Lanner Group Limited

3D Command Reference

Reference document detailing the data commands published by WITNESS for 3D visualization

Version 20

Changes

Date	Version	Description
12-Dec-12	03	Added the New Command.
		Split the <transformation> command into separate rotate, scale and translate commands.
		Removed the version attribute from all commands.
		Added the Window command.
		Added AnimationStart, AnimationUpdate and AnimationStop commands.
20-Dec-12	04	Added the Version Command.
		Renamed the <queue> element used in the Update command to <partPosition>.
		Added an attribute to the Create command specify if the path is under or over.
		Added qualification that the transformation (rotate, scale & translate) are absolute (not additive).
23-Jan-13	05	Added <geometryInfo> element containing data returned by the <load> command.
		Added the Extrude command used to specify simple shapes that are extruded to form walls/boundaries.
		Added another version of create to create a textured surface, used primarily for floors.
29-Jan-13	06	Removed the time attribute from the Extrude sub command.
		Added a constraint attribute to the queueInfo to support queue types that hang vertically.
		Added the EnumerateMaterial command
20-Feb-13	07	Removed the line extrusion command
10-May-13		Added a description table for the Enumerate materials command response XML snippet. Removed a TODO about usage of materials in extrusions, this should now be cover in the newer extrude command. Removed a question that has been answered about how the items on tracks, conveyors and sections “dangle”.
13-June-13	08	Added a new attribute to the path update command indicating entity travel direction.
19-June-13		Added new attribute to the create command indicating the type of object we are creating.
20-June-13		Added the error xml format for reading errors returned from the server.
26-June-13		Added a new appendix to describe the update command and rotate behaviour.
		Added table of contents.
TODO	09	Added the start point (x,y,z) to each line segment in the path create command
21-August-13	10	Added the Visible attribute to the Update command.
16-September-13	12	Added the create commands for text billboards, also added an update command.
10-October-13	13	Change the floor surafec to include the attributes for color1 and color2 and the singleSided flag
18-November-13	14	Remove the EnumerateMaterial command as it is not required.

12-January-14	15	Added the queue info xml protocol supplied during create.
05-March-14	16	Added the material element to the Update command. Renamed the Cx, Cy, Cz attributes in the update command for Paths.
31-March-14	17	Added a new security element to allow 3D features to be enabled / disabled
02-May-14	18	Added the transform query command, and re-added the explicit extrusion of line command in the create section.
02-July-14	19	Add the opacity attribute to the extrusion and material update commands.
08-December-14	20	Add the Start command
09-February-15	21	Added the model attribute to the new command
		Removed TODO sections that are no longer applicable or have been specified elsewhere in this document

Table of Contents

Version Command

The Version Command is a WITNESS notification sent as the first command in the sequence.

Example:

```
<version number="1" />
```

Attribute	Description
number	The version number used to version all subsequent commands.

Load Command

The Load Command is a WITNESS notification indicating that a 3D shape is required. The Load command returns the geometry bounding box.

Example:

```
<load geometry="machine.dae" />
```

Attribute	Description
geometry	The name of the 3D shape associated with a WITNESS 2D element display.

Returns one of two xml types:

```
  <geometryInfo geometry="machine.dae"
                x1="-10"
                y1="-20"
                z1="-30"
                x2="10"
                y2="20"
                z2="30">
    <queueInfo length="3.0"
               alignmentDirectionX="0.0"
               alignmentDirectionY="1.0"
               alignmentDirectionZ="0.0"
               constraint="Hang Vertical">
      <itemRotate order="xzy" x="90" y="45" z="90" />
    </queueInfo>
  </geometryInfo>
```

Attribute/Element	Description
geometryInfo	The name of the 3D shape that the bounding box information is for.
x1	The x co-ordinate of one corner of a bounding box.
y1	The y co-ordinate of one corner of a bounding box.
z1	The z co-ordinate of one corner of a bounding box.
x2	The x co-ordinate of the diagonally opposite corner of the bounding box.
y2	The y co-ordinate of the diagonally opposite corner of the bounding box.
x2	The z co-ordinate of the diagonally opposite corner of the bounding box.
queueInfo	Optional element for geometries that have embedded queue definitions.
length	The current length of the queue within the geometry (Is this altered by instance transform? Hopefully not otherwise each individual instance will

	need querying because each instance will depend on scaling)
alignmentDirectionX	The x component of the direction vector used to indicate under/over or maybe even diagonal item placement. We need this to offset parts as we add them for over slung etc.
alignmentDirectionY	The y component of the direction vector used to indicate under/over or maybe even diagonal item placement. We need this to offset parts as we add them for over slung etc. If points downwards this indicates under slung, upwards indicates over slung or can be off to the side.
alignmentDirectionZ	The z component of the direction vector used to indicate under/over or maybe even diagonal item placement. We need this to offset parts as we add them for over slung etc.
constraint	Optional attribute indicating any constrains that applies to the queue. This is an enumerated type with the only supported value being “Hang Vertical” which indicates that the queue always hangs towards the ground.
itemRotate	Optional element indicating how an item added to the queue should be rotated for the queue. Allows calculations for correct positioning of the item in the queue. For example: the normal orientation of a car body in a model may be vertical (hanging from bumper to front of car) however in a particular queue the car bodies may stack floor to roof.
order	The ordering to apply the x,y and z rotations to the added item.
x	The angle in degrees to rotate the part about its x axis.
Y	The angle in degrees to rotate the part about its y axis.
Z	The angle in degrees to rotate the part about its z axis.

Error Response

If a command to the 3D server gives an error then the result channel will not contain the expected xml snippet but will instead contain an error snippet. This means that an implementation needs to check for the error response as well as the expected response because it code be either.

```
<error code="5001" oscode="3">
       <![CDATA[Failed to load the geometry machine.dae file not found.]]/>
</error>
```

Attribute/Element	Description
error	The XML tag indicating an error element
code	This will be one of a predefined, mutually understood set of enumerated values for the error codes.
oscode	The optional os error code.
CDATA	This is a text section inside the error element giving a reasonable textual description of the error for internal logging, not for user interface.

Create Command

The Create command is a WITNESS notification to create an instance of a 3D shape. Some WITNESS element types are drawn as paths, in this case the path information is specified with the create command.

Create Example 1: create request without path information

```
<create time="0.0"
        geometry="machine.dae"
        type="machine/part/path etc "
        instanceName="w.machine001(1)-icon1">
  <queueInfo queueParent="machine.dae">
       <behaviour partPositioning="partOver/partUnder/partCentre" partRoll="0.0"
                  partPitch="0.0" partYaw="0.0"/>
       <position x="0" y="0" z="0"/>
       <direction dx="0" dy="0" dz="1"/>
  </queueInfo>
</create>
```

Create Example 2: create request with path information

```
<create time="0.0" geometry="track.dae"
        type="machine/part/path etc "
        instanceName="w.track001(1)-Path">
  <queueInfo queueParent="track.dae">
       <behaviour partPositioning="partOver/partUnder/partCentre" partRoll="0.0"
                  partPitch="0.0" partYaw="0.0"/>
       <position x="0" y="0" z="0"/>
       <direction dx="0" dy="0" dz="1"/>
  </queueInfo>
  <path startX="80" startY="0" startZ="160" cz="25">
    <line endX="160" endY="0" endZ="160" />
    <arc centerX="160" centerY="0" centerZ="60"
         angle="90" sweepDirection="counterclockwise"/>
  </path>

</create>
```

Create Example 3: Create request for a floor/surface

```
  <create time="0.0" instanceName="floor1"
       type="machine/part/path etc etc">
    <surface x1="-100.0" y1="0" z1="-150.0"
             x2="100.0" y2="0.0" z2="150.0"
             normalX="0" normalY="1.0" normalZ="0.0"
             color1="#fefefeff" color2="#efefefff" singleSided="True/False"/>

      <carpet texture="carpet1.jpg/bmp/emf/wmf"
              x1="0.0" y1="0.0" x2="200.0" y2="300.0"
              cx="20.0" cy="30.0"
              side="front | back | both"
              mode="stretch | tile | center"/>

      <carpet texture="carpet2.jpg/bmp/emf/wmf"
              x1="50.0" y1="50.0" x2="200.0" y2="300.0"
              cx="15.0" cy="25.0"
              side="front | back | both"
              mode="stretch | tile | center"/>
    </surface>
  </create>
```

Note: Multiple carpets are specified as multiple sub elements of the surface with the render stacking order as given in the XML element ordering.

Extrude Example: Using textures on extrusions

```
<create time="0.0" instanceName="extrude2"
        type="machine/part/path etc ">

  <extrude width="1" height="60" texture="steel.jpg/bmp/emf/wmf"
       cx="15.0" cy="25.0"
       mode="stretch | tile"/>

      <ellipse centerX="100" centerY="0" centerZ="200" radiusX="60" radiusZ="30" />

  </extrude>

</create>
```

NOTES

- If we want to create an extrusion with a solid colour we will create a Jpeg containing the solid colour and pass it to the Texture part of the extrusion.
- The texture will be applied to all faces of the extrusion.
- We could then remove the optional colour attribute from the extrude command as this is noew handled with the texture command.

Texture - The filename of the file containing the texture
Cx - This is the notional width of bmp/jpg/emf/wmf
Cy - This is the notional height of bmp/jpg/emf/wmf
Mode - Indicates the action to be taken when the texture is smaller than the available space, values are tile and stretch

Create Example 4: Create an extrusion of an ellipse

```
<create time="0.0" instanceName="extrude2"
       type="machine/part/path etc ">

  <extrude width="1" height="60" texture="Concrete.jpg">
      <ellipse centerX="100" centerY="0" centerZ="200" radiusX="60" radiusZ="30" />
  </extrude>

</create>
```

Create Example 5: Create an extrusion of a line

```
<create time="0.0" instanceName="extrude1">

  <extrude time="0.0" width="1" height="60" color="#990000">
      <line startX="100" startY="0" startZ="0" endX="160" endY="0" endZ="60"/>
  </extrude>

</create>
```

Create Example 6: Create an extrusion of a rectangle

```
<create time="0.0" instanceName="extrude3"
       type="machine/part/path etc ">

  <extrude width="1" height="60" color="#FF0000">
      <rectangle startX="100" startY="0" startZ="200" endX="160"
          endY="0" endZ="360" hollow="true"/>
  </extrude>

</create>
```

Create Example 7: Create request for a free standing text billboard placed at specified point.

```
  <create time="0.0" instanceName="text1" x="160" y="0" z="160"
    <text>
       <![CDATA[A character block within the XML.]]>
    </text>
  </create>
```

Create Example 7: Create request for a text billboard parented/attached to a parent instance. This will hover just above the parent during simulation run.

```
  <create time="0.0" instanceName="text2" parentInstance="Parent Instance Name"
    <text>
       <![CDATA[A character block within the XML.]]>
    </text>
  </create>
```

Element/Attribute	**Description**
Time	Simulation time.
Geometry	The name of the 3D shape associated with a WITNESS 2D element display.
instanceName	The unique name associated with a 2D display, generated by WITNESS.
Path	Optional element used to specify the path position for element types: Buffer, Conveyor, WITNESS Path element, Power and Free Sections & Tracks.
startX	The path starting x position.
startY	The path starting y position.
startZ	The path starting z position.
Length	Optional. (Internal note: Unused at present).
Height	Optional height. (Internal note: can be specified using the height in the 3D override options). If not specified the default geometries height is used.
Width	Optional width. (Internal note: derived from the display size).
Line	A path can be made up of one or more Line segments. When a Line segment is used, the starting point is the end point of the previous segment.
startX	The Line starting x position.
startY	The Line starting y position.
startZ	The Line starting z position.
endX	The Line segment ending x position.

endY	The Line segment ending y position.
endZ	The Line segment ending z position.
arc	A path can be made up of one or more Arc segments. When an Arc segment is used, the starting point is the end point of the previous segment.
Angle	The amount (in degrees) by which the ellipse is rotated from the endpoint.
sweepDirection	Specifies whether the arc is drawn in the clockwise or counterclockwise direction. Possible values [“clockwise” | “counterclockwise”].
centerX	The x component of the center point of the Arc.
centerY	The y component of the center point of the Arc.
centerZ	The z component of the center point of the Arc.
Surface	Optional element indicating a surface to create (e.g. floor)
x1	X component of one part of the diagonal corners of the surface in world coordinates
y1	Y component of one part of the diagonal corners of the surface in world coordinates
z1	Z component of one part of the diagonal corners of the surface in world coordinates
x2	X component of one part of the diagonal corners of the surface in world coordinates
y2	Y component of one part of the diagonal corners of the surface in world coordinates
z2	Z component of one part of the diagonal corners of the surface in world coordinates
normal	The X component of the surface normal, indicates the top side onto which the texture is applied.
normalY	The Y component of the surface normal, indicates the top side onto which the texture is applied.
normal	The Z component of the surface normal, indicates the top side onto which the texture is applied.
Color1	The rgba color1 for the chequered surface, whether it is 32 bit value in hexadecimal (html like). This is the material color to cover the surface front and back.
Color2	The rgba color2 for the chequered surface, whether it is 32 bit value in hexadecimal (html like). This is the material color to cover the surface front and back.
singleSided	This is a True/False value indicating whether the floor surface is doubled sided or not. True indicates single sided, False if double sided
carpet	Optional element if there is a texture to be applied
x1	The texture coordinates are specified as a box by two diagonal points on the surface. If viewing the surface from “in front” the origin is top left and the coordinates are 2D on the surface. This is the top left x component.
y1	The texture coordinates are specified as a box by two diagonal points on the surface. If viewing the surface from “in front” the origin is top left and the coordinates are 2D on the surface. This is the top left y component.
x2	The texture coordinates are specified as a box by two diagonal points on the surface. If viewing the surface from “in front” the origin is top left and the coordinates are 2D on the surface. This is the bottom right x component.
y2	The texture coordinates are specified as a box by two diagonal points on the surface. If viewing the surface from “in front” the origin is top left and

	the coordinates are 2D on the surface. This is the bottom right y component.
side	Indicates which side of the surface we are apply the texture, the front side is indicated by the surface normal. Values are front, back, both.
mode	Indicates the action to be taken when the texture is smaller than the available space, values are tile, stretch and center.
extrude	Optional element used to extrude 2D backdrop elements such as Lines, Ellipses and Rectangles which have no logical 3D mapping. These simple extrusions can be used to generate a basic skeleton of a building (walls, columns etc.)
width	The width of the extruded item. The width is spread equidistance between the startX and startZ points.
height	The height of the extrusion.
color	Optional The rgba color components for the color that is to be used on all faces of the extrusion. It is 32 bit value in hexadecimal (html like).
texture	Optional The filename containing the texture to be used on all faces of the extrusion. The texture is stretched to fit each face.
opacity	Optional The opacity of the texture. The opacity is between 0 and 1.
cx	This is the notional width of bmp/jpg/emf/wmf
cy	This is the notional height of bmp/jpg/emf/wmf
ellipse	The extrusion of an ellipse.
centerX	The Ellipses center x position.
centerY	The Ellipses center y position.
centerZ	The Ellipses center z position.
Radius	The Ellipses x radius.
Radius	The Ellipses z radius.
rectangle	The extrusion of a rectangle.
startX	The Rectangles starting x position.
startY	The Rectangles starting y position.
startZ	The Rectangles starting z position.
endX	The Rectangles ending x position.
endY	The Rectangles ending y position.
endZ	The Rectangles ending z position.
hollow	False if a solid rectangle is to be extruded or True if just the perimeter of the rectangle is to be extruded.
type	This is an indication of the type of WITNESS object the create command is associated with. For example: Buffer/machine/path etc.
x	The x coordinate of the centre point of the created object
y	The y coordinate of the centre point of the created object
z	The z coordinate of the centre point of the created object
parentInstance	When creating an object that has child relationship, this attribute names the parent instance.
text	When creating a text billboard this element will contain a CDATA text block. This is the initial text to display on the billboard.
queueInfo	This element gives informationon the queue attached to the object being created.
queueParent	The name of the node that queue is within, allow positioning of the queue on a sub-node such as the hand of a robot arm. This is optional and if not given should be taken as the top level of the geometry.
behaviour	This element is a child element of queueInfo, it details the behaviour of

	parts entering the queue.
partPositioning	An attribute of behaviour can be partOver/partUnder/partCentre how parts "sit" on the queue line, they can be above the queue, below the queue or threaded on the queue.
partRoll	An attribute of behaviour if a part should have a roll value when added to the queue, angle in degrees.
partPitch	An attribute of behaviour if a part should have a pitch value when added to the queue, angle in degrees.
partYaw	An attribute of behaviour if a part should have a yaw value when added to the queue, angle in degrees.
position	This element is a child element of queueInfo, it details the start position of the queue relative to the queueParent container node.
x	An attribute of position giving the x offset of the queue start from the queueParent container nodes centre.
y	An attribute of position giving the y offset of the queue start from the queueParent container nodes centre.
z	An attribute of position giving the z offset of the queue start from the queueParent container nodes centre.
direction	This element is a child element of queueInfo, it details the direction of the queue relative to the queueParent container node.
dx	An attribute of position giving the dx vector component of the queue direction relative to queueParent container nodes current frame.
dy	An attribute of position giving the dy vector component of the queue direction relative to queueParent container nodes current frame.
dz	An attribute of position giving the dz vector component of the queue direction relative to queueParent container nodes current frame.

Note: All coordinates (centerX, centerY, etc.) above are specified in world coordinates.

Update Command

The Update command is a WITNESS notification raised when an element changes position. The command contains sub-elements to specify separate transforms. The order these transforms appear in the xml is the order they should be applied.

Example: update the position, orientation & size of a machine instance which is made visible if not already visible.

```
<update time="0.0" instanceName="w.machine001(1)-icon1">

  <rotate order="xzy" x="90" y="45" z="90" />
  <scale x="1" y="1.2" z="2" />
  <translate x="vrX + shapeX" y="vrY + flooroffset" z="vrZ + shapeY" />

 </update>
```

Example: update the position, orientation & size of a machine instance and make it invisible.

```
<update time="0.0" instanceName="w.machine001(1)-icon1" visible="false">

  <rotate order="xzy" x="90" y="45" z="90" />
  <scale x="1" y="1.2" z="2" />
  <translate x="vrX + shapeX" y="vrY + flooroffset" z="vrZ + shapeY" />

 </update>
```

Similarly any of these transformations can be applied to a part instance.

Example: rotate a part by 90 degrees.

```
<update time="0.0" instanceName="part001-100">

  <rotate order="x" x="90" />

</update>
```

The coordinates specified for the translate command are in local coordinates.

Optionally the update command can also specify queue information for positioning dynamic entities (e.g. parts, people, etc.).

Example: update the position on a part in a queue.

```
<update time="0.0" instanceName="part001-100">

  <partPosition instanceName="w.buffer001(1)-Queue" position="0.1" reverse="true" />

</update>
```

Any created text billboard instance can be updated using the following update command.

```
<update time="0.0" instanceName="text1"
  <text>
     <![CDATA[An updated character block within the XML.]]>
  </text>
</update>
```

Any instance can have all its textures replaced with a specified material using the following update command.

```
<update time="0.0" instanceName=" w.machine001(1)-icon1"
  <material textureName="c:\images\brick.jpg" />
</update>
```

Element/Attribute	Description
Time	Simulation time.
instanceName	The unique name associated with a 2D display, generated by WITNESS.
Rotate	Optional attribute. Rotation information.
X	X-axis rotation (degrees).
Y	Y-axis rotation (degrees).
Z	Z-axis rotation (degrees).
Scale	Optional attribute. Scale information.
X	X-axis scale factor.
Y	Y-axis scale factor.
Z	Z-axis scale factor.
Translate	Optional attribute. Translate information.
X	Optional attribute. The x position in local coordinates. (Internal note: this is

		calculated as vrX + shapeX).
	Y	Optional attribute. The y position in local coordinates. (Internal note: this is calculated as vrY + floorOffset).
	Z	Optional attribute. The z position in local coordinates. (Internal note: this is calculated as vrZ + shapeY).
partPosition		Optional element used to specify the queuing position (in % percentage format).
	instanceName	The unique queue (path name) name where the shape is located.
	Position	The position of the shape in the queue expressed as a percentage along the path e.g. 0..1. In some circumstances it may be less than 0 or greater than 1. Note: Queuing positions can be greater than 1 or less than 0. There are two possible circumstances where the Position will be greater than 1 or less than 0. • When a part is entering or transitioning between continuous conveyors the part will be at a –ve percentage down the conveyor. This is because the front edge of the part is level with the rear edge of the conveyor. • If the length of the vector that describes a queuing direction is not capable of encompassing all the parts within the queue then the position parameter will be greater than 1 which signifies that the length of the vector should be increased to accommodate the part.
Reverse		This indicates whether the forward direction of the entity on the path should be reversed. This is used for bi-directional paths, it is optional and if not specified we assume a default of “false” indicating forward moving.
Visible		This indicates whether the instance should be made visible or invisible after the update has occurred. This is used for the initial orientation of the instance after a create command has been issued, it is optional and if not specified we assume a default of “true” indicating that the instance is made visible.
Text		When creating a text billboard this element will contain a CDATA text block. This is the initial text to display on the billboard.
material		Optional element used to specify the material to be used to replace all the textures on the instance.
	textureName	The full path to the texture file.
	opacity	Optional The opacity of the texture from 0 to 1.

Query Command

The Query Command is a WITNESS query information shape.

Example:

```
<query time="0.0" type="transform" instanceName="w.machine001(1)-icon1" />
```

Attribute	Description
Time	Simulation time.
type	The type of query, currently only support “transform” that queries the position, rotation and scaling of an instance.
instanceName	The unique name associated with a 2D display, generated by WITNESS. This is the

	same name used by the Create and Update commands.

The Query “transform” Command will prompt the server to send a “transform” response XML snippet that gives the position, rotation and scaling of an instance that was queried.

Example:

```
<transform x="1.0" y="2.0" z="3.0" roll="90.0" pitch="180.0" yaw="270.0" sx="1.0"
sy="2.0" sz="3.0" />
```

Attribute	Description
x,y,z	The (X,Y,Z) absolute position in the 3D world of the instance queried.
roll,pitch,yaw	The rotation orientation of the instance queried.
sx,sy,sz	The scale of instance queried.

Delete Command

The Delete Command is a WITNESS delete notification for a shape.

Example:

```
<delete time="23.0" instanceName="part001-100" />
```

Attribute	Description
time	Simulation time.
instanceName	The unique name associated with a 2D display, generated by WITNESS. This is the same name used by the Create and Update commands.

New Command

The New Command is a WITNESS notification issued when a new empty model is created, e.g. File | New model command. The model attribute will contain the fully qualified path of the model. **NOTE** it is possible that the model attribute could be empty if the user has created a model from scratch and has not saved the model before the 3D build is started.

Example:

```
<new model=”C:\3D Models\Cartons.mod”/>
```

Start Command

The Start Command is a WITNESS notification issued when a significant event has happened within WITNESS. There are currently three event types broadcast:

- “build” – We have started the static build/generate section.
- “run” – We have just started a model run.
- “session” – We have just started the server.

Examples:

```
<start time="0.0" type="build" />
<start time="3000" type="run" />
<start type="session" />
```

End Command

The End Command is a WITNESS notification issued when a significant event has happened within WITNESS. There are currently three event types broadcast:

- “build” – We have ended the static build/generate section.
- “run” – We have just completed a model run.
- “session” – We want the server to terminate.

Examples:

```
<end time="0.0" type="build" />
<end time="3000" type="run" />
<end time="3000" type="session" />
```

Window Command

The Windows Command is a WITNESS notification to supply the area to render to. It is broadcast when created and also on any size / position changes.

Example:

```
<window hwnd="19667304" />
```

Element/Attribute	Description
Hwnd	The window handle.

AnimationStart Command

The AnimationStart command is a WITNESS command that signals the start of an animation.

Example: starting an animation named “lift”.

```
<animationStart time="100.23"
                name="lift"
                instanceName="w.machine001(1)-icon1"
                duration="100"
                repeatCount="5" />
```

AnimationUpdate Command

The AnimationUpdate command is a WITNESS command issued as a hint to update an animation.

Example: updating an animation named “lift”.

```
<animationUpdate time="110.23"
                 instanceName="w.machine001(1)-icon1"
                 name="lift"  />
```
Potentially may need a “force” attribute to force a refresh in certain situations. For example: walking from point A to point B in zero time.

AnimationStop Command

The AnimationStop command is a WITNESS command issued to stop an animation.

```
<animationStop time="255.0"
               instanceName="w.machine001(1)-icon1"
               name="lift" />
```

Feature Command

The Feature command is a WITNESS command to allow enabling / disabling of features used to the 3D server. The following features are restricted to Quick 3D Pro versions:

- Stereo
- Multiple display support (F11)
- Space mouse

For example:

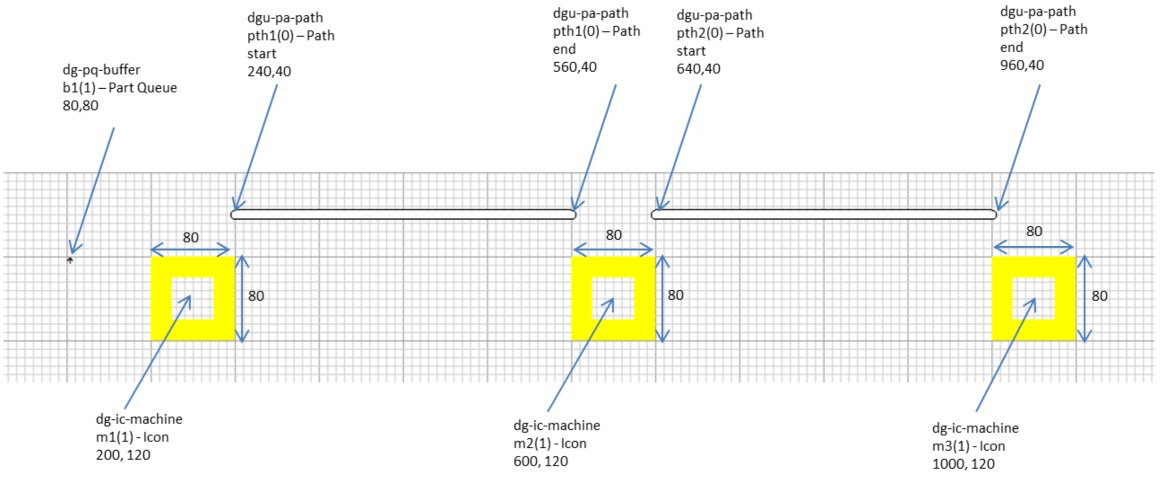
```
<feature name="stereo" enabled="true" />
```

Element/Attribute	Description
name	The name of the feature to enable / disable. The possible value are: • stereo • multipleDisplays • spaceMouse
enabled	A flag that determines if the specified feature name should be enabled or disabled.

If the Feature command is not sent from WITNESS, these features should be enabled (default).

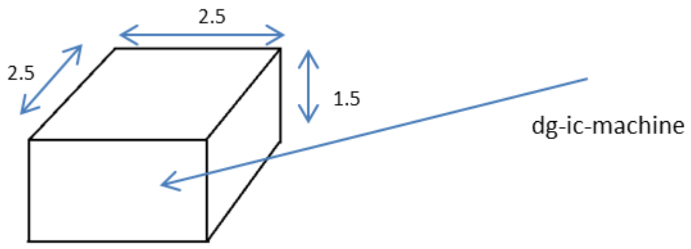
Appendix

Update Command Behavior



This is the 2D layout note all the machines are 80x80 in size and exists in a similar coordinate system.

The 3D shape we are using for dg-ic-machine is along the lines of:



This means the footprint of the loaded geometry is X x Z of 2.5 x 2.5, we require 80 x 80 to match X x Y footprint, so we scale X by 32 and Z by 32. Our system allows the user to either give a specified height (converted to scale of the geometry 3D-Y), or we have "auto-scaling" of the height by average of the width(3D-X) and depth (3D-Z) scales, in the above case we use auto-scaling and the average of 32.0, 32.0 = 32.0.

So:

3D-X = 2.5 * 32.0 = 80.0

3D-Y = 1.5 * 32.0 = 48.0 (Auto scaled)

3D-Z = 2.5 * 32.0 = 80.0

Position in our current system is then done using a translation specified in world coordinates as:

3D-X = 200.0

3D-Y = 24.0 (half the geometry height)

3D-Z = 120.0

Taking the example above, machine at (200.0, 120) and geometry of dimensions (2.5, 1.5, 2.5), this produces:

```
<load geometry="dg-ic-Machine" />
<create time="0" geometry="dg-ic-machine" instanceName="m1(1) - Icon" />
<update time="0" instanceName="m1(1) - Icon">
  <scale x="32" y="32" z="32" />
  <rotate order="xyz" x="0" y="180" z="0" />
  <translate x="200" y="24" z="-120" />
</update>
```

1. *<load geometry="dg-ic-Machine" />* – This will try to load the correct geometry "dg-ic-machine) which will respond in our case with a size of (2.5, 1.5, 2.5)

2. *<create time="0" geometry="dg-ic-machine" instanceName="m1(1) - Icon" />* - This uses the loaded geometry to create a visible instance called *"m1(1) – Icon"*

3. *<update time="0" instanceName="m1(1) - Icon">* - We are now updating instance called "m1(1) – Icon" at time 0.0

4. *<scale x="32" y="32" z="32" />* - We are scaling the instance to get it the desired X-Z footprint to match our X-Y footprint.

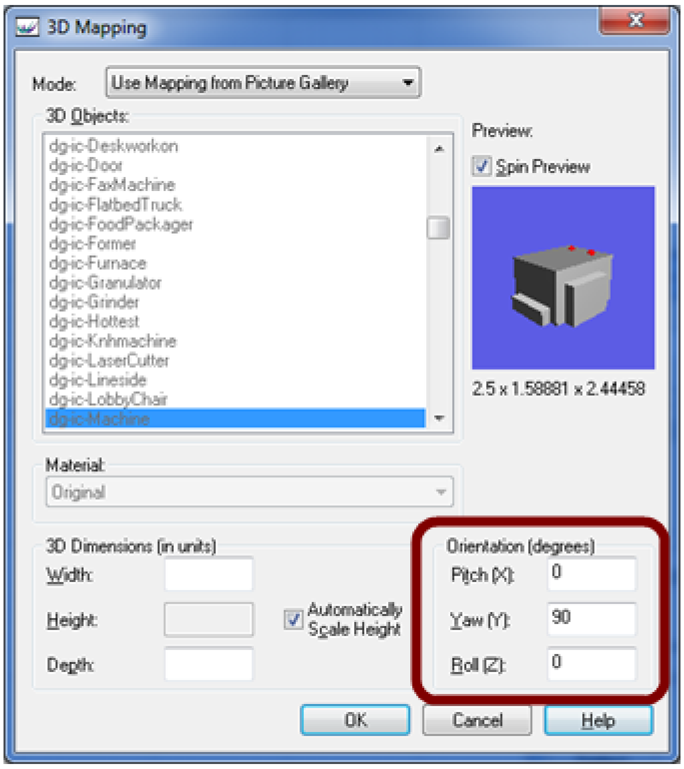
5. *<rotate order="xyz" x="0" y="180" z="0" />* - We perform any rotation to make sure the shape is pointing in the direction the simulation thinks it should be, our old system has a 180 yaw, this is probably not be required now. However this step is still required when we have icons place in 2-D with a rotation on the 2D surface.

6. *<translate x="200" y="24" z="-120" />* - This is the world coordinate position of the finished article.

7. *</update>* - This marks the end of the update block.

Rotate Command

During an update command there could be 2 rotation commands issued. The first command handles the roll, pitch and yaw overrides specified on the 3D Mapping dialog. The roll, pitch and yaw values allow the user to re-orientate the geometry that is in the standard object library without having to edit the object directly. This rotate command will only be issued if either the roll, pitch or yaw values is non-zero.


3D Mapping

Mode: Use Mapping from Picture Gallery

3D Objects:
dg-ic-Deskworkon
dg-ic-Door
dg-ic-FaxMachine
dg-ic-FlatbedTruck
dg-ic-FoodPackager
dg-ic-Former
dg-ic-Furnace
dg-ic-Granulator
dg-ic-Grinder
dg-ic-Hottest
dg-ic-Knhmachine
dg-ic-LaserCutter
dg-ic-Lineside
dg-ic-LobbyChair
dg-ic-Machine

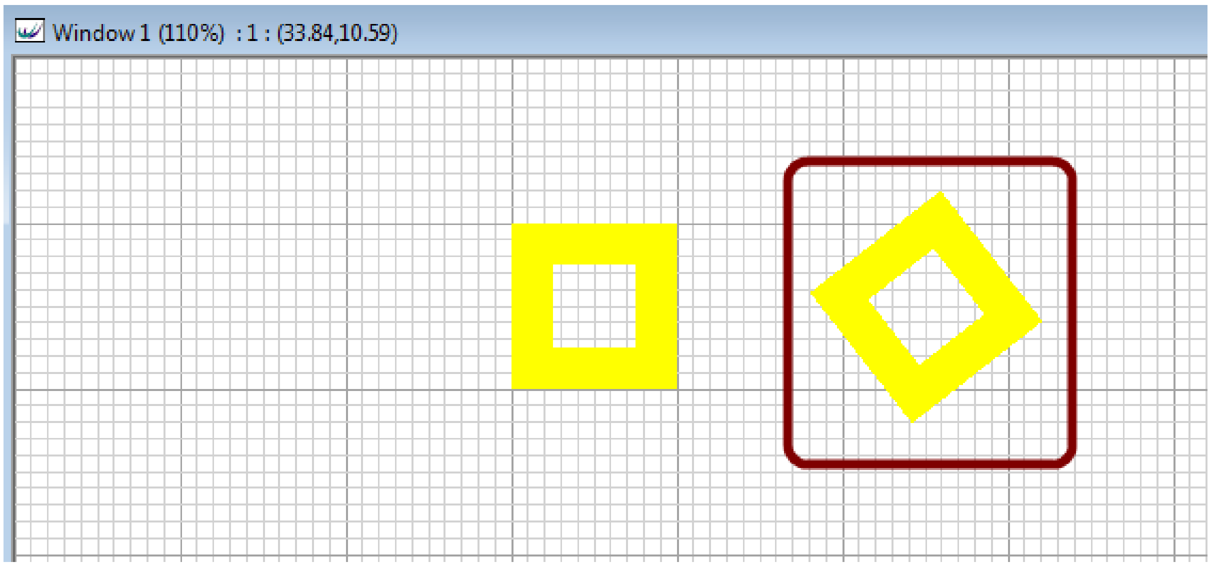
Preview:
☑ Spin Preview
2.5 x 1.58881 x 2.44458

Material: Original

3D Dimensions (in units)
Width:
Height:
Depth:
☑ Automatically Scale Height

Orientation (degrees)
Pitch (X): 0
Yaw (Y): 90
Roll (Z): 0

The second rotate command is issued when the 2D icon has been rotated by the user.



This rotate command has a default value of 180 for the yaw and then any rotation by the user of the 2D shape is added to it.