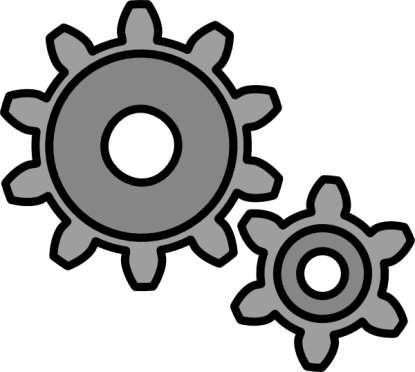
Vectric Lua Interface for Gadgets

The Script interface and associated documenta on is designed for people who are familiar with programming and is provided as a development tool for such people. It is extremely important to understand that Vectric cannot offer individual support for people wan ng to write scripts. This documenta on and the associated example scripts available from the Vectric web site are the only way you can learn about the script interface and develop your own gadgets. The appropriate area on the Vectric forum may also be useful for asking ques ons of other users.



Version 11.0 18th October 2021, Copyright Vectric Ltd.

# Contents

Introduc on...........................................................................................................................................35

Global Methods......................................................................................................................................35 IsAspire()........................................................................................................................................35 IsBetaBuild()...................................................................................................................................35 GetAppVersion()............................................................................................................................35

GetBuildVersion()...........................................................................................................................35

MessageBox (string text)...............................................................................................................35

Job related global methods...............................................................................................................36 CloseCurrentJob ()..........................................................................................................................36

CreateNewJob(string name, Box2D bounds, double thickness, bool in\_mm, bool

origin\_on\_surface).........................................................................................................................36

CreateNew2SidedJob(string name, Box2D bounds, double thickness, bool in\_mm, bool

origin\_on\_surface, SideFlipDirec on flip\_direc on).....................................................................36

CreateNewRotaryJob(string name, double length, double diameter, MaterialBlock.XYOrigin xy\_origin, bool in\_mm, bool origin\_on\_surface, bool wrapped\_along\_x\_axis)...........................36

OpenExis ngJob(string pathname)................................................................................................37

SaveCurrentJob()............................................................................................................................37

Vector object related global methods...............................................................................................37 CastCadObjectToCadBitmap(CadObject obj)................................................................................37 CastCadObjectToCadContour(CadObject obj)...............................................................................37 CastCadObjectToCadObjectGroup(CadObject obj).......................................................................37 CastCadObjectToCadPolyline(CadObject obj)...............................................................................37 CastCadObjectToCadToolpathOutline(CadObject obj)..................................................................37 CastCadObjectToCadToolpathPreview(CadObject obj).................................................................37

CastCadObjectToTxtBlock(CadObject obj).....................................................................................37 CreateCircle(double x, double y, double radius, double tolerance, double z\_value)...................38

CreateCopyOfSelectedContours(bool smash\_beziers, bool smash\_arcs, double smash\_tol)......38

Component Related Global Methods – Aspire only..........................................................................38 IsTransparent(float value)..............................................................................................................38

GetTransparentHeight().................................................................................................................38

CastComponentToComponentGroup(Component component)...................................................38

DocumentVariable Related Global Methods.....................................................................................38 IsInvalidDocumentVariableName(string name)............................................................................38 Data file loca ons related global methods........................................................................................38 GetDataLoca on()..........................................................................................................................38 GetPostProcessorLoca on()..........................................................................................................39 GetToolDatabaseLoca on()...........................................................................................................39 GetGadgetsLoca on()....................................................................................................................39 GetToolpathDefaultsLoca on().....................................................................................................39 GetBitmapTexturesLoca on()........................................................................................................39 GetVectorTexturesLoca on().........................................................................................................39 CreateCadContour (Contour ctr)...................................................................................................40 CreateCadGroup(ContourGroup ctr\_group)..................................................................................40

GetDefaultContourTolerance ().....................................................................................................40

High Level Objects..................................................................................................................................40

MaterialBlock.....................................................................................................................................40 MaterialBlock() - Constructor........................................................................................................40 . ActualXYOrigin.............................................................................................................................40 . BlcOrigin.......................................................................................................................................40 .Height............................................................................................................................................41 .InMM.............................................................................................................................................41 .MaterialBox...................................................................................................................................41 .MaterialSurfaceZ...........................................................................................................................41 .Thickness.......................................................................................................................................41 .Width.............................................................................................................................................41 .CylinderDiameter..........................................................................................................................41 .CylinderLength..............................................................................................................................41 .XYOrigin.........................................................................................................................................41 .ZOrigin...........................................................................................................................................41 .FlipDirec on..................................................................................................................................41 .JobType.........................................................................................................................................42 .Rota onAxis..................................................................................................................................42 :CalcAbsoluteZ(double z\_value).....................................................................................................42 :CalcAbsoluteZFromDepth (double z\_value).................................................................................42

: CalcDepthFromAbsoluteZ (double z\_value)................................................................................42 :SideZOrigin(UUID side\_id)............................................................................................................42 :SideMaterialSurfaceZ(UUID side\_id)............................................................................................42 :SideMaterialBox(UUID side\_id)....................................................................................................42 :SideBlcOrigin(UUID side\_id).........................................................................................................43 :CalcSideAbsoluteZ(UUID side\_id).................................................................................................43 :CalcSideDepthFromAbsoluteZ(UUID side\_id)...............................................................................43

:CalcSideAbsoluteZFromDepth(UUID side\_id)...............................................................................43

VectricJob...........................................................................................................................................45 .Exists.............................................................................................................................................45 .DocumentVariables.......................................................................................................................45 .Height............................................................................................................................................45 .IsAspire..........................................................................................................................................45 .IsBetaBuild....................................................................................................................................45 .InInches.........................................................................................................................................45 .InMM.............................................................................................................................................45 .Job Parameters.............................................................................................................................45 .LayerManager...............................................................................................................................45 .MinX..............................................................................................................................................45 .MinY..............................................................................................................................................45 .Name.............................................................................................................................................45 .PostProcessorParameters.............................................................................................................45 .Selec on........................................................................................................................................46 .SheetManager...............................................................................................................................46 .Width.............................................................................................................................................46 .XLength.........................................................................................................................................46 .YLength..........................................................................................................................................46 :ClearClipboard()............................................................................................................................46 :CreateHorizontalGuide(double y\_value, bool locked).................................................................46 :CopyToClipboard(string text)........................................................................................................46 :CreateVer calGuide(double x\_value, bool locked)......................................................................46 :DisplayToolpathPreviewForm()....................................................................................................46 :ExportSelec onToEps(string pathname)......................................................................................46 :ExportSelec onToDxf(string pathname)......................................................................................46 :ExportSelec onToSvg(string pathname)......................................................................................47 :ExportDocumentVariables(string pathname)...............................................................................47 :GetBuildVersion()..........................................................................................................................47 :GetBounds()..................................................................................................................................47 :GetAppVersion()...........................................................................................................................47 :GetWrapDiameter()......................................................................................................................47 :GroupSelec on()...........................................................................................................................47 :GuidesVisible()..............................................................................................................................47 :ImportBitmap(string pathname)...................................................................................................47 :ImportDxfDwg(string pathname)..................................................................................................47 :ImportSTLDirect(string pathname)...............................................................................................47 :ImportSVG(string pathname)........................................................................................................47 :ImportDocumentVariables(string pathname)..............................................................................48 :IsWrappedModel()........................................................................................................................48 :MoveObjectToSheet(CadObject object, UUID sheet\_id).............................................................48 :Refresh2DView ()..........................................................................................................................48 :SelectAllVectors()..........................................................................................................................48 :SetGuidesVisible(bool visible).......................................................................................................48 :SetXY\_Origin(double origin, double x\_offset, double y\_offset)...................................................48 :WrappingXValues().......................................................................................................................48 :WrapZOnSurface()........................................................................................................................48 :SwitchAc veSide()........................................................................................................................48 :SetAc veSide(UUID side\_id).........................................................................................................48 :GetSideList()..................................................................................................................................48 :GetAc veSideId()..........................................................................................................................49 :GetSideName(UUID side\_id).........................................................................................................49 :CopySelec onToOppositeSide()...................................................................................................49 :MoveSelec onToOppositeSide()..................................................................................................49 :SideComponentManager(UUID side\_id)......................................................................................49 :IsSingleSided()...............................................................................................................................49

:IsDoubleSided().............................................................................................................................49

CadLayerManager..............................................................................................................................50 .Count.............................................................................................................................................50 .IsEmpty..........................................................................................................................................50 :AddLayer(CadLayer layer, int index).............................................................................................50 :FindLayerWithName(string layer\_name).....................................................................................50 :GetAc veLayer()...........................................................................................................................50 :GetAt(POSITION pos)....................................................................................................................50 :GetHeadPosi on().........................................................................................................................50 :GetLayerWithName(string layer\_name)......................................................................................50 :GetLayerWithId(UUID layer\_id)....................................................................................................50 :GetNext(POSITION pos)................................................................................................................51 :GetPrev(POSITION pos).................................................................................................................51 :GetTailPosi on()...........................................................................................................................51 :HaveDataOnSheets().....................................................................................................................51 :RemoveLayer(CadLayer layer)......................................................................................................51

:SetAc veLayer(CadLayer layer)....................................................................................................51

CadLayer.............................................................................................................................................53 .Colour............................................................................................................................................53 .Color..............................................................................................................................................53 .Count.............................................................................................................................................53 .IsBitmapLayer................................................................................................................................53 .IsEmpty..........................................................................................................................................53 .IsSystemLayer...............................................................................................................................53 .Locked...........................................................................................................................................53 .Name.............................................................................................................................................53 .Visible............................................................................................................................................53 .Id...................................................................................................................................................53 .RawId............................................................................................................................................53 :AddObject (CadObject object, bool on\_current\_sheet)...............................................................53 :Find (UUID id)................................................................................................................................54 :GetHeadPosi on().........................................................................................................................54 :GetTailPosi on()...........................................................................................................................54 :GetNext(POSITION pos)................................................................................................................54 :GetPrev(POSITION pos).................................................................................................................54 :GetAt(POSITION pos)....................................................................................................................54 :RemoveObject (CadObject object)...............................................................................................54 :RemoveAt (POSITION pos)............................................................................................................54 :SetColour(double red, double green, double blue)......................................................................54

:SetColor (double red, double green, double blue).......................................................................55

CadSheetManager..............................................................................................................................56 .Ac veSheetId................................................................................................................................56

.NumberOfSheets..........................................................................................................................56 :CreateSheets:(integer num\_sheets, UUID sheet\_id)....................................................................56 :CreateSheets:(integer num\_sheets, UUID sheet\_id, Box2D material\_bounds)...........................56

:GetSheetIds()................................................................................................................................56

:GetSheetName(UUID sheet\_id)....................................................................................................56

CadObject...........................................................................................................................................58 .ClassName.....................................................................................................................................58 .Id...................................................................................................................................................58 .IsBitmap........................................................................................................................................58 .IsSelected......................................................................................................................................58 .IsVisible.........................................................................................................................................58 .IsLocked.........................................................................................................................................58 .LayerId...........................................................................................................................................58 .RawId............................................................................................................................................58 .RawLayerId....................................................................................................................................59 .SheetId..........................................................................................................................................59 .SideId.............................................................................................................................................59 :CanTransform(integer flags).........................................................................................................59 :Clone ()..........................................................................................................................................59 :GetBoundingBox ()........................................................................................................................59 :GetContour ()................................................................................................................................59 :InvalidateBounds()........................................................................................................................59 :SetLayer (CadLayer layer).............................................................................................................59 :Transform(Matrix2D matrix).........................................................................................................60

:GetBool(string parameter\_name, bool default\_value, bool create\_if\_not\_exist)......................60 :GetDouble(string parameter\_name, double default\_value, bool create\_if\_not\_exist)..............60 :GetInt(string parameter\_name, integer default\_value, bool create\_if\_not\_exist).....................60 :ParameterExists(string parameter\_name, utParameterType type).............................................60 :SetBool(string parameter\_name, bool value)..............................................................................60 :SetDouble(string parameter\_name, double value)......................................................................60 :SetInt(string parameter\_name, integer value).............................................................................60

:SetString(string parameter\_name, string value)..........................................................................61 CadContour........................................................................................................................................62 :InsertToolpathAtTabAtPoint(Point2D point)................................................................................62

CadObjectGroup................................................................................................................................63 .Count.............................................................................................................................................63 .IsEmpty..........................................................................................................................................63 :GetAt(POSITION pos)....................................................................................................................63 :GetHead ().....................................................................................................................................63 :GetHeadPosi on().........................................................................................................................63 :GetNext(POSITION pos)................................................................................................................63 :GetPrev(POSITION pos).................................................................................................................63

:GetTailPosi on()...........................................................................................................................63

CadBitmap..........................................................................................................................................64 .Brightness......................................................................................................................................64

CastCadObjectToCadBitmap(CadObject object)...........................................................................64 CadPolyline.........................................................................................................................................65

CadMarker.........................................................................................................................................66 CadMarker(string text, Point2D pt, integer pixel\_size) - Constructor...........................................66

.Text................................................................................................................................................66

.Posi on.........................................................................................................................................66

:SetColor(double red, double green, double blue)........................................................................66 utParameterList.................................................................................................................................68

:GetDouble(string parameter\_name, double default\_value, bool create\_if\_not\_exist)..............68 :GetInt(string parameter\_name, integer default\_value, bool create\_if\_not\_exist).....................68 :ParameterExists(string parameter\_name, utParameterType type).............................................68 :SetBool(string parameter\_name, bool default\_value, bool create\_if\_not\_exist).......................68 :SetBool(string parameter\_name, bool value)..............................................................................68 :SetDouble(string parameter\_name, double value)......................................................................69

:SetInt(string parameter\_name, integer value).............................................................................69 :SetString(string parameter\_name, string default\_value, bool create\_if\_not\_exist)...................69

:SetString(string parameter\_name, string value)..........................................................................69 luaUUID..............................................................................................................................................70 luaUUID() - Constructor.................................................................................................................70 .IsEmpty..........................................................................................................................................70 .RawId............................................................................................................................................70 :MakeEmpty().................................................................................................................................70 :CreateNew()..................................................................................................................................70 :AsString().......................................................................................................................................70 :Set(string text\_id).........................................................................................................................70 :SetId(UUID raw\_id).......................................................................................................................70

:IsEqual(luaUUID id).......................................................................................................................70

UUID\_List...........................................................................................................................................71 UUID\_List() - Constructor...............................................................................................................71 Proper es.......................................................................................................................................71 .Count.............................................................................................................................................71 .IsEmpty..........................................................................................................................................71 :AddHead( UUID id)........................................................................................................................71 :AddTail( UUID id)..........................................................................................................................71 :GetAt(pos).....................................................................................................................................71 :GetHeadPosi on().........................................................................................................................71 :GetNext(POSITION pos)................................................................................................................71 :GetPrev(POSITION pos).................................................................................................................72 :GetTailPosi on()...........................................................................................................................72 :Find(UUID id).................................................................................................................................72 :FindIndex(UUID id)........................................................................................................................72 :InsertA er(POSITION pos, UUID id)..............................................................................................72 :InsertBefore(POSITION pos, UUID id)...........................................................................................72 :RemoveAll()...................................................................................................................................72 :RemoveAt(POSITION pos).............................................................................................................72 :RemoveHead()..............................................................................................................................72 :RemoveTail().................................................................................................................................72

:SetAt(POSITION pos, UUID id)......................................................................................................72

CadObjectList.....................................................................................................................................73 CadObjectList(bool owns\_objects) - Constructor..........................................................................73 .Count.............................................................................................................................................73 .IsEmpty..........................................................................................................................................73 :AddHead( CadObject obj).............................................................................................................73 :AddTail(CadObject obj).................................................................................................................73 :CanTransform(integer flags).........................................................................................................73 :GetAt(POSITION pos)....................................................................................................................73 :GetHeadPosi on().........................................................................................................................73 :GetNext(POSITION pos)................................................................................................................73 :GetPrev(POSITION pos).................................................................................................................74 :GetTailPosi on()...........................................................................................................................74 :RemoveHead()..............................................................................................................................74 :RemoveTail().................................................................................................................................74

:Transform(Matrix2D xform).........................................................................................................74

Selec onList.......................................................................................................................................75 .Count.............................................................................................................................................75

.IsEmpty..........................................................................................................................................75 :Add ( CadObject obj, bool add\_tail, bool group\_add)..................................................................75 :AddHead( CadObject obj).............................................................................................................75 :AddTail(CadObject obj).................................................................................................................75 :CanTransform(integer flags).........................................................................................................76 :CanSelect(CadObject obj).............................................................................................................76 :Clear()............................................................................................................................................76 :GetAt(POSITION pos)....................................................................................................................76 :GetBoundingBox().........................................................................................................................76 :GetHeadPosi on().........................................................................................................................76 :GetNext(POSITION pos)................................................................................................................76 :GetPrev(POSITION pos).................................................................................................................76 :GetTailPosi on()...........................................................................................................................77 :GroupSelec onFinished().............................................................................................................77 :RemoveOnLayer(UUID layer\_id)...................................................................................................77 :Remove( CadObject obj, bool group\_remove).............................................................................77 :RemoveHead()..............................................................................................................................77 :RemoveTail().................................................................................................................................77

:Transform(Matrix2D xform).........................................................................................................77

Crea ng Vectors From Script.................................................................................................................78

Contour..............................................................................................................................................79 Contour(tolerance) - Constructor..................................................................................................79 .Area...............................................................................................................................................79 .BoundingBox2D.............................................................................................................................79

.BoundingBox3D.............................................................................................................................79 .Count.............................................................................................................................................79 .ContainsArcs.................................................................................................................................79 .ContainsBeziers.............................................................................................................................79 .CentreOfGravity............................................................................................................................79 .EndPoint3D...................................................................................................................................79

.EndPoint2D...................................................................................................................................79 .IsClockwise....................................................................................................................................80 . IsAn Clockwise............................................................................................................................80 . IsCW.............................................................................................................................................80 . IsCCW...........................................................................................................................................80

.IsClosed.........................................................................................................................................80 .IsEmpty..........................................................................................................................................80 .IsOpen...........................................................................................................................................80 .IsSinglePoint..................................................................................................................................80 .Length...........................................................................................................................................80 .StartPoint3D..................................................................................................................................80

.StartPoint2D..................................................................................................................................80 .Tolerance......................................................................................................................................80 :AppendPoint(double x, double y).................................................................................................81 :AppendPoint(double x , double y, double z)................................................................................81 :AppendPoint(Point2D pt2d).........................................................................................................81

:AppendPoint(Point3D pt3d).........................................................................................................81 :ArcTo (Point2D end\_pt2d, Point2D center\_pt2d, boolccw).........................................................81 :ArcTo (Point2D end\_pt2d, bool bulge).........................................................................................81

:ArcTo( Point3D end\_pt3d, bool bulge).........................................................................................82 :AppendContour (Contour ctr).......................................................................................................82 :AppendSpan (Span span)..............................................................................................................82 :AppendSpanAsLines(Span span, double tolerance).....................................................................82 :BezierTo(Point2D end\_pt2d, Point2D ctrl\_1\_2d, Point2D ctrl\_2\_2d)..........................................82

:BezierTo(Point3D end\_pt3d, Point2D ctrl\_1\_2d, Point2D ctrl\_2\_2d)..........................................82 :CanAppendContour(Contour ctr).................................................................................................83 :CanAppendSpan (Span span)........................................................................................................83

:Clone()...........................................................................................................................................83 :CreateTolerancedCopy(double tolerance)...................................................................................83 :CreatePolygonizedCopy(double tolerance, double max\_line\_len)..............................................83

:GetFirstSpan()...............................................................................................................................83 :GetLastSpan()................................................................................................................................83 :GetHeadPosi on().........................................................................................................................83 :GetTailPosi on()...........................................................................................................................83 :GetNext(POSITION pos)................................................................................................................83 :GetPrev (POSITION pos)................................................................................................................84 :GetAt (POSITION pos)...................................................................................................................84 :IsPointInside (Point2D point, double tolerance)..........................................................................84 :InvalidateBoundingBox()...............................................................................................................84 :LineTo(double x , double y)...........................................................................................................84 :LineTo(double x , double y, double z)...........................................................................................84 :LineTo(Point2D end\_pt2d )...........................................................................................................85 :LineTo( Point3D end\_pt3d)...........................................................................................................85 :MakeOffsetsSquare(double offset\_dist, double offset\_out, double max\_dist)...........................85

:OffsetInZ(double z\_offset)............................................................................................................85 :ReorderStartPoint(Point2D point)................................................................................................85 :ReorderStartPoint (integer span\_index, double parameter).......................................................85

:Reverse ()......................................................................................................................................85 :SetZHeight(double z\_val)..............................................................................................................86 :Smash (double tolerance, bool preserve\_arcs)............................................................................86 :Transform ( Matrix2D xform)........................................................................................................86 :RemoveHead ().............................................................................................................................86

:RemoveTail ()................................................................................................................................86

ContourGroup....................................................................................................................................87 ContourGroup(bool owns\_contours) - Constructor......................................................................87 .BoundingBox2D.............................................................................................................................87

.BoundingBox3D.............................................................................................................................87 .Count.............................................................................................................................................87 .ClosedCount..................................................................................................................................87 .IsEmpty..........................................................................................................................................87 .OpenCount....................................................................................................................................87 .OwnsContours...............................................................................................................................87 :AppendGroup(ContourGroup group)...........................................................................................87 :AddHead(Contour ctr)..................................................................................................................87 :AddTail(Contour ctr).....................................................................................................................88 :Clone()...........................................................................................................................................88 :CreateTolerancedCopy(double tolerance)...................................................................................88 :ContainsBeziers()..........................................................................................................................88 :DeleteOpenVectors()....................................................................................................................88 :GetLength()...................................................................................................................................88 :GetHeadPosi on().........................................................................................................................88 :GetTailPosi on()...........................................................................................................................88 :GetNext(POSITION pos)................................................................................................................88 :GetPrev(POSITION pos).................................................................................................................89 :GetAt(POSITION pos)....................................................................................................................89 :GetHead()......................................................................................................................................90 :GetTail().........................................................................................................................................90

:MakeOffsetsSquare(double offset\_dist, bool offset\_out, double max\_dist)...............................90 :Offset(double dist, double stepover, integer num\_offsets, bool preserve\_arcs)........................90

:OffsetInZ(double dist)...................................................................................................................90

:OffsetWithOpenVectors(double dist, double stepover, integer num\_offsets, bool

preserve\_arcs)................................................................................................................................90 :RemoveAt(POSITION pos).............................................................................................................90 :RemoveHead()..............................................................................................................................91 :RemoveTail().................................................................................................................................91 :Reverse().......................................................................................................................................91 :SetZHeight(double z\_val)..............................................................................................................91 :Smash(double tolerance, bool preserve\_arc)...............................................................................91

:Transform(Matrix2D xform).........................................................................................................91

ContourCarriage.................................................................................................................................92 ContourCarriage(Contour ctr, Point2D pt)....................................................................................92 ContourCarriage(integer span\_index, double parameter)............................................................92

.IsInvalid.........................................................................................................................................92 .SpanIndex......................................................................................................................................92 .SpanParameter.............................................................................................................................92 :Move(Contour ctr, double distance)............................................................................................92 :Posi on(Contour ctr)....................................................................................................................93 :UpdatePosi on(Contour ctr, Point2D pt).....................................................................................93 ToolpathTab.......................................................................................................................................94

Span....................................................................................................................................................95 Span(Point3D pt3d) - Constructor.................................................................................................95 .EndPoint3D...................................................................................................................................95

.EndPoint2D...................................................................................................................................95 .IsLineType.....................................................................................................................................95 .IsArcType.......................................................................................................................................95 .IsBezierType..................................................................................................................................95 .IsLeadInType.................................................................................................................................95 .IsLeadOutType..............................................................................................................................95 .IsOvercutType...............................................................................................................................95 .NumberOfControlPoints...............................................................................................................95 .StartPoint3D..................................................................................................................................95

.StartPoint2D..................................................................................................................................96 .Type...............................................................................................................................................96 :ChordLength()...............................................................................................................................96 :EndVector(bool normalise)...........................................................................................................96

:GetControlPointPosi on(integer index).......................................................................................96 :GetLength(double tolerance).......................................................................................................96 :MoveControlPoint(integer index, Point2D pt).............................................................................96 :PointAtParameter(double param, double tolerance)..................................................................96

:Reverse().......................................................................................................................................97 :StartVector(bool normalise).........................................................................................................97 :SetStartPoint3D(Point3D pt).........................................................................................................97

:SetStartPoint2D(Point2D pt).........................................................................................................97 :SetEndPoint3D(Point3D pt)..........................................................................................................97

:SetEndPoint2D(Point2D pt)..........................................................................................................97

LineSpan.............................................................................................................................................98 LineSpan(Point3D start\_pt, Point3D end\_pt) - Constructor..........................................................98 LineSpan(Point2D start\_pt, Point2D end\_pt) - Constructor..........................................................98

:ClosestParameterToPoint(Point2D pt).........................................................................................98

ArcSpan..............................................................................................................................................99

ArcSpan(Point2D start\_pt, Point2D end\_pt, Point2D pt\_on\_arc) - Constructor...........................99 ArcSpan(Point2D start\_pt, Point2D end\_pt, Point2D centre\_pt, bool ccw) - Constructor...........99 ArcSpan(Point3D start\_pt, Point3D end\_pt, Point3D centre\_pt, bool ccw) - Constructor.........100 ArcSpan(Point3D start\_pt, Point3D end\_pt, Point2D pt\_on\_arc) - Constructor.........................100 ArcSpan(Point2D start\_pt, Point2D end\_pt, double bulge) - Constructor..................................100 ArcSpan(Point3D start\_pt, Point3D end\_pt, double bulge) - Constructor..................................101

.Bulge...........................................................................................................................................101 .IsAn Clockwise...........................................................................................................................101 .IsCCW..........................................................................................................................................101 .IsClockwise..................................................................................................................................101 .IsCW............................................................................................................................................101 :ArcMidPoint()..............................................................................................................................101 :RadiusAndCentre(Point3D ret\_centre\_pt).................................................................................101 :SetBulge(bulge)...........................................................................................................................101

ArcBulgeFromMidPoint(Point2D start, Point2D end, Point2D mid)............................................102

BezierSpan........................................................................................................................................103

BezierSpan(Point2D start\_pt, Point2D end\_pt, Point2D ctrl\_pt\_1, Point2D ctrl\_pt\_1) -

Constructor..................................................................................................................................103

BezierSpan(Point3D start\_pt, Point3D end\_pt, Point2D ctrl\_pt\_1, Point2D ctrl\_pt\_1) -

Constructor..................................................................................................................................103

Span Helper Methods......................................................................................................................104 CastSpanToLineSpan(Span span).................................................................................................104 CastSpanToArcSpan(Span span)..................................................................................................104

CastSpanToBezierSpan(Span span)..............................................................................................104

Low Level Geometry............................................................................................................................105

Point2D.............................................................................................................................................106 Point2D() - Constructor................................................................................................................106

Point2D(double x, double y) - Constructor..................................................................................106 Point2D(Point2D pt) - Constructor..............................................................................................106 .IsInvalid.......................................................................................................................................106 .X..................................................................................................................................................106 .x...................................................................................................................................................106 .Y...................................................................................................................................................106 .y...................................................................................................................................................106 :IsCoincident(Point2D point, double tol).....................................................................................106 :Set(double x, double y)...............................................................................................................106 :SetInvalid()..................................................................................................................................107 Matrix2D \* Point2D.....................................................................................................................107 Point2D - Point2D........................................................................................................................107 Point2D + Vector2D.....................................................................................................................107

Point2D - Vector2D......................................................................................................................107

Point3D.............................................................................................................................................108 Point3D() - Constructor................................................................................................................108 Point3D(double x, double y, double z) - Constructor..................................................................108 Point3D(Point3D pt) - Constructor..............................................................................................108 Point3D(Point2D pt, double z).....................................................................................................108 .X..................................................................................................................................................108 .x...................................................................................................................................................108 .Y...................................................................................................................................................108 .y...................................................................................................................................................108 .Z...................................................................................................................................................108 .z...................................................................................................................................................108 .IsInvalid.......................................................................................................................................108 :IsCoincident(Point3D point, double tol).....................................................................................109 :Set(double x, double y, double z)...............................................................................................109 :Set(Point2D pt, double z)............................................................................................................109 :SetInvalid()..................................................................................................................................109 Matrix2D \* Point3D.....................................................................................................................109 Point3D - Point3D........................................................................................................................109 Point3D + Vector3D.....................................................................................................................109

Point3D - Vector3D......................................................................................................................109

Vector2D..........................................................................................................................................110 Vector2D() - Constructor..............................................................................................................110 Vector2D(double x, double y) - Constructor................................................................................110 Vector2D(Vector2D vec) - Constructor........................................................................................110 .IsInvalid.......................................................................................................................................110 .Length.........................................................................................................................................110 .LengthSq.....................................................................................................................................110

.X..................................................................................................................................................110 .x...................................................................................................................................................110 .Y...................................................................................................................................................110 .y...................................................................................................................................................110 :Cross(Vector2D vec)....................................................................................................................111 :Dot(Vector2D vec)......................................................................................................................111 :Normalize().................................................................................................................................111 :NormalTo()..................................................................................................................................111 :Set(double x, double y)...............................................................................................................111 :SetInvalid()..................................................................................................................................111 Double \* Vector2D.......................................................................................................................111 Matrix2D \* Vector2D...................................................................................................................111 Vector2D + Vector2D...................................................................................................................111 Vector2D - Vector2D....................................................................................................................111

* Vector2D....................................................................................................................................111

Vector3D..........................................................................................................................................112 Vector3D() - Constructor..............................................................................................................112 Vector3D(double x, double y, double z) - Constructor................................................................112 Vector3D(Vector3D vec) - Constructor........................................................................................112 .IsInvalid.......................................................................................................................................112 .Length.........................................................................................................................................112 .LengthSq.....................................................................................................................................112 .X..................................................................................................................................................112 .x...................................................................................................................................................112 .Y...................................................................................................................................................112 .y...................................................................................................................................................112 .Z...................................................................................................................................................112 .z...................................................................................................................................................113 :Cross(Vector3D vec)....................................................................................................................113 :Dot(Vector3D vec)......................................................................................................................113 :Dot(Point3D vec).........................................................................................................................113 :Normalize().................................................................................................................................113 :Set(double x, double y, double z)...............................................................................................113 :SetInvalid()..................................................................................................................................113 Double \* Vector3D.......................................................................................................................113 Vector3D + Vector3D...................................................................................................................113 Vector3D - Vector3D....................................................................................................................113

* Vector3D....................................................................................................................................113

Box2D...............................................................................................................................................114 Box2D() - Constructor..................................................................................................................114 Box2D (Point2D p1, Point2D p2) - Constructor...........................................................................114 Box2D(Box2D box) - Constructor.................................................................................................114 .BLC...............................................................................................................................................114 .BRC..............................................................................................................................................114 .Centre..........................................................................................................................................114

.Center..........................................................................................................................................114 .IsInvalid.......................................................................................................................................114 .MinX............................................................................................................................................114 .MinY............................................................................................................................................114 .MaxX...........................................................................................................................................114 .MaxY...........................................................................................................................................114 .XLength.......................................................................................................................................114 .YLength.......................................................................................................................................115 .MaxLength..................................................................................................................................115 .MinLength...................................................................................................................................115 .TRC..............................................................................................................................................115 .TLC...............................................................................................................................................115 :Expand(double offset\_dist).........................................................................................................115 :IsInside(Point2D point, double tol).............................................................................................115 :IsInsideOrOn(Point2D point, double tol)....................................................................................115 :IsInside(Box2D box, bool on\_counts\_as\_int, double tol)...........................................................115 :Intersects(Box2D box, double tol)..............................................................................................115 :Merge(double x, double y).........................................................................................................115 :Merge(Point2D point).................................................................................................................116 :Merge(Box2D box)......................................................................................................................116

:SetInvalid()..................................................................................................................................116

Box3D...............................................................................................................................................117

.BLC...............................................................................................................................................117 .BRC..............................................................................................................................................117 .Centre..........................................................................................................................................117

.Center..........................................................................................................................................117 .IsInvalid.......................................................................................................................................117 .MaxLength..................................................................................................................................117 .MaxX...........................................................................................................................................117 .MaxY...........................................................................................................................................117 .MaxZ............................................................................................................................................117 .MinLength...................................................................................................................................117 .MinX............................................................................................................................................117 .MinY............................................................................................................................................117 .MinZ............................................................................................................................................117 .TRC..............................................................................................................................................117 .TLC...............................................................................................................................................117 .XLength.......................................................................................................................................118 .YLength.......................................................................................................................................118 .ZLength........................................................................................................................................118 :Expand(double offset\_dist).........................................................................................................118 :IsInside(Point3D point, double tol).............................................................................................118 :IsInsideOrOn(Point3D point, double tol)....................................................................................118 :IsInside(Box3D box, bool on\_counts\_as\_int, double tol)...........................................................118 :Intersects(Box3D box, double tol)..............................................................................................118 :Merge(double x, double y, double z)..........................................................................................118 :Merge(Point3D point).................................................................................................................118 :Merge(Box3D box)......................................................................................................................119

:SetInvalid()..................................................................................................................................119

Matrix2D..........................................................................................................................................120 Iden tyMatrix2D().......................................................................................................................120 Reflec onMatrix2D(Point2D p1, Point2D p1).............................................................................120 Rota onMatrix2D(Point2D rota on\_pt, double angle)..............................................................120 ScalingMatrix2D(Vector2D scale\_vec).........................................................................................120 ScalingMatrix2D(Point2D scale\_pt, Vector2D scale\_vec)............................................................120 Transla onMatrix2D(Vector2D vec)............................................................................................120

Matrix2D \* Matrix2D...................................................................................................................120 Toolpaths.............................................................................................................................................121 ToolpathManager............................................................................................................................122 ToolpathManager - constructor..................................................................................................122 .Count...........................................................................................................................................122 .IsEmpty........................................................................................................................................122 .NumVisibleToolpaths..................................................................................................................122 :AddExternalToolpath(ExternalToolpath toolpath).....................................................................122 :CreateProfilingToolpath(............................................................................................................122 CreatePocke ngToolpath(...........................................................................................................126 :CreateDrillingToolpath(..............................................................................................................130 :CreateVCarvingToolpath(............................................................................................................133 :CreatePrismCarvingToolpath(.....................................................................................................136 :CreateFlu ngToolpath(...............................................................................................................139 :CreateRoughingToolpath( - Aspire Only.....................................................................................142 :CreateFinishingToolpath( - Aspire Only......................................................................................144 :LoadToolpathTemplate(string template\_path)..........................................................................146 :CopyToolpathWithId(UUID id, bool insert\_at\_end, bool rename)............................................147

:DeleteAllToolpath ()....................................................................................................................147 :DeleteToolpath (Toolpath toolpath)...........................................................................................147

:DeleteToolpathWithId(UUID id).................................................................................................147 :Find(UUID id)...............................................................................................................................147

:GetAt(pos)...................................................................................................................................147 :GetGroupToolpathWithIndex(UUID id, integer index)...............................................................147 :GetNext(pos)...............................................................................................................................147 :GetNumberOfToolpathsInGroup(UUID id).................................................................................148 :GetPrev(pos)...............................................................................................................................148 :GetSelectedToolpath()................................................................................................................148 :GetHeadPosi on().......................................................................................................................148 :GetTailPosi on().........................................................................................................................148 :RecalculateAllToolpaths()...........................................................................................................148 :GetAc veSideNode()...................................................................................................................149 :RecalculateToolpath(Toolpath toolpath)...................................................................................149 :SaveToolpathAsTemplate(Toolpath toolpath, string template\_path).......................................149 :SaveVisibleToolpathsAsTemplate( string template\_path).........................................................149 :ToolpathModified(Toolpath toolpath).......................................................................................149 :ToolpathWithNameExists(string name)......................................................................................149 :UndrawAllToolpath ().................................................................................................................150

:SetAllToolpathsVisibility (bool visibility).....................................................................................150

Toolpath...........................................................................................................................................151 .FirstPoint.....................................................................................................................................151 .GroupId.......................................................................................................................................151 .Id.................................................................................................................................................151 .InMM...........................................................................................................................................151 .LastPoint.....................................................................................................................................151 .Name...........................................................................................................................................151 .Notes...........................................................................................................................................151

.Posi onData................................................................................................................................151 .Tool.............................................................................................................................................151 .SheetId........................................................................................................................................151 .SideId...........................................................................................................................................151 :MachiningTime().........................................................................................................................152 :ReplaceTool(Tool tool)................................................................................................................152 :Sta s cs()....................................................................................................................................152

:Transform(Matrix2D xform).......................................................................................................152

ToolpathNode..................................................................................................................................152 :HasChildren()..............................................................................................................................153 :GetNumberOfChildren().............................................................................................................153 :GetChild(int index)......................................................................................................................153 :GetToolpath()..............................................................................................................................153

:GetName()..................................................................................................................................153

ToolDatabase...................................................................................................................................153 ToolDatabase - constructor.........................................................................................................153 :SelectTool().................................................................................................................................154

:GetTool(ToolDBId tooldb\_id)......................................................................................................154

ToolDBId...........................................................................................................................................155 ToolDBId(string sec on, string key\_prefix) – constructor...........................................................155 SaveDefaults(string sec on, string key\_prefix)...........................................................................155

LoadDefaults(string sec on, string key\_prefix)...........................................................................155 Tool..................................................................................................................................................155 Tool(string name, ToolType tool\_type) - constructor.................................................................155 .ClearStepover..............................................................................................................................156 .InMM...........................................................................................................................................156 .FeedRate.....................................................................................................................................156 .Name...........................................................................................................................................156 .Notes...........................................................................................................................................156 .PlungeRate..................................................................................................................................156 .RateUnits.....................................................................................................................................156 .RateUnitsText..............................................................................................................................156 .SpindleSpeed...............................................................................................................................156 .Stepdown....................................................................................................................................156 .Stepover......................................................................................................................................156 .ToolDB\_Loca on.........................................................................................................................156 .ToolType......................................................................................................................................157 .ToolTypeText...............................................................................................................................157 .ToolDia........................................................................................................................................157 .ToolNumber................................................................................................................................157 .VBit\_Angle...................................................................................................................................157 .ToolDBId......................................................................................................................................157 :ConvertRateUnitsTo(integer new\_units)....................................................................................157 :IsCompa bleFeedRates(Tool tool\_to\_check).............................................................................157 :IsCompa bleTool(Tool tool\_to\_check)......................................................................................157 :IsCompa bleCutDepths(Tool tool\_to\_check)............................................................................158 :FlatRadius(bool in\_mm)..............................................................................................................158 :GetBool(string parameter\_name, bool default\_value)..............................................................158 :GetDouble(string parameter\_name, double default\_value,).....................................................158 :GetInt(string parameter\_name, integer default\_value)............................................................158 :ParameterExists(string parameter\_name, utParameterType type)...........................................158 :SetBool(string parameter\_name, bool value)............................................................................158 :SetDouble(string parameter\_name, double value)....................................................................158 :SetInt(string parameter\_name, integer value)...........................................................................159 :SetString(string parameter\_name, string value)........................................................................159

:UpdateParameters()...................................................................................................................159 ToolpathPosData..............................................................................................................................160 ToolpathPosData() - Constructor.................................................................................................160 .HomeX.........................................................................................................................................160 .HomeY.........................................................................................................................................160 .HomeZ.........................................................................................................................................160 .InMM...........................................................................................................................................160 .SafeZ............................................................................................................................................160 .SafeZGap.....................................................................................................................................160 .StartZGap....................................................................................................................................160

:EnsureHomeZIsSafe()..................................................................................................................160

:SetHomePosi on(double x, double y, double z)........................................................................160

GeometrySelector............................................................................................................................161 GeometrySelector() - Constructor...............................................................................................161

.AllowOpen...................................................................................................................................161 .AllowToolDia...............................................................................................................................161 .CircleDia......................................................................................................................................161 .CircleTolerance...........................................................................................................................161 .CircleMatchAll.............................................................................................................................161 .CircleMatchCircleDia...................................................................................................................161 .CircleMatchToolDia.....................................................................................................................161 .GeometryDepthOffset................................................................................................................161 .GeometryDepthOffset Formula..................................................................................................161 .MixedGroupsOk..........................................................................................................................161 .OnlyOnLayers..............................................................................................................................161 .SelectClosed................................................................................................................................161 .SelectCircles................................................................................................................................161 .SelectOpen..................................................................................................................................162 .SetDepthFromGeometry............................................................................................................162 .ToolDia........................................................................................................................................162 .HaveLayerNames........................................................................................................................162 .LayerNameCount........................................................................................................................162 :AddLayerhName(string name)....................................................................................................162 :GetAtLayerName(POSITION pos)................................................................................................162 :GetLayerNameHeadPosi on()....................................................................................................162 :GetNextLayerName(POSITION pos)............................................................................................162 :GetPrevLayerName(POSITION pos)............................................................................................162 :GetLayerNameTailPosi on().......................................................................................................162 :FindLayerWithName(string name).............................................................................................163 :HasSelectorData(Toolpath toolpath)..........................................................................................163 :LoadSelectorData(Toolpath toolpath)........................................................................................163 :SaveSelectorData(Toolpath toolpath)........................................................................................163

:RemoveAllLayerNames().............................................................................................................163

ProfileParameterData......................................................................................................................164 ProfileParameterData() - Constructor.........................................................................................164 .Allowance....................................................................................................................................164 .AllowanceFormula......................................................................................................................164 .CutDepth.....................................................................................................................................164 .CornerSharpen............................................................................................................................164 .CreateSquareCorners..................................................................................................................164 .CutDepthFormula.......................................................................................................................164 .CutDirec on................................................................................................................................164 .KeepStartPoints..........................................................................................................................164 .Name...........................................................................................................................................164 .ProfileSide...................................................................................................................................164 .ProjectToolpath..........................................................................................................................164 .StartDepth...................................................................................................................................165

.StartDepthFormula.....................................................................................................................165

RampingData....................................................................................................................................166 RampingData() - Constructor.......................................................................................................166 .DoRamping..................................................................................................................................166 .RampAngle..................................................................................................................................166 .RampConstraint..........................................................................................................................166 .RampDistance.............................................................................................................................166 .RampMaxAngleDist.....................................................................................................................166 .RampOnLeadIn............................................................................................................................166

.RampType...................................................................................................................................166

LeadInOutData.................................................................................................................................167 LeadInData() - Constructor..........................................................................................................167 .CircularLeadRadius......................................................................................................................167 .DoLeadIn.....................................................................................................................................167 .DoLeadOut..................................................................................................................................167 .LeadType.....................................................................................................................................167 .LeadLength..................................................................................................................................167 .LinearLeadAngle..........................................................................................................................167

.OvercutDistance..........................................................................................................................167

PocketParameterData......................................................................................................................168 ProfileParameterData() - Constructor.........................................................................................168 .Allowance....................................................................................................................................168 .AllowanceFormula......................................................................................................................168 .CutDepth.....................................................................................................................................168 .CutDepthFormula.......................................................................................................................168 .CutDirec on................................................................................................................................168 .DoRamping..................................................................................................................................168 .DoRasterClearance......................................................................................................................168 .Name...........................................................................................................................................168 .ProfilePassType...........................................................................................................................168 .ProjectToolpath..........................................................................................................................168 .RampDistance.............................................................................................................................169 .RasterAllowance.........................................................................................................................169 .RasterAngle.................................................................................................................................169 .StartDepth...................................................................................................................................169 .StartDepthFormula.....................................................................................................................169

.UseAreaClearTool.......................................................................................................................169

DrillParameterData..........................................................................................................................170 DrillParameterData() - Constructor.............................................................................................170 .CutDepth.....................................................................................................................................170 .CutDepthFormula.......................................................................................................................170 .DoPeckDrill..................................................................................................................................170 .Name...........................................................................................................................................170 .PeckRetractGap...........................................................................................................................170 .ProjectToolpath..........................................................................................................................170 .StartDepth...................................................................................................................................170 .StartDepthFormula.....................................................................................................................170 VCarveParameterData.....................................................................................................................172 VCarveParameterData() - Constructor........................................................................................172 .FlatDepth.....................................................................................................................................172 .FlatDepthFormula.......................................................................................................................172 .Name...........................................................................................................................................172 .ProjectToolpath..........................................................................................................................172 .StartDepth...................................................................................................................................172 .StartDepthFormula.....................................................................................................................172 .DoFlatBo om..............................................................................................................................172

.UseAreaClearTool.......................................................................................................................172

Flu ngParameterData......................................................................................................................173 Flu ngParameterData() - Constructor.........................................................................................173 .CutDepth.....................................................................................................................................173 .CutDepthFormula.......................................................................................................................173 .FluteType....................................................................................................................................173 .Name...........................................................................................................................................173 .ProjectToolpath..........................................................................................................................173 .RampLength................................................................................................................................173 .RampRa o...................................................................................................................................173 .RampType...................................................................................................................................173 .StartDepth...................................................................................................................................173 .StartDepthFormula.....................................................................................................................173 .UseRampRa o.............................................................................................................................173

.UseSelec onOrder......................................................................................................................173

PrismCarveParameterData..............................................................................................................174 PrismCarveParameterData() - Constructor..................................................................................174 .CutDepth.....................................................................................................................................174 .CutDepthFormula.......................................................................................................................174 .CutDirec on................................................................................................................................174 .Name...........................................................................................................................................174 .StartDepth...................................................................................................................................174 .StartDepthFormula.....................................................................................................................174

:CalculateMinimumBevelDepth(Tool tool, bool show\_warnings)...............................................174 RoughingParameterData – Aspire Only...........................................................................................176 RoughingParameterData() – Constructor....................................................................................176 .Name..........................................................................................................................................176

.StartDepth...................................................................................................................................176 .StartDepthFormula.....................................................................................................................176 .CutDepth.....................................................................................................................................176 .CutDepthFormula.......................................................................................................................176 .Allowance....................................................................................................................................176 .DoZLevelRoughing......................................................................................................................176 .ZLevelStrategy.............................................................................................................................176 .ZLevelProfile................................................................................................................................176 .RasterAngle.................................................................................................................................176

.MachiningAllowance...................................................................................................................176

ExternalToolpath..............................................................................................................................177 ExternalToolpath(........................................................................................................................177

.Error............................................................................................................................................177

ExternalToolpathOp ons.................................................................................................................178 ExternalToolpathOp ons() - Constructor....................................................................................178 .CreatePreview.............................................................................................................................178

.StartDepth...................................................................................................................................178

PostPInfo..........................................................................................................................................180 .Name...........................................................................................................................................180 .FileName.....................................................................................................................................180 .Extension.....................................................................................................................................180 .SupportsArcs...............................................................................................................................180 .SupportsToolchange...................................................................................................................180 .Wrap\_X\_Axis...............................................................................................................................180

.Wrap\_Y\_Axis...............................................................................................................................180

ToolpathSaver..................................................................................................................................181 ToolpathSaver() - Constructor.....................................................................................................181 .DefaultPost.................................................................................................................................181 .NumberOfToolpaths...................................................................................................................182 :GetNumPosts()............................................................................................................................182 :AddToolpath(Toolpath toolpath)................................................................................................182 :ClearToolpathList()......................................................................................................................182 :GetNumPosts()............................................................................................................................182 :GetPostAtIndex(integer post\_index)..........................................................................................182 :GetPostWithName(string post\_name).......................................................................................182 :GetPostWithFilename(string post\_file\_name)...........................................................................182 .CreatePost(string post\_file\_path)...............................................................................................182

:SaveToolpaths(PostPInfo postp, string filename, bool output\_to\_mc).....................................182

ToolpathStats...................................................................................................................................183 .IsValid..........................................................................................................................................184 .FeedLength.................................................................................................................................184 .PlungeLength..............................................................................................................................184 .RetractLength..............................................................................................................................184 .RapidLength................................................................................................................................184

.MinimumZ...................................................................................................................................184

ToolpathSaveInfo.............................................................................................................................185 .BaseName...................................................................................................................................185 .PathName...................................................................................................................................185 .PostP...........................................................................................................................................185 .AddSheetNameToEnd.................................................................................................................185 .ShowFileDialog............................................................................................................................185

.ToolpathList................................................................................................................................185

ToolpathList.....................................................................................................................................187 .Count...........................................................................................................................................187 .IsEmpty........................................................................................................................................187 :GetAt(POSITION pos)..................................................................................................................187 :GetHeadPosi on().......................................................................................................................187 :GetNext(POSITION pos)..............................................................................................................187 :GetPrev(POSITION pos)..............................................................................................................187

:GetTailPosi on().........................................................................................................................187

User Interface......................................................................................................................................188

HTML\_Dialog....................................................................................................................................188

HTML\_Dialog(bool local\_html, string html, integer width, integer height, string dialog\_name) -

Constructor..................................................................................................................................190 .WindowWidth.............................................................................................................................190 .WindowHeight............................................................................................................................190 :AddCheckBox(string element\_ id, bool value)...........................................................................191 :AddDirectoryPicker(string element\_ id, string buddy\_name, bool buddy\_is\_edit)..................191 :AddDoubleField(string element\_ id, double value)....................................................................192 :AddDropDownList(string element\_ id, string default\_value).....................................................192 :AddDropDownListValue(string element\_ id, string value).........................................................192 :AddFilePicker(string element\_ id, string buddy\_name, bool buddy\_is\_edit)............................192 :AddIntegerField(string element\_ id, integer value)...................................................................193 :AddLabelField(string element\_ id,string value)..........................................................................193 :AddRadioGroup(string element\_ id, int default\_index).............................................................194 :AddTextField(string element\_ id, string value)..........................................................................194 :AddToolEditor(string element\_ id, string buddy\_name)............................................................194 :AddToolPicker(string element\_ id, string buddy\_name, ToolDBId tooldb\_id)..........................195 :AddToolPickerValidToolType(string element\_ id, integer tool\_type)........................................195 :ClearToolPickerValidToolType(string element\_ id)....................................................................196 :GetCheckBox(string id)...............................................................................................................196 :GetDropDownListValue(string id)...............................................................................................196 :GetDoubleField(string id)............................................................................................................196 :GetIntegerField(string id)............................................................................................................196 :GetLabelField(string id)...............................................................................................................196 :GetRadioIndex(string id).............................................................................................................196 :GetTextField(string id)................................................................................................................197 :GetTool (string element\_ id).......................................................................................................197 :SetInnerHtml(string element\_ id, string html)...........................................................................197 :SetOuterHtml(string element\_ id, string html)..........................................................................197 :UpdateCheckBox(string element\_ id, bool value)......................................................................197 :UpdateDropDownListValue(string element\_ id, string value)...................................................197 :UpdateDoubleField(string element\_ id, double value)..............................................................197 :UpdateIntegerField(string element\_ id, integer value)..............................................................198 :UpdateLabelField(string element\_ id, string value)...................................................................198 :UpdateRadioIndex(string element\_ id, integer index)...............................................................198 :UpdateTextField(string element\_ id, string value).....................................................................198

:UpdateToolPickerField(string element\_ id, Tool tool)...............................................................198

FileDialog..........................................................................................................................................199 FileDialog() - Constructor.............................................................................................................199 :FileOpen(string default\_ext,string file\_name, string filter)........................................................199

:FileSave(string default\_ext,string file\_name, string filter).........................................................199

DirectoryReader...............................................................................................................................200 DirectoryReader() - Constructor..................................................................................................200 :BuildDirectoryList(string dir\_path, bool recurse).......................................................................200 :ClearDirs()...................................................................................................................................200 :ClearFiles()..................................................................................................................................200 :DirAtIndex(integer index)...........................................................................................................200 :FileAtIndex(integer index)..........................................................................................................200 :GetFiles(string filter, bool include\_files\_in\_list, bool include\_dirs\_in\_list)...............................200

:SortDirs().....................................................................................................................................200 :SortFiles(integer sort\_order, bool reverse)................................................................................201 :NumberOfDirs()..........................................................................................................................201

:NumberOfFiles()..........................................................................................................................201

DirectoryEntry..................................................................................................................................201

.Name...........................................................................................................................................201

FileEntry...........................................................................................................................................201 .Name...........................................................................................................................................201 .FileName.....................................................................................................................................201 .FileNameNoExt...........................................................................................................................201

.Size..............................................................................................................................................201

Registry............................................................................................................................................203 Registry(string sec on\_name) - Constructor...............................................................................203 :BoolExists(string parameter\_name)...........................................................................................203 :DoubleExists(string parameter\_name).......................................................................................203 :IntExists(string parameter\_name)..............................................................................................203 :GetDouble(string parameter\_name, double default\_value)......................................................203 :GetBool(string parameter\_name, bool default\_value)..............................................................203 :GetInt(string parameter\_name, integer default\_value)............................................................203 :GetString(string parameter\_name, string value).......................................................................203 :SetBool(string parameter\_name, bool default\_value)...............................................................204 :SetBool(string parameter\_name, bool value)............................................................................204 :SetDouble(string parameter\_name, double value)....................................................................204 :SetInt(string parameter\_name, integer value)...........................................................................204 :SetString(string parameter\_name, string value)........................................................................204

:StringExists(string parameter\_name).........................................................................................204

FileDialog..............................................................................................................................................205 FileDialog() - Constructor.............................................................................................................205 :FileOpen(string default\_ext,string file\_name, string filter)........................................................205 :FileSave(string default\_ext,string file\_name, string filter).........................................................205 .Ini alDirectory............................................................................................................................206 .Directory.....................................................................................................................................206 .FileExt..........................................................................................................................................206 .FileName.....................................................................................................................................206 .FileTitle........................................................................................................................................206 .PathName...................................................................................................................................206

.Folder..........................................................................................................................................206

ProgressBar......................................................................................................................................207

ProgressBar(string text, ProgressBarMode progress\_bar\_mode) – Constructor.......................207

:SetPercentProgress(integer percent).........................................................................................207 :StepProgress().............................................................................................................................207 :SetText(string text).....................................................................................................................207

:Finished()....................................................................................................................................207

Components – Aspire Only..................................................................................................................208

CombineMode – Aspire Only...........................................................................................................208 .Add..............................................................................................................................................208 .Subtract.......................................................................................................................................208 .MergeHeighest............................................................................................................................208 .MergeLowest..............................................................................................................................209 .Mul ply.......................................................................................................................................209

.Replace........................................................................................................................................209

Relief – Aspire Only..........................................................................................................................210

Relief(integer pixel\_width, integer pixel\_height, double real\_width, double\_real\_height) –

Constructor..................................................................................................................................210 Relief(Box2D bounds, float pixel\_size) – Constructor.................................................................210 .Error...........................................................................................................................................210 .PixelWidth...................................................................................................................................210 .PixelHeight..................................................................................................................................210 .RealWidth...................................................................................................................................210 .RealHeight...................................................................................................................................210 .XPixelSize....................................................................................................................................210 .YPixelSize.....................................................................................................................................210 .Thickness.....................................................................................................................................210 .SurfaceZ......................................................................................................................................210 .Volume........................................................................................................................................211 .IsTransparent..............................................................................................................................211 :Reset(float value, bool free\_memory)........................................................................................211 :MakeValuesTransparent(float value).........................................................................................211 :ReplaceTransparentValues(float value)......................................................................................211 :PointIsOnModel(integer x, integer y).........................................................................................211 :Set(integer x, integer y, float value)...........................................................................................211 :SetLowest(integer x, integer y, float value)................................................................................211 :Get(integer x, integer y)..............................................................................................................211 :GetInterpolated(double x, double y)..........................................................................................211 :GetMinMaxZ().............................................................................................................................212 :GetTrueMinMaxZ().....................................................................................................................212 :Add(float value)..........................................................................................................................212 :Subtract(float value)...................................................................................................................212 :Mul ply(float value)...................................................................................................................212

:MergeHeighest(float value, bool is\_abs\_value, bool preserve\_transparent)............................212 :MergeLowest(float value, bool is\_abs\_value, bool preserve\_transparent)..............................212 :CombineReliefs(Relief relief, CombineMode combine\_mode)..................................................212

:BlendBetweenReliefs(Relief first\_relief, Relief second\_relief, double blend\_factor)................213

:FlipY()..........................................................................................................................................213 :FlipX()..........................................................................................................................................213 :TransposeXY().............................................................................................................................213 :RotateClockwise90()...................................................................................................................213 :RotateCounterClockwise90()......................................................................................................213 :RenderTriangle(Point3D p1, Point3D p2, Point3D p3, bool merge\_high)..................................213 :CreateSlice(float min\_z, float max\_z, bool make\_below\_min\_z\_transparent).........................213 :ColumnIsTransparent(integer x, float transparent\_value).........................................................213 :RowIsTransparent(integer y, float transparent\_value)..............................................................213 :CreateSec onCopy(integer min\_x, integer min\_y, integer max\_x, integer max\_y)..................214

:TiltRelief(Point2D anchor\_pt, Point2D direc on\_pt, double angle)..........................................214

ComponentManager – Aspire Only.................................................................................................215 :AddRelief(Relief relief, CombineMode combine\_mode, string name)......................................215 :FindObjectIdWithName(string name)........................................................................................215 :FindObjectWithId(UUID uuid).....................................................................................................215 :GetSelectedObjectIds()...............................................................................................................215 :GetTopLevelObjectIds()..............................................................................................................215 :DeleteObjectWithId(UUID uuid).................................................................................................215 :DeleteObjectsWithIds(UUID\_List uuids).....................................................................................215 :CreateBakedCopyOfObjects(UUID\_List uuids, bool delete\_originals).......................................215 :CreateBakedCopyOfObject(UUID uuid, bool delete\_originals)..................................................216 :GroupObjectsWithIds(UUID\_List uuids).....................................................................................216 :GroupSelectedObjects()..............................................................................................................216 :UnGroupSelectedObjects().........................................................................................................216 :UnGroupObjectsWithId(UUID uuid)...........................................................................................216 :Selec onCanBeGrouped()...........................................................................................................216 :Selec onCanBeUngrouped()......................................................................................................216 :CloneObjectWithId(UUID uuid, bool make\_unique)..................................................................216 :CreateReliefFromObjectWithId(UUID uuid)...............................................................................216 :ReplaceComponentRelief(UUID target\_uuid, UUID source\_uuid).............................................216 :UpdateCompositeModel()..........................................................................................................217 :CreateReliefFromCompositeModel()..........................................................................................217

:UpdatePreviews().......................................................................................................................217

Component – Aspire Only................................................................................................................218 .Id.................................................................................................................................................218 .CombineMode............................................................................................................................218 .IsTilted.........................................................................................................................................218 .UseTilt.........................................................................................................................................218 .IsFaded........................................................................................................................................218 .UseFade.......................................................................................................................................218 :Transform(Matrix2D xform, bool update\_all)............................................................................218 :SetZScaleAuto(bool auto\_scale).................................................................................................218 :SetZOffset(float z\_offset)............................................................................................................218 :SetZScale(float scale)..................................................................................................................218

:TiltWouldChange(Point2D anchor\_pt, Point2D direc on\_pt, double lt\_angle, bool do\_ lt). 219 :SetTilt(Point2D anchor\_pt, Point2D direc on\_pt, double lt\_angle)........................................219 :GetTiltData(Point2D anchor\_pt, Point2D direc on\_pt).............................................................219

:FadeWouldChange(Point2D anchor\_pt, Point2D direc on\_pt, double end\_fade\_val, bool

do\_fade).......................................................................................................................................219

:SetFade(Point2D anchor\_pt, Point2D direc on\_pt, double lt\_angle).....................................219 :GetFadeData(Point2D anchor\_pt, Point2D direc on\_pt)..........................................................219

:GetRelief()...................................................................................................................................220

ComponentGroup – Aspire Only......................................................................................................221 .GetCount.....................................................................................................................................221 .Count...........................................................................................................................................221 .IsEmpty........................................................................................................................................221 :GetHead()....................................................................................................................................221 :GetHeadPosi on().......................................................................................................................221 :GetTailPosi on().........................................................................................................................221 :GetNext(POSITION pos)..............................................................................................................221 :GetPrev(POSITION pos)..............................................................................................................221

:GetAt(POSITION pos)..................................................................................................................221

DocumentVariableList......................................................................................................................222 : DocumentVariableExists (string variable\_name)......................................................................222 : GetDocumentVariable (string variable\_name)..........................................................................222 : SetDocumentVariable (string variable\_name, double value)....................................................222 : DocumentVariable True(string variable\_name)........................................................................222 :GetHeadPosi on().......................................................................................................................222

:GetNext(POSITION pos)..............................................................................................................222

DocumentVariable...........................................................................................................................222 .Name...........................................................................................................................................223 .Value...........................................................................................................................................223

## Introduction

The Script interface and associated documenta on is designed for people who are familiar with programming and is provided as a development tool for such people. It is extremely important to understand that Vectric cannot offer individual support for people wan ng to write scripts. This documenta on and the associated example scripts available from the Vectric web site are the only way you can learn about the script interface and develop your own gadgets. The appropriate area on the Vectric forum may also be useful for asking ques ons of other users.

This document lists the objects (classes) available to Lua scripts (Gadgets) from VCarve Pro V7.5 and Aspire V4.5

An object in Lua can have both methods and proper es. A method is called using ‘:’ a er the object name e.g. job:Refresh2DView(), a property is accessed via ‘.’ e.g. job.Exists . Proper es can be either read only (R/O i.e Get Values) or read / write (R/W - Get/Set Values).

## Global Methods

Most of the ac ons you can perform from Lua are done by calling methods on objects associated with the current job. There are a few methods which operate Globaly on the applica on and they are documented here .

IsAspire()

Returns true if the script is running inside Aspire.

IsBetaBuild()

returns true if this is a Beta build rather than a release build

GetAppVersion()

Returns applica on version number as a double e.g 4.004 for Aspire V4.0

GetBuildVersion()

Returns applica on internal build verion which is the same for both Aspire and VCarve Pro. This is different to the AppVersion above as Aspire 4.004 will report e.g. 7.004 which matches VCarve Pro 7.004 as they are built off the same code base and have the same script level support.

MessageBox (string text)

Displays a message box with passed text to user. The text can include HTML forma ng.

text – string – text to display in the message box

## Job related global methods

CloseCurrentJob ()

Close the current job - same as File - Close, if there are any unsaved changes user will be prompted to save before closing.

CreateNewJob(string name, Box2D bounds, double thickness, bool in\_mm, bool origin\_on\_surface)

Creates a new job with passed name, and se ngs. Returns true if job created OK else false.

name - string - name for job (without extension) bounds - Box2D - the 2d bounding box for the job area thickness - double - thickness of material block in\_mm - bool - true if job is in mm, else in inches origin\_on\_surface - true if z zero on material surface, else on machine bed

CreateNew2SidedJob(string name, Box2D bounds, double thickness, bool in\_mm, bool origin\_on\_surface, SideFlipDirection lip\_direction)

Creates a new two sided job using the given parameters. Return true if the job was created, otherwise, false.

name - string - name for job (without extension) bounds - Box2D - the 2d bounding box for the job area thickness - double - thickness of material block in\_mm - bool - true if job is in mm, else in inches origin\_on\_surface - true if z zero on material surface, else on machine bed flip\_direc on – the axis around which we flip the job (VectricJob.AROUND\_X or VectricJob.AROUND\_Y)

CreateNewRotaryJob(string name, double length, double diameter, MaterialBlock.XYOrigin xy\_origin, bool in\_mm, bool origin\_on\_surface, bool wrapped\_along\_x\_axis)

Creates a new two sided job using the given parameters. Return true if the job was created, otherwise, false.

name - string - name for job (without extension)

length - double - the length of the material cylinder (Accessible from MaterialBlock.CylinderLength)

diameter – double – the diameter of the material cylinder (Accessible from MaterialBlock.CylinderDiameter)

xy\_origin – MaterialBlock.XYOrigin – The XY origin of the block (Maps to the enum values of MaterialBlock.XYOrigin)

in\_mm – bool – true if the job is in mm, else in inches

origin\_on\_surface – bool – true if z zero on material surface, else on machine bed

wrapped\_along\_x\_axis – bool – true if we’re wrapping along the X axis and wrapping the Y values (Accessible from MaterialBlock.Rota onAxis)

OpenExistingJob(string pathname)

Opens an exis ng CRV or CRV3D file. Returns true if file opened OK, else false pathname - string - path to file

SaveCurrentJob()

Save the current job - same as File - Save, if no path has been set for the job this method will display the File Save As dialog.

Helper methods for crea ng and accessing objects on layers ...

## Vector object related global methods

CastCadObjectToCadBitmap(CadObject obj)

Casts passed CadObject to a CadBitmap - returns a CadBitmap object to work with

obj - CadObject - Object to be cast to a CadBitmap

CastCadObjectToCadContour(CadObject obj)

Casts passed CadObject to a CadContour - returns a CadContour object to work with obj - CadObject - Object to be cast to a CadContour

CastCadObjectToCadObjectGroup(CadObject obj)

Casts passed CadObject to a CadObjectGroup - returns a CadObjectGroup object to work with obj - CadObject - Object to be cast to a CadObjectGroup

CastCadObjectToCadPolyline(CadObject obj)

Casts passed CadObject to a CadPolyline - returns a CadPolyline object to work with obj - CadObject - Object to be cast to a CadPolyline

CastCadObjectToCadToolpathOutline(CadObject obj)

Casts passed CadObject to a CadToolpathOutline - returns a CadToolpathOutline object to work with obj - CadObject - Object to be cast to a CadToolpathOutline

CastCadObjectToCadToolpathPreview(CadObject obj)

Casts passed CadObject to a CadToolpathPreview - returns a CadToolpathPreview object to work with obj - CadObject - Object to be cast to a CadBitmap

CastCadObjectToTxtBlock(CadObject obj)

Casts passed CadObject to a TxtBlock - returns a TxtBlock object obj - CadObject - Object to be cast to a TxtBlock CreateCircle(double x, double y, double radius, double tolerance, double z\_value) Create a Contour object for a circle consis ng of 4 arcs.

x – double – x coord of centre of circle y – double – y coord of centre of circle radius – double – radius of circle

tolerance – double – tolerance of contour – use 0.0 for default z\_value – double – z coord for contour

CreateCopyOfSelectedContours(bool smash\_beziers, bool smash\_arcs, double smash\_tol) Create a ContourGroup containing a copy of the currently selected contours in the job. T

smash\_beziers – bool – if true, any Bezier spans in the selec on are replaced with a series of straight line spans which match the original Bezier within the specified tolerance.

smash\_arc – bool – if true, any Arc spans in the selec on are replaced with a series of straight line spans which match the original Arc within the specified tolerance.

I – double – tolerance (max allowed devia on) to use when approxima ng arcs or beziers with a series of straight lines.

## Component Related Global Methods – Aspire only

IsTransparent( loat value)

Returns true if the specified value is considered transparent

GetTransparentHeight()

Returns as a float the heights in reliefs considered transparent. This is the value used internally to represent a ‘transparent’ point.

CastComponentToComponentGroup(Component component)

Casts passed Component to a ComponentGroup - returns a ComponentGroup object to work with

## DocumentVariable Related Global Methods

IsInvalidDocumentVariableName(string name)

Returns true if the passed name is invalid for a DocumentVariable. Document variable names must begin with a le er and then may consist of le ers, numbers and underscores.

## Data ile locations related global methods

GetDataLocation()

Returns the ‘root’ data loca on for the applica on e.g.

“C:\ProgramData\Vectric\Aspire\V4.0” For V4.0 of Aspire on Windows 7. “C:\ProgramData\Vectric\VCarve Pro\V7.0” For V7.0 of VCarve Pro on Windows 7.

This is the same folder accessible via “File – OpenApplica on Data Folder …” within the program.

GetPostProcessorLocation()

Returns the PostP folder loca on for the program e.g

“C:\ProgramData\Vectric\Aspire\V4.0\PostP” For V4.0 of Aspire on Windows 7.

GetToolDatabaseLocation()

Returns the Tool Database folder loca on for the program e.g

“C:\ProgramData\Vectric\Aspire\V4.0\ToolDatabase” For V4.0 of Aspire on Windows 7.

GetGadgetsLocation()

Returns the Gadgets folder loca on for the program e.g

“C:\ProgramData\Vectric\Aspire\V4.0\Gadgets” For V4.0 of Aspire on Windows 7.

Note that the user can also install Gadgets in their local documents folder under …

“Vectric Files\Gadgets\Aspire V4.0”

GetToolpathDefaultsLocation()

Returns the ToolpathDefaults folder loca on for the program e.g

“C:\ProgramData\Vectric\Aspire\V4.0\ToopathDefaults” For V4.0 of Aspire on Windows 7.

GetBitmapTexturesLocation()

Returns the BitmapTextures folder loca on for the program which stores the bitmaps used for displaying different material types e.g

“C:\ProgramData\Vectric\Aspire\V4.0\BitmapTextures” For V4.0 of Aspire on Windows 7.

Note that the user can also install addi onal textures in their local documents folder under …

“Vectric Files\Material Images”

GetVectorTexturesLocation()

Returns VectorTextures folder loca on for the program e.g

“C:\ProgramData\Vectric\Aspire\V4.0\VectorTextures” For V4.0 of Aspire on Windows 7.

CreateCadContour (Contour ctr)

Returns a CadContour which owns the passed Contour object.

ctr - Contour - the data which is held by the created CadContour

CreateCadGroup(ContourGroup ctr\_group)

Returns a CadContourGroup which owns the passed ContourGroup object. ctr\_group - ContourGroup - the data which is held by the created CadContourGroup

GetDefaultContourTolerance ()

Returns a double value which is the default value for contour tolerances used within the program. This is usually 0.001 if working in metric or 0.00004 if working in inches.

### High Level Objects

The objects in this sec on of the document represent the high level objects within the program. Access to the program normaly starts with the VectricJob, and from there you can access the layers holding all the geometry and the current selec on state for the job.

## MaterialBlock

This object represents the material block within the program. It holds the width, height and thickness of the job and also the XY and Z origins. There is a single instance of the material block in the program. A number of methods are provided to covert between ‘absolute’ z values and z values rela ve to the surface of the material. These are very useful what wri ng scripts for toolpath genera on as scripts need to take account of the fact that the user may change between Z0 on the machine represen ng the bo om of the material or the top of the material.

Constructors

MaterialBlock() - Constructor

A new object which referes to the single material block within the program. e.g local mtl\_block = MaterialBlock()

Proper es

. ActualXYOrigin

R/O - Point2D - XY posi on for user specified origin posi on on material (see .XY origin)

. BlcOrigin

R/O - Point3D - XYZ posi on for bo om le corner of material block

.Height

R/O - double - height of material - length in Y

.InMM

R/O - bool - true if the job / job is in MM else it is in inches

.MaterialBox

R/O - Box3D - the 3D bounding box for the material

.MaterialSurfaceZ

R/O - double - Z value for surface of material

.Thickness

R/O - double - thickness of material - length in Z

.Width

R/O - double - width of material - length in X

.CylinderDiameter

R/O - double – (Rotary Only) the diameter of the material cylinder

.CylinderLength

R/O - double – (Rotary Only) the length of the material cylinder

.XYOrigin

R/O - XYOrigin - Value indica ng posi on in material chosen as XY origin - valid values are ...

MaterialBlock.BLC - Bo om Le Corner of material

MaterialBlock.BRC - Bo om Right Corner of material

MaterialBlock.TRC - Top Right Corner of material

MaterialBlock.TLC - Top Le Corner of material

MaterialBlock.CENTRE - Centre of material (Note Bri sh spelling!)

.ZOrigin

R/O - ZOrigin - Value indica ng posi on of Z origin - valid values are ...

MaterialBlock.Z\_TOP

MaterialBlock.Z\_CENTRE

MaterialBlock.Z\_BOTTOM

.FlipDirection

R/W – FlipDirec on – (Double-Sided Only) Values indica ng how we flip the material in a two sided job.

Value values,

VectricJob.AROUND\_X

VectricJob.AROUND\_Y

This can be used for crea ng a two sided job through CreateNew2SidedJob()

.JobType

R/W – JobType – Values indica ng the value of the current job.

Value values,

MaterialBlock.SINGLE\_SIDED

MaterialBlock.DOUBLE\_SIDED

MaterialBlock.ROTARY

.RotationAxis

R/O – Rota onAxis– (Rotary Only) Values indica ng the rota on axis of the cylinder.

Value values,

MaterialBlock.X\_AXIS

MaterialBlock.Y\_AXIS

Methods

:CalcAbsoluteZ(double z\_value)

Returns an ‘absolute’ z value from a z value rela ve to the surface of the block. Z\_value can be -ve for values below the surface or +ve for values above the surface. z\_value - double - z value rela ve to surface of material

:CalcAbsoluteZFromDepth (double z\_value)

Returns a depth value below the surface from an absolute z value.

z\_value - double - absolute z value.

: CalcDepthFromAbsoluteZ (double z\_value)

Returns a depth below the surface of the material from an absolute z value. z\_value - double -Absolute z value

:SideZOrigin(UUID side\_id)

Similar to .ZOrigin, but for the given side

:SideMaterialSurfaceZ(UUID side\_id)

Similar to .MaterialSurfaceZ for the given side

:SideMaterialBox(UUID side\_id)

Similar to .MaterialBox for the given side

:SideBlcOrigin(UUID side\_id)

Similar to .BlcOrigin for the given side

:CalcSideAbsoluteZ(UUID side\_id)

Similar to :CalAbsoluteZ() for the given side

:CalcSideDepthFromAbsoluteZ(UUID side\_id)

Similar to :CalcDepthFromAbsoluteZ() for the given side

:CalcSideAbsoluteZFromDepth(UUID side\_id)

Similar to :CalcAbsoluteZFromDepth() for the given side

Example Code

--[[ ------------------- DisplayMaterialSettings ------------------------------ |

| Display information about the size and position etc of the material block

|

|

| Return Values:

| None ]]

function DisplayMaterialSettings()

local mtl\_block = MaterialBlock() local units

if mtl\_block.InMM then units = " mm" else

units = " inches" end

-- Display material XY origin local xy\_origin\_text = "invalid" local xy\_origin = mtl\_block.XYOrigin

if xy\_origin == MaterialBlock.BLC then xy\_origin\_text = "Bottom Left Corner" elseif xy\_origin == MaterialBlock.BRC then xy\_origin\_text = "Bottom Right Corner" elseif xy\_origin == MaterialBlock.TRC then xy\_origin\_text = "Top Right Corner" elseif xy\_origin == MaterialBlock.TLC then xy\_origin\_text = "Top Left Corner" elseif xy\_origin == MaterialBlock.CENTRE then -- NOTE: British spelling for Centre! xy\_origin\_text = "Centre" else

xy\_origin\_text = "Unknown XY origin value!" end

local z\_origin\_text = "invalid" local z\_origin = mtl\_block.ZOrigin

if z\_origin == MaterialBlock.Z\_TOP then z\_origin\_text = "Top of Material" elseif z\_origin == MaterialBlock.Z\_CENTRE then -- NOTE: British spelling for Centre! z\_origin\_text = "Centre of Material" elseif z\_origin == MaterialBlock.Z\_BOTTOM then z\_origin\_text = "Bottom of Material" else

z\_origin\_text = "Unknown Z origin value!" end

local xy\_origin\_pos = mtl\_block.ActualXYOrigin

-- get 3d box object describing material bounds .... local mtl\_box = mtl\_block.MaterialBox local mtl\_box\_blc = mtl\_box.BLC

-- test methods to conver z values between absolute z and relative depths local test\_val = 0.125

local depth = mtl\_block:CalcDepthFromAbsoluteZ(test\_val) local abs\_z = mtl\_block:CalcAbsoluteZFromDepth(test\_val)

DisplayMessageBox(

"Width = " .. mtl\_block.Width .. units .."\n" ..

"Height = " .. mtl\_block.Height .. units .. "\n" ..

"Thickness = " .. mtl\_block.Thickness .. units .. "\n" ..

"\n" ..

"XY Origin = " .. xy\_origin\_text .. "\n" ..

" Position = (" .. xy\_origin\_pos.x .. ", " .. xy\_origin\_pos.y .. ")\n" ..

"Z Origin = " .. z\_origin\_text .. "\n" ..

"\n" ..

"Box Width = " .. mtl\_box.XLength .. units .."\n" ..

"Box Height = " .. mtl\_box.YLength .. units .. "\n" ..

"Box Thickness = " .. mtl\_box.ZLength .. units .. "\n" ..

"Box Bottom Left Corner = (" .. mtl\_box\_blc.x ..

"," .. mtl\_box\_blc.y ..

"," .. mtl\_box\_blc.z ..

")\n" ..

"\n" ..

"Test Value = " .. test\_val .. "\n" ..

" Depth from absolute test value = " .. depth .. "\n" ..

" Absolute Z from depth test value = " .. abs\_z .. "\n" ..

"\n"

)

end

--[[ --------- main ---------------------------------

|

| Entry point for script

| ]] function main()

-- Check we have a job loaded job = VectricJob()

if not job.Exists then

DisplayMessageBox("No job loaded") return false; end

DisplayMaterialSettings()

return true; end

## VectricJob

This is the main object a script writer will interact with and represents the currently open job (file) within the applica on. There are members of the object which will tell you if there is an exis ng job open and also to allow you to create a new job.

Proper es

.Exists

R/O - bool - returns true if there is an exis ng job open

.DocumentVariables

R/O - returns the DocumentVariableList for the job. The variable list allows for user defined values to be used in formulas.

.Height

R/O - double - returns height of the job along y - same as YLength

.IsAspire

R/O - bool - returns true if the script is running inside Aspire.

.IsBetaBuild

R/O - bool - returns true if this is a Beta build rather than a release build

.InInches

R/O - bool - returns true if the current units are inches

.InMM

R/O - bool - returns true if the current units are mm

.Job Parameters

R/O - returns the utParameterList for the job. The parameter list allows scripts to store persistent data with the job.

.LayerManager

R/O - returns the CadLayerManager -for the job. This manages all the layers which in turn hold all the vector objects in the job

.MinX

R/O - double - returns X coordinate of bo om le corner of job

.MinY

R/O - double - returns Y coordinate of bo om le corner of job

.Name

R/O - string - returns the name of the current job WITHOUT the file extension e.g. ”test” not “test.crv”

.PostProcessorParameters

R/O - returns the utParameterList for the last run of a post processor. This will hold informa on such as the PostP used, the tools used and the actual toolpath files output.

.Selection

R/O - returns the Selec onList for the currently selected vectors in the job.

.SheetManager

R/O - returns the CadSheetManager for the job. This manages all the sheets in the job

.Width

R/O - double - returns width of the job along x - same as XLength

.XLength

R/O - double - returns length of the job along x - same as XLength

.YLength

R/O - double - returns length of the job along Y - same as Height

Methods

:ClearClipboard()

Clear any data on the clipboard

:CreateHorizontalGuide(double y\_value, bool locked) Create a horizontal guide

y\_value - double - y value for guide locked - bool - if true guide is locked

:CopyToClipboard(string text)

Copies passed text to clipboard. Returns true if copied to clipboard OK else false.

text -string –text to put on clipboard

:CreateVerticalGuide(double x\_value, bool locked)

Create a ver cal guide

x\_value - double - x value for guide locked - bool - if true guide is locked

:DisplayToolpathPreviewForm()

Display the toolpath preview form for simula on

:ExportSelectionToEps(string pathname)

Exports selected vectors to passed path as an EPS file. Returns true if data exported OK else false.

pathname - string - path to file

:ExportSelectionToDxf(string pathname)

Exports selected vectors to passed path as a DXF file. Returns true if data exported OK else false.

pathname - string - path to file

:ExportSelectionToSvg(string pathname)

Exports selected vectors to passed path as a SVG file. Returns true if data exported OK else false. pathname - string - path to file

:ExportDocumentVariables(string pathname)

Exports the job’s DocumentVariableList to passed path as an INI file. pathname - string - path to file

:GetBuildVersion()

Returns applica on internal build verion same for Aspire and VCarve Pro e.g. 7.004 for Aspire 4.004 which matches VCarve Pro 7.004

:GetBounds()

Return the 2D bounding box for the drawing area - a Box2D

:GetAppVersion()

Returns applica on version number as a double e.g 4.004 for Aspire V4.0

:GetWrapDiameter()

Returns wrap diameter for a wrapped model

:GroupSelection()

Returns true if the current selec on could be grouped

:GuidesVisible()

Returns true if guides are visible else false

:ImportBitmap(string pathname)

Imports a bitmap from passed path. Returns true if file imported OK else false.

pathname - string - path to file

:ImportDxfDwg(string pathname)

Imports a dxf or dwg file from passed path. Returns true if file imported OK else false.

pathname - string - path to file

NOTE: a er data is imported it is selected.

:ImportSTLDirect(string pathname)

Directly import (without orienta on form) an STL with passed path - automa cally creates a component. Returns true if file imported OK else false. NOTE: Aspire only pathname - string - path to file

:ImportSVG(string pathname)

Import an SVG file with the given path and creates the vectors and layers. Returns true if the file was imported with no issues, otherwise returns false.

pathname - string - path to file

:ImportDocumentVariables(string pathname)

Imports a DocumentVariableList from the passed path. Please note that any exis ng variables will be overwri en.

pathname - string - path to file

:IsWrappedModel()

Return true if this is a wrapped model

:MoveObjectToSheet(CadObject object, UUID sheet\_id)

Moves the passed object to the sheet with the passed id. For details of how to access the sheet ids see CadSheetManager.

:Refresh2DView ()

Refreshes the 2D view, call this a er you have created or modified geometry to update the 2d view

:SelectAllVectors()

Returns true if all vectors in the current job could be selected

:SetGuidesVisible(bool visible)

Set visibility of guides visible - bool - true if guides visible else false

:SetXY\_Origin(double origin, double x\_offset, double y\_offset) Set the 2D origin for the job - returns true if origin set OK else false.

origin - XYOrigin the four corners or the centre x\_offset - double - distance in Xof specified point from 0,0

Y\_offset - double - distance in Y of specified point from 0,0

NOTE: no data is moved by this call!

:UnGroupSelection(bool deep\_ungroup, bool preserve\_orig\_layers)

Returns true if the current selec on could be ungroup deep\_ungroup – boo l – if true ungroup Groups recursively preserve\_orig\_layers - bool – if true preserve the original object layers

:WrappingXValues()

Return true if we are wrapping X values - else wrapping Y values

:WrapZOnSurface()

Returns true if z for wrap is on surface of material, else in centre

:SwitchActiveSide()

Switch the current side and make the opposite side ac ve

:SetActiveSide(UUID side\_id)

Set the current side to be that of the given UUID

:GetSideList()

Return UUID\_List with the list of side UUIDs. This is useful for itera ng over the sides :GetActiveSideId()

Return the current side’s UUID

:GetSideName(UUID side\_id)

Return a string with the name of the current side

:CopySelectionToOppositeSide()

Copy the selected vectors and components to the opposite side and flip them.

:MoveSelectionToOppositeSide()

Move the selected vectors and components to the opposite side and flip them.

:SideComponentManager(UUID side\_id)

Return the ComponentManager for the side which has the given UUID.

If you just require the component manager of the current side, you can just carry on using

.ComponentManager

:IsSingleSided()

Return true if we have a job and it’s single-sided.

:IsDoubleSided()

Return true if we have a job and it’s double-sided.

## CadLayerManager

This object is responsible for managing all the layers within the applica on. A reference to the CadLayerManager is obtained via the LayerManager property of the VectricJob. The CadLayerManager maintains a list of all the CadLayers in the job.

Proper es

.Count

R/O - integer - Returns the number of layers being managed

.IsEmpty

R/O - bool - Returns true if there are no layers else false

Methods

:AddLayer(CadLayer layer, int index)

Add the passed layer object to the list of layers in the job and set its posi on in the list to the passed index.

Layer - CadLayer - layer to add to the job

Index - int - index of the posi on in the layer list where this layer should be inserted. Enter -1 to add the layer to the end of the list, otherwise it must be greater than 0.

:FindLayerWithName(string layer\_name)

Returns the CadLayer with passed name. If no layer exists with the passed name returns nil

Layer\_name - string - Name for layer

:GetActiveLayer()

Returns the current ac ve CadLayer. If no layer exists returns nil

:GetAt(POSITION pos)

Returns the layer at the passed posi on

pos - POSITION - current posi on in list

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list of layers

:GetLayerWithName(string layer\_name)

Returns the CadLayer with passed name. If no layer exists with the passed name a new layer is created

:GetLayerWithId(UUID layer\_id)

Returns the CadLayer with passed id. The id is usually obtained from the .RawLayerId property of a CadObject

:GetNext(POSITION pos)

Returns the layer at the current posi on AND a new value for posi on poin ng to the next item in

the list (or nil if at end of list)

pos - POSITION - current posi on in list

Example - note that GetNext(pos) is returning two values ...

local pos = layer\_manager:GetHeadPosition() local layer while pos ~= nil do

layer, pos = layer\_manager:GetNext(pos)

DO SOMETHING WITH LAYER .... end

:GetPrev(POSITION pos)

Returns the layer at the current posi on AND a new value for posi on, poin ng to the previous item

in the list (or nil if at start of list) pos - POSITION - current posi on in list

:GetTailPosition()

Returns a POSITION variable to allow access to the tail of the list of layers

:HaveDataOnSheets()

Returns true if there is data on more than one sheet.

:RemoveLayer(CadLayer layer)

Remove the passed layer object from the list of layers in the job.

Layer - CadLayer - layer to remove from the job.

:SetActiveLayer(CadLayer layer)

Sets passed layer to be the current ac ve layer. If passed layer is nil, sets first layer to be ac ve.

Layer - CadLayer - layer to be made the ac ve layer

Example Code

function SetBitmapBrightness(job, brightness)

if not job.Exists then

DisplayMessageBox("No job loaded") return false end

local layer\_manager = job.LayerManager

local pos = layer\_manager:GetHeadPosition() while pos ~= nil do local layer layer, pos = layer\_manager:GetNext(pos) if not layer.IsSystemLayer then local layer\_pos = layer:GetHeadPosition() while layer\_pos ~= nil do local object

object, layer\_pos = layer:GetNext(layer\_pos) if object.ClassName == "vcCadBitmap" then cad\_bitmap = CastCadObjectToCadBitmap(object)

MessageBox("Found bitmap - brightess = " .. cad\_bitmap.Brightness) cad\_bitmap.Brightness = brightness cad\_bitmap.Visible = visible end end end -- end of for each object on layer end -- end of for each layer

job:Refresh2DView()

## CadLayer

This object holds all the data for a layer within the applica on. Layers are created and accessed via the CadLayerManager object. A layer maintains a list of CadObjects.

Proper es

.Colour

R/W - 32 bit integer - get / set the colour for a layer as a COLORREF - 32 bit colour value

.Color

R/W - 32 bit integer - get / set the colour for a layer as a COLORREF - 32 bit colour value - this is the U.S spelling version of the method above!

.Count

R/O - integer - Returns the number of CadObjects present on the layer

.IsBitmapLayer

R/O - bool - returns true if this is a bitmap layer. The bitmap layer is used for imported bitmaps and is placed at the start of the layer list so that bitmaps are drawn before vectors.

.IsEmpty

R/O - bool - true if the layer has no objects on it.

.IsSystemLayer

R/O - bool - returns true if this is a system layer. System layers are used for holding items such as toolpath previews.

.Locked

R/W - bool - Get / set the locked property for the layer

.Name

R/W - string - get / set the name for the layer

.Visible

R/W - bool - Get / set the visible property for the layer

.Id

R/O - luaUUID - Id of this layer

.RawId

R/O - UUID - The unique iden fier for this layer

Methods

:AddObject (CadObject object, bool on\_current\_sheet)

Add passed object to this layer object - CadObject - the object (contour etc) to add to layer - object becomes property of layer

on\_current\_sheet - bool – this should always be true, and the object is created on the current sheet. If false the sheet from the object would be retained, but as this sheet property is not controllable from Lua, false should not be used in normal opera on.

:Find (UUID id)

Returns the posi on in the list for the object with the passed id. If no object is found the returned posi on is nil

id - UUID - id for object

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list of objects on layer

:GetTailPosition()

Returns a POSITION variable to allow access to the tail of the list of objects on layer

:GetNext(POSITION pos)

Returns the object at the current posi on AND a new value for posi on poin ng to the next item in

the list (or nil if at end of list)

pos - POSITION - current posi on in list

:GetPrev(POSITION pos)

Returns the object at the current posi on AND a new value for posi on, poin ng to the previous

item in the list (or nil if at start of list)

pos - POSITION - current posi on in list

:GetAt(POSITION pos)

Returns the object at the passed posi on

pos - POSITION - posi on in list

:RemoveObject (CadObject object)

Removes passed object from this layer. Object becomes property of the script. Returns object removed or nil if object was not found on layer.

Object - CadObject - the object (contour ec) to remove from layer - object becomes property of caller

:RemoveAt (POSITION pos)

Removes the CadObject at the passed posi on and returns it.

pos - POSITION - posi on in list

:SetColour(double red, double green, double blue)

Set the colour for the layer

red - double - red component of colour in range 0.0 - 1.0 green - double - red component of colour in range 0.0 - 1.0 blue - double - red component of colour in range 0.0 - 1.0 :SetColor (double red, double green, double blue)

Set the color for the layer (U.S spelling)

red - double - red component of colour in range 0.0 - 1.0 green - double - red component of colour in range 0.0 - 1.0 blue - double - red component of colour in range 0.0 - 1.0

## CadSheetManager

This object is responsible for managing all the sheets within the applica on. A reference to the CadSheetManager is obtained via the SheetManager property of the VectricJob. The CadSheetManager maintains a list of all the sheet ids in the job.

Proper es

.ActiveSheetId

R/W - UUID- Get the id of the ac ve sheet. Set this value to the id of a different sheet to change the ac ve sheet.

.NumberOfSheets

R/O - Integer- returns the number of sheets in the job.

Methods

:CreateSheets:(integer num\_sheets, UUID sheet\_id)

Create num\_sheets new sheets with the same proper es as the sheet with id sheet\_id.

num\_sheets - integer - the number of new sheets to add to the job.

sheet\_id - UUID - the id of the sheet that the new sheets should take their proper es from (e.g. dimensions, Z Zero Posi on etc.).

:CreateSheets:(integer num\_sheets, UUID sheet\_id, Box2D material\_bounds)

Create num\_sheets new sheets with the same proper es as the sheet with id sheet\_id but with width and height defined by material\_bounds.

num\_sheets - integer - the number of new sheets to add to the job. sheet\_id - UUID - the id of the sheet that the new sheets should take their proper es from. material\_bounds - Box2D - the width and height dimensions for the new sheets.

:GetSheetIds()

Returns a Lua iterator that can be used to iterate over all of the sheet ids in the job (see example below).

:GetSheetName(UUID sheet\_id)

Returns the name of the sheet with the passed id as a string.

sheet\_id - UUID - the method will return the name of the sheet with this id.

e.g.

function SetSheet() local job = VectricJob() local sheet\_manager = job.SheetManager local sheet\_ids = sheet\_manager:GetSheetIds() for id in sheet\_ids do if(sheet\_manager:GetSheetName(id) == "Sheet 1") then sheet\_manager.ActiveSheetId = id

end

end

end

## CadObject

This object is the base class for all the different types of objects which can appear on a layer within the applica on. CadObject is derived from the Object class which has only one method - ClassName. The objects derived from CadObject can include polylines, groups, text, bitmaps, toolpath previews etc.

A huge variety of named parameters can be associated with every object within the program and accessed by scripts.

Proper es

.ClassName

R/O - string - the returns the name of the class within the applica on that the object represents.

Values for this property include (but there are many more) ...

vcCadBitmap vcCadContour vcCadObjectGroup etc.

.Id

R/W - luaUUID - The unique iden fier for this object - it is EXTREMELY important that you do not set two different objects to have the same id!

.IsBitmap

R/O - bool - true if this object is a bitmap or derived from a bitmap

.IsSelected

R/O - bool - true if this object is selected. The selected state is set when an object is added to the jobs selec on list.

.IsVisible

R/W - bool - Get / Set the visibility state for an object. Normally the visibility of objects within the applica on is controlled via its layer. Be very careful not to create invisible objects from scripts which the user cannot then make visible.

.IsLocked

R/W - bool - Get / Set the locked state for an object. Normally the locked state of objects within the applica on is controlled via its layer. Be very careful not to create locked objects from scripts which the user cannot then make unlocked.

.LayerId

R/O - luaUUID - Id of layer this object belongs to. This cannot not be modified directly. To add objects to layers or move them between layers the methods of CadLayer must be used.

.RawId

R/O - UUID - The unique iden fier for the layer of this object This cannot not be modified directly. To add objects to layers or move them between layers the methods of CadLayer must be used. .RawLayerId

R/W - UUID - The unique iden fier for the layer this object belongs to.

.SheetId

R/O - UUID - The id of the sheet this object is on. To move an object to a new sheet see VectricJob:MoveObjectToSheet.

.SideId

R/O – UUID – The Id of the side which this object belongs to

To compare this to another UUID, use luaUUID(object.SideId).IsEqual(my\_other\_id)

Methods

:CanTransform(integer lags)

Returns true if object can be transformed. The value of flag is made up by combining the following values ...

1, // can object be moved ?

2, // can object be rotated ?

4, // can object be mirrored ?

8, // can object be scaled symetrically ?

16 // can object be scaled asymetrically ?

Flags - integer - value indica ng the type of valid transform being queried

:Clone ()

Return a new copy of this object, with all new id’s. The returned object is not on a layer so is not yet part of the job.

:GetBoundingBox ()

Returns a Box2D represen ng the 2d bounds of this object

:GetContour ()

Return the Contour representa on for this object - not all objects will have a contour representa on. E.g a grouped object will return nil for this call.

:InvalidateBounds()

Invalidates the cached bounding box for this object. This should be called if you modify the raw data for the object.

:SetLayer (CadLayer layer)

Adds this object to the passed layer

layer - CadLayer - layer this object should belong to

:Transform(Matrix2D matrix)

Transform the object using the passed transforma on matrix. The transforma on matrix can move (translate), scale or rotate or a combina on of all 3. matrix - Matrix2D - the transforma on matrix to apply to the object

:GetBool(string parameter\_name, bool default\_value, bool create\_if\_not\_exist)

Retrieve a Boolean flag (true / false) with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter default\_value - bool - the value which will be returned if there is no exis ng value stored create\_if\_not\_exist - bool - if true the default value will be stored in list if no exis ng value

:GetDouble(string parameter\_name, double default\_value, bool create\_if\_not\_exist)

Retrieve a double with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter default\_value - double - the value which will be returned if there is no exis ng value stored create\_if\_not\_exist - bool - if true the default value will be stored in list if no exis ng value

:GetInt(string parameter\_name, integer default\_value, bool create\_if\_not\_exist)

Retrieve an integer with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter value - integer - the value which will be returned if there is no exis ng value stored create\_if\_not\_exist - bool - if true the default value will be stored in list if no exis ng value

:ParameterExists(string parameter\_name, utParameterType type) Returns true if there is an exis ng parameter with passed name and type.

parameter\_name - string - the name of the parameter type - utParameterType - the type of parameter

:SetBool(string parameter\_name, bool value)

Store a Boolean flag (true / false) with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - bool - the value which will be stored in the parameter list

:SetDouble(string parameter\_name, double value)

Store a double with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - double - the value which will be stored in the parameter list

:SetInt(string parameter\_name, integer value)

Store an integer with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - integer - the value which will be stored in the parameter list

:SetString(string parameter\_name, string value)

Store a string with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value

value - string - the value which will be stored in the parameter list

## CadContour

This object is derived from CadObject and represents a single vector within the job. A CadContour holds a Contour object which contains the actual data for the vector shape. A CadContour is created from a Contour object using the CreateCadContour() method. To access the Contour data from a CadContour object, use the :GetContour() method from the base CadObject class.

Proper es

All the proper es for this object are documented in the CadObject base class

Methods

This object inherits all documented methods from the CadObject base class with the following addi ons

:InsertToolpathAtTabAtPoint(Point2D point)

This method inserts a toolpath tab at the point on the CadContour closest to the specified point. It returns a ToolpathTab object.

point - Point2D - The point where to place the toolpath tab

## CadObjectGroup

This object is derived from CadObject and represents a group of CadObjects within the job.

Proper es

This object inherits all documented proper es from the CadObject base class with the following addi ons

.Count

R/O – integer - returns the number of CadObjects in the group

.IsEmpty

R/O – bool – returns true if the CadObjectGroup is empty

Methods

This object inherits all documented methods from the CadObject base class with the following addi ons

:GetAt(POSITION pos)

Returns the object at the current posi on pos – POSITION – current posi on in list

:GetHead ()

Returns the object at the current posi on AND a new value for posi on poin ng to the next item in the group (or nil if at end of group)

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the group

:GetNext(POSITION pos)

Returns the object at the current posi on AND a new value for posi on poin ng to the next item in the group (or nil if at end of group) pos – POSITION – current posi on in group

:GetPrev(POSITION pos)

Returns the object at the current posi on AND a new value for posi on, poin ng to the previous item in the group (or nil if at start of group)

pos – POSITION – current posi on in list

:GetTailPosition()

Returns a POSITION variable to allow access to the tail (end) of the group

## CadBitmap

This object represents an imported bitmap in the program. It is iden fied by the object.ClassName property == “vcCadBitmap”. To convert a ‘CadObject’ to a ‘CadBitmap’ use the helper func on CastCadObjectToCadBitmap

Proper es

.Brightness

R/W - integer - value in range 0 - 255

External methods

CastCadObjectToCadBitmap(CadObject object)

Return a CadBitmap object for the passed CadObject which must have been iden fied as being a CadBitmap.

e.g

function SetBitmapBrightness(job, brightness)

if not job.Exists then

DisplayMessageBox("No job loaded") return false end

local layer\_manager = job.LayerManager

local pos = layer\_manager:GetHeadPosition() while pos ~= nil do local layer layer, pos = layer\_manager:GetNext(pos) if not layer.IsSystemLayer then local layer\_pos = layer:GetHeadPosition() while layer\_pos ~= nil do local object

object, layer\_pos = layer:GetNext(layer\_pos) if object.ClassName == "vcCadBitmap" then cad\_bitmap = CastCadObjectToCadBitmap(object)

MessageBox("Found bitmap - brightess = " .. cad\_bitmap.Brightness) cad\_bitmap.Brightness = brightness cad\_bitmap.Visible = visible end end

end -- end of for each object on layer end -- end of for each layer

job:Refresh2DView() end

## CadPolyline

This object represents a vector in the program which has been created by the polyline drawing tool. It has all the same proper es and methods as a CadContour and can be treated as one. It is iden fied by the object.ClassName property == “vcCadPolyline”. To convert a ‘CadObject’ to a

‘CadPolyline’ use the helper func on CastCadObjectToCadPolyline

## CadMarker

This object represents a ‘marker’ in the program which can be used to label items. CadMarkers are not machinable or selectable. It is derived from CadObject and has all the proper es of CadObject available.

Constructor

CadMarker(string text, Point2D pt, integer pixel\_size) - Constructor A CadMarker at passed posi on text -string – text for marker object pt -Point2D – posi on for marker pixel\_size pt -integer – size marker is drawn at in pixels

Proper es

.Text

R/W – string – Text displayed at marker

.Position

R/W – Point2D – Posi on for marker

Methods

:SetColor(double red, double green, double blue)

Set colour for marker

red - double – red value for colour in range 0-1.0 green - double – gree value for colour in range 0-1.0 blue - double – blue value for colour in range 0-1.0

Example Code

--[[ ------------------ MarkContourNodes ----------------------------

|

| Insert a marker at each node on the passed contour

| ]]

function MarkContourNodes(contour, layer)

local num\_spans = 0

local ctr\_pos = contour:GetHeadPosition() while ctr\_pos ~= nil do local span

span, ctr\_pos = contour:GetNext(ctr\_pos)

-- create a marker at start of span

local marker = CadMarker("V:" .. num\_spans, span.StartPoint2D, 3) layer:AddObject(marker, true) num\_spans = num\_spans + 1; end

-- if contour was open mark last point if contour.IsOpen then

local marker = CadMarker("V:" .. num\_spans, contour.EndPoint2D, 3) layer:AddObject(marker, true) end

end

## utParameterList

This object is used to store strings, doubles and integers as ‘proper es’ of an object such as the job. The parameter has a name such as “DocumentCreator”, a type which is either string, double (a floa ng point number such as 1.2345) or an integer ( a whole number such as 7) and the value you assign for the parameter.

For the job you would use the .JobParameters property to access the ParameterList for the job.

Methods

:GetDouble(string parameter\_name, double default\_value, bool create\_if\_not\_exist)

Retrieve a double with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter default\_value - double - the value which will be returned if there is no exis ng value stored create\_if\_not\_exist - bool - if true the default value will be stored in list if no exis ng value

:GetInt(string parameter\_name, integer default\_value, bool create\_if\_not\_exist)

Retrieve an integer with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter value - integer - the value which will be returned if there is no exis ng value stored create\_if\_not\_exist - bool - if true the default value will be stored in list if no exis ng value

:ParameterExists(string parameter\_name, utParameterType type) Returns true if there is an exis ng parameter with passed name and type.

parameter\_name - string - the name of the parameter type - utParameterType - the type of parameter

:SetBool(string parameter\_name, bool default\_value, bool create\_if\_not\_exist)

Retrieve a Boolean flag (true / false) with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter default\_value - bool - the value which will be returned if there is no exis ng value stored create\_if\_not\_exist - bool - if true the default value will be stored in list if no exis ng value

:SetBool(string parameter\_name, bool value)

Store a Boolean flag (true / false) with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - bool - the value which will be stored in the parameter list

:SetDouble(string parameter\_name, double value)

Store a double with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - double - the value which will be stored in the parameter list

:SetInt(string parameter\_name, integer value)

Store an integer with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - integer - the value which will be stored in the parameter list

:SetString(string parameter\_name, string default\_value, bool create\_if\_not\_exist)

Retrieve a string with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter default\_value - string - the value which will be returned if there is no exis ng value stored create\_if\_not\_exist - bool - if true the default value will be stored in list if no exis ng value

ParameterList.UTP\_DOUBLE

ParameterList.UTP\_INT

ParameterList.UTP\_BOOL

ParameterList.UTP\_STRING

:SetString(string parameter\_name, string value)

Store a string with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value

value - string - the value which will be stored in the parameter list

## luaUUID

Every object in the job has a unique id referred to as a UUID. As these id’s are difficult to save and process from lua, the luaUUID object wraps a windows UUID and lets it be treated as a string. Constructor

luaUUID() - Constructor

A new luaUUID with a new unique id.

Proper es

.IsEmpty

R/O - bool – returns true if this is the NULL / Empty id

.RawId

R/O - UUID – returns the raw UUID this object wraps up

Methods

:MakeEmpty()

Makes this id equal to the ‘empty’ id

:CreateNew()

Make the id hold a new unique value

:AsString()

Return a string representa on of the id

:Set(string text\_id)

Set the id using a string representa on of the id text\_id – string – text representa on of an id

:SetId(UUID raw\_id) Set the id using raw id raw\_id –UUID – id

:IsEqual(luaUUID id)

Return true if passed id is the same as this one id –UUID – id to compare

## UUID\_List

Object which holds a list of UUID’s (note NOT luaUUID’s these are ‘raw’ UUID’s).

Constructor

UUID\_List() - Constructor

Creates a new empty UUID\_List

Local uuid\_list = UUID\_List()

Properties

.Count

R/O -integer – return the number of ids in the list

.IsEmpty

R/O -bool – return true if the id list is empty

Methods

:AddHead( UUID id)

Add the passed id to the front of the list id –UUID –id to put in list

:AddTail( UUID id)

Add the passed id to the end of the list id –UUID –id to put in list

:GetAt(pos)

Returns the id at the current posi on pos – POSITION – current posi on in list

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list

:GetNext(POSITION pos)

Returns the id at the current posi on AND a new value for posi on poin ng to the next item in the

list (or nil if at end of list)

pos – POSITION – current posi on in list

Example - note that GetNext(pos) is returning two values ...

local pos = uuid\_list:GetHeadPosition() local id while pos ~= nil do

id, pos = uuid\_list:GetNext(pos) DO SOMETHING WITH ID ....

end

:GetPrev(POSITION pos)

Returns the toolpath at the current posi on AND a new value for posi on, poin ng to the previous

item in the list (or nil if at start of list)

pos – POSITION – current posi on in list

:GetTailPosition()

Returns a POSITION variable to allow access to the tail (end) of the list toolpaths

:Find(UUID id)

Return the POSITION of passed id in list. Returns nil if id not in list posi on id –UUID –id to find in list

:FindIndex(UUID id)

Return the index of passed id in list. Returns -1 if id not in list posi on id –UUID –id to find in list

:InsertAfter(POSITION pos, UUID id) Insert the id a er the passed posi on pos – POSITION –posi on in list id –UUID –id to put in list

:InsertBefore(POSITION pos, UUID id) Insert the id before the passed posi on pos – POSITION –posi on in list id –UUID –id to put in list

:RemoveAll()

Remove all ids from the list

:RemoveAt(POSITION pos)

Remove the id at the current posi on pos – POSITION – current posi on in list

:RemoveHead()

Remove the id at the front of the list and return it

:RemoveTail()

Remove the id at the end of the list and return it

:SetAt(POSITION pos, UUID id)

Replace the id at the passed posi on with the new value

pos – POSITION –posi on in list id –UUID –id to put in list

## CadObjectList

This object holds a list of CadObjects

Constructor

CadObjectList(bool owns\_objects) - Constructor Creates a new CadObjectList owns\_objects -bool – true if this list owns the objects it contains. If this is true and the list s ll contains objects when it goes out of scope the objects will be deleted.

Proper es

.Count

R/O -integer – return the number of objects in the list

.IsEmpty

R/O -bool – return true if the list is empty

Methods

:AddHead( CadObject obj)

Add the passed object to the front of the list obj –CadObject –object to put in list

:AddTail(CadObject obj)

Add the passed object to the end of the list obj –CadObject –object to put in list

:CanTransform(integer lags)

Return true if object can be transformed

flags –integer –value indica ng type of transform – these can be combined e.g 3 (1+2) will check for both Move and Rotate

1. = Move
2. = Rotate

4 = Mirror

8 = Scale Symetric

16 = Scale Asymetric

:GetAt(POSITION pos)

Returns the object at the current posi on pos – POSITION – current posi on in list

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list

:GetNext(POSITION pos)

Returns the object at the current posi on AND a new value for posi on poin ng to the next item in the list (or nil if at end of list)

pos – POSITION – current posi on in list

Example - note that GetNext(pos) is returning two values ...

local pos = obj\_list:GetHeadPosition() local object while pos ~= nil do

object, pos = obj\_list:GetNext(pos) DO SOMETHING WITH OBJECT .... end

:GetPrev(POSITION pos)

Returns the object at the current posi on AND a new value for posi on, poin ng to the previous

item in the list (or nil if at start of list)

pos – POSITION – current posi on in list

:GetTailPosition()

Returns a POSITION variable to allow access to the tail (end) of the list

:RemoveHead()

Remove the id at the front of the list and return it

:RemoveTail()

Remove the id at the end of the list and return it

:Transform(Matrix2D xform)

Transform all objects in list by passed transform xform – Matrix2D – transforma on matrix to apply to objects

## SelectionList

This object holds a list of CadObjects represen ng the current selec on in the program.

The selec on is accessed via the job …

-- Check we have a document loaded local job = VectricJob()

if not job.Exists then

DisplayMessageBox("No job loaded") return false; end

local selection = job.Selection if selection.IsEmpty then

MessageBox("Please select one or more vectors to label") return false end

Proper es

.Count

R/O -integer – return the number of objects in the list

.IsEmpty

R/O -bool – return true if the list is empty

Methods

:Add ( CadObject obj, bool add\_tail, bool group\_add) Add the passed object to the list obj –CadObject –object to put in list

add\_tail –bool –if true add to end of seelc on else insert at start

group\_add –bool –if you are adding a lot of objects to the selec on, set this true and call :GroupSelec onFinished() when you have added all the objects. This stops a lot of uneccesary upda ng while you are upda ng the selec on.

:AddHead( CadObject obj)

Add the passed object to the front of the list obj –CadObject –object to put in list

:AddTail(CadObject obj)

Add the passed object to the end of the list obj –CadObject –object to put in list :CanTransform(integer lags)

Return true if object can be transformed - these can be combined e.g 3 (1+2) will check for both

Move and Rotate

flags –integer –value indica ng type of transform –

1. = Move
2. = Rotate

4 = Mirror

8 = Scale Symetric

16 = Scale Asymetric

:CanSelect(CadObject obj)

Return true if passed object can be selected obj –CadObject –object to test

:Clear()

Clear the selec on (list is emp ed)

:GetAt(POSITION pos)

Returns the object at the current posi on pos – POSITION – current posi on in list

:GetBoundingBox()

Return a Box2D with the bounds of the selected objects

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list

:GetNext(POSITION pos)

Returns the object at the current posi on AND a new value for posi on poin ng to the next item in

the list (or nil if at end of list)

pos – POSITION – current posi on in list

Example - note that GetNext(pos) is returning two values ...

local pos = obj\_list:GetHeadPosition() local object while pos ~= nil do

object, pos = obj\_list:GetNext(pos) DO SOMETHING WITH OBJECT .... end

:GetPrev(POSITION pos)

Returns the object at the current posi on AND a new value for posi on, poin ng to the previous

item in the list (or nil if at start of list) pos – POSITION – current posi on in list :GetTailPosition()

Returns a POSITION variable to allow access to the tail (end) of the list

:GroupSelectionFinished()

Call this if you have been using :Add(…) with group\_add set to true to update the applica on when you have finished adding objects to the selec on.

:RemoveOnLayer(UUID layer\_id)

Remove any objects from the selec on which are on the layer with the passed id. The objects are just removed from the selec on, they are not deleted are changed in any other way.

:Remove( CadObject obj, bool group\_remove) Remove the passed object from the selec on obj –CadObject –object to put in list group\_remove –bool –if you are removing a lot of objects to the selec on, set this true and call :GroupSelec onFinished() when you have removed all the objects. This stops a lot of uneccesary upda ng while you are upda ng the selec on.

:RemoveHead()

Remove the id at the front of the list and return it

:RemoveTail()

Remove the id at the end of the list and return it

:Transform(Matrix2D xform)

Transform all objects in list by passed transform xform – Matrix2D – transforma on matrix to apply to objects

### Creating Vectors From Script

The previous objects (VectricJob, CadLayer, CadContour etc) represent high level objects used within the program. When the user wants to create new geometry from within a script, they will be working with lower level en es. These en es are Spans, Contours and Contour Groups.

A Span is the lowest level piece of geometry and it always has a start and end point. It is important to realize that although geometry in the main program is always 2D, the underlying reprenta on of Spans and Contours supports full 3d values. This means that when Contours are used to define toolpaths, full 3D tool moves can be represented. If you are simply crea ng geometry which will appear in the 2D view in the main program, the Z value for all points should be le at 0.0.

## Contour

This object represents a single vector within the applica on. Open and closed vectors, arcs, circles, all are based on a Contour with one or more spans. The Contour is represented as a list of spans, with a span being either a Line, Arc or Bezier.

Constructors

Contour(tolerance) - Constructor

A new contour is created within a Lua script using this contructor method e.g

local contour = Contour(0.0)

Tolerance - double - tolerance to use when performing operations - use 0.0 for default

Proper es

.Area

R/O - double - unsigned area of contour

.BoundingBox2D

R/O - Box2D - 2D bounding box for contour

.BoundingBox3D

R/O - - Box3D - 3D bounding box for contour

.Count

R/O - integer - number of spans in the contour

.ContainsArcs

R/O - bool - true if contour contains 1 or more arcs - may contain other span types as well.

.ContainsBeziers

R/O - bool - true if contour contains 1 or more beziers - may contain other span types as well.

.CentreOfGravity

R/O - Point2D - the centre of gravity for the contour

.EndPoint3D

R/O - Point3D - the 3D posi on of the contour end point - if contour is closed this will be the same as the start point.

.EndPoint2D

R/O - Point2D - the 2D posi on of the contour end point - if contour is closed this will be the same as the start point.

.IsClockwise

R/O - bool - Direc on of the contour, this is found from the signed area of the contour. For an open contour, it is assumed to be closed with a straight line when calcula ng the area.

. IsAntiClockwise

R/O - bool - Direc on of the contour, this is found from the signed area of the contour. For an open contour, it is assumed to be closed with a straight line when calcula ng the area.

. IsCW

R/O - bool - Direc on of the contour, this is found from the signed area of the contour. For an open contour, it is assumed to be closed with a straight line when calcula ng the area.

. IsCCW

R/O - bool - Direc on of the contour, this is found from the signed area of the contour. For an open contour, it is assumed to be closed with a straight line when calcula ng the area.

.IsClosed

R/O - bool - True if the contour is closed

.IsEmpty

R/O - bool - true if contour is empty - no spans

.IsOpen

R/O - bool - True if contour is open

.IsSinglePoint

R/O - bool - true if contour is just a single point

.Length

R/O - double - length of contour

.StartPoint3D

R/O - Point3D - the 3D posi on of the contour start point

.StartPoint2D

R/O - Point2D - the 2D posi on of the contour start point

.Tolerance

R/W - double - the tolerance value used when polygonising beziers and arcs. This should not be changed without very good reason.

Methods

:AppendPoint(double x, double y)

Set the 2D star ng point for the contour. Returns true if point set OK else false. This method should only be called on an EMPTY contour. The Z value for the point defaults to 0.0

x - double - X value for start point of contour y - double - Y value for start point of contour

:AppendPoint(double x , double y, double z)

Set the 3D star ng point for the contour. Returns true if point set OK else false. This method should only be called on an EMPTY contour.

x - double - X value for start point of contour y - double - Y value for start point of contour z - double - Z value for start point of contour

:AppendPoint(Point2D pt2d)

Set the 2D star ng point for the contour. Returns true if point set OK else false. This method should only be called on an EMPTY contour. The Z value for the point defaults to 0.0

Pt2d - Point2D - 2D coordinate for start point of contour

:AppendPoint(Point3D pt3d)

Set the 3D star ng point for the contour. Returns true if point set OK else false. This method should only be called on an EMPTY contour.

Pt2d - Point3D - 3D coordinate for start point of contour

:ArcTo (Point2D end\_pt2d, Point2D center\_pt2d, boolccw)

Append an arc span from the current end point of the contour to the passed posi on. Returns true if span added t OK else false. The arc is defined by the exis ng end point in the contour, the center point of the arc and the direc on of rota on. Arcs > 180 degrees will be split into two arcs.

end\_pt2d - Point2D - 2D coordinate for end point of arc center\_pt2d - Point2D - 2D coordinate for center point of arc ccw - bool - true if arc is counter clockwise from start point else clockwise

:ArcTo (Point2D end\_pt2d, bool bulge)

Append an arc span from the current end point of the contour to the passed posi on. Returns true if span added t OK else false. This method should only be called on a non-empty contour. Arcs > 180 degrees will be split into two arcs.

end\_pt2d - Point2D - 2D coordinate for end point of arc bulge - double - signed bulge factor for arc

:ArcTo( Point3D end\_pt3d, bool bulge)

Append an arc span from the current end point of the contour to the passed posi on. Returns true if span added OK else false. This method should only be called on a non-empty contour. Arcs > 180 degrees will be split into two arcs.

end\_pt3d - Point3D - 3D coordinate for end point of arc - arcs must be in XY plane bulge - double - signed bulge factor for arc

:AppendContour (Contour ctr)

Append passed contour to the end of this contour. Returns true if contour appended OK, false if contour wouldn’t append within tolerance

ctr - Contour - Contour to append

:AppendSpan (Span span)

Append passed span to the end of this contour. Returns true if span appended OK, false if span wouldn’t append within tolerance span -Span – Span to append

:AppendSpanAsLines(Span span, double tolerance)

Append passed span as line spans - passed span is NOT modified or adopted. Returns true if span appended OK, false if span wouldn’t append within tolerance

span - Span - Span to append polygonised version of tolerance - double - tolerance to use when conver ng span to lines

:BezierTo(Point2D end\_pt2d, Point2D ctrl\_1\_2d, Point2D ctrl\_2\_2d)

Append a bezier span from the current end point of the contour to the passed posi on. Returns true if span added OK else false. All points must have same z value

end\_pt2d - Point2D - 2D coordinate for end point of bezier ctrl\_1\_2d - Point2D - 2D coordinate for first control point of bezier ctrl\_2\_2d - Point2D - 2D coordinate for second control point of bezier

:BezierTo(Point3D end\_pt3d, Point2D ctrl\_1\_2d, Point2D ctrl\_2\_2d)

Append a bezier span from the current end point of the contour to the passed posi on. Returns true if span added OK else false. All points must have same z value

end\_pt3d - Point3D - 3D coordinate for end point of bezier ctrl\_1\_2d – Point2D - 2D coordinate for first control point of bezier ctrl\_2\_2d – Point2D - 2D coordinate for second control point of bezie :CanAppendContour(Contour ctr)

Return true if passed contour can be appended onto the end of this contour ctr - Contour - Contour to check if can append

:CanAppendSpan (Span span)

Return true if passed span can be appended onto the end of this contour span - Span - Span to check if can append

:Clone()

Return a new Contour which is an exact copy of this one.

:CreateTolerancedCopy(double tolerance)

Return a new Contour which has been toleranced (simplified) to the passed value. NOTE: The contour must consist of just line spans, it must NOT contain beziers or arcs, the proper es .ContainsBeziers and .ContainsArcs must be false.

tolerance - double - tolerance to use when simplifying contour.

:CreatePolygonizedCopy(double tolerance, double max\_line\_len)

Return a new Contour which has been toleranced (simplified) to the passed value. Any bezier or arc spans are polygonized before tolerancing. The max\_line\_len parameter can be used to limit the maximum length of a tolerance span.

tolerance - double - tolerance to use when simplifying contour. 0.0 means use default max\_line\_len - double - maximum length of a tolerance span , 0.0 means no limit

:GetFirstSpan()

Return the first Span in the contour

:GetLastSpan()

Return the last Span in the contour

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list of spans in the contour

:GetTailPosition()

Returns a POSITION variable to allow access to the head of the list of spans in the contour

:GetNext(POSITION pos)

Returns the Span at the current posi on AND a new value for posi on poin ng to the next item in

the list (or nil if at end of list). E.g

cur\_span, pos = offset\_vector:GetNext(pos)

pos - POSITION - current posi on in list

:GetPrev (POSITION pos)

Returns the Span at the current posi on AND a new value for posi on poin ng to the previous item

in the list (or nil if at start of list). E.g

cur\_span, pos = offset\_vector:GetPrev(pos)

pos - POSITION - current posi on in list

:GetAt (POSITION pos)

Returns the Span at the current posi on in the list. E.g .

cur\_span = offset\_vector:GetAt(pos)

pos - POSITION - current posi on in list

:IsPointInside (Point2D point, double tolerance)

Return true if passed point is inside this contour. Beziers and arcs are polygonized for this calcula on. If passed tolerance = 0.0 the default tolerance is used, else the specified tolerance. If this is called for an open contour the return value is undefined.

point - Point2D - 2D coordinate if point to check

tolerance - double - tolerance for polygonisa on - 0.0 means default

:InvalidateBoundingBox()

If you modify the data in the contour, you must call this method to invalidate the cached bounding box.

:LineTo(double x , double y)

Append a line span from the current end point of the contour to the passed posi on. Returns true if span added OK else false. This method should only be called on a non-empty contour

(use :AppendPoint() to start the contour if you want to create a contour consis ng of a series of line segments). The Z value for the point defaults to 0.0

x - double - X value for point y - double - Y value for point

:LineTo(double x , double y, double z)

Append a line span from the current end point of the contour to the passed posi on. Returns true if span added OK else false. This method should only be called on a non-empty contour

(use :AppendPoint() to start the contour if you want to create a contour consis ng of a series of line segments).

x - double - X value for point y - double - Y value for point z - double - Z value for point :LineTo(Point2D end\_pt2d )

Append a line span from the current end point of the contour to the passed posi on. Returns true if span added t OK else false. This method should only be called on a non-empty contour. The Z value for the point defaults to 0.0 end\_pt2d - Point2D - 2D coordinate for point

:LineTo( Point3D end\_pt3d)

Append a line span from the current end point of the contour to the passed posi on. Returns true if span added OK else false. This method should only be called on a non-empty contour. end\_pt3d - Point3D - 3D coordinate for point

:MakeOffsetsSquare(double offset\_dist, double offset\_out, double max\_dist)

Process contour and change any arcs which match the passed radius into 'square' offset if arc matches passed parameters. This is called AFTER a normal offset offset opera on to change normal rounded corners into ‘sharp’ corners.

offset\_dist - double - the distance contour was offset by - corners will have this radius offset\_out - bool - true if contour was offset outwards

max\_dist - double - maximum distance to allow square offset to extend (acute corners can have square offsets which extend a LONG way out)

:OffsetInZ(double z\_offset)

Add the passed z value to the Z coordinates for all spans in the contour

z\_val - double - z value to add to all span heights

:ReorderStartPoint(Point2D point)

Reorder the start point of a closed contour to be as close as possible to the passed point. No points are inserted into the contour, the exis ng spans are just reordered. point - Point2D - 2D coordinate of point to define contour start

:ReorderStartPoint (integer span\_index, double parameter)

Reorder a closed the contour, inser ng a point (and hence spli ng a span if necessary).

span\_index - integer - index of span holding new start point parameter - double - parameter in range 0-1.0 of point on span

:Reverse ()

Reverse the direc on of the contour - the start becomes the end and the direc on of all spans are reversed.

:SetZHeight(double z\_val)

Set the Z coordinates for all spans in contour to the passed value z\_val - double - z height for contour

:Smash (double tolerance, bool preserve\_arcs)

Smash beziers (and op onally arcs) into straight lines within the passed tolerance.

tolerance - double - tolerance to use when polygonizing beziers and arcs preserve\_arcs - bool - if true arcs aren’t smashed, only beziers

:Transform ( Matrix2D xform)

Transform the contour using the passed transforma on matrix. The transforma on matrix can move (translate), scale or rotate or a combina on of all 3.

matrix - Matrix2D - the transforma on matrix to apply to the contour

:RemoveHead ()

Remove the first span in the contour and return it

:RemoveTail ()

Remove the last span in the contour and return it

## ContourGroup

This object represents a group of Contour objects. The contours are held in a double linked list which can be iterated through using the GetHeadPosi on/ GetTailPosi on and GetNext / GetPrev methods.

Constructor

ContourGroup(bool owns\_contours) - Constructor Create a new empty ContourGroup object.

owns\_contour -bool -If true, this object will own all the contours placed inside it and will delete them when the group is deleted. If this flag is false the contours in the group will not be deleted when the group is deleted.

Proper es

.BoundingBox2D

R/O - Box2D - returns the 2D bounding box for all contours in group

.BoundingBox3D

R/O - Box3D - returns the 3D bounding box for all contours in group

.Count

R/O - integer - returns the number of contours in the group

.ClosedCount

R/O- integer - returns the number of closed contours in the group

.IsEmpty

R/O - bool - true if the group is empty - doesn’t contain any contours

.OpenCount

R/O - integer - returns the number of open contours in the group

.OwnsContours

R/O - bool - true if the group owns the contours it holds

Methods

:AppendGroup(ContourGroup group)

Append the contours in the passed group onto the end of this group. Passed group is deleted.

group -ContourGroup – Contour Group object being added to group

:AddHead(Contour ctr)

Add passed contour to the Front / Head / Start of this group ctr -Contour - Contour object being added to group :AddTail(Contour ctr)

Add passed contour to the Back / Tail / End of this group ctr -Contour - Contour object being added to group

:Clone()

Return a copy of this group and its contents. The new group will own its copies off the contours.

:CreateTolerancedCopy(double tolerance)

Create a new controur group with all arcs and beziers in the original group replaced with straight line segments which match the original spans within the passed tolerance.

tolerance -double -max devia on from original contour permi ed when replacing arcs and Bezier with straight lines.

:ContainsBeziers()

Returns true if one or more contours in the group contains one or more Bezier spans

:DeleteOpenVectors()

Delete all open vectors in group

:GetLength()

Return the length of all contours in the group

:GetHeadPosition()

Return a POSITION object poin ng to the first Contour in the group which can be used to iterate through the group contents

:GetTailPosition()

Return a POSITION object poin ng to the last Contour in the group which can be used to iterate through the group contents

:GetNext(POSITION pos)

Returns BOTH the contour at the current posi on and a new pos value poin ng to the next contour in the group. If the end of the list has been reached the posi on will be returned as nil.

pos-POSITION - Current posi on in the group to get contour from

Example Code

-- iterate through the group and contours in group local count = 0 local num\_spans = 0 local span\_length = 0.0 local tolerance = 0.0001

local pos = group:GetHeadPosition() while pos ~= nil do local contour

contour , pos = group:GetNext(pos)

-- iterate through each span in the contour local ctr\_pos = contour:GetHeadPosition() while ctr\_pos ~= nill do local span

span, ctr\_pos = contour:GetNext(ctr\_pos) num\_spans = num\_spans + 1;

span\_length = span\_length + span:GetLength(0.0001) end

count = count + 1 end

NOTE: contour , pos = group:GetNext(pos)

- both the contour and a new value for pos are returned by the method

:GetPrev(POSITION pos)

Returns BOTH the contour at the current posi on and a new pos value poin ng to the previous contour in the group. If the start of the list has been reached the posi on will be returned as nil.

pos-POSITION - Current posi on in the group to get contour from

Example Code

-- iterate through the group and contours in group local span\_length = 0.0 local tolerance = 0.0001

local pos = group:GetTailPosition() while pos ~= nil do local contour

contour , pos = group:GetPrev(pos)

-- iterate through each span in the contour local ctr\_pos = contour:GetTailPosition() while ctr\_pos ~= nill do local span

span, ctr\_pos = contour:GetPrev(ctr\_pos) span\_length = span\_length + span:GetLength(0.0001) end

end

NOTE: contour , pos = group:GetPrev(pos)

- both the contour and a new value for pos are returned by the method

:GetAt(POSITION pos)

Returns the contour at the current posi on. Pos is le unchanged

pos-POSITION - Current posi on in the group to get contour from

:GetHead()

Return the contour at the head of the list, the contour remains in the group

:GetTail()

Return the contour at the tail of the list, the contour remains in the group

:MakeOffsetsSquare(double offset\_dist, bool offset\_out, double max\_dist) Used to ‘square’ offsets on external corners a er an offset opera on.

offset\_dist -double - distance contours in group were offset offset\_out - bool - if true the offset opera on was an offset outwards max\_dist -double - max distance to allow square corners to extend out. Offsets of acute corners can extend a long way from the original vector.

:Offset(double dist, double stepover, integer num\_offsets, bool preserve\_arcs)

Create a new contour group generated by offse ng the contours in this group. The group should only contain closed non-overlapping vectors.

dist -double -distance to offset by. +ve values will offset outwards, -ve values inwards stepover -double -if doing mul ple offset, this is the value for the 2nd and subsequent offsetd num\_offsets -integer -number of offsets to perform

preserve\_arcs -bool -if true arcs are preserved, else arcs replaced by approximated lines

:OffsetInZ(double dist)

Add the passed z dist to the z height of every span in every contour in group dist-double -z value to add to every span in every contour in group

:OffsetWithOpenVectors(double dist, double stepover, integer num\_offsets, bool preserve\_arcs)

Create a new contour group generated by offse ng the contours in this group. If the group contains open vectors these are offset individually and merged back into the final offset. When open offsets are merged back in this may result in crossing vectors.

dist -double -distance to offset by. +ve values will offset outwards, -ve values inwards stepover -double -if doing mul ple offset, this is the value for the 2nd and subsequent offsetd num\_offsets -integer -number of offsets to perform preserve\_arcs -bool -if true arcs are preserved, else arcs replaced by approximated lines

:RemoveAt(POSITION pos)

Returns the contour at the current posi on, removing it from the group. Pos is INVALID a er this call and should not be used again.

pos-POSITION - Current posi on in the group to remove contour from

:RemoveHead()

Remove the first contour in the group and return it

:RemoveTail()

Remove the last contour in the group and return it

:Reverse()

Reverse the direc on of every contour in the group

:SetZHeight(double z\_val)

Set the z height of every span in every contour to the passed value.

z\_val -double -z value for every span in every contour in group

:Smash(double tolerance, bool preserve\_arc)

Replaces all beziers (and op onally arcs) in all contours with a straight line representa on within passed tolerance.

tolerance -double -max devia on from original contour permi ed when replacing arcs and Bezier with straight lines. preserve\_arcs - bool - if true arcs are not smashed, only beziers

:Transform(Matrix2D xform)

Transform all contours in the group using the passed matrix xform-Matrix2D -matrix to transform all contours with

## ContourCarriage

This object is used to move along a contour a fixed distance at a me irrespec ve of the number of spans in the contour or the type of spans.

Constructor

ContourCarriage(Contour ctr, Point2D pt)

Creates a ‘carriage’ for the passed contour and posi ons it at the nearest point on the contour to the passed point.

ctr-Contour -The contour that the carriage travels along pt-Point2D -The carriage is posi oned on the contour at the closest posi on to this point

ContourCarriage(integer span\_index, double parameter)

Creates a ‘carriage’ for a contour and set its posi ons for the span with the passed index at the passed parameter posi on (in range 0-1.0) on the span. These values are normaly specified as 0,0.0 to posi on the carriage at the start of the first span in the contour.

span\_index-integer -Index of span, carriage will be posi oned at parameter-double -Parameter along span carriage will be posi oned at

e.g local cursor = ContourCarriage(0, 0.0)

Proper es

.IsInvalid

R/O - bool - Returns true if current posi on is invalid

.SpanIndex

R/O - integer - Return span index for current posi on of carriage

.SpanParameter

R/O - double - Return span parameter (in range 0-1.0) for current posi on of carriage

Methods

:Move(Contour ctr, double distance)

Moves the carriage the requested distance along the contour. Returns true if manage to move requested distance successfully. If the contour is closed we will wrap around at the end, if it is open and you a empt to move past the end this method will return false.

ctr-Contour -The contour this carriage rides along

distance - double –The distance to move along the contour, nega ve values will move back

towards the start

:Position(Contour ctr)

Returns a Point2D represen ng the current posi on on the carriage on the passed contour ctr-Contour -The contour this carriage rides along

:UpdatePosition(Contour ctr, Point2D pt)

Update the posi on of the carriage to the nearest posi on on the contour to the passed point.

ctr-Contour -The contour that the carriage travels along pt-Point2D -The carriage is posi oned on the contour at the closest posi on to this point

Example Code

[[ -------------- MarkContourParameterSteps --------------------------------- |

| Insert a marker at each parameter step around the contour and display the

| distance from the start point

| ]]

function MarkContourParameterSteps(num\_steps, contour, layer) local cursor = ContourCarriage(0, 0.0)

local contour\_length = contour.Length local step\_dist = contour\_length / num\_steps

local end\_index = num\_steps if contour.IsOpen then end\_index = end\_index + 1 end

for i = 1, end\_index do

local ctr\_pos = cursor:Position(contour)

local marker = CadMarker("D:" .. (step\_dist \* (i - 1)), ctr\_pos, 3) layer:AddObject(marker, true) cursor:Move(contour, step\_dist) end end

## ToolpathTab

This object is derived from ContourCarriage and represents a ToolpathTab. A ToolpathTab is created from a CadContour object using the InsertToolpathTabAtPoint() method.

Proper es

All the proper es for this object are documented in the ContourCarriage base class

Methods

All the methods for this object are documented in the ContourCarriage base class

## Span

This object is the base class for spans in a Contour. Spans are either lines, arcs or beziers or a single

‘point’

Constructors

Span(Point3D pt3d) - Constructor

A new span represen ng a single point is created within a Lua script using this contructor method pt3d - Point3D - 3d position for a point

e.g

local pt = Point3D(0, 0, 0) local pt\_span = Span(pt)

NOTE: most spans are created as either a LineSpan, ArcSpan or BezierSpan.

Proper es

.EndPoint3D

R/O - Point3D - The 3D end point of the span

.EndPoint2D

R/O - Point3D - The 2D end point of the span

.IsLineType

R/O - bool - true if span is a line

.IsArcType

R/O - bool - true if span is an arc

.IsBezierType

R/O - bool - true if span is a bezier

.IsLeadInType

R/O - bool - true if span is a lead in

.IsLeadOutType

R/O - bool - true if span is a lead out

.IsOvercutType

R/O - bool - true if span is part of an overcut

.NumberOfControlPoints

R/O - integer - number of control points for span (2 for Bezier, 1 for an arc (the mid point), 0 for a line

.StartPoint3D

R/O - Point3D - The 3D start point of the span

.StartPoint2D

R/O - Point2D - The 2D start point of the span

.Type

R/O - integer - enum iden fying type of the span

Usual values are Span.NormalLine, Span.Arc and Span.Bezier

Methods

:ChordLength()

Return the straight line distance between start and end points of the span

:EndVector(bool normalise)

Return a Vector2D represen ng the direc on of the span at the end.

normalise - bool - if true, the returned vector is normalised

:GetControlPointPosition(integer index)

Return a Point2D with posi on of requested control point (use .NumberOfControlPoints to get count). This call is only valid on a Bezier or an Arc. For a Bezier 0 returns first control point, 1 returns the second. For an arc, 0 returns arc mid point. index - integer - index of control point

:GetLength(double tolerance)

Return a double holding the length of the span. The passed tolerance is used for beziers to approximate length to given tolerance. tolerance - double - tolerance to use when polygonizing beziers to find length

:MoveControlPoint(integer index, Point2D pt)

This is only valid for an arc or Bezier and a empts to move the specified control point to the passed posi on. This call is only valid on a Bezier or an Arc. For a Bezier, index 0 modifies first control point, 1 modifies the second. For an arc, 0 modifies the arc mid point. If the control point can’t be moved to the passed posi on (e.g would make an invalid arc) returns false, else true.

index - integer - index of control point pt - Point2D - posi on for control point

:PointAtParameter(double param, double tolerance)

Return a Point2D represen ng the point on the span at passed parameter in range 0 - 1.0. Passed tolerance is used for polygonisa on for Bezier spans.

parameter - double - parameter in range 0 - 1.0 to find point on span at tolerance - double - tolerance to use when polygonizing beziers :Reverse()

Reverse the span - swap start and end points and for arcs swap sign of bulge, for beziers swap control points.

:StartVector(bool normalise)

Return a Vector2D represen ng the direc on of the span at the start.

normalise - bool - if true, the returned vector is normalised

:SetStartPoint3D(Point3D pt)

Set start point for span - be EXTREMELY careful if you call this method on spans in a contour as gaps will open between adjoining spans leading to an invalid contour. pt - Point3D - 3D posi on for point

:SetStartPoint2D(Point2D pt)

Set start point for span - be EXTREMELY careful if you call this method on spans in a contour as gaps will open between adjoining spans leading to an invalid contour. pt - Point2D - 2D posi on for point

:SetEndPoint3D(Point3D pt)

Set end point for span - be EXTREMELY careful if you call this method on spans in a contour as gaps will open between adjoining spans leading to an invalid contour. pt - Point3D - 3D posi on for point

:SetEndPoint2D(Point2D pt)

Set end point for span - be EXTREMELY careful if you call this method on spans in a contour as gaps will open between adjoining spans leading to an invalid contour.

pt - Point2D - 2D posi on for point

## LineSpan

This object represents a line span in a Contour. It is derived from the Span class and all methods and proper es of the span class are valid on this object as well. As well as crea ng spans to append to a contour you can use the LineTo methods to avoid span crea on.

Constructors

LineSpan(Point3D start\_pt, Point3D end\_pt) - Constructor

A new span represen ng a line is created within a Lua script using this contructor method

start\_pt - Point3D – 3D posi on for start point of span end\_pt - Point3D – 3D posi on for end point of span

e.g

local start\_pt = Point3D(0, 0, 0) local end\_pt = Point3D(1.0, 0, 0) local line\_span = LineSpan(start\_pt, end\_pt)

LineSpan(Point2D start\_pt, Point2D end\_pt) - Constructor

A new span represen ng aline is created within a Lua script using this contructor method

start\_pt - Point2D - 2d posi on for start point of span end\_pt - Point2D - 2d posi on for end point of span

e.g

local start\_pt = Point2D(0, 0) local end\_pt = Point2D(1.0, 0) local line\_span = LineSpan(start\_pt, end\_pt)

Methods

:ClosestParameterToPoint(Point2D pt)

Return the parameter of the point on the line which is closest to the passed point pt - Point2D - 2d posi on of point finding closest parameter to

## ArcSpan

This object represents an arc span in a Contour. It is derived from the Span class and all methods and proper es of the span class are valid on this object as well. As well as crea ng spans to append to a contour you can use the ArcTo methods to avoid span crea on. Arcs are represented internally using the start and end points and a ‘bulge factor’ which controls the shape of the arc. The bulge represents a ra o between the chord length of an arc and its height (bulge = (2\*arc\_height) / chord\_length).

Constructors

ArcSpan(Point2D start\_pt, Point2D end\_pt, Point2D pt\_on\_arc) - Constructor

A new span represen ng an arc is created within a Lua script using this contructor method

start\_pt – Point2D – 2D posi on for start point of span

end\_pt – Point2D – 2D posi on for end point of span- must have same Z values as start pt\_on\_arc- Point2D – 2D posi on for a point on arc NOTE: arcs can be a maximum of 180 degrees

e.g

local start\_pt = Point2D(0, 0) local end\_pt = Point2D(1.0, 0) local pt\_on\_arc = Point2D(0.5, 0.3) local arc\_span = ArcSpan(start\_pt, end\_pt, pt\_on\_arc)

ArcSpan(Point2D start\_pt, Point2D end\_pt, Point2D centre\_pt, bool ccw) - Constructor A new span represen ng an arc is created within a Lua script using this contructor method

start\_pt – Point2D – 2D posi on for start point of span end\_pt – Point2D – 2D posi on for end point of span- must have same Z values as start centre\_pt – Point2D – 2D posi on for centre point of arc (not mid-point)

ccw - bool - true if arc is Counter Clockwise

e.g

local start\_pt = Point2D(0, 0) local end\_pt = Point2D(1.0, 0) local center\_pt = Point2D(0.5, 0) local arc\_span = ArcSpan(start\_pt, end\_pt, center\_pt, false)

NOTE: arcs can be a maximum of 180 degrees

ArcSpan(Point3D start\_pt, Point3D end\_pt, Point3D centre\_pt, bool ccw) - Constructor A new span represen ng an arc is created within a Lua script using this contructor method

start\_pt - Point3D - 3d posi on for start point of span

end\_pt - Point3D - 3d posi on for end point of span- must have same Z values as start centre\_pt - Point3D - 3d posi on for centre point of arc (not mid-point)

ccw - bool - true if arc is Counter Clockwise

e.g

local start\_pt = Point3D(0, 0, 0) local end\_pt = Point3D(1.0, 0, 0) local center\_pt = Point3D(0.5, 0, 0) local arc\_span = ArcSpan(start\_pt, end\_pt, center\_pt, false)

NOTE: arcs can be a maximum of 180 degrees

ArcSpan(Point3D start\_pt, Point3D end\_pt, Point2D pt\_on\_arc) - Constructor

A new span represen ng an arc is created within a Lua script using this contructor method

start\_pt - Point3D – 3D posi on for start point of span

end\_pt - Point3D – 3D posi on for end point of span- must have same Z values as start pt\_on\_arc- Point2D – 2D posi on for a point on arc NOTE: arcs can be a maximum of 180 degrees

e.g

local start\_pt = Point3D(0, 0, 0) local end\_pt = Point3D(1.0, 0, 0) local pt\_on\_arc = Point2D(0.5, 0.3) local arc\_span = ArcSpan(start\_pt, end\_pt, pt\_on\_arc)

ArcSpan(Point2D start\_pt, Point2D end\_pt, double bulge) - Constructor

A new span represen ng an arc is created within a Lua script using this contructor method

start\_pt – Point2D – 2D position for start point of span

end\_pt – Point2D – 2D position for end point of span - must have same Z values as start bulge - double - buge factor for arc

e.g

local start\_pt = Point2D(0, 0) local end\_pt = Point2D(1.0, 0) local arc\_span = ArcSpan(start\_pt, end\_pt, 1.0) ArcSpan(Point3D start\_pt, Point3D end\_pt, double bulge) - Constructor

A new span represen ng an arc is created within a Lua script using this contructor method

start\_pt - Point3D – 3D position for start point of span

end\_pt - Point3D – 3D position for end point of span - must have same Z values as start bulge - double - buge factor for arc

e.g

local start\_pt = Point3D(0, 0, 0) local end\_pt = Point3D(1.0, 0, 0) local arc\_span = ArcSpan(start\_pt, end\_pt, 1.0)

NOTE: arcs can be a maximum of 180 degrees

Proper es

.Bulge

R/O - double - signed bulge for arc

.IsAntiClockwise

R/O - bool - true if arc is an clockwise (same as .IsCCW)

.IsCCW

R/O - bool - true if arc is counter clockwise (same as . IsAn Clockwise)

.IsClockwise

R/O - bool - true if arc is clockwise (same as . IsCW )

.IsCW

R/O - bool - true if arc is clockwise (same as . IsClockwise)

Methods

:ArcMidPoint()

Returns a Point3D holding the mid point of the arc

:RadiusAndCentre(Point3D ret\_centre\_pt)

Calculate the centre point of the arc and the radius. The method returns the radius and also sets the posi on of the passed point to the centre point posi on of the arc.

ret\_centre\_pt - Point3D - returned 3d posi on for centre point of arc (not mid-point) Example Code

local arc = ArcSpan(Point3D(0,0,0), Point3D(1,0,0), 1.0) local centre = Point3D()

local radius = arc:RadiusAndCentre(centre) MessageBox("Arc Radius = " .. radius ..

" Centre Point = " .. centre.x .. "," .. centre.y )

:SetBulge(bulge)

Set the bulge factor for this arc

Bulge - double - bulge factor for arc

Global helper methods

ArcBulgeFromMidPoint(Point2D start, Point2D end, Point2D mid) Return the bulge factor for an arc through the specified 3 points

start -Point2D -Start point for arc end -Point2D -End point for arc

mid -Point2D -mid point for arc ( NOT centre point)

## BezierSpan

This object represents a bezier span in a Contour. It is derived from the Span class and all methods and proper es of the span class are valid on this object as well. As well as crea ng spans to append to a contour you can use the BezierTo methods to avoid span crea on.

Constructors

BezierSpan(Point2D start\_pt, Point2D end\_pt, Point2D ctrl\_pt\_1, Point2D ctrl\_pt\_1) - Constructor

A new span represen ng a bezier is created within a Lua script using this contructor method

start\_pt – Point2D – 2D posi on for start point of span

end\_pt – Point2D – 2D posi on for end point of span- must have same Z values as start ctrl\_pt\_1 - Point2D – 2D posi on for first control point ctrl\_pt\_2 - Point2D – 2D posi on for second control point e.g

local start\_pt = Point2D(0, 0) local end\_pt = Point2D(1.0, 0) local ctrl\_1 = Point2D(0.2, 0.2) local ctrl\_2 = Point2D(0.8, 0.2) local bez\_span = BezierSpan(start\_pt, end\_pt, ctrl\_1, ctrl\_2)

BezierSpan(Point3D start\_pt, Point3D end\_pt, Point2D ctrl\_pt\_1, Point2D ctrl\_pt\_1) - Constructor

A new span represen ng a bezier is created within a Lua script using this contructor method

start\_pt - Point3D – 3D posi on for start point of span

end\_pt - Point3D – 3D posi on for end point of span- must have same Z values as start ctrl\_pt\_1 - Point2D – 2D posi on for first control point ctrl\_pt\_2 - Point2D – 2D posi on for second control point e.g

local start\_pt = Point3D(0, 0, 0) local end\_pt = Point3D(1.0, 0, 0) local ctrl\_1 = Point2D(0.2, 0.2) local ctrl\_2 = Point2D(0.8, 0.2) local bez\_span = BezierSpan(start\_pt, end\_pt, ctrl\_1, ctrl\_2)

## Span Helper Methods

The following methods are used to convert a Span object to the specific span type

CastSpanToLineSpan(Span span)

Returns a LineSpan object for the passed span object. The passed span objects .IsLineType property must be true. span - Span - the span we are cas ng to a line

CastSpanToArcSpan(Span span)

Returns a LineSpan object for the passed span object. The passed span objects. IsArcType property must be true. span - Span - the span we are cas ng to an arc

CastSpanToBezierSpan(Span span)

Returns aBezierSpan object for the passed span object. The passed span objects .IsBezierType property must be true.

span - Span - the span we are cas ng to a bezier

### Low Level Geometry

To make programming scripts easier, Vectric provide access to a range of low level geometric primi ves which can greatly simplify wri ng scripts which involve geometry manipula on.

The Point2D and Point3D primi ves are widely used to set the end points of spans and while building geometry. The Box2D and Box3D objects are extremely useful for determining the size and posi on of objects. The Vector objects can make a lot of tasks which rely on calcula on of distances between points and transforma on of objects much easier.

## Point2D

This object represents a 2D point (X and Y values only). It is used by many other objects and can also be used with other geometry objects such as Vector2D to perform calcula ons.

Constructors

Point2D() - Constructor

This construct an un-ini alised point with invalid x and y values which can be tested for using the .IsInvalid property.

Point2D(double x, double y) - Constructor A new point with the specified X and Y values x -double - the value for the x coordinate of the point y -double - the value for the y coordinate of the point

Point2D(Point2D pt) - Constructor

A new point with the same X and Y values as the passed point pt -Point2D - an exis ng point to copy X and Y values from

Proper es

.IsInvalid

Returns true if the point is invalid - not ini alised or :SetInvalid called

.X

R/W - double -x coordinate for point

.x

R/W - double -x coordinate for point

.Y

R/W - double -y coordinate for point

.y

R/W - double -y coordinate for point

Methods

:IsCoincident(Point2D point, double tol)

Returns true if passed point is coincident with this point within passed tolerance point -Point2D -point to check for coincidence

tol -double -maximum distance apart point can be and s ll be considered coincident

:Set(double x, double y)

Set both the x and y values for the point x -double -new x value for point y -double -new y value for point :SetInvalid()

Set the point to invalid state -X and Y will be lost

Opera ons

Matrix2D \* Point2D

Mul plying a matrix by a point returns a transformed point

Point2D - Point2D

Subtrac ng a point from another point will return a Vector2D with the difference between the 2 points.

Point2D + Vector2D

Adding a vector to a point will return a point displaced from original point by adding the vector

Point2D - Vector2D

Subtrac ng a vector from a point will return a point displaced from original point by subtrac ng the vector

## Point3D

This object represents a 3D point (X,Y and Z values). It is used by many other objects and can also be used with other geometry objects such as Vector3D to perform calcula ons.

Constructors

Point3D() - Constructor

This construct an un-ini alised point with invalid x, y and z values which can be tested for using the .IsInvalid property.

Point3D(double x, double y, double z) - Constructor A new point with the specified X, Y and Z values x -double - the value for the x coordinate of the point y -double - the value for the y coordinate of the point z -double - the value for the z coordinate of the point

Point3D(Point3D pt) - Constructor

A new point with the same X, Y and Z values as the passed point pt -Point3D - an exis ng point to copy X, Y and Z values from

Point3D(Point2D pt, double z)

Create a new point with x and y values from passed 2D point and new z value pt -Point2D -2D point holding x and y values for point z -double -new z value for point

Proper es

.X

R/W - double -x coordinate for point

.x

R/W - double -x coordinate for point

.Y

R/W - double -y coordinate for point

.y

R/W - double -y coordinate for point

.Z

R/W - double -z coordinate for point

.z

R/W - double -z coordinate for point

.IsInvalid

Returns true if the point is invalid - not ini alised or :SetInvalid called

Methods

:IsCoincident(Point3D point, double tol)

Returns true if passed point is coincident with this point within passed tolerance point -Point3D -point to check for coincidence tol -double -maximum distance apart point can be and s ll be considered coincident

:Set(double x, double y, double z)

Set new x, y and z values for the point x -double -new x value for point y -double -new y value for point z -double -new z value for point

:Set(Point2D pt, double z)

Set new x and y values from passed 2D point and new z value pt -Point2D -2D point holding new x and y values for point z -double -new z value for point

:SetInvalid()

Set the point to invalid state –X,Y and Z will be lost

Opera ons

Matrix2D \* Point3D

Mul plying a matrix by a point returns a transformed point. The Z value is unchanged

Point3D - Point3D

Subtrac ng a point from another point will return a Vector3D with the difference between the 2 points.

Point3D + Vector3D

Adding a vector to a point will return a point displaced from original point by adding the vector

Point3D - Vector3D

Subtrac ng a vector from a point will return a point displaced from original point by subtrac ng the vector

## Vector2D

This object represents a 2D vector (X and Y values only). It is used by many other objects and can also be used with other geometry objects such as Point2D to perform calcula ons.

Constructors

Vector2D() - Constructor

This construct an un-ini alised vector with invalid x and y values which can be tested for using the .IsInvalid property.

Vector2D(double x, double y) - Constructor A vector point with the specified X and Y values x -double - the value for the x coordinate of the vector y -double - the value for the y coordinate of the vector

Vector2D(Vector2D vec) - Constructor

A new vector with the same X and Y values as the passed vector vec -Vector2D - an exis ng vector to copy X and Y values from

Proper es

.IsInvalid

Returns true if the vector is invalid - not ini alised or :SetInvalid called

.Length

Return the length of the vector

.LengthSq

Return the length squared of the vector. If comparing distance between points etc. using LengthSq instead of Length will be faster as it avoids the square root opera on required to find the exact length.

.X

R/W - double -x value for vector

.x

R/W - double -x value for vector

.Y

R/W - double -y value for vector

.y

R/W - double -y value for vector

Methods

:Cross(Vector2D vec)

Return the cross product (a Vector3D) of this vector and passed vector

:Dot(Vector2D vec)

Return the dot product (a double value) of this vector and passed vector

:Normalize()

Normalize this vector to make it a unit vector

:NormalTo()

Return a new vector ‘normal’ to this vector found by rota ng vector 90 degrees clockwise

:Set(double x, double y)

Set both the x and y values for the vector x -double -new x value for vector y -double -new y value for vector

:SetInvalid()

Set the vector to invalid state -X and Y will be lost

Opera ons

Double \* Vector2D

Returns a new vector mul plied by double value

Matrix2D \* Vector2D

Mul plying a matrix by a point returns a transformed vector

Vector2D + Vector2D

Adding a vector to a vector will return a new vector.

Vector2D - Vector2D

Subtrac ng a vector from a point will return a new vector.

- Vector2D

Negates x,y values of vector (reverses vector direc on)

## Vector3D

This object represents a 3D vector (X, Y and Z values). It is used by many other objects and can also be used with other geometry objects such as Point3D to perform calcula ons.

Constructors

Vector3D() - Constructor

This construct an un-ini alised vector with invalid x, y and z values which can be tested for using the .IsInvalid property.

Vector3D(double x, double y, double z) - Constructor A vector point with the specified values x -double - the value for the x coordinate of the vector y -double - the value for the y coordinate of the vector z -double - the value for the z coordinate of the vector

Vector3D(Vector3D vec) - Constructor

A new vector with the same X, Y and Z values as the passed vector vec -Vector3D - an exis ng vector to copy values from

Proper es

.IsInvalid

Returns true if the vector is invalid - not ini alised or :SetInvalid called

.Length

Return the length of the vector

.LengthSq

Return the length squared of the vector. If comparing distance between points etc. using LengthSq instead of Length will be faster as it avoids the square root opera on required to find the exact length.

.X

R/W - double -x value for vector

.x

R/W - double -x value for vector

.Y

R/W - double -y value for vector

.y

R/W - double -y value for vector

.Z

R/W - double -z value for vector

.z

R/W - double -z value for vector

Methods

:Cross(Vector3D vec)

Return the cross product (a Vector3D) of this vector and passed vector

:Dot(Vector3D vec)

Return the dot product (a double value) of this vector and passed vector

:Dot(Point3D vec)

Return the dot product (a double value) of this vector and passed point. The X,Y,Z values of the point are treated exactly the same as for the Vector3D version of this method.

:Normalize()

Normalize this vector to make it a unit vector

:Set(double x, double y, double z)

Set the x, y and z values for the vector x -double -new x value for vector y -double -new y value for vector z -double -new z value for vector

:SetInvalid()

Set the vector to invalid state -X, Y and Z will be lost

Opera ons

Double \* Vector3D

Returns a new vector mul plied by double value

Vector3D + Vector3D

Adding a vector to a vector will return a new vector.

Vector3D - Vector3D

Subtrac ng a vector from a point will return a new vector.

- Vector3D

Negates x,y and z values of vector (reverses vector direc on)

## Box2D

This object represents a 2d bounding box for an object or group of objects.

Constructors

Box2D() - Constructor

This construct an un-ini alised box with invalid values which can be tested for using the .IsInvalid property.

Box2D (Point2D p1, Point2D p2) - Constructor

Create a box which bounds the 2 passed in points

P1 -Point2D - first point within the box

P1 -Point2D - second point within the box

Box2D(Box2D box) - Constructor

A new Box2D with the same values as the passed box box -Box2D - an exis ng box to copy values from

Proper es

.BLC

R/O - Point2D - a point represen ng the bo om le corner of the bounding box

.BRC

R/O - Point2D - a point represen ng the bo om right corner of the bounding box

.Centre

R/O - Point2D - a point represen ng the centre of the bounding box (Bri sh spelling)

.Center

R/O - Point2D - a point represen ng the centre of the bounding box (U.S spelling)

.IsInvalid

R/O - bool - returns true if the box is in an invalid state - not ini alised or SetInvalid called

.MinX

R/O - double - the minimum X value for the bounding box

.MinY

R/O - double - the minimum Y value for the bounding box

.MaxX

R/O - double - the maximum X value for the bounding box

.MaxY

R/O - double - the maximum Y value for the bounding box

.XLength

R/O - double - the length in X for the bounding box

.YLength

R/O - double - the length in Y for the bounding box

.MaxLength

R/O - double - the length of the longest side for the bounding box

.MinLength

R/O - double - the length of the shortest side for the bounding box

.TRC

R/O - Point2D - a point represen ng the top right corner of the bounding box

.TLC

R/O - Point2D - a point represen ng the top le corner of the bounding box

Methods

:Expand(double offset\_dist)

Expand this box on all sides by passed amount offset\_dist -double - distance to offset box by

:IsInside(Point2D point, double tol)

Return true if passed point is inside this box within passed tolerance (which may be 0.0).

point -Point2D - point to test to see if inside box

tol - double - tolerance value to use when checking if inside box

:IsInsideOrOn(Point2D point, double tol)

Return true if passed point is inside or on the edge of this box within passed tolerance (which may be 0.0).

point -Point2D - point to test to see if inside or on box tol - double - tolerance value to use when checking if inside box

:IsInside(Box2D box, bool on\_counts\_as\_int, double tol)

Return true if passed box intersects this box within passed tolerance (which may be 0.0). box -Box2D - box to test to see if intersects this box

on\_counts\_as\_int - bool - if true boxes with shared edges count as intersec ons tol - double - tolerance value to use when checking if inside box

:Intersects(Box2D box, double tol)

Return true if passed box is inside this box within passed tolerance (which may be 0.0). box -Box2D - box to test to see if inside this box

tol - double - tolerance value to use when checking if inside box

:Merge(double x, double y)

Merge a point with the passed x and y values into the box upda ng limits

x -double - x value for point being merged y -double - y value for point being merged

:Merge(Point2D point)

Merge a point into the box upda ng limits point -Point2D - point being merged

:Merge(Box2D box)

Merge another box into this box, upda ng the limits box -Box2D - box being merged

:SetInvalid()

Set the box to invalid state

## Box3D

This object represents a 3D bounding box for objects.

Proper es

.BLC

R/O - Point3D - a point represen ng the bo om le corner of the bounding box

.BRC

R/O - Point3D - a point represen ng the bo om right corner of the bounding box

.Centre

R/O - Point3D - a point represen ng the centre of the bounding box (Bri sh spelling)

.Center

R/O - Point3D - a point represen ng the centre of the bounding box (U.S spelling)

.IsInvalid

R/O - bool - returns true if the box is in an invalid state - not ini alised or SetInvalid called

.MaxLength

R/O - double - the length of the longest side for the bounding box

.MaxX

R/O - double - the maximum X value for the bounding box

.MaxY

R/O - double - the maximum Y value for the bounding box

.MaxZ

R/O - double - the maximum Z value for the bounding box

.MinLength

R/O - double - the length of the shortest side for the bounding box

.MinX

R/O - double - the minimum X value for the bounding box

.MinY

R/O - double - the minimum Y value for the bounding box

.MinZ

R/O - double - the minimum Z value for the bounding box

.TRC

R/O - Point3D - a point represen ng the top right corner of the bounding box

.TLC

R/O - Point3D - a point represen ng the top le corner of the bounding box

.XLength

R/O - double - the length in X for the bounding box

.YLength

R/O - double - the length in Y for the bounding box

.ZLength

R/O - double - the length in Z for the bounding box

Methods

:Expand(double offset\_dist)

Expand this box on all sides by passed amount offset\_dist -double - distance to offset box by

:IsInside(Point3D point, double tol)

Return true if passed point is inside this box within passed tolerance (which may be 0.0).

point -Point3D - point to test to see if inside box

tol - double - tolerance value to use when checking if inside box

:IsInsideOrOn(Point3D point, double tol)

Return true if passed point is inside or on the edge of this box within passed tolerance (which may be 0.0).

point -Point3D - point to test to see if inside or on box tol - double - tolerance value to use when checking if inside box

:IsInside(Box3D box, bool on\_counts\_as\_int, double tol)

Return true if passed box intersects this box within passed tolerance (which may be 0.0).

box -Box2D - box to test to see if intersects this box

on\_counts\_as\_int - bool - if true boxes with shared edges count as intersec ons tol - double - tolerance value to use when checking if inside box

:Intersects(Box3D box, double tol)

Return true if passed box is inside this box within passed tolerance (which may be 0.0). box -Box2D - box to test to see if inside this box tol - double - tolerance value to use when checking if inside box

:Merge(double x, double y, double z)

Merge a point with the passed values into the box upda ng limits x -double - x value for point being merged y -double - y value for point being merged z -double - z value for point being merged

:Merge(Point3D point)

Merge a point into the box upda ng limits point -Point3D - point being merged :Merge(Box3D box)

Merge another box into this box, upda ng the limits box -Box3D - box being merged

:SetInvalid()

Set the box to invalid state

## Matrix2D

This object represents a matrix which can be used to transform 2D points - transla on, rota on scaling etc.

Constructors

Matrix2D objects are usually constructed by the u lity methods listed below.

IdentityMatrix2D()

Returns a Matrix2D which is an iden ty matrix.

Re lectionMatrix2D(Point2D p1, Point2D p1)

Returns a Matrix2D which is performs reflec on about a line between the passed points.

P1 -Point2D - the start point of the line about which reflec on will take place

P2 -Point2D - the end point of the line about which reflec on will take place

RotationMatrix2D(Point2D rotation\_pt, double angle)

Returns a Matrix2D which is performs the rota on about the specified point by the specified amount.

rota on\_pt -Point2D - the point about which rota on will take place angle -double - the angle to rotate by in degrees, posi ve values are CCW.

ScalingMatrix2D(Vector2D scale\_vec)

Returns a Matrix2D which performs the scaling around the origin (0,0) by the specified amount. scale\_vec -Vector2D - the X and Y values of the vector specify the amount to scale in X and Y

ScalingMatrix2D(Point2D scale\_pt, Vector2D scale\_vec)

Returns a Matrix2D which is performs the scaling about the specified point by the specified amount.

scaling\_pt -Point2D - the point about which scaling will take place scale\_vec -Vector2D - the X and Y values of the vector specify the amount to scale in X and Y

TranslationMatrix2D(Vector2D vec)

Returns a Matrix2D which is performs the transla on specified by the vector.

vec -Vector2D - the X and Y values of the vector specify the distances to translate in X and Y

Opera ons

Matrix2D \* Matrix2D

Mul plying a matrix by a matrix returns a new matrix Example Code

local p1 = Point2D(10 ,10)

local xform = RotationMatrix2D(Point2D(0,0) 90) local p3 = xform \* p1

DisplayMessageBox("Transformed p1 = " .. p3.X .. " , " .. p3.Y)

### Toolpaths

The objects documented in this sec on allow the script writer to create toolpaths within the program. There are two types of toolpaths which can be created from scripts.

1. Standard Vectric toolpaths. These are the same toolpaths you would have created within the program using the Profile. Pocke ng, VCarvingn etc. forms. From script you can fill out all the same parameters that you would manually in the program, and the user can edit the toolpaths within the interface to change parameters and recalculate. These toolpaths are created for the currently selected vectors (the selec on can also be controlled from script).
2. External toolpaths. These are toolpaths which are created en rely from within the script. The script creates contours which represent the movement of the tool and hence has complete freedom to do whatever it wants to do. As these toolpaths are not related to the geometry in the main program, the user can not recalculate them, but would need to delete them and re-run the script to recreate a new version.

## ToolpathManager

The ToolpathManager object is used to handle all the toolpaths within the program. Like the

MaterialBlock object there is one instance within the program which is accessed by any ToolpathManager object. For the ini al release, the ToolpathManager is only used to create new toolpaths, load templates and recalculate all toolpaths, for future releases Vectric may offer more access to exis ng toolpaths depending on demand.

Constructor

ToolpathManager - constructor

Returns a new object which refers to the single toolpath manager within the program. e.g local toolpath\_mgr = ToolpathManager()

Proper es

.Count

R/O - integer - the number of toolpaths

.IsEmpty

R/O -bool - true if there are no toolpaths

.NumVisibleToolpaths

R/O - integer - the number of visible toolpaths

Methods

:AddExternalToolpath(ExternalToolpath toolpath)

Adds the passed toolpath to the list of toolpaths in the job. Returns true if the toolpath was added OK, else false. See the documenta on for ExternalToolpath for informa on on how to create the actual toolpath itself.

toolpath -ExternalToolpath - toolpath created in a lua script which is added to job

:CreatePro ilingToolpath(

String name,

Tool tool,

ProfileParameterData profile\_data,

RampingData ramping\_data,

LeadInOutData lead\_data,

ToolpathPosData pos\_data,

GeometrySelector geometry\_selector,

bool create\_2d\_preview, bool interac ve

)

Creates a profiling toolpath for the currently selected vectors. Returns the UUID for the toolpath created.

name-string - Name for the toolpath to be created

tool -Tool -Tool to use for the toolpath

profile\_data -ProfileParameterData -Se ngs for profiling depth, inside / outside etc

ramping\_data -RampingData -Se ngs for ramping lead\_data -LeadInOutData-Se ngs for lead in / out pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc

geometry\_selector-GeometrySelector -Can be used to automa cally select vectors on layers etc

create\_2d\_preview-bool -If true create preview vectors in 2d view interac ve-bool -If true display warnings etc to user Example Code - from Profile\_Vectors\_On\_Layer.lua

function CreateLayerProfileToolpath

( layer\_name, name , start\_depth, cut\_depth, tool\_dia, tool\_stepdown, tool\_in\_mm

)

-- clear current selection local selection = job.Selection selection:Clear()

-- get layer

local layer = job.LayerManager:FindLayerWithName(layer\_name)

if layer == nil then

DisplayMessageBox("No layer found with name = " .. layer\_name) return false end

-- select all closed vectors on the layer

if not SelectVectorsOnLayer(layer, selection, true, false, true) then DisplayMessageBox("No closed vectors found on layer " .. layer\_name) return false end

-- Create tool we will use to machine vectors local tool = Tool(

"Lua End Mill",

Tool.END\_MILL -- BALL\_NOSE, END\_MILL, VBIT

)

tool.InMM = tool\_in\_mm tool.ToolDia = tool\_dia tool.Stepdown = tool\_stepdown tool.Stepover = tool\_dia \* 0.25 tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, INCHES\_SEC ...

tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only tool.ClearStepover = tool\_dia \* 0.5 -- used for vbit only

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData()

pos\_data:SetHomePosition(0, 0, 5.0) pos\_data.SafeZGap = 5.0

-- Create object used to pass profile options local profile\_data = ProfileParameterData()

-- start depth for toolpath profile\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth profile\_data.CutDepth = cut\_depth

-- direction of cut - ProfileParameterData.

-- CLIMB\_DIRECTION or ProfileParameterData.CONVENTIONAL\_DIRECTION profile\_data.CutDirection = ProfileParameterData.CLIMB\_DIRECTION

-- side we machine on - ProfileParameterData.

-- PROFILE\_OUTSIDE, ProfileParameterData.PROFILE\_INSIDE or

-- ProfileParameterData.PROFILE\_ON

profile\_data.ProfileSide = ProfileParameterData.PROFILE\_OUTSIDE

-- Allowance to leave on when machining profile\_data.Allowance = 0.0

-- true to preserve start point positions, false to reorder start

-- points to minimise toolpath length profile\_data.KeepStartPoints = false

-- true if want to create 'square' external corners on toolpath profile\_data.CreateSquareCorners = false

-- true to perform corner sharpening on internal corners (only with v-bits) profile\_data.CornerSharpen = false

-- true to use tabs (position of tabs must already have been defined on vectors) profile\_data.UseTabs = false -- length for tabs if being used profile\_data.TabLength = 5.0 -- Thickness for tabs if being used profile\_data.TabThickness = 1.0

-- if true then create 3d tabs else 2d tabs profile\_data.Use3dTabs = true

-- if true in Aspire, project toolpath onto composite model profile\_data.ProjectToolpath = false

-- Create object used to control ramping local ramping\_data = RampingData() -- if true we do ramping into toolpath ramping\_data.DoRamping = false

-- type of ramping to perform RampingData.RAMP\_LINEAR , RampingData.RAMP\_ZIG\_ZAG

-- or RampingData.RAMP\_SPIRAL

ramping\_data.RampType = RampingData.RAMP\_ZIG\_ZAG

-- how ramp is contrained - either by angle or distance RampingData.CONSTRAIN\_DISTANCE

-- or RampingData.CONSTRAIN\_ANGLE

ramping\_data.RampConstraint = RampingData.CONSTRAIN\_ANGLE -- if we are constraining ramp by distance, distance to ramp over ramping\_data.RampDistance = 100.0

-- if we are contraining ramp by angle , angle to ramp in at (in degrees) ramping\_data.RampAngle = 25.0

-- if we are contraining ramp by angle, max distance to travel before 'zig zaging'

-- if zig zaging

ramping\_data.RampMaxAngleDist = 15

-- if true we restrict our ramping to lead in section of toolpath ramping\_data.RampOnLeadIn = false

-- Create object used to control lead in/out local lead\_in\_out\_data = LeadInOutData()

-- if true we create lead ins on profiles (not for profile on) lead\_in\_out\_data.DoLeadIn = false

-- if true we create lead outs on profiles (not for profile on)

lead\_in\_out\_data.DoLeadOut = false

-- type of leads to create LeadInOutData.LINEAR\_LEAD or LeadInOutData.CIRCULAR\_LEAD lead\_in\_out\_data.LeadType = LeadInOutData.CIRCULAR\_LEAD

-- length of lead to create lead\_in\_out\_data.LeadLength = 10.0 -- Angle for linear leads

lead\_in\_out\_data.LinearLeadAngle = 45 -- Radius for circular arc leads lead\_in\_out\_data.CirularLeadRadius = 5.0

-- distance to 'overcut' (travel past start point) when profiling lead\_in\_out\_data.OvercutDistance = 0.0

-- Create object which can be used to automatically select geometry local geometry\_selector = GeometrySelector()

-- if this is true we create 2d toolpaths previews in 2d view, if false we dont local create\_2d\_previews = true

-- if this is true we will display errors and warning to the user local display\_warnings = true

-- Create our toolpath local toolpath\_manager = ToolpathManager()

local toolpath\_id = toolpath\_manager:CreateProfilingToolpath( name, tool, profile\_data, ramping\_data, lead\_in\_out\_data, pos\_data, geometry\_selector, create\_2d\_previews, display\_warnings )

if toolpath\_id == nil then

DisplayMessageBox("Error creating toolpath") return false

end return true end

CreatePocketingToolpath(

String name,

Tool tool,

Tool area\_clear\_tool,

PocketParameterData pocket\_data,

ToolpathPosData pos\_data, GeometrySelector geometry\_selector, bool create\_2d\_preview, bool interac ve

)

Creates a pocke ng toolpath for the currently selected vectors. Returns the UUID for the toolpath created.

name-string - Name for the toolpath to be created tool -Tool -Tool to use for the toolpath - must not be nil area\_clear\_tool -Tool -Op onal tool for area clearance can be nil pocket\_data -PocketParameterData -Se ng for pocke ng depth, style etc pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc

geometry\_selector-GeometrySelector -Can be used to automa cally select vectors on layers etc create\_2d\_preview-bool -If true create preview vectors in 2d view interac ve-bool -If true display warnings etc to user

Example Code - from Create\_Pocke ng\_Toolpath.lua

-- VECTRIC LUA SCRIPT require “strict”

--[[ ------------ CreatePocketingToolpath ------------

|

| Create a Pocketing toolpath within the program for the currently selected vectors

| Parameters:

| name, -- Name for toolpath

| start\_depth -- Start depth for toolpath below surface of material

| cut\_depth -- cut depth for pocket toolpath

| tool\_dia -- diameter of end mill to use

| tool\_stepdown -- stepdown for tool

| tool\_stepover\_percent -- percentage stepover for tool

| tool\_in\_mm -- true if tool size and stepdown are in mm

|

| Return Values:

| true if toolpath created OK else false

| ]]

function CreatePocketingToolpath( name , start\_depth, cut\_depth, tool\_dia, tool\_stepdown, tool\_stepover\_percent, tool\_in\_mm

)

-- Create tool we will use to machine vectors local tool = Tool(

"Lua End Mill",

Tool.END\_MILL -- BALL\_NOSE, END\_MILL, VBIT

)

tool.InMM = tool\_in\_mm tool.ToolDia = tool\_dia tool.Stepdown = tool\_stepdown

tool.Stepover = tool\_dia \* (tool\_stepover\_percent / 100) tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, INCHES\_SEC,...

tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only

tool.ClearStepover = tool\_dia \* (tool\_stepover\_percent / 100) -- used for vbit only

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData() pos\_data:SetHomePosition(0, 0, 5.0) pos\_data.SafeZGap = 5.0

-- Create object used to pass pocketing options local pocket\_data = PocketParameterData()

-- start depth for toolpath pocket\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth pocket\_data.CutDepth = cut\_depth

-- direction of cut for offet clearance - ProfileParameterData.CLIMB\_DIRECTION or -- ProfileParameterData.CONVENTIONAL\_DIRECTION - NOTE: enum from ProfileParameterData pocket\_data.CutDirection = ProfileParameterData.CLIMB\_DIRECTION

-- Allowance to leave on when machining pocket\_data.Allowance = 0.0

-- if true use raster clearance strategy , else use offset area clearance pocket\_data.DoRasterClearance = true -- angle for raster if using raster clearance pocket\_data.RasterAngle = 0

-- type of profile pass to perform PocketParameterData.PROFILE\_NONE , -- PocketParameterData.PROFILE\_FIRST orPocketParameterData.PROFILE\_LAST pocket\_data.ProfilePassType = PocketParameterData.PROFILE\_LAST

-- if true we ramp into pockets (always zig-zag) pocket\_data.DoRamping = false -- if ramping, distance to ramp over pocket\_data.RampDistance = 10.0

-- if true in Aspire, project toolpath onto composite model pocket\_data.ProjectToolpath = false

-- Create object which can be used to automatically select geometry local geometry\_selector = GeometrySelector()

-- if this is true we create 2d toolpaths previews in 2d view, if false we dont local create\_2d\_previews = true

-- if this is true we will display errors and warning to the user local display\_warnings = true

-- if we are doing two tool pocketing define tool to use for area clearance local area\_clear\_tool = nill

-- we just create a tool twice as large for testing here area\_clear\_tool = Tool(

"Lua Clearance End Mill",

Tool.END\_MILL -- BALL\_NOSE, END\_MILL, VBIT )

area\_clear\_tool.InMM = tool\_in\_mm area\_clear\_tool.ToolDia = tool\_dia \* 2 area\_clear\_tool.Stepdown = tool\_stepdown \* 2 area\_clear\_tool.Stepover = tool\_dia \* 2 \*(tool\_stepover\_percent / 100) area\_clear\_tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, INCHES\_SEC..

area\_clear\_tool.FeedRate = 30 area\_clear\_tool.PlungeRate = 10 area\_clear\_tool.SpindleSpeed = 20000 area\_clear\_tool.ToolNumber = 1

area\_clear\_tool.VBitAngle = 90.0 -- used for vbit only

area\_clear\_tool.ClearStepover = tool\_dia\*2\*(tool\_stepover\_percent/100) -- used for vbit

-- Create our toolpath

local toolpath\_manager = ToolpathManager()

local toolpath\_id = toolpath\_manager:CreatePocketingToolpath( name, tool, area\_clear\_tool, pocket\_data, pos\_data, geometry\_selector, create\_2d\_previews, display\_warnings )

if toolpath\_id == nill then

DisplayMessageBox("Error creating toolpath") return false

end return true end

--[[ ------------ main ------------

|

| Entry point for script

| ]]

function main()

-- Check we have a job loaded job = VectricJob()

if not job.Exists then

DisplayMessageBox("No job loaded") return false; end

local selection = job.Selection if selection.IsEmpty then

MessageBox("Please select one or more vectors to pocket") return false end

local start\_depth = 0.0 local cut\_depth = 5.0 local tool\_dia = 12.0 local tool\_stepdown = 2.5 local tool\_stepover\_percent = 50.0 local tool\_in\_mm = true

local success = CreatePocketingToolpath(

"Lua Pocketing Toolpath", start\_depth, cut\_depth, tool\_dia, tool\_stepdown, tool\_stepover\_percent, tool\_in\_mm )

return success; end

129 :CreateDrillingToolpath(

String name,

Tool tool,

DrillParameterData drilling\_data,

ToolpathPosData pos\_data, GeometrySelector geometry\_selector, bool create\_2d\_preview, bool interac ve

)

Creates a drilling toolpath for the currently selected vectors. Returns the UUID for the toolpath created.

name-string - Name for the toolpath to be created

tool -Tool -Tool to use for the toolpath

drill\_data DrillParameterData -Se ng for drilling depth, peck etc pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc

geometry\_selector-GeometrySelector -Can be used to automa cally select vectors on layers etc create\_2d\_preview-bool -If true create preview vectors in 2d view interac ve-bool -If true display warnings etc to user

Example Code - from Create\_Drilling\_Toolpath.lua

-- VECTRIC LUA SCRIPT require “strict”

--[[ ------- CreateDrillingToolpath -------------------------------------------------- |

| Create a drilling toolpath within the program for the currently selected vectors

| Parameters:

| name, -- Name for toolpath

| start\_depth -- Start depth for toolpath below surface of material

| cut\_depth -- cut depth for drilling toolpath

| retract\_gap -- distance to retract above surface for pecks

| tool\_dia -- diameter of drill to use

| tool\_stepdown -- stepdown for tool

| tool\_in\_mm -- true if tool size and stepdown are in mm

|

| Return Values:

| true if toolpath created OK else false | ]]

function CreateDrillingToolpath( name , start\_depth, cut\_depth, retract\_gap, tool\_dia, tool\_stepdown, tool\_in\_mm)

-- Create tool we will use to machine vectors local tool = Tool(

"Lua Drill",

Tool.THROUGH\_DRILL -- BALL\_NOSE, END\_MILL, VBIT, THROUGH\_DRILL

)

tool.InMM = tool\_in\_mm tool.ToolDia = tool\_dia tool.Stepdown = tool\_stepdown tool.Stepover = tool\_dia \* 0.25

tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, INCHES\_SEC,

-- INCHES\_MIN, FEET\_MIN tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000

tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only tool.ClearStepover = tool\_dia \* 0.5 -- used for vbit only

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData()

pos\_data:SetHomePosition(0, 0, 5.0) pos\_data.SafeZGap = 5.0

-- Create object used to pass profile options local drill\_data = DrillParameterData() -- start depth for toolpath drill\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth drill\_data.CutDepth = cut\_depth

-- if true perform peck drilling drill\_data.DoPeckDrill = retract\_gap > 0.0

-- distance to retract above surface when peck drilling drill\_data.PeckRetractGap = retract\_gap

-- if true in Aspire, project toolpath onto composite model drill\_data.ProjectToolpath = false

-- Create object which can be used to automatically select geometry local geometry\_selector = GeometrySelector()

-- if this is true we create 2d toolpaths previews in 2d view,

-- if false we dont local create\_2d\_previews = true

-- if this is true we will display errors and warning to the user local display\_warnings = true

-- Create our toolpath local toolpath\_manager = ToolpathManager()

local toolpath\_id = toolpath\_manager:CreateDrillingToolpath( name, tool, drill\_data, pos\_data, geometry\_selector, create\_2d\_previews, display\_warnings )

if toolpath\_id == nil then

DisplayMessageBox("Error creating toolpath") return false

end return true end

--[[ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ main -------------------------------------------------- |

| Entry point for script

| ]] function main()

-- Check we have a job loaded

job = VectricJob()

if not job.Exists then

DisplayMessageBox("No job loaded") return false; end

local selection = job.Selection if selection.IsEmpty then

MessageBox("Please select one or more vectors to drill") return false end

local start\_depth = 0.0 local cut\_depth = 5.0 local tool\_dia = 3.0 local retract\_gap = 1.0 local tool\_stepdown = 1.0 local tool\_in\_mm = true

local success = CreateDrillingToolpath(

"Lua Drilling Toolpath", start\_depth, cut\_depth, retract\_gap, tool\_dia, tool\_stepdown, tool\_in\_mm )

return success; end

:CreateVCarvingToolpath(

String name,

Tool tool,

Tool area\_clear\_tool,

VCarveParameterData vcarve\_data,

PocketParameterData pocket\_data,

ToolpathPosData pos\_data,

GeometrySelector geometry\_selector, bool create\_2d\_preview,

bool interac ve

)

Creates a vcarving toolpath for the currently selected vectors. Returns the UUID for the toolpath created.

name-string - Name for the toolpath to be created tool -Tool -Tool to use for the toolpath - must not be nil area\_clear\_tool -Tool -Op onal tool for area clearance can be nil vcarve\_data -VCarveParameterData -Se ngs for vcarving pocket\_data -PocketParameterData -Se ng for pocke ng depth, style etc pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc

geometry\_selector-GeometrySelector -Can be used to automa cally select vectors on layers etc create\_2d\_preview-bool -If true create preview vectors in 2d view interac ve-bool -If true display warnings etc to user

Example Code - from Create\_VCarving\_Toolpath.lua

-- VECTRIC LUA SCRIPT require “strict”

--[[ -------------- CreateVCarvingToolpath --------------

|

| Create a VCarving toolpath within the program for the currently selected vectors

| Parameters:

| name, -- Name for toolpath

| start\_depth -- Start depth for toolpath below surface of material

| flat\_depth -- flat depth - if 0.0 assume not doing flat bottom

| vbit\_angle -- angle of vbit to use

| vbit\_dia -- diameter of VBit to use

| vbit\_stepdown -- stepdown for tool

| tool\_stepover\_percent - percentage stepover for tool

| tool\_in\_mm -- true if tool size and stepdown are in mm

|

| Return Values:

| true if toolpath created OK else false

| ]]

function CreateVCarvingToolpath( name , start\_depth, flat\_depth, vbit\_angle, vbit\_dia, vbit\_stepdown, tool\_stepover\_percent, tool\_in\_mm)

-- Create tool we will use to machine vectors local tool = Tool(

"Lua VBit",

Tool.VBIT -- BALL\_NOSE, END\_MILL, VBIT

)

tool.InMM = tool\_in\_mm tool.ToolDia = vbit\_dia tool.Stepdown = vbit\_stepdown tool.Stepover = vbit\_dia \* (tool\_stepover\_percent / 100)

tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, INCHES\_SEC,

-- INCHES\_MIN, FEET\_MIN tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only

tool.ClearStepover = vbit\_dia \* (tool\_stepover\_percent / 100) \* 2 -- used for vbit only

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData()

pos\_data:SetHomePosition(0, 0, 5.0) pos\_data.SafeZGap = 5.0

-- Create object used to pass pocketing options - used for area clearance only local vcarve\_data = VCarveParameterData()

-- start depth for toolpath vcarve\_data.StartDepth = start\_depth

-- flag indicating if we are creating a flat bottomed toolpath vcarve\_data.DoFlatBottom = flat\_depth > 0.0

-- cut depth for toolpath this is depth below start depth vcarve\_data.FlatDepth = flat\_depth

-- if true in Aspire, project toolpath onto composite model vcarve\_data.ProjectToolpath = false

-- set flag indicating we are using flat tool vcarve\_data.UseAreaClearTool = true

-- Create object used to pass pocketing options - used for area clearance only local pocket\_data = PocketParameterData()

-- start depth for toolpath pocket\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth pocket\_data.CutDepth = flat\_depth

-- direction of cut for offet clearance - ProfileParameterData.CLIMB\_DIRECTION -- or ProfileParameterData.CONVENTIONAL\_DIRECTION - NOTE: enum from ProfileParameterData pocket\_data.CutDirection = ProfileParameterData.CLIMB\_DIRECTION

-- if true use raster clearance strategy , else use offset area clearance pocket\_data.DoRasterClearance = false -- angle for raster if using raster clearance pocket\_data.RasterAngle = 0

-- type of profile pass to perform PocketParameterData.PROFILE\_NONE , -- PocketParameterData.PROFILE\_FIRST orPocketParameterData.PROFILE\_LAST pocket\_data.ProfilePassType = PocketParameterData.PROFILE\_LAST

-- if this is true we create 2d toolpaths previews in 2d view, if false we dont local create\_2d\_previews = true

-- if this is true we will display errors and warning to the user local display\_warnings = true

-- if we are doing two tool pocketing define tool to use for area clearance local area\_clear\_tool = nil

-- we just create a 10mm end mill

area\_clear\_tool = Tool(

"Lua Clearance End Mill",

Tool.END\_MILL -- BALL\_NOSE, END\_MILL, VBIT

)

area\_clear\_tool.InMM = true area\_clear\_tool.ToolDia = 10 area\_clear\_tool.Stepdown = 3 area\_clear\_tool.Stepover = 3

area\_clear\_tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, -- INCHES\_SEC, INCHES\_MIN, FEET\_MIN area\_clear\_tool.FeedRate = 30 area\_clear\_tool.PlungeRate = 10 area\_clear\_tool.SpindleSpeed = 20000 area\_clear\_tool.ToolNumber = 2

-- Create object which can be used to automatically select geometry local geometry\_selector = GeometrySelector()

-- Create our toolpath

local toolpath\_manager = ToolpathManager()

local toolpath\_id = toolpath\_manager:CreateVCarvingToolpath( name, tool, area\_clear\_tool, vcarve\_data, pocket\_data, pos\_data, geometry\_selector, create\_2d\_previews, display\_warnings )

if toolpath\_id == nil then

DisplayMessageBox("Error creating toolpath") return false

end return true end

--[[ -------------- main ------------------------------

| Entry point for script

| ]] function main()

-- Check we have a job loaded job = VectricJob()

if not job.Exists then

DisplayMessageBox("No job loaded") return false; end

local selection = job.Selection if selection.IsEmpty then

MessageBox("Please select one or more vectors to VCarve") return false end

local start\_depth = 0.0 local flat\_depth = 5.0 local vbit\_angle = 90.0 local tool\_dia = 32.0 local tool\_stepdown = 2.5 local tool\_stepover\_percent = 2.0 local tool\_in\_mm = true

local success = CreateVCarvingToolpath(

"Lua VCarving Toolpath", start\_depth, flat\_depth,

vbit\_angle, tool\_dia, tool\_stepdown, tool\_stepover\_percent, tool\_in\_mm )

return success; end

:CreatePrismCarvingToolpath(

String name,

Tool tool,

PrismCarveParameterData prism\_data,

ToolpathPosData pos\_data,

GeometrySelector geometry\_selector, bool create\_2d\_preview,

bool interac ve

)

Creates a prism carving toolpath for the currently selected vectors. Returns the UUID for the toolpath created.

name-string - Name for the toolpath to be created

tool -Tool -Tool to use for the toolpath

prism\_data -PrismParameterData -Se ngs for prism carving pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc

geometry\_selector-GeometrySelector -Can be used to automa cally select vectors on layers etc create\_2d\_preview-bool -If true create preview vectors in 2d view interac ve-bool -If true display warnings etc to user

Example Code - from Create\_Prism\_CarvingToolpath.lua

-- VECTRIC LUA SCRIPT require “strict”

--[[ ------------------- CreatePrismToolpath -------------------

|

| Create a prism toolpath within the program for the currently selected vectors

| Parameters:

| name, -- Name for toolpath

| start\_depth -- Start depth for toolpath below surface of material

| cut\_depth -- cut depth for drilling toolpath vbit\_angle -- angle of vbit to use | vbit\_dia -- diameter of VBit to use

| vbit\_stepdown -- stepdown for tool

| tool\_stepover\_percent - percentage stepover for tool

| tool\_in\_mm -- true if tool size and stepdown are in mm

|

| Return Values:

| true if toolpath created OK else false

| ]]

function CreatePrismToolpath( name , start\_depth, cut\_depth, vbit\_angle,

vbit\_dia, vbit\_stepdown, tool\_stepover\_percent, tool\_in\_mm)

-- Create tool we will use to machine vectors local tool = Tool(

"Lua VBit",

Tool.VBIT -- BALL\_NOSE, END\_MILL, VBIT

)

tool.InMM = tool\_in\_mm tool.ToolDia = vbit\_dia tool.Stepdown = vbit\_stepdown

tool.Stepover = vbit\_dia \* (tool\_stepover\_percent / 100) tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, -- INCHES\_SEC, INCHES\_MIN, FEET\_MIN tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only

tool.ClearStepover = vbit\_dia \* (tool\_stepover\_percent / 100) \* 2 -- used for vbit only

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData()

pos\_data:SetHomePosition(0, 0, 5.0) pos\_data.SafeZGap = 5.0

-- Create object used to pass profile options local prism\_data = PrismCarveParameterData()

-- start depth for toolpath prism\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth prism\_data.CutDepth = cut\_depth

-- direction of cut for offet clearance - ProfileParameterData.CLIMB\_DIRECTION -- or ProfileParameterData.CONVENTIONAL\_DIRECTION - NOTE: enum from ProfileParameterData prism\_data.CutDirection = ProfileParameterData.CLIMB\_DIRECTION

-- calculate the minimum cut depth to fully form the bevel on the current

-- selection with the current tool

local min\_bevel\_depth = prism\_data:CalculateMinimumBevelDepth(tool, true) if min\_bevel\_depth > cut\_depth then

DisplayMessageBox(

"A prism will not be fully formed with a depth of " .. cut\_depth .. "\r\n" ..

"A depth of " .. min\_bevel\_depth .. " is required to fully form the prism"

) end

-- Create object which can be used to automatically select geometry local geometry\_selector = GeometrySelector()

-- if this is true we create 2d toolpaths previews in 2d view, if false we dont local create\_2d\_previews = true

-- if this is true we will display errors and warning to the user local display\_warnings = true

-- Create our toolpath local toolpath\_manager = ToolpathManager()

local toolpath\_id = toolpath\_manager:CreatePrismCarvingToolpath( name, tool, prism\_data, pos\_data, geometry\_selector, create\_2d\_previews, display\_warnings )

if toolpath\_id == nil then

DisplayMessageBox("Error creating toolpath") return false

end

return true end

--[[ ------------------- main -------------------

|

| Entry point for script

| ]] function main()

-- Check we have a job loaded job = VectricJob()

if not job.Exists then

DisplayMessageBox("No job loaded") return false; end

local selection = job.Selection if selection.IsEmpty then

MessageBox("Please select one or more vectors to bevel carve") return false end

local start\_depth = 0.0 local cut\_depth = 20.0 local vbit\_angle = 90.0 local tool\_dia = 32.0 local tool\_stepdown = 2.5 local tool\_stepover\_percent = 2.0 local tool\_in\_mm = true

local success = CreatePrismToolpath(

"Lua Prism Toolpath", start\_depth, cut\_depth, vbit\_angle, tool\_dia, tool\_stepdown, tool\_stepover\_percent, tool\_in\_mm )

return success; end

:CreateFlutingToolpath(

String name,

Tool tool,

Flu ngParameterDataflu ng\_data,

ToolpathPosData pos\_data,

GeometrySelector geometry\_selector, bool create\_2d\_preview, bool interac ve

)

Creates a flu ng toolpath for the currently selected vectors. Returns the UUID for the toolpath created.

name-string - Name for the toolpath to be created tool -Tool -Tool to use for the toolpath flu ng\_data -Flu ngParameterData -Se ngs for flu ng pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc

geometry\_selector-GeometrySelector -Can be used to automa cally select vectors on layers etc create\_2d\_preview-bool -If true create preview vectors in 2d view interac ve-bool -If true display warnings etc to user

Example Code - from Create\_Flu ng\_Toolpath.lua

-- VECTRIC LUA SCRIPT require “strict”

--[[ ----------------- CreateFlutingToolpath -----------------

|

| Create a drilling toolpath within the program for the currently selected vectors

| Parameters:

| name, -- Name for toolpath

| start\_depth -- Start depth for toolpath below surface of material

| cut\_depth -- cut depth for toolpath

| tool\_dia -- diameter of tool to use

| tool\_stepdown -- stepdown for tool

| tool\_in\_mm -- true if tool size and stepdown are in mm

|

| Return Values:

| true if toolpath created OK else false

| ]]

function CreateFlutingToolpath( name , start\_depth, cut\_depth, tool\_dia, tool\_stepdown, tool\_in\_mm)

-- Create tool we will use to machine vectors local tool = Tool(

"Lua Ball Nose", Tool.BALL\_NOSE -- BALL\_NOSE, END\_MILL, VBIT, THROUGH\_DRILL

)

tool.InMM = tool\_in\_mm tool.ToolDia = tool\_dia tool.Stepdown = tool\_stepdown tool.Stepover = tool\_dia \* 0.25

tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, INCHES\_SEC,

-- INCHES\_MIN, FEET\_MIN tool.FeedRate = 30

tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only tool.ClearStepover = tool\_dia \* 0.5 -- used for vbit only

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData()

pos\_data:SetHomePosition(0, 0, 5.0) pos\_data.SafeZGap = 5.0

-- Create object used to pass fluting options local fluting\_data = FlutingParameterData()

-- start depth for toolpath fluting\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth fluting\_data.CutDepth = cut\_depth

-- type of fluting FULL\_LENGTH, RAMP\_START or RAMP\_START\_END fluting\_data.FluteType = FlutingParameterData.RAMP\_START\_END

-- type of ramping RAMP\_LINEAR, RAMP\_SMOOTH fluting\_data.RampType = FlutingParameterData.RAMP\_LINEAR

-- if true use ratio field for controling ramp length else absolute length value fluting\_data.UseRampRatio = false

-- length of ramp as ratio of flute length(range 0 - 1.0) -- (for start and end - ratio is of half length) fluting\_data.RampRatio = 0.2

-- length to ramp over - if UseRampRatio == false fluting\_data.RampLength = 15

-- if true in Aspire, project toolpath onto composite model fluting\_data.ProjectToolpath = false

-- Create object which can be used to automatically select geometry local geometry\_selector = GeometrySelector()

-- if this is true we create 2d toolpaths previews in 2d view, if false we dont local create\_2d\_previews = true

-- if this is true we will display errors and warning to the user local display\_warnings = true

-- Create our toolpath local toolpath\_manager = ToolpathManager()

local toolpath\_id = toolpath\_manager:CreateFlutingToolpath( name, tool, fluting\_data, pos\_data, geometry\_selector, create\_2d\_previews, display\_warnings )

if toolpath\_id == nil then

DisplayMessageBox("Error creating toolpath") return false

end return true

end

--[[ ----------------- main -----------------

|

| Entry point for script

| ]] function main()

-- Check we have a job loaded job = VectricJob()

if not job.Exists then

DisplayMessageBox("No job loaded") return false; end

local selection = job.Selection if selection.IsEmpty then

MessageBox("Please select one or more vectors to flute") return false end

local start\_depth = 0.0 local cut\_depth = 5.0 local tool\_dia = 15.0 local tool\_stepdown = 5.0 local tool\_in\_mm = true

local success = CreateFlutingToolpath(

"Lua Fluting Toolpath", start\_depth, cut\_depth, tool\_dia, tool\_stepdown, tool\_in\_mm )

return success; end

:CreateRoughingToolpath( - Aspire Only String name,

Tool tool,

RoughingParameterData roughing\_ data,

ToolpathPosData pos\_data,

GeometrySelector geometry\_selector, bool interac ve

)

Creates a roughing toolpath. Returns the UUID for the toolpath created.

NOTE: This method is only available in Aspire

name-string - Name for the toolpath to be created

tool -Tool -Tool to use for the toolpath

roughing\_data -RoughingParameterData -Se ngs for roughing, strategy etc pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc

geometry\_selector-GeometrySelector -Can be used to automa cally select vectors on layers etc interac ve-bool -If true display warnings etc to user Example Code from 05\_Roughing\_Toolpath.lua

function CreateRoughingToolpath()

-- Toolpath name

local name = "Lua Roughing Toolpath"

-- Metric unit parameters

local tool\_dia = 6.35 -- Quarter of an inch local tool\_stepdown = 5 local tool\_stepover\_percent = 40

-- Create tool we will use to machine vectors local tool = Tool("Lua End Mill", Tool.END\_MILL)

tool.InMM = true tool.ToolDia = tool\_dia tool.Stepdown = tool\_stepdown

tool.Stepover = tool\_dia \* (tool\_stepover\_percent / 100)

tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, INCHES\_SEC, INCHES\_MIN, FEET\_MIN tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only

tool.ClearStepover = tool\_dia \* (tool\_stepover\_percent / 100) -- used for vbit only

-- we will set home position and safe z relative to material block size local mtl\_block = MaterialBlock() local mtl\_box = mtl\_block.MaterialBox local mtl\_box\_blc = mtl\_box.BLC

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData()

pos\_data:SetHomePosition(mtl\_box\_blc.x, mtl\_box\_blc.y, mtl\_box.TRC.z + (mtl\_block.Thickness

\* 0.2))

pos\_data.SafeZGap = mtl\_block.Thickness \* 0.1

-- Pocketing parameters

local start\_depth = 0 local cut\_depth = mtl\_box.ZLength

-- Create object used to pass roughing options local roughing\_data = RoughingParameterData()

-- start depth for toolpath roughing\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth roughing\_data.CutDepth = cut\_depth

-- Machining allowance

roughing\_data.MachiningAllowance = 0.0

-- Allowance to leave on when machining - (this is different from machining allowance) roughing\_data.Allowance = 0.0

-- if true use z level roughing roughing\_data.DoZLevelRoughing = true -- z level clearance strategy

roughing\_data.ZLevelStrategy = RoughingParameterData.RASTER\_X roughing\_data.ZLevelProfile = RoughingParameterData.LAST

-- angle for raster if using raster clearance roughing\_data.RasterAngle = 0

-- if true we ramp into pockets (always zig-zag) roughing\_data.DoRamping = false -- if ramping, distance to ramp over roughing\_data.RampDistance = 10.0

-- Create object which can used to automatically select geometry on layers etc local geometry\_selector = GeometrySelector()

-- if this is true we will display errors and warning to the user local display\_warnings = true

-- Create our toolpath

local toolpath\_manager = ToolpathManager()

local toolpath\_id = toolpath\_manager:CreateRoughingToolpath( name, tool, roughing\_data, pos\_data, geometry\_selector, display\_warnings )

if not toolpath\_id then

DisplayMessageBox("Error creating toolpath") end end

:CreateFinishingToolpath( - Aspire Only String name,

Tool tool,

PocketParameterData pocket\_data,

ToolpathPosData pos\_data, GeometrySelector geometry\_selector, bool create\_2d\_preview,

bool interac ve

)

Creates a finishing toolpath for the currently selected vectors. Returns the UUID for the toolpath created.

NOTE: This method is only available in Aspire

name-string - Name for the toolpath to be created

tool -Tool -Tool to use for the toolpath

pocket\_data -PoketParameterData - Se ng for pocke ng depth, style etc pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc

geometry\_selector-GeometrySelector -Can be used to automa cally select vectors on layers etc create\_2d\_preview-bool -If true create preview vectors in 2d view interac ve-bool -If true display warnings etc to user

Example Code from 04\_Finishing\_Toolpath.lua

function CreateFinishingToolpath()

-- Toolpath name

local name = "Lua Finishing Toolpath"

-- Metric unit parameters

local tool\_dia = 3.175 -- 8th of an inch local tool\_stepdown = 5 local tool\_stepover\_percent = 10

-- Create tool we will use to machine vectors local tool = Tool("Lua Ball Nose", Tool.BALL\_NOSE)

tool.InMM = true tool.ToolDia = tool\_dia tool.Stepdown = tool\_stepdown

tool.Stepover = tool\_dia \* (tool\_stepover\_percent / 100)

tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, INCHES\_SEC, INCHES\_MIN, FEET\_MIN tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only

tool.ClearStepover = tool\_dia \* (tool\_stepover\_percent / 100) -- used for vbit only

-- we will set home position and safe z relative to material block size local mtl\_block = MaterialBlock() local mtl\_box = mtl\_block.MaterialBox local mtl\_box\_blc = mtl\_box.BLC

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData()

pos\_data:SetHomePosition(mtl\_box\_blc.x, mtl\_box\_blc.y, mtl\_box.TRC.z +

(mtl\_block.Thickness \* 0.2))

pos\_data.SafeZGap = mtl\_block.Thickness \* 0.1

-- Pocketing parameters local start\_depth = 0 local cut\_depth = mtl\_box.ZLength

-- Create object used to pass pocketing options local pocket\_data = PocketParameterData()

-- start depth for toolpath pocket\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth pocket\_data.CutDepth = cut\_depth

-- direction of cut for offet clearance - ProfileParameterData.CLIMB\_DIRECTION or ProfileParameterData.CONVENTIONAL\_DIRECTION - NOTE: enum from ProfileParameterData pocket\_data.CutDirection = ProfileParameterData.CLIMB\_DIRECTION

-- Allowance to leave on when machining pocket\_data.Allowance = 0.0

-- if true use raster clearance strategy , else use offset area clearance pocket\_data.DoRasterClearance = true -- angle for raster if using raster clearance pocket\_data.RasterAngle = 0

-- type of profile pass to perform PocketParameterData.PROFILE\_NONE , PocketParameterData.PROFILE\_FIRST orPocketParameterData.PROFILE\_LAST pocket\_data.ProfilePassType = PocketParameterData.PROFILE\_LAST

-- if true we ramp into pockets (always zig-zag) pocket\_data.DoRamping = false -- if ramping, distance to ramp over pocket\_data.RampDistance = 10.0

-- if true in Aspire, project toolpath onto composite model pocket\_data.ProjectToolpath = false

-- Create object which can used to automatically select geometry on layers etc local geometry\_selector = GeometrySelector()

-- if this is true we create 2d toolpaths previews in 2d view, if false we dont local create\_2d\_previews = true

-- if this is true we will display errors and warning to the user local display\_warnings = true

-- Create our toolpath

local toolpath\_manager = ToolpathManager()

local toolpath\_id = toolpath\_manager:CreateFinishingToolpath( name, tool, pocket\_data, pos\_data, geometry\_selector, create\_2d\_previews, display\_warnings )

if not toolpath\_id then

DisplayMessageBox("Error creating toolpath") end end

:LoadToolpathTemplate(string template\_path)

Load a toolpath template from passed file. Returns true if template loaded OK, else false template\_path - string - Full path to template file to load

Example Code

-- VECTRIC LUA SCRIPT require "strict"

--[[ -------------- main ---------------------

|

| Entry point for script

| ]] function main()

-- Check we have a job loaded local job = VectricJob()

if not job.Exists then

DisplayMessageBox("No job loaded") return false; end

local template\_path = "c:\\temp\\TestToolpathTemplate.ToolpathTemplate" local toolpath\_manager = ToolpathManager()

if not toolpath\_manager:LoadToolpathTemplate(template\_path) then

MessageBox("Failed to load template " .. template\_path) return false

end

MessageBox("Loaded template " .. template\_path)

local calc\_result = toolpath\_manager:RecalculateAllToolpaths() if calc\_result == nil then

MessageBox("Recalculate all toolpaths failed") else

MessageBox("Results from recalculate all\n" .. calc\_result) end

return true; end

:CopyToolpathWithId(UUID id, bool insert\_at\_end, bool rename)

Copy (duplicate) toolpath with passed id - returns copied toolpath and adds to toolpath list

id -UUID - Id of toolpath to copy

insert\_at\_end -bool -If true, new toolpath is inserted at end of list, else following original to copy rename -bool -If true, new toolpath is automa cally renamed, else keeps same name

:DeleteAllToolpath ()

Delete ALL toolpaths in job.

:DeleteToolpath (Toolpath toolpath)

Delete passed toolpath - make sure you do not refer to toolpath in script a er this call!

toolpath -Toolpath - Toolpath to delete

:DeleteToolpathWithId(UUID id) Delete toolpath with passed id id - UUID -id of toolpath to delete

:Find(UUID id)

Returns the POSITION in the list of the toolpath with hte passed id - nil if no toolpath with id found id - UUID -id of toolpath to find posi on for

:GetAt(pos)

Returns the toolpath at the passed posi on pos - POSITION - current posi on in list

:GetGroupToolpathWithIndex(UUID id, integer index)

Return the ‘n’th toolpathof the passed group id. Toolpaths created together such as the two toolpaths from a flat bo om v-carve with two tools share a unique groupid property.

id -UUID - Id of toolpath group

index -integer -Index (inrange 0 to n-1) of toolpath in group to return

:GetNext(pos)

Returns the toolpath at the current posi on AND a new value for posi on poin ng to the next item

in the list (or nil if at end of list)

pos - POSITION - current posi on in list

Example - note that GetNext(pos) is returning two values ...

local pos = toolpath\_manager:GetHeadPosition() local toolpath while pos ~= nil do

toolpath, pos = toolpath \_manager:GetNext(pos) DO SOMETHING WITH TOOLPATH ....

end

:GetNumberOfToolpathsInGroup(UUID id)

Return the number of toolpaths which have the passed group id. Toolpaths created together such as the two toolpaths from a flat bo om v-carve with two tools share a unique groupid property. id -UUID - Id of toolpath group to count members for

:GetPrev(pos)

Returns the toolpath at the current posi on AND a new value for posi on, poin ng to the previous

item in the list (or nil if at start of list) pos - POSITION - current posi on in list

:GetSelectedToolpath()

Returns the currently selected toolpath or nil if no toolpath currently selected

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list of toolpaths

:GetTailPosition()

Returns a POSITION variable to allow access to the tail of the list of toolpaths

:RecalculateAllToolpaths()

Recalculate all toolpaths in the job. Returns a string containing a list of all the toolpaths recalculated if it succeeds, else nil.

WARNING: Recalcula ng all toolpaths will destroy the original toolpaths and create new ones with the same ID. Do not try and reuse a toolpath variable a er this call. See :RecalculateToolpath() for informa on on how to retrieve a specific toolpath a er it has been recalculated. Example Code local toolpath\_manager = ToolpathManager()

local calc\_result = toolpath\_manager:RecalculateAllToolpaths() if calc\_result == nil then

MessageBox("Recalculate all toolpaths failed") else

MessageBox("Results from recalculate all\n" .. calc\_result) end

:GetActiveSideNode()

Returns a ToolpathNode variable represen ng the ac ve side of the job

:RecalculateToolpath(Toolpath toolpath)

Recalculates passed toolpath. Returns true if toolpath recalculated ok toolpath - Toolpath - Toolpath to recalculate

WARNING: The passed toolpath is invalid a er this call as a new toolpath with the same id is created internally. If you wish to con nue accesing the toolpath a er recalcula ng, save its id BEFORE calling :RecalculateToolpath() and then use :Find(id) to find the posi on of the recalculated toolpath and use :GetAt(posi on) to return the recalculated toolpath. Example Code

local orig\_id = luaUUID() orig\_id:SetId(toolpath.Id) local recalced\_toolpath = nil

if toolpath\_manager:RecalculateToolpath(toolpath) then local recalced\_pos = toolpath\_manager:Find(orig\_id.RawId) if recalced\_pos == nil then

MessageBox("Failed to find toolpath after calculation") return false else

recalced\_toolpath = toolpath\_manager:GetAt(recalced\_pos) end end

:SaveToolpathAsTemplate(Toolpath toolpath, string template\_path)

Save a toolpath as a template to passed file. Returns true if template saved, else false. If passed toolpath is part of a group (e.g flat bo om vcarving) all toolpaths in group are saved to the template

toolpath - Toolpath - Toolpath to save as a template template\_path - string - Full path to template file to save

:SaveVisibleToolpathsAsTemplate( string template\_path)

Save all visible toolpaths as a template to passed file. Returns true if template saved, else false.

template\_path - string - Full path to template file to save

:ToolpathModi ied(Toolpath toolpath)

This method must be called if parameters for a toolpath are modified from script. E.g tool parameters are changed. If this method is not called the ui will not show the changes you have made.

toolpath - Toolpath - Toolpath which has been modified

:ToolpathWithNameExists(string name)

Returns true if there is one or more exis ng toolpaths with passed name name -string - Toolpath name to check for

id -UUID - Id of toolpath to delete

:UndrawAllToolpath ()

Undraw (set visibility to false) ALL toolpaths in job.

:SetAllToolpathsVisibility (bool visibility) Set visibility for all toolpaths in job.

visibility -bool - true to draw all toolpaths, false to stop drawing them

## Toolpath

The Toolpath object represents a toolpath within the program. Toolpaths are never created directly, they are created using the CreateXXX methods of ToolpathManager.

Proper es

.FirstPoint

R/O- Point3D- Posi on of first point in toolpath - start of first plunge move

.GroupId

R/O-UUID - returns group id of toolpath, if this toolpath is part of a group (e.g from flat bo omed vcarving). All toolpaths in a group will have the same group id

.Id

R/O-UUID - returns id of toolpath, all toolpaths have a unique id

.InMM

R/O- bool - flag indica ng if toolpath is in mm or inches

.LastPoint

R/O- Point3D- Posi on of last point in toolpath - end of last retract move

.Name

R/W- string - name of toolpath - call ToolpathManager:ToolpathModified() a er changing

.Notes

R/W- string - notes for toolpath - call ToolpathManager:ToolpathModified() a er changing

.PositionData

R/O- ToolpathPosData- Posi on data (home pos, safe z etc) for toolpath

.Tool

R/O- Tool- returns a Tool object which references the tool in the toolpath and can be used to update

the tool data. See Tool:UpdateToolParameters()for more details. Also call ToolpathManager:ToolpathModified() a er changing tool values.

.SheetId

R/O- UUID- The id of the sheet that this toolpath belongs to

.SideId

R/O – UUID – The side id on which this toolpath belongs

To compare this to another UUID, use luaUUID(toolpath.SideId).IsEqual(my\_other\_id)

Methods

:MachiningTime()

Return es mated machining me for toolpath. This es mate will use the Rapid Feed rate and Scale Factor set within the program on the ‘Toolpaths Summary’ page.

:ReplaceTool(Tool tool)

Replace the exis ng tool with the passed tool.

tool -Tool -tool which will replace the current tool in the toolpath

:Statistics()

Return a ToolpathStats object with informa on about toolpath such as tolal length of feed rate moves etc.

:Transform(Matrix2D xform)

Transform the toolpath using passed xform. Only transla on and rota on are supported xform -Matrix2D -transforma on matrix to apply to toolpath

## ToolpathNode

Anything represented in the Toolpaths List in the so ware has an associated ToolpathNode (Toolpaths, Merged Toolpaths, Array Copy Toolpaths and Toolpath Groups). ToolpathNodes are used to represent toolpaths in the toolpath manager and can be used to determine parent-child rela onships between toolpaths e.g. to find all the children of a merged toolpath.

In order to access the node represen ng the toolpaths of the ac ve side of a job, ToolpathManager:GetAc veSideNode() is used.

Methods

:HasChildren()

Return a bool indica ng whether this ToolpathNode has any children.

:GetNumberOfChildren()

Return an int equal to the number of children belonging to this ToolpathNode.

:GetChild(int index)

Return a ToolpathNode child of the current ToolpathNode.

index - integer - index of the child

:GetToolpath()

Return the Toolpath which is associated with this ToolpathNode.

:GetName()

Return the name string of this ToolpathNode. Example Code - from Setup\_Sheet.lua

local function TraverseNodesAndFillBoxes(node, level) local toolpathisonsheet = ToolpathIsOnSheet(node, sheet\_number)

if (level > 0 and toolpathisonsheet) then CreateAndFillToolpathBox(node, level) end

if node:HasChildren() then

local num\_children = node:GetNumberOfChildren() for i = 0, (num\_children - 1) do local childnode = node:GetChild(i) level = level + 1

TraverseNodesAndFillBoxes(childnode, level) level = level -1 end end

if (level > 0 and toolpathisonsheet) then

CloseBox() end end

local tpmanager = ToolpathManager()

local activesidenode = tpmanager:GetActiveSideNode()

TraverseNodesAndFillBoxes(activesidenode, 0)

## ToolDatabase

The ToolDatabase object gives access to the single Tool database within the program. Currently this object can only be used to select an exis ng tool from the database.

Constructor

ToolDatabase - constructor

Returns a new object which gives access to the single Tool database for the program. e.g

local tooldb = ToolDatabase()

Methods

:SelectTool()

Displays the Tool Database dialog and allows the user to choose a tool which is returned from this method.

e.g

local tool\_database = ToolDatabase(); local tool = tool\_database:SelectTool();

:GetTool(ToolDBId tooldb\_id)

Return the tool from the database with the given Id.

## ToolDBId

This is an object used to locate a selected tool with its loca on in the tool database. It is retrieved by the tool through Tool.ToolDBId and is useful for saving and loading defaults.

Constructor

ToolDBId(string section, string key\_pre ix) – constructor Create an Id object from the contents of the registry

Sec on – string – sec on name in the registry

Key\_prefix – string – prefix for the registry keys (could be empty) in case mul ple tools are used in the same gadget

Methods

SaveDefaults(string section, string key\_pre ix)

Save the Id to the registry

Sec on – string – sec on name in the registry

Key\_prefix – string – prefix for the registry keys (could be empty) in case mul ple tools are used in the same gadget

LoadDefaults(string section, string key\_pre ix)

Load the Id from the registry

Sec on – string – sec on name in the registry

Key\_prefix – string – prefix for the registry keys (could be empty) in case mul ple tools are used in the same gadget

## Tool

The Tool object represents a tool within the program. All the methods for crea ng toolpaths take one or more Tool objects as arguments. Any of the supported tool types can be accessed from the ToolDatabase as shown in the sec on describing the ToolDatabase:SelectTool() method. In addi on the most common tool types (BallNose, EndMill, VBit and Drill) can be created programma cally from Lua.

Constructor

Tool(string name, ToolType tool\_type) - constructor

Create a new tool. Only a limited number of tool types are supported for crea on from script. To access the full range of tools use the ToolDatabase:SelectTool() method to allow users to select tools from the tool database. name-string - Name for the too to be created tool\_type -integer -Type of tool to create. Valid values are ...

Tool.BALL\_NOSE

Tool.END\_MILL

Tool.VBIT

Tool.THROUGH\_DRILL

NOTE: If you change ANY parameters / proper es on a tool retrieved from an EXISTING toolpath, you MUST call UpdateParameters() a er you have finished edi ng tool proper es to flush the changes through to the toolpath.

Proper es

.ClearStepover

R/W - double - Clearance pass stepover for tool - VBits only

.InMM

R/W - bool - flag indica ng if tool is in mm or inches

.FeedRate

R/W - double - Feedrate for tool - see RateUnits for the units

.Name

R/O - string - returns name of tool

.Notes

R/W - string – Notes field for tool . (Aspire 4.015/VCP 7.015 onwards only)

.PlungeRate

R/W - double - Feedrate for tool - see RateUnits for the units

.RateUnits

R/W - integer - Feed rate units for tool. Valid values are

Tool.MM\_SEC

Tool.MM\_MIN

Tool.METRES\_MIN

Tool.INCHES\_SEC

Tool.INCHES\_MIN

Tool.FEET\_MIN

.RateUnitsText

R/O -string - Feed rate units for tool as a string for display e.g “mm/sec” or “inch/sec”

.SpindleSpeed

R/W - integer - Spindle speed for tool

.Stepdown

R/W - double - Stepdown for tool

.Stepover

R/W - double -Stepover for tool

.ToolDB\_Location

R/O - string - returns path to tool in tool database - not including tool name

.ToolType

R/O - integer - Returns an integer indica ng the type of the tool – values are …

Tool.BALL\_NOSE

Tool.END\_MILL

Tool.RADIUSED\_END\_MILL

Tool.VBIT

Tool.ENGRAVING

Tool.RADIUSED\_ENGRAVING

Tool.THROUGH\_DRILL

Tool.FORM\_TOOL

Tool.DIAMOND\_DRAG

Tool.RADIUSED\_FLAT\_ENGRAVING

.ToolTypeText

R/O - string - Returns a string indica ng the type of the tool

.ToolDia

R/W - double - Diameter of tool

.ToolNumber

R/W - integer - Tool number for tool

.VBit\_Angle

R/W - double - Included angle for VBit tools

.ToolDBId

R/O - ToolDBId – The Id describing the loca on in the tool database

Methods

:ConvertRateUnitsTo(integer new\_units)

Convert the feed and plunge rates for the tool into the passed units.

new\_units-integer -new units for feed and plunge rate. Valid values are the same as for .RateUnits above

:IsCompatibleFeedRates(Tool tool\_to\_check)

Return true if this tool is has same feed, plunge and spindles speeds as passed tool. Does not check geometry or cut depths (see: IsCompa bleTool and :IsCompa bleCutDepthss) tool\_to\_check-Tool -tool we check for compa bility

:IsCompatibleTool(Tool tool\_to\_check)

Return true if this tool is compa ble (same type and geometry) as passed tool. Does not check cut depths or feedrates (see :IsCompa bleCutDepths and :IsCompa bleFeedRates for this data) tool\_to\_check-Tool -tool we check for compa bility

:IsCompatibleCutDepths(Tool tool\_to\_check)

Return true if this tool is has same cut depths as passed tool. Does not check geometry or feedrates

(see: IsCompa bleTool and :IsCompa bleFeedRates)

Tool\_to\_check-Tool -tool we check for compa bility

:FlatRadius(bool in\_mm)

Retrieve the flat radius for tools flat tools otherwise returns zero.

in\_mm- bool -If true return the p radius in millimetres

:GetBool(string parameter\_name, bool default\_value)

Retrieve a Boolean flag (true / false) with the passed name, if no value with passed name returns passed default value. (Aspire 4.015/VCP 7.015 onwards only)

parameter\_name - string - the name of the parameter default\_value - bool - the value which will be returned if there is no exis ng value stored

:GetDouble(string parameter\_name, double default\_value,)

Retrieve a double with the passed name, if no value with passed name returns passed default value.

(Aspire 4.015/VCP 7.015 onwards only)

parameter\_name - string - the name of the parameter default\_value - double - the value which will be returned if there is no exis ng value stored

:GetInt(string parameter\_name, integer default\_value)

Retrieve an integer with the passed name, if no value with passed name returns passed default value. (Aspire 4.015/VCP 7.015 onwards only)

parameter\_name - string - the name of the parameter value - integer - the value which will be returned if there is no exis ng value stored

:ParameterExists(string parameter\_name, utParameterType type)

Returns true if there is an exis ng parameter with passed name and type. (Aspire 4.015/VCP 7.015 onwards only)

parameter\_name - string - the name of the parameter type - utParameterType - the type of parameter

:SetBool(string parameter\_name, bool value)

Store a Boolean flag (true / false) with the passed name and value. (Aspire 4.015/VCP 7.015 onwards only)

parameter\_name - string - the name which will be used to store and retrieve the value value - bool - the value which will be stored in the parameter list

:SetDouble(string parameter\_name, double value)

Store a double with the passed name and value. (Aspire 4.015/VCP 7.015 onwards only)

parameter\_name - string - the name which will be used to store and retrieve the value value - double - the value which will be stored in the parameter list

:SetInt(string parameter\_name, integer value)

Store an integer with the passed name and value. (Aspire 4.015/VCP 7.015 onwards only)

parameter\_name - string - the name which will be used to store and retrieve the value value - integer - the value which will be stored in the parameter list

:SetString(string parameter\_name, string value)

Store a string with the passed name and value. (Aspire 4.015/VCP 7.015 onwards only)

parameter\_name - string - the name which will be used to store and retrieve the value value - string - the value which will be stored in the parameter list

:TipRadius(bool in\_mm)

Retrieve the p radius for tools of type Tool.RADIUSED\_END\_MILL or Tool.RADIUSED\_ENGRAVING otherwise returns zero.

in\_mm- bool -If true return the p radius in millimetres

:UpdateParameters()

If edi ng a tool on an exis ng toolpath update feedrates and other parameters set on the tool to new values .

NOTE: You MUST call this method if you have edited any parameters on an exis ng tool, otherwise your changes will be ignored!

Example Code

-- Create tool we will use to machine vectors local tool = Tool(

"Lua End Mill",

Tool.END\_MILL -- BALL\_NOSE, END\_MILL, VBIT

)

tool.InMM = tool\_in\_mm tool.ToolDia = tool\_dia tool.Stepdown = tool\_stepdown tool.Stepover = tool\_dia \* 0.25

tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, -- INCHES\_SEC, INCHES\_MIN, FEET\_MIN tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only tool.ClearStepover = tool\_dia \* 0.5 -- used for vbit only

## ToolpathPosData

This object is used to pass data to the various toolpath crea on func on in the ToolpathManager object to specify the home posi on and safe z etc.

Constructor

ToolpathPosData() - Constructor

Creates a new ToolpathPosData object with default values.

Proper es

.HomeX

R/O - double - X value for the home posi on (where tool starts from)

.HomeY

R/O - double - Y value for the home posi on (where tool starts from)

.HomeZ

R/O - double - Z value for the home posi on (where tool starts from)

.InMM

R/O - bool - True if program working in mm, else inches

.SafeZ

R/O - double - Absolute Z value for the safe z moves (rapid moves)

.SafeZGap

R/W - double - Get / Set the SafeZ gap - distance above surface of material for rapid moves

.StartZGap

R/W - double - Get / Set the StartZ gap - distance above surface of material when plunges change from rapid to plunge federate.

Methods

:EnsureHomeZIsSafe()

As Home posi on is specified as an absolute value, it is possible to set it programma cally to an invalid value within the block. This method will ensure that it is always at least ‘SafeZ’ above the material surface.

:SetHomePosition(double x, double y, double z)

Set the posi on from which the tool path starts and usually returns to.

x -double -X value for tool home posi on y -double -Y value for tool home posi on z -double -Z value for tool home posi on

## GeometrySelector

This object is used to pass data to the various toolpath crea on func on in the ToolpathManager object to specify automa c selec on of geometry for the toopath when it is calculated. The default constructor for the object will leave the GeometrySelector inac ve, and the toolpath will calculate using the currently selected vectors.

Constructor

GeometrySelector() - Constructor

Creates a new GeomterySelector object with default values.

Proper es

.AllowOpen

R/W - bool – if true selector will allow open vectors in the selec on

.AllowToolDia

R/W - bool – if true selector will use .ToolDia field when selec ng vectors

.CircleDia

R/W - double –Diameter of circles we select if .CircleMatchCircleDia and .SelectCircles is true

.CircleTolerance

R/W - double –tolerance we use when deciding if a circle matches the ToolDia / CircleDia criteria

.CircleMatchAll

R/W - bool – if true we select all circles if .SelectCircles is true

.CircleMatchCircleDia

R/W - bool – if true we select all circles matching .CircleDia if .SelectCircles is true

.CircleMatchToolDia

R/W - bool – if true we select all circles that match the diameter of the tool used for the toolpath if .SelectCircles is true

.GeometryDepthOffset

R/W - double –Offset value to add to depth

.GeometryDepthOffset Formula

R/W - string –Formula for depth offset from geometry

.MixedGroupsOk

R/W - bool – if true groups which only match some criteria are selected

.OnlyOnLayers

R/W - bool – if true only vectors on the layers added with :AddLayerName are selected.

.SelectClosed

R/W - bool – if true closed vectors are selected

.SelectCircles

R/W - bool – if true we are only selec ng circles

.SelectOpen

R/W - bool – if true open vectors are selected

.SetDepthFromGeometry

R/W - bool – if true depth for toolpath is set from selected geometry (which must have been imported from DXF files with depth set)

.ToolDia

R/W - double –Current diameter of circles we select if .CircleMatchToolDia and .SelectCircles is true

Layer List Access

.HaveLayerNames

R/O - bool – true if the geometry selector has a list of layer names

.LayerNameCount

R/O - integer – number of layers

:AddLayerhName(string name)

Add passed layer name to list of names selector uses if .OnlyOnLayers is true. name - string – name of layer to add to list

:GetAtLayerName(POSITION pos)

Returns the layer name at the current posi on pos - POSITION - current posi on in list

:GetLayerNameHeadPosition()

Returns a POSITION variable to allow access to the head of the list of layer names

:GetNextLayerName(POSITION pos)

Returns the layer name at the current posi on AND a new value for posi on poin ng to the next item in the list (or nil if at end of list)

pos - POSITION - current posi on in list

:GetPrevLayerName(POSITION pos)

Returns the layer name at the current posi on AND a new value for posi on, poin ng to the previous item in the list (or nil if at start of list)

pos - POSITION - current posi on in list

:GetLayerNameTailPosition()

Returns a POSITION variable to allow access to the tail (end) of the list of layer names :FindLayerWithName(string name)

Returns the POSITION of layer with passed name or nil if no layer in list with name found name - string – name of layer to find posi on of

Methods

:HasSelectorData(Toolpath toolpath)

Returns true if the passed toolpath has selector data stored with it toolpath -Toolpath –toolpath to check for data

:LoadSelectorData(Toolpath toolpath)

Loads the selector data from passed toolpath into this object data toolpath -Toolpath –toolpath to get data from

:SaveSelectorData(Toolpath toolpath)

Saves the selector data to passed toolpath toolpath -Toolpath –toolpath to set data on

:RemoveAllLayerNames()

Remove all layer names associated with this selector.

## Pro ileParameterData

This object is used to hold the se ngs for a profile toolpath.

Constructor

Pro ileParameterData() - Constructor

Create a new object ready to have its parameters set

e.g local profile\_data = ProfileParameterData()

Proper es

.Allowance

R/W - double - allowance to leave on profile when calcula ng toolpath

.AllowanceFormula

R/W - string -Formula for allowance to leave on profile when calcula ng toolpath

.CutDepth

R/W - double -Final cu ng depth below start depth

.CornerSharpen

R/W - bool - True if want to create 3D ‘corner sharpening’ moves for internal corners

.CreateSquareCorners

R/W - bool - true if want to create ‘square’ external corners

.CutDepthFormula

R/W - string –Op onal formula for final cu ng depth below start depth. The formula is the same as can be entered in the CutDepth field on the normal toolpath page and has access to the same variables (e.g “z \* 0.5” to set cut depth to half the material thickness).

.CutDirection

R/W - integer - Cu ng direc on for toolpath. Valid values are ... ProfileParameterData.CLIMB\_DIRECTION

ProfileParameterData.CONVENTIONAL\_DIRECTION

.KeepStartPoints

R/W - bool - If true, start points of vectors are maintained, else start points are op mised

.Name

R/W - string - The name for the toolpath

.Pro ileSide

R/W - integer - Side of the vector to cut on. Valid values are ...

ProfileParameterData.PROFILE\_OUTSIDE

ProfileParameterData.PROFILE\_INSIDE

ProfileParameterData.PROFILE\_ON

.ProjectToolpath

R/W - bool - If true and used with Aspire, toolpath is projected onto model surface a er calcula on .StartDepth

R/W - double - Start depth for toolpath below material surface

.StartDepthFormula

R/W - string – Op onal formula for start depth for toolpath below material surface. The formula is the same as can be entered in the Start Depth field on the normal toolpath page and has access to the same variables (e.g “z \* 0.5” to set start depth to half the material thickness).

Example Code

-- Create object used to pass profile options local profile\_data = ProfileParameterData()

-- start depth for toolpath profile\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth profile\_data.CutDepth = cut\_depth

-- direction of cut - ProfileParameterData.CLIMB\_DIRECTION or

-- ProfileParameterData.CONVENTIONAL\_DIRECTION

profile\_data.CutDirection = ProfileParameterData.CLIMB\_DIRECTION

-- side we machine on - ProfileParameterData.PROFILE\_OUTSIDE,

-- ProfileParameterData.PROFILE\_INSIDE or ProfileParameterData.PROFILE\_ON profile\_data.ProfileSide = ProfileParameterData.PROFILE\_OUTSIDE

-- Allowance to leave on when machining profile\_data.Allowance = 0.0

-- true to preserve start point positions, false to reorder start points to

-- minimise toolpath length profile\_data.KeepStartPoints = false

-- true if want to create 'square' external corners on toolpath profile\_data.CreateSquareCorners = false

-- true to perform corner sharpening on internal corners (only with v-bits) profile\_data.CornerSharpen = false

-- if true in Aspire, project toolpath onto composite model profile\_data.ProjectToolpath = false

## RampingData

This object is used to hold parameters rela ng to ramping for toolpaths which support ramping.

Constructor

RampingData() - Constructor

Create a new object ready to have its parameters set

e.g local ramping\_data = RampingData()

Proper es

.DoRamping

R/W - bool - If true ramping is performed. If false all other fields are ignored and tool plunges straight down.

.RampAngle

R/W - double -angle (in degrees) to ramp at if constrained by angle

.RampConstraint

R/W - integer - How the ramp is constrained, either by angle or distance. Valid values are RampingData.CONSTRAIN\_DISTANCE RampingData.CONSTRAIN\_ANGLE

.RampDistance

R/W - double - distance to ramp over if constrained by distance

.RampMaxAngleDist

R/W - double -max distance to ramp over if constrained by angle

.RampOnLeadIn

R/W - bool - If true ramps are created on the lead in moves

.RampType

R/W - integer - Type of ramping to perform. Valid values are ...

RampingData.RAMP\_LINEAR

RampingData.RAMP\_ZIG\_ZAG

RampingData.RAMP\_SPIRAL

Example Code

-- Create object used to control ramping local ramping\_data = RampingData() -- if true we do ramping into toolpath ramping\_data.DoRamping = false

-- type of ramping to perform RampingData.RAMP\_LINEAR , RampingData.RAMP\_ZIG\_ZAG

-- or RampingData.RAMP\_SPIRAL

ramping\_data.RampType = RampingData.RAMP\_ZIG\_ZAG

-- how ramp is contrained - either by angle or distance RampingData.CONSTRAIN\_DISTANCE

-- or RampingData.CONSTRAIN\_ANGLE

ramping\_data.RampConstraint = RampingData.CONSTRAIN\_ANGLE -- if we are constraining ramp by distance, distance to ramp over ramping\_data.RampDistance = 100.0

-- if we are contraining ramp by angle , angle to ramp in at (in degrees) ramping\_data.RampAngle = 25.0

-- if we are contraining ramp by angle, max distance to travel before

-- 'zig zaging' if zig zaging ramping\_data.RampMaxAngleDist = 15

-- if true we restrict our ramping to lead in section of toolpath ramping\_data.RampOnLeadIn = false

## LeadInOutData

This object is used to hold parameters rela ng to lead in / out for toolpaths which support leads.

Constructor

LeadInData() - Constructor

Create a new object ready to have its parameters set

e.g local lead\_data = LeadInOutData()

Proper es

.CircularLeadRadius

R/W - double - radius for circular leads

.DoLeadIn

R/W - bool - If true we do lead in else all other fields for lead ins are ignored

.DoLeadOut

R/W - bool - If true we do lead out else all other fields for lead outs are ignored

.LeadType

R/W - integer - Type of lead to create. Valid values are ...

LeadInOutData.LINEAR\_LEAD

LeadInOutData.CIRCULAR\_LEAD

.LeadLength

R/W - double - Length for lead in / out

.LinearLeadAngle

R/W - double - Angle in degrees for linear leads from a perpendicular lead

.OvercutDistance

R/W - double - If greater than 0.0 tool will con nue past entry point by this distance at end of profile

Example Code

-- Create object used to control lead in/out local lead\_in\_out\_data = LeadInOutData()

-- if true we create lead ins on profiles (not for profile on) lead\_in\_out\_data.DoLeadIn = false

-- if true we create lead outs on profiles (not for profile on) lead\_in\_out\_data.DoLeadOut = false

-- type of leads to create LeadInOutData.LINEAR\_LEAD or LeadInOutData.CIRCULAR\_LEAD lead\_in\_out\_data.LeadType = LeadInOutData.CIRCULAR\_LEAD

-- length of lead to create lead\_in\_out\_data.LeadLength = 10.0 -- Angle for linear leads

lead\_in\_out\_data.LinearLeadAngle = 45 -- Radius for circular arc leads lead\_in\_out\_data.CirularLeadRadius = 5.0 -- distance to 'overcut' (travel past start point) when profiling lead\_in\_out\_data.OvercutDistance = 0.0

## PocketParameterData

This object is used to hold the se ngs for a pocke ng toolpath and also the clearance tool sec on of a flat bo omed v carving toolpath.

Constructor

Pro ileParameterData() - Constructor

Create a new object ready to have its parameters set

e.g local pocket\_data = PocketParameterData()

Proper es

.Allowance

R/W - double - allowance to leave on pocket sides when calcula ng toolpath

.AllowanceFormula

R/W - string -Formula for allowance to leave on pocket sides when calcula ng toolpath

.CutDepth

R/W - double -Final cu ng depth below start depth

.CutDepthFormula

R/W - string -Formula for cu ng depth below start depth

.CutDirection

R/W - integer - Cu ng direc on for toolpath. Valid values are ... ProfileParameterData.CLIMB\_DIRECTION

ProfileParameterData.CONVENTIONAL\_DIRECTION

Note: Direc on constants use ProfileParameterData. NOT PocketParameterData.

.DoRamping

R/W - bool - If true ramp entry to pockets (always zig-zag)

.DoRasterClearance

R/W - bool - if true doing raster area clearance, else offset

.Name

R/W - string - The name for the toolpath

.Pro ilePassType

R/W - integer - type of profile pass to perform . Valid values are ...

PocketParameterData.PROFILE\_NONE PocketParameterData.PROFILE\_FIRST

PocketParameterData.PROFILE\_LAST

.ProjectToolpath

R/W - bool - If true and used with Aspire, toolpath is projected onto model surface a er calcula on .RampDistance

R/W - double - distance to ramp over if doing ramping

.RasterAllowance

R/W - double - allowance to leave on pocket edge between rasters

.RasterAngle

R/W - double - Angle in degrees to create raster toolpaths at

.StartDepth

R/W - double - Start depth for toolpath below material surface

.StartDepthFormula

R/W - string -Formula for start depth for toolpath below material surface

.UseAreaClearTool

R/W - bool - If true use a larger tool for area clearance

Example Code

-- Create object used to pass pocketing options - used for area clearance only local pocket\_data = PocketParameterData()

-- start depth for toolpath pocket\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth pocket\_data.CutDepth = flat\_depth

-- direction of cut for offet clearance - ProfileParameterData.CLIMB\_DIRECTION

-- or ProfileParameterData.CONVENTIONAL\_DIRECTION - NOTE: enum from ProfileParameterData pocket\_data.CutDirection = ProfileParameterData.CLIMB\_DIRECTION

-- if true use raster clearance strategy , else use offset area clearance pocket\_data.DoRasterClearance = true

-- angle for raster if using raster clearance pocket\_data.RasterAngle = 0

-- type of profile pass to perform PocketParameterData.PROFILE\_NONE , -- PocketParameterData.PROFILE\_FIRST orPocketParameterData.PROFILE\_LAST pocket\_data.ProfilePassType = PocketParameterData.PROFILE\_LAST

## DrillParameterData

This object is used to hold the se ngs for a drilling toolpath .

Constructor

DrillParameterData() - Constructor

Create a new object ready to have its parameters set

e.g local drill\_data = DrillParameterData()

Proper es

.CutDepth

R/W - double -Final cu ng depth below start depth

.CutDepthFormula

R/W - string -Formula for cut depth below start depth

.DoPeckDrill

R/W - bool - if true we will do peck drilling

.Name

R/W - string - The name for the toolpath

.PeckRetractGap

R/W - double - distance above surface to retract to when peck drilling

.ProjectToolpath

R/W - bool - If true and used with Aspire, toolpath is projected onto model surface a er calcula on

.StartDepth

R/W - double - Start depth for toolpath below material surface

.StartDepthFormula

R/W - string -Formula for start depth for toolpath below material surface

Example Code

-- Create object used to pass drilling options local drill\_data = DrillParameterData() -- start depth for toolpath drill\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth drill\_data.CutDepth = cut\_depth

-- if true perform peck drilling drill\_data.DoPeckDrill = retract\_gap > 0.0

-- distance to retract above surface when peck drilling drill\_data.PeckRetractGap = retract\_gap

-- if true in Aspire, project toolpath onto composite model drill\_data.ProjectToolpath = false

171

## VCarveParameterData

This object is used to hold the se ngs for a vcarving toolpath .

Constructor

VCarveParameterData() - Constructor

Create a new object ready to have its parameters set

e.g local vcarve\_data = VCarveParameterData()

Proper es

.FlatDepth

R/W - double -If DoFlatBo om is true the toolpath will not cut any deeper than this depth below StartDepth.

.FlatDepthFormula

R/W - string -Formula for flat depth - see .FlatDepth for when used

.Name

R/W - string - The name for the toolpath

.ProjectToolpath

R/W - bool - If true and used with Aspire, toolpath is projected onto model surface a er calcula on

.StartDepth

R/W - double - Start depth for toolpath below material surface

.StartDepthFormula

R/W - string -Formula for start depth for toolpath below material surface

.DoFlatBottom

R/W - bool - If true the toolpath depth will be limited to the value specified in .FlatDepth and the bo om of the carving will be flat.

.UseAreaClearTool

R/W - bool - If true, the area clearance tool will be used to clear the flat areas of the toolpath

Example Code

-- Create object used to pass vcarving options local vcarve\_data = VCarveParameterData()

-- start depth for toolpath vcarve\_data.StartDepth = start\_depth

-- flag indicating if we are creating a flat bottomed toolpath vcarve\_data.DoFlatBottom = flat\_depth > 0.0

-- cut depth for toolpath this is depth below start depth vcarve\_data.FlatDepth = flat\_depth

-- if true in Aspire, project toolpath onto composite model vcarve\_data.ProjectToolpath = false

## FlutingParameterData

This object is used to hold the se ngs for a flu ng toolpath.

Constructor

FlutingParameterData() - Constructor

Create a new object ready to have its parameters set

e.g local fluting\_data = FlutingParameterData()

Proper es

.CutDepth

R/W - double -Final cu ng depth below start depth

.CutDepthFormula

R/W - string -Formula for final cu ng depth below start depth

.FluteType

R/W - integer - Type of flute to create. Valid values are ... Flu ngParameterData.FULL\_LENGTH

Flu ngParameterData.RAMP\_START

Flu ngParameterData.RAMP\_START\_END

.Name

R/W - string - The name for the toolpath

.ProjectToolpath

R/W - bool - If true and used with Aspire, toolpath is projected onto model surface a er calcula on

.RampLength

R/W - double -Length of ramp. This is only used if .UseRampRa o is false.

.RampRatio

R/W - double -Length of ramp as ra o of flute length (range 0-1.0). For ramp at start end ra o is of half length. This is only used if .UseRampRa o is true.

.RampType

R/W - integer - Type of ramping to perform. Valid values are ...

Flu ngParameterData.RAMP\_LINEAR

Flu ngParameterData.RAMP\_SMOOTH

.StartDepth

R/W - double - Start depth for toolpath below material surface

.StartDepthFormula

R/W - string -Formula for start depth for toolpath below material surface

.UseRampRatio

R/W - bool -If true, use .RampRa o field for controling ramp length, else absolute value from .RampLength

.UseSelectionOrder

R/W - bool - If true use selec on order of vectors as machining order

## PrismCarveParameterData

This object is used to hold the se ngs for a prism carving toolpath.

Constructor

PrismCarveParameterData() - Constructor

Create a new object ready to have its parameters set

e.g local prism\_data = PrismCarveParameterData()

Proper es

.CutDepth

R/W - double -Final cu ng depth below start depth

.CutDepthFormula

R/W - string -Formula for cut depth below start depth

.CutDirection

R/W - integer - Cu ng direc on for toolpath. Valid values are ... ProfileParameterData.CLIMB\_DIRECTION

ProfileParameterData.CONVENTIONAL\_DIRECTION

.Name

R/W - string - The name for the toolpath

.StartDepth

R/W - double - Start depth for toolpath below material surface

.StartDepthFormula

R/W - string -Formula for start depth for toolpath below material surface

Note: Direc on constants use ProfileParameterData. NOT PrismCarveParameterData.

Methods

:CalculateMinimumBevelDepth(Tool tool, bool show\_warnings)

Return the minimum cut depth required with the passed tool for the currently selected vectors. Tool -Tool -Tool to use for the toolpath show\_warnings -bool -if true display warning to user if tool not suitable of vectors not selected

Example Code

-- Create object used to pass toolpath options local prism\_data = PrismCarveParameterData()

-- start depth for toolpath prism\_data.StartDepth = start\_depth

-- cut depth for toolpath this is depth below start depth prism\_data.CutDepth = cut\_depth

-- direction of cut for offet clearance - ProfileParameterData.CLIMB\_DIRECTION or -- ProfileParameterData.CONVENTIONAL\_DIRECTION - NOTE: enum from ProfileParameterData prism\_data.CutDirection = ProfileParameterData.CLIMB\_DIRECTION

-- calculate the minimum cut depth to fully form the bevel on the current selection

-- with the current tool

local min\_bevel\_depth = prism\_data:CalculateMinimumBevelDepth(tool, true) if min\_bevel\_depth > cut\_depth then

DisplayMessageBox("A prism will not be fully formed with a depth of " .. cut\_depth ..

"\r\n" ..

"A depth of " .. min\_bevel\_depth ..

" is required to fully form the prism"

)

## RoughingParameterData – Aspire Only

This object is used to hold se ngs for a 3D roughing toolpath.

Constructors

RoughingParameterData() – Constructor

Create a new object ready to have its parameters set

Proper es

.Name

R/W – string – The name for the toolpath

.StartDepth

R/W – double – Start depth for the toolpath below the material surface

.StartDepthFormula

R/W – string – Formula for start depth for toolpath below material surface

.CutDepth

R/W – double – Final cu ng depth below the start depth

.CutDepthFormula

R/W – string – Formula for final cu ng depth below start depth

.Allowance

R/W – double – allowance to leave for a finishing toolpath to clear

.DoZLevelRoughing

R/W – bool – If true do Z level roughing

.ZLevelStrategy

R/W – integer – type of Z level roughing to perform. Valid values are:

RoughingParameterData.RASTER\_X RoughingParameterData.RASTER\_Y

RoughingParameterData.OFFSET

.ZLevelPro ile

R/W – integer – type of roughing pass to perform. Valid values are:

RoughingParameterData.LAST

RoughingParameterData.FIRST

RoughingParameterData.NONE

.RasterAngle

R/W – double – The angle to raster at

.MachiningAllowance

R/W – double – Allowance to leave on when roughing

## ExternalToolpath

This object is used to create a toolpath within the program with the user supplying ALL data for the toolpath including the 3D moves.

WARNING

This is an extremely powerful capability as you can control the complete movement of the tool. However it is extremely important to realize how dangerous this can be, the program makes no checks on the data supplied via this method, so the user must make sure that the toolpath is doing what they expect. Scripts that use this capability should be rigorously tested with different jobs with both the Z origin on the material top and bo om, as well as all varia ons of origin posi on and in both mm and inches.

Note: Once the toolpath is created, use the AddExternalToolpath method of ToolpathManager to add it to the program.

Constructor

ExternalToolpath(

string name,

Tool tool,

ToolpathPosData pos\_data,

ExternalToolpathOp ons op ons,

ContourGroup contours

)

Create a new toolpath object

Name - string - Name for the toolpath

Tool -Tool -Tool to use for the toolpath

pos\_data-ToolpathPosData -Se ngs for home posi on, safe z etc op ons -ExternalToolpathOp ons -Se ngs for external toolpath contours - ContourGroup - the contours describing the actual toolpath

Proper es

.Error

R/O- bool - true if an error occurred while crea ng the toolpath

## ExternalToolpathOptions

This object is used to hold the se ngs for an external toolpath.

Constructor

ExternalToolpathOptions() - Constructor

Create a new object ready to have its parameters set

Proper es

.CreatePreview

R/W - bool - if true, create preview in 2d view for this toolpath

.StartDepth

R/W - double - Start depth for toolpath below material surface Example Code

function CreateToolpath(name , vectors, start\_depth)

-- Create tool we will use to machine vectors local tool = Tool(

"Lua End Mill",

Tool.END\_MILL -- BALL\_NOSE, END\_MILL, VBIT

) tool.InMM = true tool.ToolDia = 3.0 tool.Stepdown = 2.0 tool.Stepover = 1.0

tool.RateUnits = Tool.MM\_SEC -- MM\_SEC, MM\_MIN, METRES\_MIN, -- INCHES\_SEC, INCHES\_MIN, FEET\_MIN tool.FeedRate = 30 tool.PlungeRate = 10 tool.SpindleSpeed = 20000 tool.ToolNumber = 1

tool.VBitAngle = 90.0 -- used for vbit only tool.ClearStepover = 1.0 -- used for vbit only

-- Create object used to set home position and safez gap above material surface local pos\_data = ToolpathPosData() pos\_data:SetHomePosition(0, 0, 5.0) pos\_data.SafeZGap = 5.0

-- object used to pass data on toolpath settings toolpath\_options = ExternalToolpathOptions() toolpath\_options.StartDepth = start\_depth toolpath\_options.CreatePreview = true

-- Create our toolpath local toolpath = ExternalToolpath( name, tool, pos\_data, toolpath\_options, vectors )

if toolpath:Error() then

DisplayMessageBox("Error creating toolpath") return end

local toolpath\_manager = ToolpathManager() success = toolpath\_manager:AddExternalToolpath(toolpath) return success end

## PostPInfo

This object is used access a post processor within the program. A post processor is acquired from the ToolpathSaver objects :GetPostWithName(), :GetPostWithFilename() or :GetPostAtIndex() methods.

Proper es

.Name

R/O -string – the name of the post processor. This is the name displayed in the dropdown list within the program

.FileName

R/O -string – the filen name of the post processor. This does not include the path to the PostP directory.

.Extension

R/O -string – the file extension to use for toolpaths saved using this post processor. Does not include the ‘.’ E.g “txt” NOT “.txt”

.SupportsArcs

R/O -bool – true if the post processor supports arcs. If arcs aren’t supported, any arc moves in the toolpath are output as a series of straight lines approxima ng the arc within tolerance.

.SupportsToolchange

R/O -bool – true if the post processor supports tool changing.

.Wrap\_X\_Axis

R/O -bool – true if the post processor wraps X moves onto a rotary axis

.Wrap\_Y\_Axis

R/O -bool – true if the post processor wraps Y moves onto a rotary axis

## ToolpathSaver

This object is used to save toolpaths en rely from script. It gives access to the set of PostProcessors available to the program and holds a list of toolpaths to be saved.

IMPORTANT: It is the script writer’s responsibility to ensure that all the toolpaths added to the

ToolpathSaver object are suitable for saving to one file using the selected post-processor. The ToolpathSaver gives low-level access to the toolpath output from the program and will write what ever toolpaths you specify to a single file irrespec ve of the tool geometry or the post processors support for tool changing. It is the responsibility of the script writer to ensure that a tool changing post is selected if mul ple toolpaths using different tools are output. The :IsCompa ble method of the Tool object can be used to check if tools used for different toolpaths are suitable for outpu ng to a single file if a tool changer is not available.

Example Code

--[[ ------------------------ DoAllToolpathsUseSameTool ----------------------------------|

| Check if all the toolpaths in the passed list use the same tool. If they do, we can

| save all the toolpaths to the same file, regardles of whether the post supports

| toolchanging. If the toolpaths use different tools and the post does not support tool | changing, the toolpaths will need to be saved individually

| ]]

function DoAllToolpathsUseSameTool(toolpath\_ids, toolpath\_manager) local tool = nil

for i,toolpath\_id in ipairs(toolpath\_ids) do local pos = toolpath\_manager:Find(toolpath\_id) if pos == nil then

MessageBox("Failed to find toolpath " .. i) return false end

local toolpath = toolpath\_manager:GetAt(pos) -- do we have an existing tool to check against?

if tool == nil then

-- no this is first toolpath - save a copy of tool to check against tool = toolpath.Tool else

-- check if this tool is compatible with previous tools if not tool:IsCompatibleTool(toolpath.Tool) then return false -- we have a mixture of tools end end

end

return true end

Constructor

ToolpathSaver() - Constructor

Create a new object which can be used to save toolpaths

Proper es

.DefaultPost

R/O -PostPInfo – the currently selected post processor within the program

.NumberOfToolpaths

R/O -integer – the number of toolpaths currently held by this object ready for saving. Toolpaths are added to the object using the :AddToolpaths() method.

Methods

:GetNumPosts()

Return number of post processers the program knows about. This is usually used to iterate through all the posts using :GetPostAtIndex

:AddToolpath(Toolpath toolpath)

Add the passed toolpath to the list of toolpaths to save managed by this object. Retunrs true if toolpath added to list ok.

toolpath -Toolpath –toolpath to be saved

:ClearToolpathList()

Clear the list of toolpaths to save managed by this object.

:GetNumPosts()

Return number of post processers the program knows about. This is usually used to iterate through all the posts using :GetPostAtIndex

:GetPostAtIndex(integer post\_index)

Return the post processor (PostPInfo) for the specified index. post\_index - integer –index of post to return. The index go from 0 to num\_posts - 1

:GetPostWithName(string post\_name)

Return the post processor (PostPInfo) with the specified name. This is the name displayed in the drop down list in the program. post\_name - string –name of post to return.

:GetPostWithFilename(string post\_ ile\_name)

Return the post processor (PostPInfo) with the specified file name. This is the name of the .pp file on disk without the path.

post\_file\_name - string – filename of post to return.

.CreatePost(string post\_ ile\_path)

Create a new post processor from the given file. This is the full path to the .pp file. Returns a PostPInfo (or nil if something went wrong).

post\_file\_path – string – path to the .pp file

:SaveToolpaths(PostPInfo postp, string ilename, bool output\_to\_mc)

Save all the toolpaths which have been added with :AddToolpath using passed post processor to passed filename, Returns true if file saved OK, false if an error occurs.

postp -PostPInfo – Post processor to use to save file filename -string – Full pathname to file to be saved. This MUST include the extension you want for the file as well.

output\_to\_mc -bool –if true AND the post processor supports direct output to the machine the file will be sent to the machine a er saving. Example Code

local toolpath\_manager = ToolpathManager() if toolpath\_manager.IsEmpty then

MessageBox("There are no toolpaths to save") return false end

local output\_path = “c:\\temp\\test\_toolpath\_output.txt”

-- get post processor to use for saving ... local toolpath\_saver = ToolpathSaver();

local post = toolpath\_saver:GetPostWithName(“Text Output Arcs (mm) (\*.txt)”) if post == nil then

MessageBox("Failed to load Post Processor with name " .. post\_name) return false end

-- save toolpath to file ... local toolpath\_names = "" -- we keep a list of toolpaths we save if selected\_only then

local toolpath = toolpath\_manager:GetSelectedToolpath() if toolpath == nil then

MessageBox("There is no currently selected toolpath") return false end

toolpath\_saver:AddToolpath(toolpath) toolpath\_names = toolpath.Name .."\n" else

-- we are saving all toolpaths ...

local pos = toolpath\_manager:GetHeadPosition() while pos ~= nil do local toolpath

toolpath, pos = toolpath\_manager:GetNext(pos) toolpath\_saver:AddToolpath(toolpath)

toolpath\_names = toolpath\_names .. toolpath.Name .."\n" end end

local success = toolpath\_saver:SaveToolpaths(post, output\_path, false) if not success then

MessageBox("Failed to save toolpath(s)\n\n" .. toolpath\_names ..

"\nto file\n" .. output\_path ) return false end

MessageBox("Saved toolpath(s)\n\n" .. toolpath\_names .. "\nto file\n" .. output\_path .. "\nUsing post\n" .. post.Name)

## ToolpathStats

This object is used to hold informa on about a calculated toolpath. This object is returned by the Toolpath:Sta s cs() method

Proper es

.IsValid

R/O -bool – true if the sta s cs in this object are valid. If toolpath has not yeet been calculated this would return fals

.FeedLength

R/O -double –total length of feed rate moves in the toolpath in either mm or inches depending on units of job.

.PlungeLength

R/O -double –total length of plunge moves in the toolpath in either mm or inches depending on units of job.

.RetractLength

R/O -double –total length of retract moves in the toolpath in either mm or inches depending on units of job.

.RapidLength

R/O -double –total length of rapid moves in the toolpath in either mm or inches depending on units of job.

.MinimumZ

R/O -double – the z value of the deepest move in the toolpath in either mm or inches depending on units of job.

## ToolpathSaveInfo

This object is used only within the GetNameForToolpathFile.lua configura on handler. This file can be used to customise the behavior of the file save mechanism for toolpath. Just before the program displays the File Save dialog when saving a toolpath, it will check if a file called

GetNameForToolpathFile.lua was present when the program started. The path to the file varies depending on the opera ng system (the equivalent loca on of public “My Documents”) and the program name and version. For Aspire V8.0 on Windows 7, this path is …

C:\Users\Public\Documents\Vectric Files\Config\Aspire 8.0\ GetNameForToolpathFile.lua

For VCarve Pro V8.0 it would be ..

C:\Users\Public \Documents\Vectric Files\Config\VCarve Pro 8.0\ GetNameForToolpathFile.lua

An example of a Lua file which just displays some informa on about the file being saved is shown later.

Proper es

.BaseName

R/W -string – base name for toolpath to save without extension. Before saving the default extension from .PostP will be added

.PathName

R/W -string – The full pathname to the file to be saved. On calling the lua script this field will be empty. If the Lua script sets this value it will be used as the default path for the file open dialog.

.PostP

R/O -PostPInfo – The post processor the user has chosen to save the toolpath(s) with.

.AddSheetNameToEnd

R/W bool – If this is true the sheet name will be appended to the toolpath filename rather than prepended when saving toolpaths in a mul -sheet project.

.ShowFileDialog

R/W -bool – If the Lua script sets this to true AND .PathName is not empty, The File Save As dialog will not be displayed by the program and the toolpath will be wri en to the file specified in .PathName . If the file in .PathName does NOT have an extension, the default extension from the PostP will be added automa cally.

.ToolpathList

R/O -ToolpathList – The list of toolpaths being saved.

Example Code – GetNameForToolpathFile.lua

-- VECTRIC LUA SCRIPT require "strict"

--[[ --- GetNameForToolpathFile -------------------------------------------------|

| Entry point for script

| ]]

function GetNameForToolpathFile(save\_info)

local tc\_support;

if save\_info.PostP.SupportsToolchange then tc\_support = "Toolchanges Are Supported" else

tc\_support = "Toolchanges Not Supported" end

MessageBox(

"GetNameForToolpathFile:" ..

"\nNumber of Toolpaths = " .. save\_info.ToolpathList.Count ..

"\nDefault Name = " .. save\_info.BaseName ..

"\nPost Processor Name = " .. save\_info.PostP.Name ..

"\nPost Processor Extension = " .. save\_info.PostP.Extension ..

"\nPost Processor " .. tc\_support )

-- save\_info.BaseName = "Lua Toolpath Name"

-- save\_info.PathName = "d:\\temp\\Lua Toolpath Name"

-- save\_info.ShowFileDialog = false return true end

--[[ -------------- main -------------------------------------------------- |

| Entry point for script

| ]] function main(script\_path)

return true end

## ToolpathList

This object holds a list of toolpaths for another object.

Proper es

.Count

R/O -integer – return the number of toolpaths in the list

.IsEmpty

R/O -bool – return true if the toolpath list is empty

Methods

:GetAt(POSITION pos)

Returns the toolpath at the current posi on pos – POSITION – current posi on in list

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list of toolpaths

:GetNext(POSITION pos)

Returns the toolpath at the current posi on AND a new value for posi on poin ng to the next item

in the list (or nil if at end of list)

pos – POSITION – current posi on in list

Example - note that GetNext(pos) is returning two values ...

local pos = save\_info.ToolpathList:GetHeadPosition() local toolpath while pos ~= nil do

toolpath, pos = save\_info.ToolpathList:GetNext(pos) DO SOMETHING WITH TOOLPATH .... end

:GetPrev(POSITION pos)

Returns the toolpath at the current posi on AND a new value for posi on, poin ng to the previous

item in the list (or nil if at start of list)

pos – POSITION – current posi on in list

:GetTailPosition()

Returns a POSITION variable to allow access to the tail (end) of the list toolpaths

User Interface

## HTML\_Dialog

This object is used display a dialog to the user. The dialog is defined using a HTML web page and can collect informa on from the user as well as display it. The system works by associa ng variables within the script with HTML elements within the dialog. When the dialog is closed, the current values of edit boxes, check boxes etc can be examined. As well as passively collec ng all the data once the dialog is closed, ‘callbacks’ into the lua code can be made when the user clicks on a bu on, selects a file or tool or chooses from a list.

Bu on Callbacks

If a bu on on the dialog has class=”LuaBu on”, when the user clicks on the bu on, HTML\_Dialog will search the calling script for a method with the signature

func on OnLuaBu on\_BUTTON\_ID(dialog)

where BUTTON\_ID is the id of the bu on pressed and ‘dialog’ is the HTML\_Dialog object the bu on was pressed on.

e.g HTML

<BUTTON CLASS="LuaBu on" ID="TestBu on1" type=bu on>Press Me!</BUTTON></td> e.g Lua

function OnLuaButton\_TestButton1(dialog)

MessageBox("User pressed TestButton1")

local cur\_num\_machines = dialog:GetIntegerField("MachineCount")

return true end

NOTE: The callback func ons MUST have “return true” at the end or you will get script errors!

If a specific handler is not found for the bu on an a empt will be made to call a ‘generic’ handler with the signature

function OnLuaButton\_XXXX(element\_id, dialog)

MessageBox(Button with id = “ .. element .. “ was pressed!”) local cur\_num\_machines = dialog:GetIntegerField("MachineCount") return true end

e.g HTML

<BUTTON CLASS="LuaBu on" ID="TestBu on1" type=bu on>Press Me!</BUTTON></td> e.g Lua

function OnLuaButton\_TestButton1(dialog) MessageBox("User pressed TestButton1")

local cur\_num\_machines = dialog:GetIntegerField("MachineCount")

return true end

DirectoryPicker Callbacks

When a DirectoryPicker is used to pick a directory, HTML\_Dialog will will search the calling script for a method with the signature func on OnDirectoryPicker\_PICKER\_ID(dialog)

where PICKER\_ID is the id of the directory picker bu on pressed and ‘dialog’ is the HTML\_Dialog object the bu on was pressed on.

FilePicker Callbacks

When a FilePicker is used to pick a file, HTML\_Dialog will will search the calling script for a method with the signature func on OnFilePicker\_PICKER\_ID(dialog)

where PICKER\_ID is the id of the file picker bu on pressed and ‘dialog’ is the HTML\_Dialog object the bu on was pressed on.

ToolPicker Callbacks

When a ToolPicker is used to pick a tool, HTML\_Dialog will will search the calling script for a method with the signature func on OnToolPicker\_PICKER\_ID(dialog)

where PICKER\_ID is the id of the tool picker bu on pressed and ‘dialog’ is the HTML\_Dialog object the bu on was pressed on.

Selector / DropDownList Callbacks

When a Selector / DropDownList is used to pick a value, HTML\_Dialog will will search the calling script for a method with the signature func on OnLuaSelector\_ELEMENT\_ID(dialog)

where ELEMENT\_ID is the id of the selector / dropdown list bu on pressed and ‘dialog’ is the HTML\_Dialog object the list was seelcted on.

If a specific handler is not found for the list an a empt will be made to call a ‘generic’ handler with the signature

function OnLuaSelector\_XXXX(element\_id, dialog) local selected\_value = dialog:GetDropDownListValue(element\_id)

MessageBox(“Value “ .. selected\_value ..

“ was selected from List with id = “ .. element\_id) return true end

Constructor

HTML\_Dialog(bool local\_html, string html, integer width, integer height, string dialog\_name) - Constructor

Create a new dialog object , specifying the source for the HTML and the dialog size and tle.

local\_html -bool –true if the next argument (html) contains all the html code for the dialog in the string. If false the html string is the path to the html file.

html -string – local\_html was true, contains all the html code for the dialog in the string. If local\_html was false the html string is the path to the html file. The path must be fully qualified and is usually built up from the scripts loca on as shown in the example below.

width -integer – width of the dialog in pixels height -integer – height of the dialog in pixels dialog\_name -string – name for dialog displayed in tle bar

e.g

function main(script\_path)

local html\_path = "file:" .. script\_path .. about.htm"

local dialog = HTML\_Dialog(false, html\_path, 800, 600, “About Gadgets”) dialog:ShowDialog() end

Proper es

.WindowWidth

R/O-integer –Width of dialog window in pixels when user closed the dialog.

.WindowHeight

R/O-integer –Height of dialog window in pixels when user closed the dialog.

Methods

:AddCheckBox(string element\_ id, bool value)

Sets a check box element (input box displaying a check box) in the HTML with id=”element\_id” to passed value.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – bool – true to ‘check box, false to leave unchecked e.g – HTML

<INPUT class="LuaBu on" type="checkbox" checked ID="CreatePreviewCheck" value='ac ve'>Create a preview e.g – Lua

dialog:AddCheckBox("CreatePreviewCheck”, true)

:AddDirectoryPicker(string element\_ id, string buddy\_name, bool buddy\_is\_edit)

Sets a bu on on the dialog to act as a ‘directory’ chooser. When the user presses the bu on a dialog is displayed to allow them to select a directory / folder . When the selec on dialog is closed the ‘buddy’ field (which can either be an edit box or a label) is updated with the path to the directory/folder chosen.

NOTE: The <Bu on > element MUST have a class = “DirectoryPicker” set

element\_id -string –Id of the bu on element in HTML to use (id=”element\_id” in HTML). The bu on MUST have class = “DirectoryPicker” buddy\_name – string – id of associated label or edit field which will hold the value the user picks

buddy\_is\_edit – bool – true if the ‘buddy’ field is an edit field (added with AddTextField), false if ‘buddy’ is a label field (added with AddLabelField).

e.g HTML for picker associated with an edit field …

<input name="tex ield" type="text" size="40" maxlength="128" ID="DirNameEdit"> <BUTTON CLASS="DirectoryPicker" ID="ChooseDirBu on1" type=bu on>Choose...</BUTTON>

e.g Lua for picker associated with an editable text field …

dialog:AddTextField("DirNameEdit", "c:\\temp"); dialog:AddDirectoryPicker("ChooseDirBu on1", "DirNameEdit", true);

e.g HTML for picker associated with a label field …

<span id="DirNameLabel"></span>

<BUTTON CLASS="DirectoryPicker" ID="ChooseDirBu on2" type=bu on>Choose...</BUTTON>

e.g Lua for picker associated with a label field …

dialog:AddLabelField("DirNameLabel", "c:\\");

dialog:AddDirectoryPicker("ChooseDirBu on2", "DirNameLabel", false);

:AddDoubleField(string element\_ id, double value)

Sets a text box element (text input box displaying a double value) in the HTML with id=”element\_id” to passed value.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – double – value to set for element in HTML

e.g HTML

<input name="tex ield" type="text" size="8" maxlength="8" ID="MachineCost"> e.g Lua dialog:AddDoubleField("MachineCost", 6495.12)

:AddDropDownList(string element\_ id, string default\_value)

Sets a drop downlis n the HTML (type=”radio”) with id=”element\_id” to passed value.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – string – ini al value to set for element in HTML e.g HTML

<select id="ProductList" >

<op on value="1">first</op on>

<op on value="2" selected>second</op on>

<op on value="3" >third</op on> </select>

e.g Lua

dialog:AddDropDownList("ProductList ","Aspire") dialog:AddDropDownListValue("ProductList ", "Cut2D") dialog:AddDropDownListValue("ProductList ", "VCarve Pro") dialog:AddDropDownListValue("ProductList ", "Aspire") dialog:AddDropDownListValue("ProductList ", "Cut3D") dialog:AddDropDownListValue("ProductList ", "PhotoVCarve")

:AddDropDownListValue(string element\_ id, string value)

Add a value to a dropdown list n the HTML (type=”radio”) with id=”element\_id” to passed value.

element\_id -string –Id of the select list in HTML to add value to (id=”element\_id” in HTML) value – string –value to add to list of values

:AddFilePicker(string element\_ id, string buddy\_name, bool buddy\_is\_edit)

Sets a bu on on the dialog to act as a ‘file’ chooser. When the user presses the bu on a dialog is displayed to allow them to select a file. When the selec on dialog is closed the ‘buddy’ field (which can either be an edit box or a label) is updated with the path to file chosen.

NOTE: The <Bu on > element MUST have a class = “FilePicker” set

element\_id -string –Id of the bu on element in HTML to use (id=”element\_id” in HTML). The bu on MUST have class = “FilePicker” buddy\_name – string – id of associated label or edit field which will hold the value the user picks

buddy\_is\_edit – bool – true if the ‘buddy’ field is an edit field (added with AddTextField), false if ‘buddy’ is a label field (added with AddLabelField).

e.g HTML for picker associated with an edit field …

<input name="tex ield" type="text" size="40" maxlength="128" ID="FileNameEdit"> <BUTTON CLASS="FilePicker" ID="ChooseFileBu on1" type=bu on>Choose...</BUTTON>

e.g Lua for picker associated with an editable text field …

dialog:AddTextField("FileNameEdit", "c:\\temp");

dialog:AddDirectoryPicker("ChooseFileBu on1", "FileNameEdit", true);

e.g HTML for picker associated with a label field …

<span id="FileNameLabel"></span>

<BUTTON CLASS="FilePicker" ID="ChooseFileBu on2" type=bu on>Choose...</BUTTON>

e.g Lua for picker associated with a label field …

dialog:AddLabelField("FileNameLabel", "c:\\");

dialog:AddDirectoryPicker("ChooseFileBu on2", "FileNameLabel", false);

:AddIntegerField(string element\_ id, integer value)

Sets a text box element (text input box displaying an integer value) in the HTML with id=”element\_id” to passed value.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – integer – value to set for element in HTML

e.g HTML

<input name="tex ield" type="text" size="4" maxlength="8" ID="MachineCount"> e.g Lua dialog:AddIntegerField("MachineCount", 1)

:AddLabelField(string element\_ id,string value)

Sets a text element in the HTML with id=”element\_id” to passed value

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – string – Value to set for element in HTML

e.g – HTML

<span id="WelcomeText">Welcome to our first test HTML dialog <br>created via Lua </spe.g – Lua e.g Lua dialog:AddLabelField("WelcomeText", "Welcome to my <b>Lua</b> HTML dialog")

:AddRadioGroup(string element\_ id, int default\_index)

Sets a radio group in the HTML (type=”radio”) with id=”element\_id” to passed value. Note that the radio index is one based, so the first item in the group is 1.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) default\_index – integer – index of radio bu on to check in HTML – 1 based e.g – HTML

<br> Type of Machine<br><br>

<input class="LuaBu on" type="radio" name="MachineType" ID="MachineRadio1"> Engraver

<input class="LuaBu on" type="radio" name="MachineType" ID="MachineRadio2"> Router

<input class="LuaBu on" type="radio" name="MachineType" ID="MachineRadio3"> CNC Mill

<br>

e.g – Lua

dialog:AddRadioGroup("MachineType", 2)

:AddTextField(string element\_ id, string value)

Sets a text element (text input box) in the HTML with id=”element\_id” to passed value.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – string – value to set for element in HTML

e.g HTML

<input name="tex ield" type="text" size="20" maxlength="8" ID="MachineName"> e.g Lua

dialog:AddTextField("MachineName", "Rosy")

:AddToolEditor(string element\_ id, string buddy\_name)

Sets a bu on on the dialog to act as a ‘tool’ editor. When the user presses the bu on, the tool editor dialog is displayed to allow them to edit the tool chosen by the associated ‘buddy’ ToolPicker. The ‘buddy’ field which MUST be a ToolPicker added with AddToolPickers has the path to the tool chosen by the tool picker.

element\_id -string –Id of the bu on element in HTML to use (id=”element\_id” in HTML). The bu on MUST have class = “ToolPicker”

buddy\_name – string – id of associated tool picker which will hold the path to the tool the user picks.

:AddToolPicker(string element\_ id, string buddy\_name, ToolDBId tooldb\_id)

Sets a bu on on the dialog to act as a ‘tool’ chooser. When the user presses the bu on the tool database dialog is displayed to allow them to select a tool. When the selec on dialog is closed the ‘buddy’ field which MUST be a label added with AddLabelField is updated with the path to the tool chosen.

element\_id -string –Id of the bu on element in HTML to use (id=”element\_id” in HTML). The bu on MUST have class = “ToolPicker”

buddy\_name – string – id of associated label field which will hold the path to the tool the user picks.

Tooldb\_id – ToolDBId – id object describing the loca on in the tool database – usually saved from a previous selec on by the tool picker, can be retrieved from Tool.ToolDBId property.

e.g HTML for picker associated with a label field …

<span id="ToolNameLabel>Tool Name</span>

<input id="ToolChooseBu on" class="ToolPicker" name="ToolChooseBu on" type="bu on" value="Tool...">

e.g Lua for picker associated with a label field …

dialog:AddLabelField("ToolNameLabel ", ""); dialog:AddToolPicker("ToolChooseBu on", "ToolNameLabel", ToolDBId()) dialog:AddToolPickerValidToolType("ToolChooseBu on", Tool.END\_MILL) dialog:AddToolPickerValidToolType("ToolChooseBu on", BALL\_NOSE)

:AddToolPickerValidToolType(string element\_ id, integer tool\_type)

Used to add valid tool types for the tool databse. See below for a list of valid values

element\_id -string –Id of the toolpicker element in HTML to add valid type to (id=”element\_id” in HTML). The bu on MUST have class = “ToolPicker” tool\_type – integer – Integer indica ng a type of tool which will be valid for selec on – choose from ..

Tool.BALL\_NOSE

Tool.END\_MILL

Tool.RADIUSED\_END\_MILL

Tool.VBIT

Tool.ENGRAVING

Tool.RADIUSED\_ENGRAVING

Tool.THROUGH\_DRILL

Tool.FORM\_TOOL

Tool.DIAMOND\_DRAG

Tool.RADIUSED\_FLAT\_ENGRAVING

:ClearToolPickerValidToolType(string element\_ id)

Used to clear list of valid tool types for the tool databse.

element\_id -string –Id of the toolpicker element in HTML to clear (id=”element\_id” in HTML). The bu on MUST have class = “ToolPicker”

:GetCheckBox(string id)

Returns a bool containing current value of a check box in the HTML with id=”element\_id” . The field must have been created with a previous call to AddCheckBox before the dialog was displayed. element\_id -string –Id of the element in HTML to get to value for (id=” element\_id” in HTML)

:GetDropDownListValue(string id)

Returns a string containing current value of a dropdown list (select element) in the HTML with id=”element\_id” . The field must have been created with a previous call to AddDropDownList before the dialog was displayed.

element\_id -string –Id of the element in HTML to get value for (id=” element\_id” in HTML)

:GetDoubleField(string id)

Returns a double containing current value of a text input box displaying a double value in the HTML with id=”element\_id” . The field must have been created with a previous call to AddDoubleField before the dialog was displayed.

element\_id -string –Id of the element in HTML to get value for (id=” element\_id” in HTML)

:GetIntegerField(string id)

Returns an integer containing current value of a text input box displaying an integer value in the HTML with id=”element\_id” . The field must have been created with a previous call to AddIntegerField before the dialog was displayed.

element\_id -string –Id of the element in HTML to get value for (id=” element\_id” in HTML)

:GetLabelField(string id)

Returns a string containing current value of an element in the HTML with id=”element\_id” . The field must have been created with a previous call to AddLabelField before the dialog was displayed. element\_id -string –Id of the element in HTML to get value for (id=” element\_id” in HTML)

:GetRadioIndex(string id)

Returns an integer containing current value of a radio group in the HTML with id=”element\_id” . The field must have been created with a previous call to AddRadioGroup before the dialog was displayed.

element\_id -string –Id of the element in HTML to get value for (id=” element\_id” in HTML) :GetTextField(string id)

Returns a string containing current value of a text input box element in the HTML with id=”element\_id” . The field must have been created with a previous call to AddTextField before the dialog was displayed. element\_id -string –Id of the element in HTML to get value for (id=” element\_id” in HTML)

:GetTool (string element\_ id)

Returns the Tool selected by the ToolPicker with the passed id.

element\_id -string –Id of the bu on element in HTML to use (id=”element\_id” in HTML). The bu on MUST have class = “ToolPicker”

:SetInnerHtml(string element\_ id, string html) Sets the ‘inner html’ for an element on the dialog.

element\_id -string –Id of the element in HTML to set inner html for (id=” element\_id” in HTML) html – string – new ‘inner’ html for the element

:SetOuterHtml(string element\_ id, string html)

Sets the ‘outer html’ for an element on the dialog. This can replace the complete element defini on

element\_id -string –Id of the element in HTML to set outer html for (id=” element\_id” in HTML) html – string – new ‘outer’ html for the element

:UpdateCheckBox(string element\_ id, bool value)

Update check box in the HTML with id=”element\_id” to passed value. The field must have been created with a previous call to AddCheckBox before the dialog was displayed. This method is used within ‘callbacks’ from the dialog, while the dialog is s ll visible.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – bool – new value to set for element in HTML

:UpdateDropDownListValue(string element\_ id, string value)

Update dropdown list in the HTML with id=”element\_id” to passed value. The field must have been created with a previous call to AddDropDownList before the dialog was displayed and the value being set must be a value that was added to the list using AddDropDownListValue. This method is used within ‘callbacks’ from the dialog, while the dialog is s ll visible.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – string – new value to set for dropdown list

:UpdateDoubleField(string element\_ id, double value)

Update text input box displaying a double value in the HTML with id=”element\_id” to passed value. The field must have been created with a previous call to AddDoubleField before the dialog was displayed. This method is used within ‘callbacks’ from the dialog, while the dialog is s ll visible.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – double – new value to set for element in HTML

:UpdateIntegerField(string element\_ id, integer value)

Update text input box displaying an integer value in the HTML with id=”element\_id” to passed value. The field must have been created with a previous call to AddIntegerField before the dialog was displayed. This method is used within ‘callbacks’ from the dialog, while the dialog is s ll visible.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – double – new value to set for element in HTML

:UpdateLabelField(string element\_ id, string value)

Update a text element in the HTML with id=”element\_id” to passed value. The field must have been created with a previous call to AddLabelField before the dialog was displayed. This method is used within ‘callbacks’ from the dialog, while the dialog is s ll visible.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – string – new value to set for element in HTML

:UpdateRadioIndex(string element\_ id, integer index)

Update radio group in the HTML with id=”element\_id” to passed value. The field must have been created with a previous call to AddRadioGroup before the dialog was displayed. This method is used within ‘callbacks’ from the dialog, while the dialog is s ll visible.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – integer – new one based index to set for element in HTML

:UpdateTextField(string element\_ id, string value)

Update a text element (text input box) in the HTML with id=”element\_id” to passed value. The field must have been created with a previous call to AddTextField before the dialog was displayed. This method is used within ‘callbacks’ from the dialog, while the dialog is s ll visible.

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) value – string – new value to set for element in HTML

:UpdateToolPickerField(string element\_ id, Tool tool)

Update a tool picker element in the HTML with id=”element\_id” with values from passed tool. The field must have been created with a previous call to AddToolPickerField before the dialog was displayed. This method is used within ‘callbacks’ from the dialog, while the dialog is s ll visible if you change a tool value in the handler. (Aspire 4.015/ VCP 4.015 and later)

element\_id -string –Id of the element in HTML to set to value (id=”element\_id” in HTML) tool – Tool –tool with new values to set for element in HTML

## FileDialog

This object is used to display either a File Open or a File Save dialog box to the user.

Constructor

FileDialog() - Constructor

Create a new object used for displaying a File Open / Save dialog

local file\_dialog = FileDialog()

Methods

:FileOpen(string default\_ext,string ile\_name, string ilter) Display a File Open dialog – this is used to select an Exis ng file.

default\_ext – string – default extension of files to display e.g “txt” file\_name – string –path to default file top open. Can just be path or include filename filter – string –filter string to display for dialog file type.The string is build up of pairs of ‘file descrip on’ followed by ‘file extension’ separated by ‘|’. The filter list is terminated by ‘||’ . E.g “Toolpaths (\*.tap) | \*.tap||”.

:FileSave(string default\_ext,string ile\_name, string ilter)

Display a File Save dialog – this is usually used to select a new file for saving to but user can choose to overwite an exis ng file.

default\_ext – string – default extension of files to display e.g “txt”

file\_name – string –path to default file to save. Can just be path or include filename filter – string –filter string to display for dialog file type

Example Code

local post\_name = dialog:GetDropDownListValue("PostNameSelector"); local toolpath\_saver = ToolpathSaver();

local post = toolpath\_saver:GetPostWithName(post\_name) if post == nil then

MessageBox("Failed to load Post Processor with name " .. post\_name) return false end

-- get name for output file local file\_dialog = FileDialog()

local dir = dialog:GetTextField("PostOutputFolderEdit")

-- MessageBox("Ext = " .. post.Extension .. "\nFolder = " .. dir) local file\_filter = "Toolpaths (\*." .. post.Extension .. ")|" ..

“\*.” .. post.Extension .. "||"

if not file\_dialog:FileSave( post.Extension, dir .. "\\\*" , file\_filter ) then

MessageBox("No file to save to selected") return false end

local output\_path = file\_dialog.PathName

## DirectoryReader

This object is used to build a list of files and directories matching various criteria.

Constructor

DirectoryReader() - Constructor

Create a new object used for building lists of files

local dir\_reader = DirectoryReader()

Methods

:BuildDirectoryList(string dir\_path, bool recurse) Build a list of directories we will search for files

dir\_path – string – path to build list of files from recurse – bool – if true include sub directories as well

:ClearDirs()

Clear list of directories

:ClearFiles()

Clear list of files

:DirAtIndex(integer index)

Return DirectoryEntry for directory at passed index

index – integer – index of directory entry to get. Entries are indexed from 1 to :NumberOfDirs()

:FileAtIndex(integer index)

Return FileEntry for file at passed index index – integer – index of file entry to get. Entries are indexed from 1 to :NumberOfFiles()

:GetFiles(string ilter, bool include\_ iles\_in\_list, bool include\_dirs\_in\_list)

Add files / directories matching passed filter to list held by this object. Note this method can be called mul ple mes to add different sets of files

filter – string – Filter (eg “\*.dxf”) to use for finding matching files include\_files\_in\_list – bool - if true we collect names of matching files include\_dirs\_in\_list – bool - if true we collect names of matching directories

:SortDirs()

Sort list of directories into alphabe cal order

:SortFiles(integer sort\_order, bool reverse)

Sort list of files into order sort\_order – integer – Sor ng algorithm to use, op ons are

DirectoryReader.SORT\_NONE

DirectoryReader.SORT\_ALPHA

DirectoryReader.SORT\_WRITE\_DATE DirectoryReader.SORT\_SIZE

reverse – bool – if true sort in reverse order

:NumberOfDirs()

Return the number of directories found

:NumberOfFiles()

Return the number of files found

## DirectoryEntry

This object is returned from the :GetDirAtIndex() method of DirectoryReader and holds informa on about a sub directory found while building the list of files and directories.

Proper es

.Name

R/O – string – Name of the directory

## FileEntry

This object is returned from the :GetFileAtIndex() method of DirectoryReader and holds informa on about a file found while building the list of files and directories.

Proper es

.Name

R/O – string – Name of the file – this is the full pathname to the file

.FileName

R/O – string – Name of the file

.FileNameNoExt

R/O – string – Name of the file without the extension

.Size

R/O – integer – Size of the file in bytes

Example Code – from DXF\_Batch\_Processor.Lua

--[[ ----------- DirectoryProcessor

-------------------------------------------------

|

| Given a root directory and file filter, call the passed function with

| every file which matches the filter

|

| The do\_sub\_dirs flag indicates if sub-directories should be processed as

| well as the root directory

| ]]

function DirectoryProcessor(job, dir\_name, filter, do\_sub\_dirs, function\_ptr)

local num\_files\_processed = 0;

local directory\_reader = DirectoryReader() local cur\_dir\_reader = DirectoryReader()

directory\_reader:BuildDirectoryList(dir\_name, do\_sub\_dirs) directory\_reader:SortDirs()

local number\_of\_directories = directory\_reader:NumberOfDirs()

for i = 1, number\_of\_directories do

local cur\_directory = directory\_reader:DirAtIndex(i)

-- get contents of current directory

-- dont include sub-dirs, use passed filter

cur\_dir\_reader:BuildDirectoryList(cur\_directory.Name, false) cur\_dir\_reader:GetFiles(filter, true, false)

-- call passed method for each file ... local num\_files\_in\_dir = cur\_dir\_reader:NumberOfFiles() for j=1, num\_files\_in\_dir do

local file\_info = cur\_dir\_reader:FileAtIndex(j) if not function\_ptr(job, file\_info.Name) then return -1 end

num\_files\_processed = num\_files\_processed + 1 end

-- empty out our directory object ready for next go cur\_dir\_reader:ClearDirs() cur\_dir\_reader:ClearFiles()

end

return num\_files\_processed end

## Registry

This object is used to save and load values from the registry. This allows gadgets to persist se ngs across different runs of the gadget. All the values are stored under a single ‘sec on’ in the registry, so it is important that each gadget uses a unique name for its sec on. The sec ons are unique to each product, so the same gadget run under both Aspire and VCarve Pro will store their se ngs in different areas.

Constructor

Registry(string section\_name) - Constructor

Create a new object used for reading and wri ng values

Sec on\_name –string – Name of sec on in registry to store all values under

local dir\_reader = DirectoryReader(“MyGadget”)

Methods

:BoolExists(string parameter\_name)

Return true if there is an exis ng bool parameter with passed name

:DoubleExists(string parameter\_name)

Return true if there is an exis ng double parameter with passed name

:IntExists(string parameter\_name)

Return true if there is an exis ng int parameter with passed name

:GetDouble(string parameter\_name, double default\_value)

Retrieve a double with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter default\_value - double - the value which will be returned if there is no exis ng value stored

:GetBool(string parameter\_name, bool default\_value)

Retrieve a bool with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter value - integer - the value which will be returned if there is no exis ng value stored

:GetInt(string parameter\_name, integer default\_value)

Retrieve an integer with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter

value - integer - the value which will be returned if there is no exis ng value stored

:GetString(string parameter\_name, string value)

Retrieve a string with the passed name , if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter

value - string - the value which will be returned if there is no exis ng value stored

:SetBool(string parameter\_name, bool default\_value)

Retrieve a Boolean flag (true / false) with the passed name, if no value with passed name returns passed default value

parameter\_name - string - the name of the parameter

default\_value - bool - the value which will be returned if there is no exis ng value stored

:SetBool(string parameter\_name, bool value)

Store a Boolean flag (true / false) with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - bool - the value which will be stored in the parameter list

:SetDouble(string parameter\_name, double value)

Store a double with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - double - the value which will be stored in the parameter list

:SetInt(string parameter\_name, integer value)

Store an integer with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - integer - the value which will be stored in the parameter list

:SetString(string parameter\_name, string value)

Store a string with the passed name and value

parameter\_name - string - the name which will be used to store and retrieve the value value - string - the value which will be stored in the parameter list

:StringExists(string parameter\_name)

Return true if there is an exis ng string parameter with passed name

### FileDialog

This object is used to display either a File Open or a File Save dialog box to the user.

Constructor

FileDialog() - Constructor

Create a new object used for displaying a File Open / Save dialog

local file\_dialog = FileDialog()

Methods

:FileOpen(string default\_ext,string ile\_name, string ilter)

Display a File Open dialog – this is used to select an Exis ng file. Returns true if user selects a file else false if they cancel the dialog.

default\_ext – string – The default filename extension. If the user does not include an extension in the Filename edit box, the extension specified here is automa cally appended to the filename. If this parameter is empty (“”), no file extension is appended.

file\_name – string –path to default file to open. Can just be path or include filename or just be le empty (“”), usualy use “\*.ext” at end of string to preselect all files of a par cular extension.

filter – string –filter string to display for dialog file type.The string is build up of pairs of ‘file descrip on’ followed by ‘file extension’ separated by ‘|’. The filter list is terminated by ‘||’ . E.g “Toolpaths (\*.tap) | \*.tap||”.

:FileSave(string default\_ext,string ile\_name, string ilter)

Display a File Save dialog – this is usually used to select a new file for saving to but user can choose to overwite an exis ng file. Returns true if user selects a file else false if they cancel the dialog.

default\_ext – string – The default filename extension. If the user does not include an extension in the Filename edit box, the extension specified here is automa cally appended to the filename. If this parameter is empty (“”), no file extension is appended.

file\_name – string –path to default file to open. Can just be path or include filename or just be le empty (“”), usualy use “\*.ext” at end of string to preselect all files of a par cular extension.

filter – string –filter string to display for dialog file type.The string is build up of pairs of ‘file descrip on’ followed by ‘file extension’ separated by ‘|’. The filter list is terminated by ‘||’ . E.g “Toolpaths (\*.tap) | \*.tap||”. Example Code

local file\_dialog = FileDialog() if not file\_dialog:FileSave(

"tap",

"d:\\Temp\\\*.tap" ,

"Toolpaths (\*.tap)|\*.tap|"

) then

MessageBox("No file to save to selected") return false

end

MessageBox("User chose file ...\n" .. file\_dialog.PathName)

Proper es

.InitialDirectory

R/W – string – folder/directory the dialog should display when first opened

The proper es below are used to access the data on the file chosen and are only valid a er the dialog has returned from :FileOpen() or FileSave()

.Directory

R/O – string – directory/folder (and drive) of the file the user chose (same as .Folder)

.FileExt

R/O – string – extension of the file chosen

.FileName

R/O – string – filename without the path but including the extension

.FileTitle

R/O – string – name of file without path or extension

.PathName

R/O – string – The full pathname to the file the user

.Folder

R/O – string – folder/directory (and drive) of the file the user chose (same as .Directory)

## ProgressBar

This object is used to display a progress bar in the host program while doing me consuming tasks.

Constructor

ProgressBar(string text, ProgressBarMode progress\_bar\_mode) – Constructor Creates a new progress bar and displays it in the host program.

text – string – The text displayed to the le of the progress bar.

progress\_bar\_mode-integer-Type of progresss bar to create. Valid values are...

ProgressBar.LINEAR – Use this when you can compute your percentage progress while processing. ProgressBar.PINGPONG – Use this when you can’t compute your percentage progress.

Methods

:SetPercentProgress(integer percent)

Call this moderately frequently to update the progress bar with your percentage progress. Should be used in conjunc on with ProgressBar.LINEAR

percent-integer-The percentage task comple on between 0 and 100, does not have to be monotonically increasing in value. Progress bars can be reappropriated for different tasks by using SetText().

:StepProgress()

Call this moderately frequently to update the progress bar when you can’t complete your percentage progress. Should be used in conjunc on with ProgressBar.PINGPONG

:SetText(string text)

Sets the text displayed to the le of the progress bar.

text-string-The text to be displayed.

:Finished()

This should always be called when you’re finished with a progress bar.

### Components – Aspire Only

Components form the basis of 3D modeling within Aspire. A Component is created when a Relief is added to the ComponentManager. In order to reduce memory consump on, the Components use a Relief referencing scheme coordinated by the ComponentManager. In short several Components can reference a single Relief.

Example Code from 01\_Crea on.lua

-- Check for the existence of a job local job = VectricJob() if not job.Exists then return false end

-- Try and create a relief local pixel\_width = 1000 local pixel\_height = 1000 local width = 5.0 local height = 5.0

local relief = Relief(pixel\_width, pixel\_height, width, height)

-- Set its data

local max\_r = 0.5 \* math.sqrt(pixel\_width \* pixel\_width + pixel\_height \* pixel\_height) for y = 0, pixel\_height - 1 do

-- We want our waves to eminate from the centre local y\_r = (2 \* y - pixel\_height) / 2 for x = 0, pixel\_width - 1 do

-- We want our waves to eminate from the centre local x\_r = (2 \* x - pixel\_width) / 2

-- Compute our normalized radial position from the model centre local r = math.sqrt(x\_r \* x\_r + y\_r \* y\_r) / max\_r

-- Our amplitude is a gaussian function of our radial position local amp = math.exp(-10.0 \* (r - 0.3) \* (r - 0.3))

--Set the height from a cos function

local z = 0.5 + 0.2 \* amp \* math.cos(2 \* math.pi \* 5 \* r) relief:Set(x, y, z) end end

-- Add the relief to the manager local component\_manager = job.ComponentManager component\_manager:AddRelief(relief, CombineMode.Add, "01 Creation")

## CombineMode – Aspire Only

This is an ‘enum’ which represents the values used to specify a combine mode for many opera ons. Combine modes instruct Aspire how overlapping components should interact with each other.

Proper es

.Add

R/O – When a component overlaps with another at a pixel combine their heights addi vely

.Subtract

R/O – When a component overlaps with another at a pixel combine their heights subtrac vely

.MergeHeighest

R/O – When a component overlaps with another at a pixel combine their heights by taking the highest value

.MergeLowest

R/O – When a component overlaps with another at a pixel combine their heights by taking the lowest value

.Multiply

R/O – When a component overlaps with another at a pixel combine them by mul plying their heights together

.Replace

R/O – When a component overlaps with another at a pixel combine replacing pixels

## Relief – Aspire Only

Reliefs are two dimensional grids of pixels with a height value associated at each pixel. They are used to represent 2D height fields that can have transparent regions. Reliefs can be created programma cally and converted into Components by adding a relief to the ComponentManager or obtained by querying a Component for its relief. The posi on of a relief is set using the Transform methond on the associated component once it is created.

Constructors

Relief(integer pixel\_width, integer pixel\_height, double real\_width, double\_real\_height) –

Constructor

pixel\_width-integer–The width of the relief in pixels pixel\_height-integer –The height of the relief in pixels real\_width-double –The width of the relief in real world units real\_height-double –The height of the relief in real world units

Relief(Box2D bounds, loat pixel\_size) – Constructor bounds-Box2D –The real world dimensions of the relief pixel\_size-float –The side length of a pixel in real world units

Proper es

.Error

R/O – bool – true if there was an error during relief construc on

.PixelWidth

R/O – integer – the width in pixels of the relief

.PixelHeight

R/O – integer – the height in pixels of the relief

.RealWidth

R/O – float – the real width of the relief

.RealHeight

R/O – float – the real height of the relief

.XPixelSize

R/O – integer – the width of a pixel in real units

.YPixelSize

R/O – integer – the height of a pixel in real units

.Thickness

R/O – float – the thickness of the block in real units - 0.0 for reliefs created by Lua

.SurfaceZ

R/O – float – returns the surface z in real units – 0.0 for reliefs created by Lua

.Volume

R/O – float – returns the volume of the relief

.IsTransparent

R/O – bool – returns true if the relief consists en rely of transparent pixels

Methods

:Reset( loat value, bool free\_memory)

Sets all heights in the relief

value- float –The height to set all pixels in the relief free\_memory-bool –Has no effect, ignored

:MakeValuesTransparent( loat value)

Makes all parts of the relief at the given height transparent value-float –The height of the pixels to be made transparent

:ReplaceTransparentValues( loat value)

Replaces all transparent pixels with the given height value-float-The height that all transparent pixels will become

:PointIsOnModel(integer x, integer y)

Returns true if x is between zero and the pixel width and y is between zero and the pixel height

x-integer -The x coordinate of the pixel to test y-integer-The y coordinate of the pixel to test

:Set(integer x, integer y, loat value)

Sets the height for the relief at the point to the specified value

x-integer -The x coordinate of the pixel to set y-integer-The y coordinate of the pixel to set value-float-The height value to assign to the pixel

:SetLowest(integer x, integer y, loat value)

Sets the height for the relief at the point to whichever is lower, the current value or the passed value. Returns true if the height was changed

x-integer-The x coordinate of the pixel to set y-integer-The y coordinate of the pixel to set value-float-The height value to try and assign to the pixel

:Get(integer x, integer y)

Returns the height for the relief at the point in pixel coordinates

x-integer -The x coordinate of the pixel to get y-integer-The y coordinate of the pixel to get

:GetInterpolated(double x, double y)

Returns the interpolated height for the relief for the specified coordinate

x-double -The x coordinate in pixel units y-double -The y coordinate in pixel units

:GetMinMaxZ()

Returns the maximum and minimum z values of the relief taking into account the thickness and a boolean flag indica ng if the model is transparent

local transparent, min\_z, max\_z = relief:GetMinMaxZ()

:GetTrueMinMaxZ()

Returns the maximum and minimum z values of the relief ignoring the thickness and a boolean flag indica ng if the model is transparent

local transparent, min\_z, max\_z = relief:GetTrueMinMaxZ()

:Add( loat value)

Adds onto all non transparent pixels vaIue-float -The height to add onto each non transparent pixel

:Subtract( loat value)

Subtracts from all non transparent pixels vaIue-float -The height to subtract from each non transparent pixel

:Multiply( loat value)

Mul ply all non transparent pixels vaIue-float -The height to mul ply each non transparent pixel

:MergeHeighest( loat value, bool is\_abs\_value, bool preserve\_transparent)

Merge heighest over the en re relief

value-float -The height value to merge into the relief is\_abs\_value-bool -If true the relief’s Z offset is taken into account preserve\_transparent-bool - If true transparent pixels remain transparent

:MergeLowest( loat value, bool is\_abs\_value, bool preserve\_transparent)

Merge lowest of the en re relief

value-float -The height value to merge into the relief is\_abs\_value-bool -If true the relief’s Z offset is taken into account preserve\_transparent-bool - If true transparent pixels remain transparent

:CombineReliefs(Relief relief, CombineMode combine\_mode)

Combines the relief

relief-Relief -The relief to combine into this

combine\_mode- CombineMode -How the relief should be combined :BlendBetweenReliefs(Relief irst\_relief, Relief second\_relief, double blend\_factor) Returns a new relief that is a blended version of the two passed reliefs

first\_relief-Relief -The first relief to blend

second\_relief-Relief -The second relief to blend, must have the same dimensions as the first blend\_factor-double –0.0 all first, 1.0 all second

:FlipY()

Flips the relief mirroring it ver cally

:FlipX()

Flips the relief mirroring it horizontally

:TransposeXY()

Transposes the relief’s pixels

:RotateClockwise90()

Rotates the relief 90 degrees clockwise

:RotateCounterClockwise90()

Rotates the relief 90 degrees counter clockwise

:RenderTriangle(Point3D p1, Point3D p2, Point3D p3, bool merge\_high)

Renders a triangle into the relief

p1-Point3D -The first point of the triangle p2-Point3D -The second point of the triangle p3-Point3D -The third point of the triangle merge\_high-bool –If true merge the triangle high

:CreateSlice( loat min\_z, loat max\_z, bool make\_below\_min\_z\_transparent) Returns a relief sliced between the specified Z range

min\_z-float-The minimum Z value to include max\_z-float-The maximum Z value to include

make\_below\_min\_z\_transparent-bool-If true heights below min\_z become transparent

:ColumnIsTransparent(integer x, loat transparent\_value)

Returns true if the column consists en rely of pixels that are of the transparent value

x-integer-The column pixel coordinate

transparent\_value-float-The Z value to consider transparent

:RowIsTransparent(integer y, loat transparent\_value)

Returns true if the row consists en rely of pixels that are of the transparent value

x-integer-The row pixel coordinate

transparent\_value-float-The Z value to consider transparent

:CreateSectionCopy(integer min\_x, integer min\_y, integer max\_x, integer max\_y) Returns a subset of the relief between the coordinate ranges

min\_x-integer-The minimum x pixel coordinate to include min\_y-integer- The minimum y pixel coordinate to include max\_x-integer- The maximum x pixel coordinate to include max\_y-integer- The maximum y pixel coordinate to include

:TiltRelief(Point2D anchor\_pt, Point2D direction\_pt, double angle)

Tilts the relief

anchor\_pt-Point2D –Start of vector determining lt direc on direc on\_pt-Point2D –End of vector determining lt direc on angle-double –The the angle to lt in degrees

## ComponentManager – Aspire Only

This object is responsible for managing all the components within the applica on. A reference to the ComponentManager is obtained via the ComponentManager property of the VectricJob. The ComponentManager manages all the Components in the job.

local component\_manager = job.ComponentManager

Methods

:AddRelief(Relief relief, CombineMode combine\_mode, string name)

Creates a Component from a Relief and returns the UUID of the component created

relief-Relief- The relief to add as a component

combineMode- CombineMode -The combine mode to assign to the component name -string-The name to give to the component created from the relief

:FindObjectIdWithName(string name)

Returns the UUID of the component with the specified name name-string - Name for the component to find

:FindObjectWithId(UUID uuid)

Returns the Component with the given id uuid- UUID-The unique id of the component to find

:GetSelectedObjectIds()

Returns a UUID\_List containing the UUIDs of the currently selected components

:GetTopLevelObjectIds()

Returns a UUID\_List containing the UUIDs of the top level components in the component tree

:DeleteObjectWithId(UUID uuid)

Returns true if the Component with the given is UUID deleted uuid- UUID-The unique id of the component to delete

:DeleteObjectsWithIds(UUID\_List uuids)

Returns true if the Components with the UUIDs in the list are successfully deleted uuids- UUID\_List -The unique ids of the components to delete

:CreateBakedCopyOfObjects(UUID\_List uuids, bool delete\_originals) Returns the UUID of the created component uuids-UUID\_List- The list of component ids to bake

delete\_orignal-bool –If true delete the original components

:CreateBakedCopyOfObject(UUID uuid, bool delete\_originals) Returns the UUID of the created Component

uuid-UUID - The id of the component to bake delete\_orignal-bool -If true delete the original components

:GroupObjectsWithIds(UUID\_List uuids)

Group the Components with the ids in the list. Returns the UUID of the grouped component uuids- UUID\_List -The unique ids of the components to group

:GroupSelectedObjects()

Group the currently selected components. Returns the UUID of the grouped component

:UnGroupSelectedObjects()

Ungroup the currently selected components. Returns true if the ungroup was successful

:UnGroupObjectsWithId(UUID uuid)

Ungroup the Component with the given UUID in the list. Returns true if the component was ungrouped uuid- UUID -The unique id of the component to ungroup

:SelectionCanBeGrouped()

Returns true if the selec on can be grouped

:SelectionCanBeUngrouped()

Returns true if the selec on can be ungrouped

:CloneObjectWithId(UUID uuid, bool make\_unique) Returns the cloned version of the Component

uuid- UUID - The id of the component to clone

make\_unique-bool –If true make a deep copy of the component

:CreateReliefFromObjectWithId(UUID uuid)

Returns a Relief cloned from the Component with the given id uuid- UUID - The id of the component to clone a relief from

:ReplaceComponentRelief(UUID target\_uuid, UUID source\_uuid)

Returns true if the ComponentManager was able to replace the target component’s Relief with that of the source

target\_uuid-UUID - The id of the component whose relief we are to replace

source\_uuid-UUID - The id of the component whose relief we are to use as the replacement

:UpdateCompositeModel()

Updates the Composite Model. This method should always be called when you have completed a set of Component opera ons to ensure that the Composite Model displayed to the user is up to date.

:CreateReliefFromCompositeModel()

Creates a Relief from the Compsite Model

:UpdatePreviews()

Updates the Component Previews in the 2D view. This method should always be called when you have completed a set of Component opera ons to ensure that the Component Previews are correctly posi oned

## Component – Aspire Only

This object is the base class for all the different types of components managed by the

ComponentManager. There are two classes of Component available to Lua; Component itself and ComponentGroup. Components reference the Reliefs that were used to create them. It is possible to have mul ple Components referring to the same original Relief and this should be kept in mind while manipula ng Component data.

Proper es

.Id

R/O - UUID - Returns the id of the component

.CombineMode

R/W - CombineMode - The combine mode of the component

.IsTilted

R/O – bool -Returns true if the component is lted

.UseTilt

R/W – bool – Is the component using lt

.IsFaded

R/O – bool – Returns true if the component is faded

.UseFade

R/W – bool – Is the component using fade

Methods

:Transform(Matrix2D xform, bool update\_all) Tranforms the component

xform- Matrix2D -The transform

update\_all -bool-Update the en re bounds of the component tree

:SetZScaleAuto(bool auto\_scale)

Determine whether transforming the component using :Transform() will automa cally scale it in the Z axis, as well.

auto\_scale - bool – Flag which determine whether Z is scaled.

:SetZOffset( loat z\_offset)

Set the Z offset of the component z\_offset-float-The amount by which to offset the component

:SetZScale( loat scale)

Scale the Z heights of the component

scale-float-The amount by which to scale the heights of the component

:TiltWouldChange(Point2D anchor\_pt, Point2D direction\_pt, double tilt\_angle, bool do\_tilt)

Returns true if the parameters would change the current lt

anchor\_pt-Point2D -Start of vector determining lt direc on direc on\_pt- Point2D -End of vector determining lt direc on

lt\_angle-double -The angle to lt at in degrees

do\_ lt -bool -If true we want to apply a lt

:SetTilt(Point2D anchor\_pt, Point2D direction\_pt, double tilt\_angle) Returns true if the component was lted successfully

anchor\_pt-Point2D -Start of vector determining lt direc on direc on\_pt- Point2D-End of vector determining lt direc on

lt\_angle-double -The angle to lt at in degrees

:GetTiltData(Point2D anchor\_pt, Point2D direction\_pt)

Returns the lt angle and the passed points are updated to contain the component’s lt anchor and direc ons points

anchor\_pt-Point2D -Start of vector determining lt direc on direc on\_pt- Point2D -End of vector determining lt direc on

local anchor\_pt = Point2D() local direction\_pt = Point2D()

local tilt\_angle = component:GetTiltData(anchor\_pt, direction\_pt)

:FadeWouldChange(Point2D anchor\_pt, Point2D direction\_pt, double end\_fade\_val, bool do\_fade)

Returns true if the parameters would change the current fade

anchor\_pt-Point2D -Start of vector determining fade direc on direc on\_pt- Point2D -End of vector determining fade direc on end\_fade\_val-double -The final value of the fade do\_fade-bool -If true we want to apply a fade

:SetFade(Point2D anchor\_pt, Point2D direction\_pt, double tilt\_angle) Returns true if the component was faded successfully

anchor\_pt-Point2D -Start of vector determining fade direc on direc on\_pt- Point2D -End of vector determining fade direc on end\_fade\_val-double -The final value of the fade

:GetFadeData(Point2D anchor\_pt, Point2D direction\_pt)

Returns the end fade value and the passed points are updated to contain the component’s fade anchor and direc ons points anchor\_pt-Point2D -Start of vector determining fade direc on

direc on\_pt- Point2D -End of vector determining fade direc on

local anchor\_pt = Point2D() local direction\_pt = Point2D()

local tilt\_angle = component:GetTiltData(anchor\_pt, direction\_pt)

:GetRelief()

Returns the Relief for the component. If the component is a ComponentGroup then this will return nil. Please note any changes made to this relief will affect the appearance of any other components that reference this Relief

## ComponentGroup – Aspire Only

This object represents a group of Components. A ComponentGroup may contain ComponentGroups.

Proper es

.GetCount

R/O – integer – returns the number of components in the group

.Count

R/O – integer – returns the number of components in the group

.IsEmpty

R/O – bool – returns true if the group is empty

Methods

:GetHead()

Return the component at the head of the group, the component remains in the group

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list of components in the group

:GetTailPosition()

Returns a POSITION variable to allow access to the tail of the list of components in the group

:GetNext(POSITION pos)

Returns the component at the current posi on AND a new value for posi on poin ng to the next item in the group (or nil if at end of group)

pos - POSITION - current posi on in group

:GetPrev(POSITION pos)

Returns the component at the current posi on AND a new value for posi on poin ng to the previous item in the group (or nil if at end of group)

pos - POSITION - current posi on in group

:GetAt(POSITION pos)

Returns the component at the passed posi on pos - POSITION - posi on in group

## DocumentVariableList

This object is responsible for managing all the document variables associated with a Job. A reference to a DocumentVariableList is obtained via the DocumentVariables property of the VectricJob. local document\_variable\_list = job.DocumentVariables

Methods

: DocumentVariableExists (string variable\_name)

Returns true if a DocumentVariable with the passed name exists

Variable\_name - string - Name for variable

: GetDocumentVariable (string variable\_name)

Returns a DocumentVariable with the passed name if exists otherwise nil

Variable\_name - string - Name for variable

: SetDocumentVariable (string variable\_name, double value)

Sets the DocumentVariable with the specified name to the passed value. It either creates a new variable if one with the passed name doesn’t exist or updates the current one to the new value. Please see the documenta on for IsInvalidDocumentVariableName as to what cons tutes a valid variable name.

Variable\_name - string - Name for variable

Value - double – value of the variable

: DocumentVariable True(string variable\_name)

Returns true if a DocumentVariable with the passed name is deleted otherwise false. Please note DocumentVariables considered to be in use cannot be deleted.

Variable\_name - string - Name for variable

:GetHeadPosition()

Returns a POSITION variable to allow access to the head of the list of DocumentVariables in the list

:GetNext(POSITION pos)

Returns the DocumentVariable at the current posi on AND a new value for posi on poin ng to the next item in the list (or nil if at end of list) pos - POSITION - current posi on in group

## DocumentVariable

This object is a name-value pair that represents a Document Variable within a VectricJob.

Proper es

.Name

R/O – string – returns the name of the DocumentVariable

.Value

R/O – double – returns the value of the DocumentVariable