButton>
<SplitButton X="32" Y="0" Width="4" Height="4" Command="Measure.Distance" Text="translation:ribbon_Measure" Icon="resource:Brush" />
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Measure.Distance" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Measure.HorizontalDistance" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Measure.VerticalDistance" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Measure.Angle" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Measure.Arc" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="10" Width="8" Height="2" Command="Measure.BoltSpacing" Text="command:ShortText" Icon="resource:Brush" />
</SplitButton>
<SimpleButton X="36" Y="0" Width="6" Height="2" Command="Edit.Copy" Text="command:FullText" Icon="resource:Brush" />
<DropDownButton X="36" Y="2" Width="6" Height="2" Text="translation:lbl_Copy_special" Icon="resource:Brush" />
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Edit.CopySpecial.Linear" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Edit.CopySpecial.Mirror" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Edit.CopySpecial.Rotate" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Edit.CopySpecial.ToAnotherObject" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="8" Width="8" Height="3" Command="Edit.CopySpecial.AllContentToAnotherObject" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="11" Width="8" Height="2" Command="Edit.CopySpecial.ToAnotherPlane" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="13" Width="8" Height="2" Command="Edit.CopySpecial.FromAnotherModel" Text="command:ShortText" Icon="resource:Brush" />
</DropDownButton>
<SimpleButton X="42" Y="0" Width="6" Height="2" Command="Edit.Move" Text="command:FullText" Icon="resource:Brush" />
<DropDownButton X="42" Y="2" Width="6" Height="2" Text="translation:lbl_Move_special" Icon="resource:Brush" />
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Edit.MoveSpecial.Linear" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Edit.MoveSpecial.Rotate" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Edit.MoveSpecial.Mirror" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Edit.MoveSpecial.ToAnotherPlane" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Edit.MoveSpecial.ToAnotherObject" Text="command:ShortText" Icon="resource:Brush" />
</DropDownButton>
<SplitButton X="48" Y="0" Width="3" Height="4" Command="Modeling.CreateRectangularGrid" Text="translation:ribbon_Grid" Icon="resource:Brush" />
  <SimpleButton X="0" Y="0" Width="8" Height="2" Command="Modeling.CreateRectangularGrid" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="Modeling.CreateRadialGrid" Text="command:ShortText" Icon="resource:Brush" />
  <SimpleButton X="0" Y="4" Width="8" Height="2" Command="Modeling.AddGridLine" Text="command:FullText" Icon="resource:Brush" />
</SplitButton>
```

```

<DropDownButton X="51" Y="0" Width="3" Height="4" Text="translation:abl_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="comm
<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="com
<SimpleButton X="0" Y="16" Width="8" Height="2" Command="Points.AddTangentToCircle" Text="command
<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" Tex
<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" Te
</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="comm
<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="comr
<SimpleButton X="0" Y="6" Width="8" Height="2" Command="Modeling.AddConstructionArc" Text="comm
<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="com
<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="command
<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" Tex
<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="com
<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" Tex
<SimpleButton X="0" Y="17" Width="8" Height="2" Command="View.CreateModelViewOnPartBackPlane" Te
<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:S
<SimpleButton X="0" Y="24" Width="8" Height="2" Command="View.CreatePartDefaultViews" Text="comm
<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="com
<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:E
  <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:SI
</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="comr
  <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.ShadedWirefr
  <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="comn
</SplitButton>
<SimpleButton X="34" Y="0" Width="8" Height="2" Command="View.TogglePerspective" Text="command:Full
<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:abl_Zoom" Icon="resource:Brush.Cc
  <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="cc
<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:abl_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="comr
    <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="con
    <SimpleButton X="0" Y="6" Width="8" Height="2" Command="Drawing.CreateCastUnitDrawing" Text="comr
    <SimpleButton X="0" Y="8" Width="8" Height="2" Command="Drawing.CreateGeneralArrangementDrawing"
  </SplitButton>
  <DropDownButton X="13" Y="0" Width="8" Height="2" Text="translation:abl_Perform_numbering" Icon="resc
    <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="cc
    <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:abl_Numbering_settings" Icon="resc
    <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:SI
<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:FullT
<SimpleButton X="21" Y="0" Width="3" Height="4" Command="LockManager.ObjectLocks" Text="command:
<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:
<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="comr
</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:St
<SimpleButton X="18" Y="2" Width="8" Height="2" Command="Analysis.MergeSelectedNodes" Text="comm
<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="comm
<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="comn
<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="resol
<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:
<SimpleButton X="0" Y="6" Width="8" Height="2" Command="Loads.UniformLoadProperties" Text="comm
```

```

<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
<CheckBox X="42" Y="0" Width="8" Height="1" Command="Analysis.ToggleShowNodeNumbers" Text="co
<CheckBox 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="command
  <SimpleButton X="0" Y="2" Width="8" Height="2" Command="TrimbleConnect.ProjectModel" Text="comman
  <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="comi
<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:St
<SimpleButton X="20" Y="0" Width="2" Height="2" Command="TrimbleConnect.AdjustTsView" Text="commar
<SimpleButton X="20" Y="2" Width="2" Height="2" Command="TrimbleConnect.SelectTsObjects" Text="comi
<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="□□□□" IsCollapsed="false" IsUserDefined="false">
  <SimpleButton X="0" Y="0" Width="3" Height="4" Command="Plugin.CatalogPluginComponentItem?SC
  □□□□□□□□ " Text="□□□□" Icon="D:\□□□□ \BIM□□□□ \TSEP□□ \□□□□□□ \□□
  \favicon.ico" ShowText="true" ShowIcon="true" />
  <CheckBox X="3" Y="0" Width="3" Height="4" Command="Plugin.CatalogPluginComponentItem?SC
  □□□□□□ 32□□ " Text="□□□□" Icon="D:\□□□□ \BIM□□□□ \TSEP□□ \□□□□□□ \□□
  \favicon (3).ico" ShowText="true" ShowIcon="true" />
  <CheckBox X="6" Y="0" Width="3" Height="4" Command="Plugin.CatalogPluginComponentItem?SC
  □□□□□□ 29□ " Text="□□□□" Icon="D:\□□□□ \BIM□□□□ \TSEP□□ \□□□□□□ \□□
  \favicon (2).ico" ShowText="true" ShowIcon="true" />
  <CheckBox X="9" Y="0" Width="3" Height="4" Command="Plugin.CatalogPluginComponentItem?SC
  □□□□□□ 38□□□ " Text="□□□□" Icon="D:\□□□□ \BIM□□□□ \TSEP□□ \□□□□□□ \□□
  \favicon (1).ico" ShowText="true" ShowIcon="true" />
</Tab>
</Ribbon>

```



```

string str = "□□□□ \r\n\r\n";
DrawingHandler drawingHandler = new DrawingHandler();//□□□□□□□□
var drawings= drawingHandler.GetDrawings();//□□□□□□□□
while (drawings.MoveNext())
{
  str += "□□□□□□ " + drawings.Current.Name + "\r\n\r\n";
  str += "□□□□□□ " + drawings.Current.GetType() + "\r\n\r\n";
  str += "□□□□□□ " + drawings.Current.Mark + "\r\n\r\n";
  str += "□□□□□□ " + ((drawings.Current as Drawing).IsFrozen == true ? "□□ " : "□□□□ ") + "\r\n\r\n";
}

```

```

    str += "        " + ((drawings.Current as Drawing).IsLocked == true ? "   " : "    ") + "\n\n";
    str += "    1 " + drawings.Current.Title1 + "\n\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.BringToFront = true;
        base.OnStartupNextInstance(eventArgs);
    }
}

```

```

/// <summary>
/// ██████████
/// </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();
        }
    }
}
}

```



```

Part beam = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART) as Part;//██████
string property = "████ \r\n\r\n";
//████
property += "██ : " + beam.GetType().Name + "\r\n\r\n";//██
property += "██ : " + beam.Name + "\r\n\r\n";//██
property += "██ : " + beam.Profile.ProfileString + "\r\n\r\n";//██
property += "██ : " + beam.Material.MaterialString + "\r\n\r\n";//██
property += "██ : " + beam.Finish + "\r\n\r\n";//██
property += "██ : " + beam.Class + "\r\n\r\n";//██
property += "████ : " + beam.PartNumber.Prefix + "\r\n\r\n";//████
property += "████ : " + beam.PartNumber.StartNumber + "\r\n\r\n";//████
property += "████ : " + beam.AssemblyNumber.Prefix + "\r\n\r\n";//████
property += "████ : " + beam.AssemblyNumber.StartNumber + "\r\n\r\n";//████

property += "████████ : " + beam.Position.Depth + "\r\n\r\n";//████████
property += "████ : " + beam.Position.DepthOffset + "\r\n\r\n";//████

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

```

```

property += "Rotation" : " + beam.Position.Rotation + "\r\n\r\n";//Rotation
property += "RotationOffset" : " + beam.Position.RotationOffset + "\r\n\r\n";//RotationOffset
property += "Plane" : " + beam.Position.Plane + "\r\n\r\n";//Plane
property += "PlaneOffset" : " + beam.Position.PlaneOffset + "\r\n\r\n";//PlaneOffset

property += "StartPointOffset.Dx" : " + ((Beam)beam).StartPointOffset.Dx + "\r\n\r\n";//StartPointOffset.Dx
property += "StartPointOffset.Dy" : " + ((Beam)beam).StartPointOffset.Dy + "\r\n\r\n";//StartPointOffset.Dy
property += "StartPointOffset.Dz" : " + ((Beam)beam).StartPointOffset.Dz + "\r\n\r\n";//StartPointOffset.Dz
property += "EndPointOffset.Dx" : " + ((Beam)beam).EndPointOffset.Dx + "\r\n\r\n";//EndPointOffset.Dx
property += "EndPointOffset.Dy" : " + ((Beam)beam).EndPointOffset.Dy + "\r\n\r\n";//EndPointOffset.Dy
property += "EndPointOffset.Dz" : " + ((Beam)beam).EndPointOffset.Dz + "\r\n\r\n";//EndPointOffset.Dz
property += "Angle" : " + beam.DeformingData.Angle + "\r\n\r\n";//Angle
property += "Angle2" : " + beam.DeformingData.Angle2 + "\r\n\r\n";//Angle2
property += "Cambering" : " + beam.DeformingData.Cambering + "\r\n\r\n";//Cambering
property += "Shortening" : " + beam.DeformingData.Shortening + "\r\n\r\n";//Shortening
}

textBox1.Text = property;
//Rotation
property = "";
string pos = "";
int pos2 = 0;
double pos1 = 0.0;
Phase phase;
property += "PART_POS" : " + "\r\n\r\n";
property += "PART_POS" : " + beam.GetReportProperty("PART_POS", ref pos) + " " + pos + "\r\n\r\n";
property += "Phase" : " + beam.GetPhase(out phase) + " " + phase.PhaseName + "\r\n\r\n";
property += "WEIGHT" : " + beam.GetReportProperty("WEIGHT", ref pos1) + " " + pos1 + "\r\n\r\n";
property += "VOLUME" : " + beam.GetReportProperty("VOLUME", ref pos1) + " " + pos1 + "\r\n\r\n";
property += "LENGTH" : " + beam.GetReportProperty("LENGTH", ref pos1) + " " + pos1 + "\r\n\r\n";
property += "NUMBER" : " + beam.GetAssembly().GetReportProperty("NUMBER", ref pos2) + " " + pos2 + "\r\n\r\n";
property += "TOP_LEVEL" : " + beam.GetReportProperty("TOP_LEVEL", ref pos) + " " + pos + "\r\n\r\n";
property += "BOTTOM_LEVEL" : " + beam.GetReportProperty("BOTTOM_LEVEL", ref pos) + " " + pos + "\r\n\r\n";
property += "POSITION_CODE" : " + beam.GetAssembly().GetReportProperty("POSITION_CODE", ref pos) + " " + pos + "\r\n\r\n";
textBox2.Text = property;

```



```

Weld weld = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_WELD) as Weld;//Weld
string basicproperty = "WeldType" : "\r\n\r\n";
basicproperty += "WeldType" : " + weld.GetType().Name + "\r\n\r\n";
basicproperty += "AroundWeld" : " + (weld.AroundWeld == true ? "Yes" : "No") + "\r\n\r\n";
basicproperty += "ShopWeld" : " + (weld.ShopWeld == true ? "Yes" : "No") + "\r\n\r\n";
basicproperty += "Position" : " + weld.Position + "\r\n\r\n";
basicproperty += "IntermittentType" : " + weld.IntermittentType + "\r\n\r\n";

basicproperty += "ConnectAssemblies" : " + (weld.ConnectAssemblies == true ? "Yes" : "No") + "\r\n\r\n";//ConnectAssemblies
basicproperty += "Placement" : " + weld.Placement + "\r\n\r\n";
//basicproperty += "Preparation" : " + weld.Preparation + "\r\n\r\n";//Preparation

basicproperty += "PrefixAboveLine" : " + weld.PrefixAboveLine + "\r\n\r\n";//PrefixAboveLine

```

```

basicproperty += "XXXXXXXX" + weld.TypeAbove + "\n\n\n";//XXXXXX
basicproperty += "XXXXXX" + weld.SizeAbove + "\n\n\n";//XXXX
basicproperty += "XXXXXX" + weld.AngleAbove + "\n\n\n";//XXXX
basicproperty += "XXXXXX" + weld.ContourAbove + "\n\n\n";//XXXXXX
basicproperty += "XXXXXXXX" + weld.FinishAbove + "\n\n\n";//XXXX
basicproperty += "XXXXXX" + weld.RootFaceAbove + "\n\n\n";
basicproperty += "XXXXXX" + weld.EffectiveThroatAbove + "\n\n\n";
basicproperty += "XXXXXX" + weld.RootOpeningAbove + "\n\n\n";
basicproperty += "XXXXXX" + weld.IncrementAmountAbove + "\n\n\n";
basicproperty += "XXXX" + weld.LengthAbove + "\n\n\n";
basicproperty += "XXXX" + weld.PitchAbove + "\n\n\n";

```

```

basicproperty += "XXXXXX" + weld.PrefixBelowLine + "\n\n\n";//XXXXXX
basicproperty += "XXXXXXXX" + weld.TypeBelow + "\n\n\n";//XXXXXX
basicproperty += "XXXXXX" + weld.SizeBelow + "\n\n\n";//XXXX
basicproperty += "XXXXXX" + weld.AngleBelow + "\n\n\n";//XXXX
basicproperty += "XXXXXX" + weld.ContourBelow + "\n\n\n";//XXXXXX
basicproperty += "XXXXXXXX" + weld.FinishBelow + "\n\n\n";//XXXX
basicproperty += "XXXXXX" + weld.RootFaceBelow + "\n\n\n";
basicproperty += "XXXXXX" + weld.EffectiveThroatBelow + "\n\n\n";
basicproperty += "XXXXXX" + weld.RootOpeningBelow + "\n\n\n";
basicproperty += "XXXXXX" + weld.IncrementAmountBelow + "\n\n\n";
basicproperty += "XXXX" + weld.LengthBelow + "\n\n\n";
basicproperty += "XXXX" + weld.PitchBelow + "\n\n\n";

```

```

basicproperty += "NDTXXXX" + weld.NDTInspection + "\n\n\n";//NDTXXXX
basicproperty += "XXXXXX" + weld.ElectrodeClassification + "\n\n\n";//XXXXXX
basicproperty += "XXXXXX" + weld.ElectrodeStrength + "\n\n\n";//XXXX
basicproperty += "XXXXXX" + weld.ElectrodeCoefficient + "\n\n\n";//XXXX
basicproperty += "XXXX" + weld.ProcessType + "\n\n\n";//XXXX
basicproperty += "XX" + weld.ReferenceText + "\n\n\n";
textBox3.Text = basicproperty;
//XXXX
double ww1 = 0.0;
string ww = "";
basicproperty = "XXXX \n\n\n";

```

```

basicproperty += "XXXX" + weld.GetReportProperty("WELD_NUMBER", ref ww) + " " + ww + "\n\n\n";
basicproperty += "XXXX" + weld.GetReportProperty("WELD_FATHER_NUMBER", ref ww) + "
" + ww + "\n\n\n";
basicproperty += "XX" + weld.GetReportProperty("LENGTH", ref ww1) + " " + ww1 + "\n\n\n";
textBox4.Text = basicproperty;

```



```

double num = 0.0;
string number = "";
Assembly assembly = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_OBJECT) as Assembly;//
XXXX
string basicproperty = "XXXX \n\n\n";
basicproperty += "XX" + assembly.GetAssemblyType() + "\n\n\n";
basicproperty += "XX" + assembly.GetReportProperty("ASSEMBLY_WEIGHT", ref num) + "

```

```

" + num + "\r\n\r\n";
basicproperty += "[]" + assembly.GetReportProperty("ASSEMBLY_VOLUME", ref num) + "[]"
" + num + "\r\n\r\n";
basicproperty += "[]" + assembly.GetReportProperty("LENGTH", ref num) + "[]" + num + "\r\n\r\n";
basicproperty += "[]" + assembly.GetReportProperty("NUMBER", ref num) + "[]" + num + "\r\n\r\n";
basicproperty += "[]" + assembly.GetReportProperty("ASSEMBLY_TOP_LEVEL", ref number) + "[]"
" + number + "\r\n\r\n";
basicproperty += "[]" + assembly.GetReportProperty("ASSEMBLY_BOTTOM_LEVEL", ref number) + "[]"
" + number + "\r\n\r\n";
basicproperty += "[]" + assembly.GetReportProperty("ASSEMBLY_POSITION_CODE", ref number) + "[]"
" + number + "\r\n\r\n";
textBox6.Text = basicproperty;
basicproperty = "[] \r\n\r\n";
basicproperty += "[]" + assembly.GetReportProperty("ASSEMBLY_POS", ref number) + "[]"
" + number + "\r\n\r\n";
basicproperty += "[]" + 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("{} {}"
", i) + part.GetReportProperty("PART_POS", ref number) + "[]" + number + "\r\n\r\n";
        basicproperty += string.Format("{} {}"
", i) + part.Profile.ProfileString + "\r\n\r\n";
        if (part.GetWelds().GetSize() > 0)
        {
            var welds = part.GetWelds();
            while (welds.MoveNext())
            {
                basicproperty += string.Format("{} {}"
", i) + (welds.Current as Weld).GetReportProperty("WELD_NUMBER", ref number) + "[]" + number + "\r\n\r\n";
            }
        }
    }
}

if (part.GetType() == typeof(ContourPlate))
{
    basicproperty += string.Format("{} {}"
", i) + part.GetReportProperty("PART_POS", ref number) + "[]" + number + "\r\n\r\n";
    basicproperty += string.Format("{} {}"
", i) + part.Profile.ProfileString + "\r\n\r\n";
    if (part.GetWelds().GetSize() > 0)
    {
        var welds = part.GetWelds();
        while (welds.MoveNext())
        {
            basicproperty += string.Format("{} {}"
", i) + (welds.Current as Weld).GetReportProperty("WELD_NUMBER", ref number) + "[]" + number + "\r\n\r\n";
        }
    }
}

```

```

}
i++;
}

```

```
textBox5.Text = basicproperty;
```



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

```


```

```

string basicproperty = "      \r\n\r\n";
basicproperty += "      : " + boltArray.PartToBeBolted.GetPartMark() + "\r\n\r\n"
+ "      : " + boltArray.PartToBoltTo.GetPartMark() + "\r\n\r\n"
+ "      : " + boltArray.BoltSize + "\r\n\r\n"
+ "      : " + (boltArray.BoltSize + boltArray.Tolerance) + "\r\n\r\n"
+ "      : " + boltArray.BoltType + "\r\n\r\n"
+ "      : " + boltArray.BoltStandard + "\r\n\r\n"
+ "      : " + boltArray.BoltStandard + "\r\n\r\n"
+ "      : " + boltArray.Length + "\r\n\r\n"
+ "      : " + boltArray.ExtraLength + "\r\n\r\n"
+ "      : " + boltArray.CutLength + "\r\n\r\n"
+ "      : " + boltArray.ThreadInMaterial + "\r\n\r\n"
+ "      : " + boltArray.Position.Plane + "\r\n\r\n"
+ "      : " + boltArray.Position.Rotation + "-" + boltArray.Position.RotationOffset + "\r\n\r\n"
+ "      : " + (boltArray.Bolt == true ? " " : " ") + "\r\n\r\n"
+ "      1 : " + (boltArray.Washer1 == true ? " " : " ") + "\r\n\r\n"
+ "      1 : " + (boltArray.Washer2 == true ? " " : " ") + "\r\n\r\n"
+ "      2 : " + (boltArray.Washer3 == true ? " " : " ") + "\r\n\r\n"
+ " 1 : " + (boltArray.Nut1 == true ? " " : " ") + "\r\n\r\n"
+ " 2 : " + (boltArray.Nut2 == true ? " " : " ") + "\r\n\r\n"
+ " 5 : " + (boltArray.Hole1 == true ? " " : " ") + "\r\n\r\n"
+ "      : " + (boltArray.HoleType == BoltGroup.BoltHoleTypeEnum.HOLE_TYPE_OVERSIZED ? " " : " ") + "\r\n\r\n"
+ "      : " + (boltArray.BoltSize + boltArray.Tolerance + boltArray.SlottedHoleX) + "\r\n\r\n"
+ "      : " + boltArray.RotateSlots + "\r\n\r\n"
+ "x      : " + boltArray.GetBoltDistX(0).ToString() + "\r\n\r\n"
+ "y      : " + boltArray.GetBoltDistY(0).ToString() + "\r\n\r\n"
+ "  Dx : " + boltArray.StartPointOffset.Dx + "\r\n\r\n"
+ "  Dx : " + boltArray.EndPointOffset.Dx + "\r\n\r\n";
textBox8.Text = basicproperty;
int t = 0;
string str = "";
basicproperty = "      \r\n\r\n";

basicproperty += "      "
+ boltArray.GetReportProperty("SECONDARY_1.ASSEMBLY_POS", ref str) + "-" + str + "\r\n\r\n";
basicproperty += "      "
+ boltArray.GetReportProperty("CONNECTED_ASSEMBLIES", ref str) + "-" + str + "\r\n\r\n";
basicproperty += "      "
+ boltArray.GetReportProperty("CONNECTED_PARTS", ref str) + "-" + str + "\r\n\r\n";
basicproperty += "      " + boltArray.GetReportProperty("NUMBER", ref t) + "-" + t + "\r\n\r\n";

```

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



```
string t1 = "", t2 = "", t3 = "", t4 = "", t5 = "";
CatalogHandler catalogHandler = new CatalogHandler();//[ ]

if (catalogHandler.GetConnectionStatus())//[ ]
{
    ComponentItemEnumerator componentItemEnumerator = catalogHandler.GetComponentItems();//
    [ ]
    //t1 = componentItemEnumerator.GetSize().ToString() + "\r\n\r\n";//[ ]
    while (componentItemEnumerator.MoveNext())//[ ]
    {
        if (componentItemEnumerator.Current.Name == "DkBoltCommon")
        {
            t1 += "[ ] : " + (componentItemEnumerator.Current as ComponentItem).Name + "\r\n\r\n"
                + "[ ] : " + componentItemEnumerator.Current.Name + "\r\n\r\n"
                + "[ ] UI:" + componentItemEnumerator.Current.UIName + "\r\n\r\n"
                + "[ ] : " + componentItemEnumerator.Current.Type;
        }
        componentItemEnumerator.Current.Select(componentItemEnumerator.Current.Name, componentItemEnum
        t2 += "[ ] : " + componentItemEnumerator.Current.Number + "\r\n\r\n";//[ ]
        t3 += "UI[ ] : " + componentItemEnumerator.Current.UIName + "\r\n\r\n";//[ ] UI[ ]
        switch (componentItemEnumerator.Current.Type.ToString()) //[ ]
switch(){case :beak;....default :break;}//[ ]
{
    case "UNKNOWN":
        t4 += "[ ] " + "\r\n\r\n";
        break;
    case "CONNECTION":
        t4 += "[ ] " + "\r\n\r\n";
        break;
    case "COMPONENT":
        t4 += "[ ] " + "\r\n\r\n";
        break;
    case "SEAM":
        t4 += "[ ] " + "\r\n\r\n";
        break;
    case "DETAIL":
        t4 += "[ ] " + "\r\n\r\n";
        break;
    case "CUSTOM_PART":
        t4 += "[ ] " + "\r\n\r\n";
        break;
    case "DRAWING_PLUGIN":
        t4 += "[ ] " + "\r\n\r\n";
        break;
    default:
        t4 += "[ ] " + "\r\n\r\n";
        break;
}
```

```

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

```



```

string str = "";
int t = 0;
Part castA = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART) as Part;//[ ]
Assembly cast = castA.GetAssembly();//[ ]
string castproperty = "[ ] \r\n\r\n";
castproperty += "[ ] " + castA.GetAssembly().GetType().Name + "\r\n\r\n";
castproperty += "[ ] ID[ ] " + castA.Identifier.ToString() + "\r\n\r\n";//[ ]
castproperty += "[ ] " + castA.GetReportProperty("PART_POS", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + castA.GetReportProperty("NUMBER", ref t) + "[ ] " + t + "\r\n\r\n";
castproperty += "[ ] " + castA.GetReportProperty("PHASE.NAME", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + castA.Name + "\r\n\r\n";
castproperty += "[ ] " + castA.GetAssembly().GetReportProperty("POSITION_CODE", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + castA.GetReportProperty("TOP_LEVEL", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + castA.GetReportProperty("BOTTOM_LEVEL", ref str) + "[ ] " + str + "\r\n\r\n";
textBox12.Text = castproperty;
castproperty = "[ ] \r\n\r\n";
castproperty += "[ ] " + cast.GetAssemblyType().ToString() + "\r\n\r\n";
castproperty += "[ ] ID[ ] " + cast.Identifier.ToString() + "\r\n\r\n";//[ ]
castproperty += "[ ] " + cast.GetReportProperty("CAST_UNIT_POS", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + cast.GetReportProperty("NUMBER", ref t) + "[ ] " + t + "\r\n\r\n";
castproperty += "[ ] " + cast.GetReportProperty("PHASE.NAME", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + cast.GetReportProperty("MAINPART.NAME", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + cast.GetReportProperty("CAST_UNIT_POSITION_CODE", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + cast.GetReportProperty("TOP_LEVEL", ref str) + "[ ] " + str + "\r\n\r\n";
castproperty += "[ ] " + cast.GetReportProperty("BOTTOM_LEVEL", ref str) + "[ ] " + str + "\r\n\r\n";
textBox11.Text = castproperty;

```



```

ArrayList arrayList = new ArrayList();//[ ]
Part castA = new Picker().PickObject(Picker.PickObjectEnum.PICK_ONE_PART) as Part;//[ ]
int t = 0;
double st = 0.0;
string str = "";
string castproperty = "[ ] \r\n\r\n";
string castproperty2 = "[ ] \r\n\r\n";
var rebars = castA.GetChildren();//[ ] \[ ]
//while (rebars.MoveNext())
//{

```

```

// if (rebars.Current.GetType() == typeof(RebarGroup)
//     || rebars.Current.GetType() == typeof(SingleRebar)
//     || rebars.Current.GetType() == typeof(RebarSet))//
████████████████████████████████████████
// {
//     arrayList.Add(rebars.Current);
//     castproperty += "██████" + rebars.Current.GetType() + "\r\n\r\n";
//     castproperty += "██████" + rebars.Current.GetReportProperty("REBAR_POS", ref str) + "█
" + str + "\r\n\r\n";
//     castproperty += "██████" + rebars.Current.GetReportProperty("NUMBER", ref t) + "█ " + t + "\r\n\r\n";
//     castproperty += "██████" + rebars.Current.GetReportProperty("GRADE", ref str) + "█
" + str + "\r\n\r\n";
//     castproperty += "██████" + rebars.Current.GetReportProperty("SIZE", ref str) + "█ " + str + "\r\n\r\n";
//     castproperty += "██████" + rebars.Current.GetReportProperty("LENGTH", ref st) + "█
" + st + "\r\n\r\n";
//     castproperty += "██████" + rebars.Current.GetReportProperty("WEIGHT", ref st) + "█
" + st + "\r\n\r\n";
//     castproperty += "██████" + rebars.Current.GetReportProperty("WEIGHT", ref st) + "█
" + st * t + "\r\n\r\n";

//     //████████████████████████████████████████
//     castproperty2 += "██████" + rebars.Current.GetReportProperty("SHAPE", ref str) + "█
" + str + "\r\n\r\n";
// }

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

```

██████████

```

string str = "██████ \r\n\r\n";
DrawingHandler drawingHandler = new DrawingHandler();//██████████
var drawings= drawingHandler.GetDrawings();//██████████
while (drawings.MoveNext())
{
    str += "██████" + drawings.Current.Name + "\r\n\r\n";
    str += "██████" + drawings.Current.GetType() + "\r\n\r\n";
    str += "██████" + drawings.Current.Mark + "\r\n\r\n";
    str += "██████" + ((drawings.Current as Drawing).IsFrozen == true ? "██ " : "██ ") + "\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;

```

██████████

```

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;//██████
string str = "██████ \r\n\r\n";//██████
str += "██ :" + part.Profile.ProfileString + "\r\n\r\n";
str += "██ :" + part.Material.MaterialString + "\r\n\r\n";

```

```

str += "[] : " + part.GetReportProperty("LENGTH", ref length) + "-" + length + "\n\n";
str += "[] : " + part.GetReportProperty("WEIGHT", ref weight) + "-" + weight + "\n\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 = "[] \n\n";
str += "[] " + "PL" + thickness + "*" + width + "\n\n";
str += "[] " + length + "\n\n";
//weights += (7.85 * thickness * width * length / 1000000);
//str += "[] " + weights + "\n\n";
//[]
part.GetReportProperty("PROFILE.WEB_THICKNESS", ref thickness);
part.GetReportProperty("WEB_WIDTH", ref width);
part.GetReportProperty("WEB_LENGTH", ref length);
str += "[] " + "PL" + thickness + "*" + width + "\n\n";
str += "[] " + length + "\n\n";
//weights += (7.85 * thickness * width * length / 1000000);
//str += "[] " + (7.85 * thickness * width * length / 1000000) + "\n\n";
//[]
part.GetReportProperty("FLANGE_THICKNESS_B", ref thickness);
part.GetReportProperty("FLANGE_WIDTH_B", ref width);
part.GetReportProperty("FLANGE_LENGTH_B", ref length);
str += "[] " + "PL" + thickness + "*" + width + "\n\n";
str += "[] " + length + "\n\n";
//weights += (7.85 * thickness * width * length / 1000000);
//str += "[] " + (7.85 * thickness * width * length / 1000000) + "\n\n";
//str += "[] " + weights;
textBox19.Text = str;

```



```

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);
}

```



```
pickobjects.Current.GetReportProperty("PROFILE_TYPE", ref str);
```

□□ #4

□ □ □□ 31 □□ 2024 20:49:14

□ □ □□ 1 □□ 2025 14:53:26