Release PCB-Investigator 8.0 Automation Interface update

New features:

IAction

New ID_Action_Item : ID_SURFACEOUTLINE
Change between outline and full filling of surface drawing.

IAttribute

All properties of the component. The List contains IEDA_PRPs to handle names and values directly.
static List<IEDA_PRP> GetAllProperties(ICMPObject Parent)

Get all attributes of a package definition.
static List<IAttributeElement> GetAllAttributes(IPackageSpecificsD Parent, IPCBIWindow Window)

Create a list of all existing attributes of the current net.
static List<IAttributeElement> GetAllAttributes(INet Net, IPCBIWindow Window)

Set an package attribute to the Parent package specifics. This do not set the "has changed" flag in Step and IPCBIWindow, you can use PackagePropertyOrAttributeHasChanged and NeedToSave to set it manualy.
static bool SetAttribute(IAttributeElement Attribute, IPackageSpecificsD Parent)

Add or override the attribute for the parent net.
static bool SetAttribute(IAttributeElement Attribute, INet Parent)

Set a component property to the component object.
static bool SetProperty(IEDA_PRP Property, ICMPObject Parent)

Set a package property to the relevant package. This do not set the "has changed" flag in Step and IP-CBIWindow, you can use PackagePropertyOrAttributeHasChanged and NeedToSave to set it manualy.
static bool SetProperty(IEDA_PRP Property, IPackageSpecificsD Parent)

Remove a property from the package identified by name. This do not set the "has changed" flag in Step and IPCBIWindow, you can use PackagePropertyOrAttributeHasChanged and NeedToSave to set it manualy.
static bool RemoveProperty(string PropertyName, IPackageSpecificsD Parent)

Set the need to save flag of current step and relevant package list.
static void PackagePropertyOrAttributeHasChanged(IStep Step)

Remove attribute from parent, identified by attribute enum.
static bool RemoveAttribute(FeatureAttributeEnum AttributeEnum, INet Parent)

Remove attribute of the relevant INet.
static bool RemoveAttribute(string DisplayName, INet Parent, IPCBIWindow Window)

Get all properties of the parent component object.
static Dictionary<string, Automation.IEDA_PRP> GetProperties(ICMPObject Parent)

Get all properties of the parent component object.
static Dictionary<string, Automation.IEDA_PRP> GetProperties(IPackageSpecificsD Parent)

Remove Attribute from Package identified by feature attribute enum.
static bool RemoveAttribute(FeatureAttributeEnum AttributeEnum, IPackageSpecificsD Parent)

The ODB++ conform display name of the attribute.
Special Feature Attributes need the parent to load the look up table of special display names.
static string GetDisplayName(IPCBIWindow Parent, FeatureAttributeEnum AttributeEnum)

Create List of all Net Attribute Enums defined in ODB++.
static List<FeatureAttributeEnum> GetAllNetAttributeEnums()

IAttributeElement

User Attributes are handled differently, you need this static method to create the user attribute and set the value.
static IAttributeElement CreateUserAttribute(string UserAttributeName, string Value, ODBEntityEnum EntityType, IPCBIWindow Parent)

Clone the attribute element and set the value and type.
IAttributeElement Clone()

IAutomation

Add an entry to the error log, this can be a warning too.
static void AddToErrorLog(string WarningOrError, bool IsError = true)

Distance around selection for zoom to selection feature (see PCB-Investigator options dialog page "display").
static int ZoomAreaInflateValue

Ask PCB-Investigator for checking ODB++ license.
static bool IsODBAllowed()

Ask PCB-Investigator how it is running. As Demo version or not?
static bool IsDemoVersion

Some drawing options are moved to IDrawingParameters (see details there).

ICMPLayer

Returns true if layer has an image overlay
override bool HasOverlay()

Returns the Overlay color at a certain position
override Color GetOverlayColor(double worldX, double worldY)

ICMPObject

[read only] Bounds in mils of board location.
PCBI.MathUtils.PointD CenterPoint

The bounds of the component body (without Pins).
MathUtils.RectangleD GetBodyBoundsD()

GetComponentAtributes returns a Dictionary of strings with all component attributes.
Dictionary<string, IEDA_PRP> GetComponentAttributeClasses()

Creates outline as IPolyClass. This is simple outline with only closed outline parts (e.g. genCad files contain often single lines or non closed structures).
PCBI.MathUtils.IPolyClass GetSimplePolygonOutline(bool IncludePins)

IDrawingParameters

```
IDrawingParameters:

Use XOR Colormixing also for Components
        bool XorComponentLayer = true;

List of Layers (with additional settings) to draw. The order is important because of the ColorMix-
ingMode.
        List<ILayerParameters> SortedLayersToDraw;
Draw the contour of the board
        bool DrawProfile = true;
Draw the contour of the board
        bool DrawOrigin = true;
Color of the Profile Pen
        Color ProfileColor = Color.FromArgb(255, 255, 255);
Fill the contour of the board with the BoardBrushColor
        bool DrawBoardColored = false;
Color which is used for filling the contour of the PCB
        Color BoardBrushColor = Color.Green;
Background color
        Color BackColor = Color.Black;
If more than one layer is drawn, this mode determines how the different overlapping colors are mixed
        DrawingMode ColorMixingMode = DrawingMode.xor;
If set, drills are drawn at the end without mixing the colors
        bool NoColorMixtureForDrills = false;
Draw ODB++ Zones (if defined in the data set)
```

```
        bool DrawODBZones = false;
Set of Parameters for drawing components
        IComponentParameters ComponentParameters;
Set of Parameters for drawing the selection
        ISelectionParameters SelectionParameters;
Set of Parameters for drawing normal objects (Line/Arc/Surface/Text/Pad)
        IFeatureParameters FeatureParameters;
Draw additional panel information (if a step with Step and Repeat is active)
        bool PanelInfoInProfile;
Draw all instances of a Step and Repeat panel
        bool FlattenStep;

IComponentParameters :

Color for components on TOP layer
        Color ComponentColorTop = Color.FromArgb(250, 200, 100);
Color for components on BOT layer
        Color ComponentColorBot = Color.FromArgb(250, 100, 200);
Label Setting for Components
        ComponentLabel ComponentLabel;
Show a label in each Pin
        bool ShowPinLabel;
Type of the Pin Label
        ComponentPinLabelEnum PinLabelType;
Show Component Pins or hide them totally
        bool ShowPins;
Color of the Component Label
        Color ComponentLabelColor;
Mirror all label texts of a component
        bool MirrorLabels = false;// SpecialDrawForCMPsWithMirroredText = false;
Highlight the first Pin by filling the pin
        bool HighlightFirstPin = false;
Fill components with component color and this opacity (0-255)
        int ComponentAlpha
Font Family for all Labels
        string FontName = "Arial";
Allow different colors per Pin (can be set via the API)

bool AllowColorPins = true;
Show the insert point for each component
        bool ShowInsertPoint = true;
Multiplicator for the Insert-Point Location (Component Property COMPONENT_INSERT_CENTER)
        double InsertPointUnitFactor = 1;
Pen Width in mils for components (0-5)
        int PenWidth

IFeatureParameters :

Draw a X into all drills (must be a Drill Layer)
        bool DrawXInDrills = true; //drawDrillX
Minimum size of drills in pixel to draw the X
        int DrawXMinDrillSizePX = 10; //in Pixel!
DrawMode for all objects (Filled/Outline/Both)
        DrawModeEnum DrawMode = DrawModeEnum.Filled;
Force "Outline" DrawMode for Surfaces
        bool DrawOnlySurfaceFrame = false;// ShowSurfacesFull = true;
Color for the outline, if DrawMode=Both (Standard=150/150/120/120)
        Color BothDrawModeOutlineColor = Color.FromArgb(150, 150, 120, 120);
Draws an additional Frame for colored objects in Fill-Mode at the end (=> Even if partly overdrawn
by other objects, the colored frame is stimm visible)
        bool DrawFrameForColoredObjects = true;
```

```
IInfoLayerParameters :

Info layer on/off
        bool ShowInfoLayer = false;
Display info overlay on lines
        bool ShowLineInfo = true;
Display info overlay on SMD Pads
        bool ShowSMDInfo = true;
Display info overlay on Drill Pads
        bool ShowPadInfo = true;
Display info overlay on Surfaces (only Size)
        bool ShowSurfaceInfo = true;
Draw additional frame for overlayed objects
        bool HighlightOutline = true;
Type of information for the overlay
        InfoLayerEnum InfoToShow = InfoLayerEnum.NETNAME;
Font Family for all Labels
        string FontName = "Verdana";
Unit MM or Mils for Size Label
        bool UnitMM = true;
Color of Lines
        Color LineColor = Color.FromArgb(220, 180, 255, 230);
Color of Line's outline
        Color OutLineColor = Color.FromArgb(220, 200, 255, 255);
Color of line label
        Color LineTextColor = Color.FromArgb(220, 220, 220, 200);
Color of Pads
        Color OutPadColor = Color.FromArgb(220, 255, 255, 200);
Color of pad label
        Color PadTextColor = Color.White;

ISelectionParameters :

Contrast for non-selected Objects if SelectionModeFeatures is DRAW_ONLY_SELECTION (0-255, 0=Hide)
        int ContrastFeatures
Contrast for non-selected Components if SelectionModeCMPs is DRAW_ONLY_SELECTION (0-255, 0=Hide)
        int ContrastComponents
Drawing mode for selected Objects
        SelectionModeEnum SelectionModeFeatures = SelectionModeEnum.DRAW_SELECTION_HIGHLIGHTED;
Drawing mode for selected Components
        SelectionModeEnum SelectionModeCMPs = SelectionModeEnum.DRAW_SELECTION_HIGHLIGHTED;
Show or Hide non-selected Surfaces if SelectionModeFeatures is DRAW_ONLY_SELECTION
        bool ShowTransparentSurfaces = false;
Color of the Selection if SelectionModeFeatures is not DRAW_ONLY_SELECTION
        Color SelectionColor
Hatch of Fill the Selection if SelectionModeFeatures is not DRAW_ONLY_SELECTION
        bool HatchSelection
Hatch Style of the Selection if HatchSelection=true and SelectionModeFeatures is not
DRAW_ONLY_SELECTION
        System.Drawing.Drawing2D.HatchStyle HatchStyleSelection
Selection color for components on the TOP Side
        Color ComponentTopSelectionColor = Color.Empty;
Selection color for components on the BOT Side
        Color ComponentBotSelectionColor = Color.Empty;
Selection color for component labels
        Color ComponentTextSelectionColor = Color.Empty;
```

```
ILayerParameters :
Settings for the info overlay of this layer
        IInfoLayerParameters InfoLayerParameters;
Color of this layer
        Color Color;
GET: Name of the Layer
        string LayerName
```

IEDA_PRP

```
The EDA property name.
        string NAME
This string value is the first entry of the PRP string or first number if the PRP string value was
empty.
        string VALUE
List of all numbers in this EDA property.
        List<double> Ns
Combine string and numbers to one "Value String". This starts with '' if the string part is empty,
all values are splitted with ' '.
        string ToValueString()
Set Value in correct spelling with 'xxx' (This trim empty spaces at the beginning and end).
        void SetValue
First two points of the Ns array interpretet as point with double values.
        MathUtils.PointD Location
```

IFilter

Copy of symbol from one layer to other layer.

static int AddToolDefinitionSpecial(Automation.IODBLayer LayerDst, Automation.IODBLayer LayerSrc, int ShapeIndexOnSrcLayer)

Creates a new Special Symbol and returns the toolNr. for the wanted layer

static int AddToolDefinitionSpecial(Automation.IODBLayer Layer, IPCBIWindow pcbi, string cadConformName, List<IODBObject> ObjectList, double offsetX, double offsetY, out IFilter.ToolDefinition outToolDefintion)

Add Symbol to layer, this works only if the shape index is ok!

static int AddToolDefinitionToLayer(IODBLayer ParentLayer,ToolDefinition SymbolToAdd)

IMath

Calculates the Center for a given Start/End Point, Angle and direction

static PointD CalculateCenter(PointD Begin, PointD End, double absAngle, bool cw)

IMatrix

List of all layers names with option to set names to lower.
List<string> GetAllLayerNames(bool NamesToLower)

Create a list of all layernames with types. This put all layernames in the keys of the dictionary and the matrix layer types are the values of the dictionary.
Dictionary<string, MatrixLayerType> GetAllLayerNamesAndTypes(bool NamesToLower)

Delete a layer.
bool DelateLayer(string Layername, bool Save, bool AddUndoItem)

IPackageSpecifics

Simple package outline, this is only one big polygon or the containing rectangle if no "good" rectangle is available.
IPolyClass GetSimplePackageOutline(bool includePins, double flattenValue=0)

List of pin polygons.
List<IPolyClass> GetPinPolys(double flattenValue = 0)

IODBLayer

Create a list of Lines in structure of the Text.
E.G. a X contains two lines, one from left down to right up and one from left up to right down.
Attention: The Text is not added to the layer, you have to SetSpecifics() of each line to add them to this layer!
List<IODBObject> GetTextLines(string Text, MathUtils.RectangleD BoundsForText)

Works only with extra licence!
Creates a high resolution Bitmap from the clipping Rectangle and save it.
bool AOIHighResolutionBMPExport(string FullPath, RectangleF ClippingRectangle, bool AntiAlias, int DPI, bool drawOnlySelected)

Removes negative objects on the layer by polygonizing cutted objects, so that the result looks same, but without negatives objects.
bool PolygonizeCuttedObjectsOnLayer(bool OnlySelected)

Returns true if layer has an image overlay
override bool HasOverlay()

Returns the Overlay color at a certain position
override Color GetOverlayColor(double worldX, double worldY)

IPCBIWindow

Lists Gerber, Excellon and Sieb and Meyer Files in the CAM input.
void OpenCAMInputAndSetListOfFiles(List<string> ListFileLocations)

Open CAM input and lists all Files on the window.
void OpenCAMInputWithFileFromDirectory(string GerberDirectoryPath)

Generate Net List with option to override existing nets and add components.
void GenerateNetList(bool AskForOverwriteOldNetNr = false)

Load zip data from stream e.g. use MemoryStream wit zip as byte[] included.
You can use folowing code to try it out:

```
using (FileStream dataStream = new FileStream(@"D:\Daten\odb.zip", FileMode.Open))
{
        MemoryStream memStream = new MemoryStream();
        await dataStream.CopyToAsync(memStream);
        memStream.Seek(0, SeekOrigin.Begin);
        bool res = await window.LoadDataFromZipStreamAsync(memStream);
}
```

async System.Threading.Tasks.Task<bool> LoadDataFromZipStreamAsync(Stream DataStream)

Load data async from all available formats (depending on your license).
async System.Threading.Tasks.Task<bool> LoadDataAsync(string FullPath)

Opens the Job Library Dialog. If the "regID" is a registered Commend of an "IPluginOpenDesign" plugin, the corresponding Tabpage of the "IPluginOpenDesign" is initially shown.
void OpenJobLibrary(int regID)

Set or get the internal bool for asking user to save the design changes at closing project.
This is only important if changes are made, some smaller changes do not set this bool. Most changes like delete of objects or add new objects set this bool immidiatly.
bool NeedToSave

Delegate for job open request.
delegate void JobOpenRequest(string Path, string StepName);

Delegate for property changing or adding to IObject Parent.
delegate void PropertyChangedEventHandler(Automation.IEDA_PRP Property, IObject Parent);

Occurs when a property is added or changed to the parent IObject.
event PropertyChangedEventHandler PCBIPropertyAddedOrChanged;

Delegate for attribute changing or adding to IObject Parent.
delegate void AttributeChangedEventHandler(Automation.IAttributeElement Attribute, IObject Parent);

Change enable for action item.
void CallHostChangeEnable(ID_ActionItem id, bool enable)

Change button check type.
void CallHostChangeButton(ID_ActionItem id, bool check)

Returns the Preview and Icon Image of the currently loaded design
bool GetJobPreviewImages(out Bitmap Preview200px, out Bitmap Icon32px)

Draw only outline of surfaces or fill them.
(Fill is default, reset after loading new project to fill(true))
bool ShowFilledSurfaces

The unit factor for calculating the package insert point.
The insert point is most in the middle of the package. You see the insert point marked with a small yellow circle inside of the package outline.
e.g. 39.37f = mm; 1 = mils
double ComponentInsertPointUnitFactor

The first pin is drawn with transparent color, if you want to turn off this set the value to false.
This is saved in options and PCB-Investigator remembers last setting.
Value can only be used after creating an IPCBIWindow instance.
bool HighlightFirstPinOfCMPs

Surfaces in transparent mode are hidden by default; you can change this by using ShowTransparentSurfaces or ask for the value if it is changed.
bool ShowTransparentSurfaces

The inner color transpary of CMPs (alpha level between 0 and 255).
int CMP_TransparyLevel

Drill Tool Mixed Color setting, change for easy finding of drills.
Value can only be used after creating an IPCBIWindow instance.
bool DrillMixedColor

IPin

GetIPinPosition returns the global coordinates on the layer (worldcoordinates), with translation (mirrored, etc.)
PCBI.MathUtils.PointD GetIPinPositionD(ICMPObject IcmpObject)

Create a list of lines and arc who build the pin outline.
List<IObjectSpecificsD> GetOutlineD(ICMPObject ParentCMP)

Gets the INetObject for this Pin/Component
INetObject GetNetObject(ICMPObject component)

IPolygon

Create IPolyClass from Rectangle.
static IPolyClass FromRectangle(RectangleD Rect)

Draws the Polygon on the given Graphics by using the current World-Client transformation of the IP-CBIWindow.
void Draw(System.Drawing.Graphics g, System.Drawing.Pen pen, IPCBIWindow pcbi)

ITextSpecificsD

Get IPolyClass from this polygon.
PCBI.MathUtils.IPolyClass GetPolygonOutline()

ISurfaceSpecificsD

Count of edge elements of the current polygon (this is only the newest one after start Polygon method).
int CurrentElementCount

IPackageSpecificsD

Count of all pins of this package.
int PinCount

Remove all pins of this package and clear outline polygon.
Be carefull with this method, all old pin information are lost after "RemovePinList".
void RemovePinList()

Replace exiting pin on pin position "PinNumber", with ellipse pin definition.
void ReplacePinEllipse(int PinNumber, PointD PinPostion, float diameter, string PackagePinName)

Replace existing pin with an rectangle pin definition.
void ReplacePinRectangle(int PinNumber, RectangleD Rect, string PackagePinName)

Replace a pin of this package by new pin definition as IPolyClass.
void ReplacePinContour(int PinNumber, IPolyClass PinOutline, PointD Position, string PackagePin-Name)

Creates outline as IPolyClass without extra elements (only main body or biggest closed part).
PCBI.MathUtils.IPolyClass GetSimplePolygonOutlineWithoutPins()

Bounds of the package outline.
RectangleD GetBounds()

Override the standard hash code with internal hash code calculation.
override int GetHashCode()

The HashCode of this package.
int HashCode

Override the standard quals method to find internal objects correct.
override bool Equals(object obj)

IStep

Asynchronius loading of layer. This can be dangoures! Please use only if you have experiance with async methods.
async System.Threading.Tasks.Task<ILayer> GetLayerAsync(string LayerName)

Creates a unioned Polygon of all Profile Objects
MathUtils.IPolyClass GetPCBOutlinePoly()

Get the step header origin point.
This value is only used for step and repeat positions (has no effect for standard root steps).
PCBI.MathUtils.PointD GetOriginStepHDR()

Gets a bitmap for the chosen rectangle area, expands the choosen area if it is too small.
Bitmap GetBitmap(PCBI.Automation.DrawingParameters.IDrawingParameters setting, RectangleF DetailRectangle, int Width, int Height, out RectangleF DrawnRectangle)

Select all objects of the choosen net.
void SelectNetElements(string NetName)

A seperate licence is necessary to use AOI!
Creates a Bitmap from all layer in LayerList.
bool AOIHighResolutionBMPExport(List<IODBLayer> LayerList, List<bool> ColorList, string FullPath, RectangleF ClippingRectangle, bool AntiAlias, bool DrawProfil, int DPI, AOIMatrixSize InternalMatrixSize, bool UseMultyThreading, bool InvertImageColors, out int imageSizeModification, bool drawOnlySelected, bool DrawSurfaceFrame, PCBI.Automation.IPCBIWindow.ProgressChanged onProgressChanged = null)

A seperate licence is necessary to use AOI!
Creates a TIF from all layer in LayerList.
bool AOIHighResolutionTIFExport(List<IODBLayer> LayerList, List<bool> ColorList, string FullPath, MathUtils.RectangleD ClippingRectangle, int DPI, AOIMatrixSize InternalMatrixSize, int MaxThreadCount, bool drawOnlySelected, bool InvertImageColors, int TifTagPhotometric = 1, int TifTagOrientation = 4, PCBI.Automation.IPCBIWindow.ProgressChanged onProgressChanged = null)

A seperate licence is necessary to use AOI!
Fills the MemoryStream with rgb image data (3 bytes/pixel). Layer-Color is used.
bool AOIHighResolutionMemoryExport(ref MemoryStream ms, IODBLayer Layer, MathUtils.RectangleD ClippingRectangle, int DPI, int OverSampling, bool IgnoreOverSampling4LinesArc, AOIMatrixSize InternalMatrixSize, bool UseMultiThreading, bool drawOnlySelected, Color BackColor, bool AllowSpecialObjectColor, out int widthPx, out int heightPx, PCBI.Automation.IPCBIWindow.ProgressChanged onProgressChanged = null)

Remove package from current ODB++ structure (eda file of this step).
It's important that no component use the relevant package.
This method use the selected package name to identify components on the PCB for usage.
bool RemovePackage(IPackageSpecificsD package)

This method checks whether a feature file is existing and check the size for 0 KB. If there is a file with some information (not necessary elements on the layer) it returns false.
bool IsLayerFileEmptyOrMissing(string Layername)

Remove net list from current step.

void RemoveNetList()

Updates the component pin net information according to the copper net information.

void UpdateNetComponentConnection()