

# Update IAutomation Interface PCB - Investigator 6.0

---

## Action:

Get all registered items including internal items. You have plugins at the end and all ID\_ActionItems at the beginning.

```
Dictionary<string, PCBI.Plugin.Interfaces.IRegisterItem> GetRegisteredItems()
```

Get List of new Commands to find updates easily. This is special by first start with new plug-ins.

```
List<PCBI.Plugin.Interfaces.IRegisterItem> GetNewCommands()
```

Get Action ID for the keys.

```
ID_ActionItem GetIDActionItem(System.Windows.Forms.Keys keys)
```

PCB-Investigator has a list of standard key settings, you can ask the standard key for a ID\_ActionItem with this method.

```
System.Windows.Forms.Keys GetStandardKeyFor(ID_ActionItem id)
```

Active layer of layer list will be set to a color from color dialog.

```
ID_SET_COLOR_TO_ACTIVE_LAYER
```

Open favorites list.

```
ID_IMPORT_FAVORITES
```

Open recent design dialog to select from last opened designs.

```
ID_IMPORT_RECENT
```

Open browse dialog to select design in folder tree view.

```
ID_IMPORT_BROWSE
```

toggle between metric and mils

```
ID_CHANGE_UNIT_To_MM
```

toggle between metric and mils

ID\_CHANGE\_UNIT\_To\_MILS

Toggles Strokes on/off

ID\_TOGGLE\_STROKES

Clear PCB outline from active step.

ID\_CLEAR\_OUTLINE

Calculate bounds of the current step.

ID\_GET\_PROFILE\_BOUNDS

Combine all elements on the active layer and set them to the PCB outline.

ID\_ADD\_ACTIVE\_LAYER\_TO\_

Create new layer with elements of the PCB outline.

ID\_CREATE\_LAYER\_FROM\_OUTLINE

Open dialog to create outline by mouse clicks.

ID\_SET\_OUTLINE\_BY\_LINES

Set outline by calculating the bounds around of all elements.

ID\_SET\_OUTLINE\_BY\_ALL\_LAYER\_OBJECTS

Open dialog to check whether nets are connected to components and select them if there are nets.

ID\_SELECT\_ALL\_NETS\_OF\_COMPONENT

Polygonize selection connect all selected elements by intersecting points and make one or more surfaces of it.

ID\_POLYGONIZE\_SELECTION

Flatten of step and repeat data, this contains all sub steps to put them on a new step with all data in one step.

ID\_DO\_FLATTEN\_STEP\_AND\_REPEAT

Add selection to outline, this combine the selected elements and add them to the PCB outline.

ID\_ADD\_TO\_OUTLINE\_FROM\_SELECTION

Show size on info layer.

ID\_INFO\_LAYER\_SIZE

Show netname on info layer.

ID\_INFO\_LAYER\_

Show freetext on info layer.

ID\_INFO\_LAYER\_FREETEXT

Show info on lines.

ID\_INFO\_LAYER\_LINES

Show info on pads.

ID\_INFO\_LAYER\_PADS

Show info on SMDs.

ID\_INFO\_LAYER\_SMDS

Show info on surfaces.

ID\_INFO\_LAYER\_SURFACE

Show info layer for all active layers.

ID\_INFO\_LAYER\_ALLLAYERS

Show highlighting off info objects in draw only selected mode.

ID\_INFO\_LAYER\_OUTLINE\_ONLY\_SELECTED

Deactivate info layer.

ID\_INFO\_LAYER\_OFF

## Automation:

Special PCB-Investigator license check of registered plugin, this is with license key for checking your own handled key.

Internal key can be checked with `GetAddInLicence(PlugInName)`.

```
bool CheckAddInLicence(string PlugInName, string LicenceKey)
```

Set the language for PCB-Investigator e.g. English.

```
void SetLanguagePackage(string Language)
```

Some ODB++ output tool use more than one component files e.g. components2, in some cases they are the newest version in other the lowest number is newest version.

This depends on the output tool and with this option it is possible to select highest number of components file or lowest.

```
bool ComponentFileUseHighest
```

## ICMPLayer:

Returns all objects in the specified rectangle Area.

List<IObject> GetAllObjectInRectangle(PCBI.MathUtils.RectangleD Area)

Is top component layer or bottom layer?

On bottom layer all components are mirrored in X direction, because user see them from false direction.

bool IsTopComponentLayer()

## ICMPObject:

Returns true if the SecondObject have a Point in this object.

bool IsPointOfSecondObjectIncluded(IODBObject SecondObject, bool IncludePins)

Rotation of the component.

float Rotation

Height of the component

double CompHEIGHT

[read only] Toleranz of the components value.

string Toleranz

[read only] Bounds in mils of board location.

PCBI.MathUtils.RectangleD Bounds

[read only] If available is a list of variants of the component.

ArrayList Variants

[read only] Count of different variants.

int VariantCount

[read only] Mirror in X direction of the component.

bool MirrorX

[read only] Mirror in Y direction of the component.

bool MirrorY

Tag of the component, can be filled with any object.

object Tag

Create an image of the component.

Bitmap GetImageOfCMP(Size ImageSize)

## IFilter:

Create a empty package

IPackageSpecificsD CreatePackageD(string PackageName)

## IMath:

Check the lines (Line1 and Line2) are parallel?

bool LinesAreParallel(PointF StartLine1, PointF EndLine1, PointF StartLine2, PointF EndLine2, double errLevel)

Calculates the distance between a line and a point (PointDistance).

double DistancePointToLine(PointF PointDistance, PointF LineP1, PointF LineP2, out PointD DestinationPoint)

## IMatrix:

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()
```

## IObject:

Create IPolyClass object from outline.

```
IPolyClass GetPolyClassOutline()
```

## IODBLayer:

Returns all objects in the specified rectangle Area.

```
List<IObject> GetAllObjectInRectangle(PCBI.MathUtils.RectangleD Area)
```

Create IPolyClass outline form shape index.

```
PCBI.MathUtils.IPolyClass GetPolygonOutlineFromShapeIndex(int shapeIndex)
```

Returns the ODBString for the ShapeIndex or Empty String

```
string GetSymbolNameFromShapeIndex(int ShapeIndex)
```

## IODBObject:

This is the global net number of current feature (from graphically connection definition). Default value is -1.

```
int GlobalNetNumber
```

Gives back the Tye of the object.

```
IObjectType Type
```

Get or set the object color.

Default added color action to undo buffer, for ignoring undo buffer use ObjectColorTemporary.

Color ObjectColor

[read only] Is the Object selected in PCB-Investigator.

bool IsSelected

[Free Text field filled from different tools with different information. This can be modified via Automation Interface from scripts and own software components.

string FreeText

Space to add optional objects.

object Tag

Polarity of feature, this can be Positive as default and negative if the feature "cut out" parts of the PCB.

bool Positive

[read only] Bounds in mils of board location.

MathUtils.RectangleD Bounds

Set the component color temporary, without event for need save changes.

Caution: If you save while the object exists is the color added!

void ObjectColorTemporary(Color ObjectColor)

Set a string attribute to the IODBObject.

SetAttribute(string Value)

Set a new attribute for this IODBObject with definition for type and value.

It can be a double, int, bool, enum (string of value) or string attribute.

SetAttribute(FeatureAttributeEnum FeatureAttribute, string Value, bool AddUndo = false)

Set a new attribute for this IODBObject with definition for type and value.

It can be a double, int, bool, enum (string of value) or string attribute.

```
void SetAttribute(FeatureAttributeEnum FeatureAttribute, object Value, bool AddUndo = false)
```

## enum LoadInformation (new element)

With the current license no right to open the selected format.

NoRights

## IPCBIWindow:

Occurs when elements are deleted from active project data.

```
event EventHandler PCBIElementsDeleted;
```

Informs about the progress for certain operations

```
delegate void ProgressChanged(int percent);
```

Occurs if user click button close all windows

```
event EventHandler PCBICloseAllWindows;
```

Get information of job, details of selection, components and repeats.

```
void GetJobInfo(out int CountSelectedSignalObjects, out int CountSelectedCMPs, out bool HasComponentLayers, out bool HasStepAndRepeat)
```

Show pin information on components or not.

```
void ActivatePinSpecialInfo()
```

Get or set special pin info for components.

This overwrites UseGeometryPinNames and set it instead of.

```
bool ShowPinSpecialInfo
```



## IPin:

Set special text to pin, this must be activated in component view setup or IPCBIWindow.ActivatePinSpecialInfo().

```
string PinSpecialInfo
```

Gets the position of the pin on the component without translation etc.

```
PCBI.MathUtils.PointD GetIPinPositionGeometryD()
```

## IPolygon:

Compares two polygons.

```
bool Equals(object obj)
```

Compares two polygons.

```
bool isEqualToTol(object obj)
```

Gets the HashCode.

```
int GetHashCode()
```

Determines whether the IPolyClass is null.

```
bool EqualsNull(object obj)
```

override the operator == to compare two IPolyClass

```
bool operator ==(IPolyClass a, IPolyClass b)
```

override the operator !=

```
bool operator !=(IPolyClass a, IPolyClass b)
```

If there are double edges in the polygon outline, they are removed. The original polygon will be changed!

Some other checks and operations need "clean" polygons without double edges, if necessary call this method first.

```
void RemoveDoubleEdges()
```

If the SecondObject intersect with this Object it returns true.

```
bool DoesIntersect(IPolyClass SecondObject)
```

If the SecondObject have a Point in this object it returns true;

```
bool IsPointOfSecondObjectIncluded(IPolyClass SecondObject)
```

## ISpecificsD:

All changed to Properties.

## IPackageSpecificsD :

To use AddPolygonPin you need a surface to work.

```
ISurfaceSpecificsD CreatePolygonPinSurfaceD()
```

Get the GraphicsPath of the package.

```
GraphicsPath GetGraphicsPath()
```

Get the GraphicsPath of the Pins of the package.

```
GraphicsPath GetGraphicsPathPins()
```

Create a list of pins in this package.

```
List<IPolyClass> GetPinPolys()
```

Get a list of all pin for this package.

```
List<IPin> GetGeometryPinList()
```

Creates outline as IPolyClass.

```
PCBI.MathUtils.IPolyClass GetPolygonOutlineWithoutPins()
```

Creates a list of all outline elements.

```
List<IODBObject> GetOutlineList()
```

New class **PinPadInfo**

Index of the used tool.

int ShapeIndex

The location of the pinpad.

PointD Location

### IStep:

Get the step bounds including all substeps.

```
MathUtils.RectangleD GetBoundsD()
```

Gets a bitmap for the chosen rectangle area, expands the chosen area if it is too small.

```
Bitmap GetBitmap(List<ILayer> Layers, RectangleF DetailRectangle, int Width, int Height, bool DrawPCBOutline, bool FillBoardOutline, bool ShowComponentDetails, bool IgnoreSelection)
```

A separate 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 DrawProfile, int DPI, AOIMatrixSize InternalMatrixSize, bool UseMultithreading, bool InvertImageColors, out int imageSizeModification, bool DrawSurfaceFrame = false, PCB.Automation.IPCBIWindow.ProgressChanged onProgressChanged = null)
```

A separate 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 InvertImageColors, int TifTagPhotometric = 1, int TifTagOrientation = 4, PCB.Automation.IPCBIWindow.ProgressChanged onProgressChanged = null)
```

## PCBStyleLibrary:

Many new Icons and Images