**LPT**

Laser Projection Technologies, Inc.

8 Delta Drive, Londonderry, NH, USA 03053
Phone: (603) 421-0209, FAX: (603) 421-0072
Website: www.lptcorp.com

RayTracer

Remote API

Manual


```
Command Prompt - java -jar api.jar                                    _ □ ×

C:\mike>java -jar api.jar
******************* Main Menu ***************************
* 1 Connect to socket         11 Pause Projecting         *
* 2 Disconnect from socket    12 Cont. Projecting          *
* 3 Login to Raytracer        13 Next Layer                *
* 4 Set Job                   14 Previous Layer            *
* 5 Set Tool                  15 Get Operator Version      *
* 6 Set Part                  16 Get Client Version        *
* 7 Set LaserGroup            17 Set TimeOut               *
* 8 Set Layers                18 Get Orientation Info      *
* 9 Auto align                19 Set Orientation Info      *
*10 Project                                                *
******************************************************
Enter command: 1
Host: 10.0.0.63
Port: 10101
Connected with server tlp001.lptcorp.com/10.0.0.63:10101
Successfully retrieved the OperatorAPI instance
Successfully added the AsyncListener
******************* Main Menu ***************************
* 1 Connect to socket         11 Pause Projecting         *
* 2 Disconnect from socket    12 Cont. Projecting          *
* 3 Login to Raytracer        13 Next Layer                *
* 4 Set Job                   14 Previous Layer            *
* 5 Set Tool                  15 Get Operator Version      *
* 6 Set Part                  16 Get Client Version        *
* 7 Set LaserGroup            17 Set TimeOut               *
* 8 Set Layers                18 Get Orientation Info      *
* 9 Auto align                19 Set Orientation Info      *
*10 Project                                                *
******************************************************
Enter command:
```

RayTracer Remote API Manual Rev. 2

RayTracer API Setup

Any API-enabled software must be run in conjunction with the RayTracer Operator program. The API controls RayTracer Operator; the latter controls the projector. API cannot control the projector directly.

Hence, the user must launch the RayTracer Operator program when using an API. A TCP socket connection must be established between the API program and RayTracer Operator. Usually, an IP address and a Port number are required to establish such a connection. In order to ascertain the connection has been established, the user should left-click the mouse on the **Options** dropdown menu on the RayTracer Operator interface and choose **Socket Connection**. The following window should pop up:
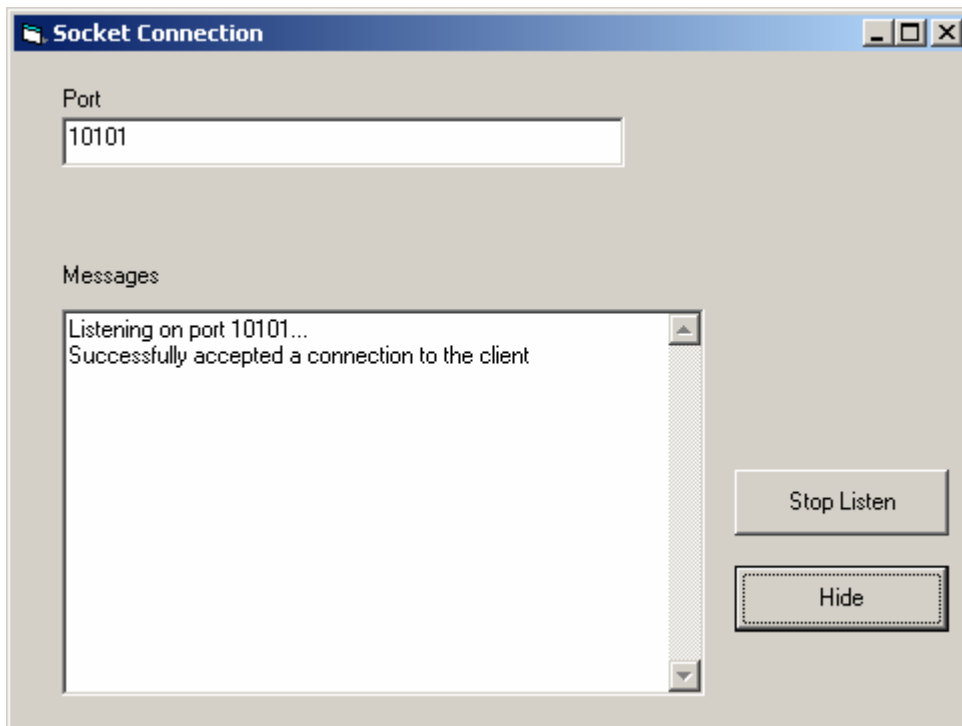
Port 10101 is the default, but it can easily be changed by entering another number. Clicking on **Hide** will close the window properly.

Note also that the projector has to be manually aligned with respect to the tool using the RayTracer Operator interface before any operations can be undertaking using the API program. Please perform **Manual Alignment** prior to working with the projector.

8 Delta Drive, Londonderry, NH, USA 03053
Phone: (603) 421-0209, FAX: (603) 421-0072
Website: www.lptcorp.com

Laser Projection Technologies, Inc.

API Commands

Once a TCP socket connection is established, text commands can be sent to RayTracer Operator, thus enabling the projector to be run remotely.

Here is an example of an LPT-designed API program:

Each of the text commands is sent via a socket to RayTracer Operator. In most cases these commands are the equivalent of pressing the associated controls on Operator's user interface. Thus, sending the "NextLayer" command is equivalent to pressing the "Next Layer" button on the operator screen).

Each command is preceded by ">" and followed by '\r\n', where "\r" and "\n" are control characters of 'return' and 'newline' respectively. For example: ">NextLayer\r\n"

Any status or error messages that are the result of a command will be returned on the socket. These will take the form of ASCII messages (the same messages that would be in the pop-up message boxes were Operator to be run manually).

8 Delta Drive, Londonderry, NH, USA 03053
Phone: (603) 421-0209, FAX: (603) 421-0072
Website: www.lptcorp.com

Laser Projection Technologies, Inc.

The following is a list of commands.

GetOperatorVersion: Returns the version of RayTracer Operator being run. This command has no parameters.

Login: Changes or reloads the database or user for the Operator.

Syntax: Login,<user>,<password>,<database>
Example: "Login,Admin,Admin,D:\databases\LPT_DEMO_CDS"
     This would set the current database to D:\databases\LPT_DEMO_CDS and log in as the
     default "Admin" user (with a password of "Admin")

GetOperatorDBInfo: Returns the Users.xml path and the current job

SetJob: Changes or reloads the job for the Operator. The specified job must exist within the current database.

Syntax: SetJob,<job>
Example: "SetJob,E_SIZE_CDS"

SetTool: Changes or reloads the tool for the job. The specified tool must exist within the current database and be included in the current job.

Syntax: SetTool,<tool>
Example: "SetTool,E_SIZE_CDS_tool"

SetPart: Changes or reloads parts. The specified part must exist within the current database and be included in the current job.

Syntax: SetPart,<part>
Example: "SetPart,E_SIZE_CDS_part"

SetLaserGroup: Changes or reloads the current LaserGroup. The specified laser group must exist within the current database, and be listed in the current job. If any of the projectors in the laser group are not available, a message to that effect will be sent back.

Syntax: SetLaserGroup,<laserGroup>
Example: "SetLaserGroup,LPT50233"

SetLayers: will set the layers within the part that Operator is using. These layers must exist within the current part.

Syntax: SetLayers,<comma-separated layer list>
Example: "SetLayers,Circle,Diamond,Square"

StartAutoAlignment: Initiates automatic alignment of the projector. It assumed that the projector is connected and that job, tool and laser group are all valid. Any errors or problems with the alignment (e.g. not sufficient accuracy) will be returned as warning messages. This command has no parameters.

RayTracer Remote API Manual Rev. 2

StartProjection: Initiates the projection of the currently selected layer(s), in the current part, projected by the projectors in the current laser group. This command has no parameters.

PauseProjection: Pauses the current projection. This command has no parameters.

ContinueProjection: Restarts the projection previously paused. This command has no parameters.

NextLayer: Initiates projection of the next layer in the part. If the current layer is the last layer listed, no action will be taken. This command has no parameters.

PreviousLayer: Initiates projection of the previous layer in the part. If the current layer is the first layer listed, no action will be taken. This command has no parameters.

GetLaserCount: Returns the number of projectors in the current LaserGroup. This command has no parameters.

GetLaserNameType: Returns the type and name of the specified projector number. The number is a reference to the position of the projector in the projector list (1 = first, 2 = second, etc).

Syntax: GetLaserNameType,<projector #>
Example:GetLaserNameType,1

SetCrossField: Turns the center field cross for the specified projector on or off (1=on, 0=off)

Syntax: SetCrossField,<projector #>,<1/0>
Example: SetCrossField,1,1

SetFieldOfView: Turns the field of view (FOV) for the specified projector on or off (1=on, 0=off)

Syntax: SetFieldOfView,<projector #>,<1/0>
Example: SetFieldOfView,1,1

8 Delta Drive, Londonderry, NH, USA 03053
Phone: (603) 421-0209, FAX: (603) 421-0072
Website: www.lptcorp.com

LPT

Laser Projection Technologies, Inc.

System Orientation

The last two commands involve getting/setting the projector's position and orientation with respect to the tool. The position is the [XYZ] location of the projector's point of origin within the tool's coordinate system. The orientation is a [3x3] matrix describing the projector's X,Y and Z axes within the tool's coordinate system. It can also be thought of as a rotation transform between the tool and projector coordinate systems, just as the position can be thought of as a linear translation between the two.

The XYZ position of the projector is given as a position of its origin with respect to the tool coordinate frame. The origin of the projector's coordinate system coincides with the center of the Y-Galvanometer mirror. The projector's orientation matrix is defined as follows:

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \begin{bmatrix} i_{XT} & j_{XT} & k_{XT} \\ i_{YT} & j_{YT} & k_{YT} \\ i_{ZT} & j_{ZT} & k_{ZT} \end{bmatrix} \tag{1}$$

Where: $i_{XT},\ i_{YT},\ i_{PZT},$ are directional cosines of the projector's frame basis vector *i*
(projections of the vector *i* onto the tool's coordinate system axes $X_T,\ Y_T,\ Z_T$).

$j_{XT},\ j_{YT},\ j_{ZT},$ are directional cosines of the projector's frame basis vector *j*
(projections of the vector *j* onto the tool's coordinate system axes $X_T,\ Y_T,\ Z_T$).

$k_{XT},\ k_{YT},\ k_{ZT},$ are directional cosines of the projector's frame basis vector *k*
(projections of the vector *k* onto the tool's coordinate system axes $X_T,\ Y_T,\ Z_T$).

GetSystemOrientation: Returns four lines of comma-separated text. Line 1 is the XYZ position of the specified projector, whereas lines 2, 3, and 4 are its 3x3 orientation matrix.

GetSystemOrientation,<projector #>
Example: "GetSystemOrientation,1" would get the orientation of the 1st projector in the Laser Group.

SetSystemOrientation: Allows user to specify the position and orientation of the projector.

SetSystemOrientation,<projector #>, <X>,<Y>,<Z>,<comma-separated 3x3 orientation matrix>
Example: "SetSystemOrientation,1,0,0,180,1,0,0,0,1,0,0,0,1" would set the system orientation of the 1st projector in the laser group to 15 ft in Z from tool's origin, oriented parallel to the tool's X-Y plane, with the projector right-side up since the orientation matrix is the identity matrix in this case.

Creating XML Files

Using a new tool, or displaying a new projection using the remote API and RayTracer Operator is done by adding or modifying the XML files in the database that Operator is currently attached to.
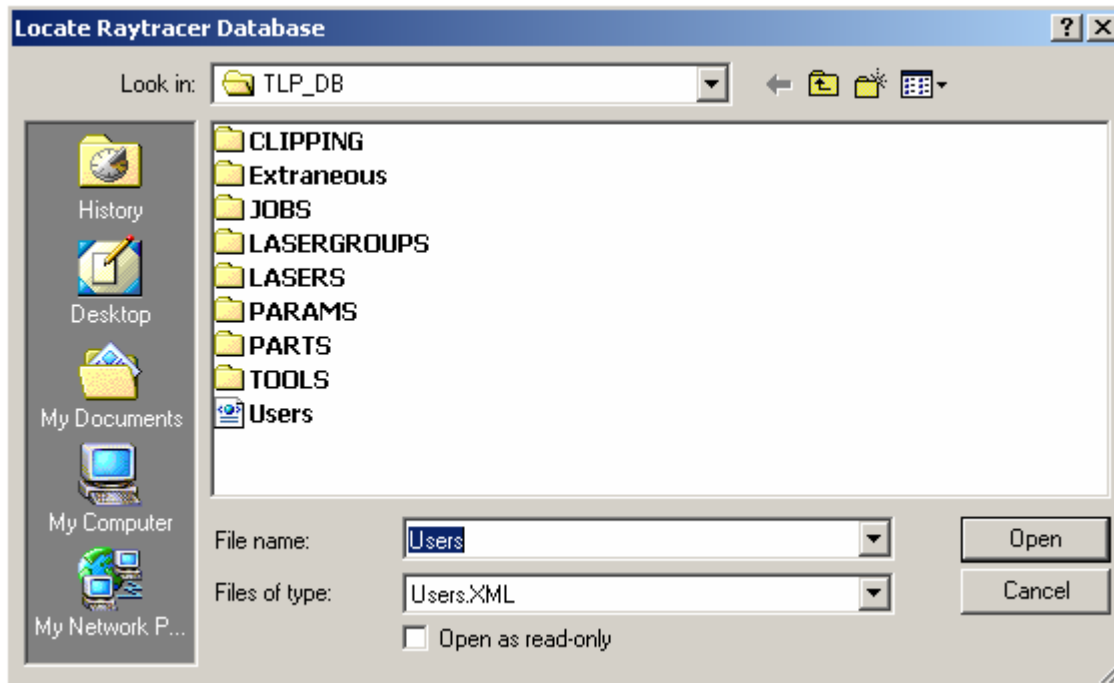
A RayTracer database consists of a **USERS.XML** file and directories containing the XML files for the various objects in the database. Each directory has an index file containing a list of all the files in that directory. This index file is used both for its contents (a list of XML files) and its timestamp. The timestamp is used by Operator to determine whether it needs to download changes from the database.

In order for Operator to receive new data, two procedures must be performed:

- New data has to be added to the database, which involves either creating a new XML file or modifying an existing one.

- Operator has to download the new part, tool, job, etc. from the database, which involves changing the timestamp of the appropriate index file (and its contents, in case a file has been added or removed), and sending a command to Operator to reload the appropriate list. For example, if a part were modified, the **PARTSINDEX.XML** file would need its timestamp updated and a *SetPart* command would have to be sent to Operator in order for the part to be reloaded with the new data.

The Job XML file is the container object for the rest of the associated data. It contains references to parts, tools, parameters, laser groups, etc. Whenever any of these objects are added or removed, the Job file needs to be updated to reflect the changes, and the **JOBINDEX.XML** file will need to be

RayTracer Remote API Manual Rev. 2

touched so that it has a new timestamp, thereby showing Operator that the new copy must be loaded when the *SetJob* command is sent.

Sample Job File

```
- <XML>
  - <JOB NAME="sample_job" DBVERSION="3.9" MODIFIED="2008/08/19
      14:53:11" PARAMETER="Eng_Default" TKR_NUMBER_REQ="0"
      DISPLAY_TOOL_CODE="0" DISPLAY_DATE_TIME="0"
      PROJECTAFTERALIGNMENT="0" DISPLAY_MEDIA_STATUS="0"
      DISPLAY_TOOL_UNIT_NUMBER="0" REQUESTOFFSET="0"
      OFFSET_FILE_PATH="" SHOWINSTRUCTIONS="0" DONTPEEK="0"
      REMEMBER_ALIGNMENT="0" ALIGNMENT_EXPIRATION=""
      SEQUENTIALPROJECTION="0" SEMISEQUENTIALPROJECTION="0"
      ZOOMONNEXT="0" VIRTUAL_TOOL_ALIGNMENT="0" PAUSE_DETAIL="0"
      OPERATOR_TYPE_REASON="0" OPERATORMAYSHUTDOWN="0"
      SLEEPYTIME="" CHANGE_LASER="1" CHANGE_PART="1"
      CHANGE_TOOL="1">
    - <LASER_GROUPS>
        <LASERGROUP NAME="sample_lasergroup_50233" />
      </LASER_GROUPS>
    - <TOOLS>
        <TOOL NAME="sample_tool_cds" />
      </TOOLS>
    - <PARTS>
        <PART NAME="sample_part_cds" />
      </PARTS>
    </JOB>
  </XML>
```

The most important entries for the use of an application sending remote commands to RayTracer Operator are the following:

* JOB NAME="**sample_job**" – which specifies the name of this job, which must be the same as the filename (sample_job.xml)
* PARAMETER="**Eng_Default**" – which specifies Eng_Default.xml as the parameter set being used.
* <LASERGROUP NAME="**sample_lasergroup_50233**" /> – which specifies sample_lasergroup_50233.xml as the lasergroup being used.
* <TOOL NAME="**sample_tool_cds**" /> – which specifies sample_tool_cds.xml as the tool being used.
* <PART NAME="**sample_part_cds**" /> – which specifies sample_part_cds.xml as the part being used.

The job file can contain more than one entry for lasergroup, tool and part.

Sample LaserGroup File

```
- <XML>
  - <LASERGROUP NAME="sample_lasergroup_50233">
      <LASERGROUPITEM LASERNAME="lpt50233" />
    </LASERGROUP>
  </XML>
```

This, in turn, references the **Laser** file lpt50233.xml which is the specification of the projector used. That file is as follows:

```
- <XML>
    <LASER NAME="lpt50233" />
  </XML>
```

Sample Tool File

```
- <XML>
  - <TOOL DBVERSION="3.9" MODIFIED="2008/08/19 16:36:14"
      NAME="sample_tool_cds" COORDSYSTEM="RIGHT" STATUS=""
      TOOLCODE="">
    <TARGET LASERNUMBER="1" NAME="T 1" DVAL=""
      XVAL="1.000000" YVAL="1.000000" ZVAL="0.000000"
      NUMBER="1" SEARCHRATIO="1.0" />
    <TARGET LASERNUMBER="1" NUMBER="2" NAME="T 2" DVAL=""
      XVAL="1.000000" YVAL="33.000000" ZVAL="0.000000"
      SEARCHRATIO="1.0" />
    <TARGET LASERNUMBER="1" NUMBER="3" NAME="T 3" DVAL=""
      XVAL="43.000000" YVAL="33.000000" ZVAL="0.000000"
      SEARCHRATIO="1.0" />
    <TARGET LASERNUMBER="1" NUMBER="4" NAME="T 4" DVAL=""
      XVAL="43.000000" YVAL="1.000000" ZVAL="0.000000"
      SEARCHRATIO="1.0" />
    <TARGET LASERNUMBER="1" NUMBER="5" NAME="T 5" DVAL=""
      XVAL="21.000000" YVAL="1.000000" ZVAL="0.000000"
      SEARCHRATIO="1.0" />
    <TARGET LASERNUMBER="1" NUMBER="6" NAME="T 6" DVAL=""
      XVAL="1.000000" YVAL="19.000000" ZVAL="0.000000"
      SEARCHRATIO="1.0" />
    <TARGET LASERNUMBER="1" NUMBER="7" NAME="T 7" DVAL=""
      XVAL="23.000000" YVAL="33.000000" ZVAL="0.000000"
      SEARCHRATIO="1.0" />
    <TARGET LASERNUMBER="1" NUMBER="8" NAME="T 8" DVAL=""
      XVAL="43.000000" YVAL="17.000000" ZVAL="0.000000"
      SEARCHRATIO="1.0" />
  </TOOL>
</XML>
```

This file lists all 8 targets used in the Circle/Diamond/Square tool included with LPT projectors.

- [Tool] NAME – the name of this tool, which must be the same as the xml file name.
- LASERNUMBER – the "in-group" number of the laser which is to measure this target. The only time the value is other than "1" is when a multiple-projector setup (clipping) is involved. Then each target is assigned the number of the projector that will scan it.
- NUMBER – These numbers must start at 1 and be sequential. They can be thought of as a unique identifier for the target.
- [Target] NAME – the target name is often useful for user to know during manual alignment. The names need not be specified or be unique, although both are recommended.
- XVAL, YVAL, ZVAL are the X,Y,Z coordinates of the target. DVAL is used to designate the target as a specific drift-check candidate. If none are specified, Operator automatically chooses which to use when drift-check is enabled.

RayTracer Remote API Manual Rev. 2

Sample Part File

```
- <XML>
  - <PART NAME="sample_small_part" STATUS="" DBVERSION="3.9"
      FILEDATE="8/19/2008 4:44:38 PM" INVERTNORMALS=""
      HIGHESTPLY="4" MODIFIED="2008/08/19 16:44:38" TopLayers="2"
      DEFAULTLASER="0" USEASSIGNEDLASERS="">
    - <PLY NAME="L1-Diamond">
        <POINT NUMBER="1" NAME="" XVAL="22" YVAL="32" ZVAL="0"
          IVAL="0" JVAL="0" KVAL="0" />
        <POINT NUMBER="2" NAME="" XVAL="37" YVAL="17" ZVAL="0"
          IVAL="0" JVAL="0" KVAL="0" />
        <POINT NUMBER="3" NAME="" XVAL="22" YVAL="2" ZVAL="0"
          IVAL="0" JVAL="0" KVAL="0" />
        <POINT NUMBER="4" NAME="" XVAL="7" YVAL="17" ZVAL="0"
          IVAL="0" JVAL="0" KVAL="0" />
        <POINT NUMBER="5" NAME="" XVAL="22" YVAL="32" ZVAL="0"
          IVAL="0" JVAL="0" KVAL="0" />
        <INSTRUCTIONS />
        <NOTES />
      </PLY>
    - <PLY NAME="L2-X">
        <INSTRUCTIONS />
        <NOTES />
      - <PLY NAME="L3-/">
          <POINT NUMBER="1" NAME="" XVAL="7" YVAL="2"
            ZVAL="0" IVAL="0" JVAL="0" KVAL="0" />
          <POINT NUMBER="2" NAME="" XVAL="37" YVAL="32"
            ZVAL="0" IVAL="0" JVAL="0" KVAL="0" />
          <INSTRUCTIONS />
          <NOTES />
        </PLY>
      - <PLY NAME="L4-\">
          <POINT NUMBER="1" NAME="" XVAL="7" YVAL="32"
            ZVAL="0" IVAL="0" JVAL="0" KVAL="0" />
          <POINT NUMBER="2" NAME="" XVAL="37" YVAL="2"
            ZVAL="0" IVAL="0" JVAL="0" KVAL="0" />
          <INSTRUCTIONS />
          <NOTES />
        </PLY>
      </PLY>
    </PART>
  </XML>
```

This part file has two top-level plies, one of which is "X", composed of two more lower-level plies. Part files are potentially the most complicated of all LPT database XML files, since they can be hierarchical, with plies containing other plies.

The main entries for the use of an application sending remote commands to Raytracer Operator are the following:

- PART tag:
  - NAME – the name of the part; must be the same as the XML file name.
  - HIGHESTPLY – total number of plies of all levels within the part.
  - TopLayers – total number of top-level plies within the part

- PLY tag:
  - NAME – the name of the ply. It **must** start with "L-#" where # is a unique number of this ply. These numbers do not have to be in a specific order within the part, but they must start at 1 and there may not be any numerical gaps. The highest number has to correspond to the "HIGHESTPLY" value in the PART tag. Operator and Administrator do not display the "L-#" part of the name. It is suggested that the application sending commands to Operator follow their example.

- POINT tag:
  - NUMBER – specifies a unique number **within the ply** and must be sequential, starting at 1.
  - XVAL, YVAL, ZVAL – X,Y,Z coordinates of the CAD point.
  - IVAL, JVAL, KVAL – the I,J,K surface normal vector. If there is no normal vector, all these should be set to "0".

In each case, the correctness of these files can be verified by placing them into a database file structure and accessing them via RayTracer Administrator. Selecting File->Database Cleanup will also check the structure of each file. The final test, of course, is to use them in RayTracer Operator to align to a tool and project.

Measurement with the LPT10 & LPT12

The measurement commands cannot directly supply XYZ coordinates of the measured feature because the projector only has the mirror angles that point to the measured position. Instead, the calling program is supplied with a description of a **RAY**. This ray consists of an X,Y,Z position, in tool coordinates, that corresponds to the 'start' of the laser beam, and a normalized I,J,K directional vector, also in tool coordinates, that corresponds to the direction of the beam towards the measured position.

This is necessary because the projector has no internal information about the shape of the object being projected upon (the 'tool'). It is up to the CAD system making these remote API measurement calls to trace the ray described by **[X, Y, Z, I, J, K]** until it intersects with the surface of the object being measured. The procedure is to first align to the tool, then issue a measure command. For some features two or more rays are needed to describe the feature (e.g. two are needed to describe an edge feature).

Sometimes the scan may contain more than one of the features being looked for. In this case an **ASSIST POINT** must be specified. This is an [x, y] pixel coordinate (in the scan image itself) that identifies which of the possible features is the one of interest. It doesn't have to be exactly on the feature, just closer to the feature of interest than any of the other features in the scan. There is a 'reprocess feature' command so that an assist point can be specified without having to rescan the region.

The currently supported features are:

- **Intersection:** This is the intersection of two linear edges of a region that is optically contrasted (lighter or darker) to the rest of the scan. It is defined by a single point.
- **Dot:** This is a circular or elliptical region that is optically contrasted to the rest of the scan. It is defined by three points: a center point and two points on the edge. The edge points correspond to the major and minor axes of the ellipse. The circular region appears as an ellipse to the projector because it will usually be at some angle to the projector.
- **Edge:** This is the linear edge between two optically contrasted regions. It is defined by two points along the edge.
- **Raw Contour:** This is the non-linear (free-form) edge of an optically contrasted region. It is determined by the strongest string of edge pixels that are seen in the scan. The contour itself is defined by a ray for each pixel in the scan that comprises the edge.

API Measurement Commands:

MoveToXYZ: Position a laser cursor (a '+' shape) at a specified [X,Y,Z] coordinate. Note that there is no projector number associated with this command. This is because all projectors in the current laser group will project a cursor to the same point when this command is sent.

Syntax: MoveToXYZ,<X>,<Y>,<Z>,<flag, 1=start, 4=stop>

Example: MoveToXYZ,1.0,2.0,3.0,1
　　　　This would start the display of the 'plus' cursor at [1,2,3].

Example: MoveToXYZ,2.0,3.0,4.0,0
　　　　This would move the already-visible cursor [2,3,4]

Example: MoveToXYZ,2.0,3.0,4.0,4
This would stop display of the cursor – the [XYZ] doesn't really matter here.

ScanFeature: This command takes two XYZ coordinates which describe the opposite corners of a rectangle that denotes the area of interest. Depending on flags supplied with the command, this rectangle (the 'scan-area' or 'scan-box') either the outline of the rectangle will be projected, to show the area of interest (the 'scan-box' or 'scan-cursor'), or the entire area will be scanned, and the results processed to find the specified feature.

The way it is used is that the first time it is called (with a 'start' flag) it will start displaying the scan-box. To move or re-size the scan-box, successive calls are made with new coordinates. To turn off the scan box a call is made with a 'stop' flag. At some point the user has the scan-box positioned and sized to their liking, and another 'ScanFeature' command is sent with the 'scan' flag set, along with information about what feature is being looked for.

Syntax: "ScanFeature, <projector#>,<feature type>,<flags>,<# scan areas>,PNT,<XYZ coord1>,<XYZ coord2>,<assist info>,EOP"

<projector#>: index of the projector in the laser group (usually 1, unless you're using a multi-projector setup)

<feature type>: Note, that the feature type doesn't matter unless a scan is actually being done.
   0: retro target
   1: intersection (intersection of two linear edges)
   2: dot (planar circle)
   3: edge (linear border between two areas)
   4: raw contour (free-form contour between two areas)
   5: line (thin line -- not currently implemented)

<flags> ('do a scan' can be OR ed with start or stop)
   0x0000 - no flags set -- show scan box only.
   0x1000 - Do a scan and look for the specified feature
   0x2000 - First call -- start displaying scan box
   0x4000 - Final call -- stop displaying scan box

<# scan areas> Not currently implemented. Should always be set to 1.

<XYZ coord1> One corner of the scan box in tool (object being projected upon) coordinates

<XYZ coord2> Opposite corner of the scan box in tool coordinates

<assist info> The assist info values are extra, optional information that helps the feature recognition algorithms find the correct feature. Specifying zeros ('0.0') will cause defaults to be used. There are (currently) two values to be specified.

The first is an optional 'expected' angle (in degrees) used for **Intersection** features. If zero is specified, a default of 90 is used. This value is currently ignored by other features, although it is intended that **Dot** features will be able to take expected size and shape values at some future time.

The second value is actually an [x,y] assist point in image coordinates. The 'X' coordinate is in the high-order word (most significant 16 bits) and the 'Y' is in the low order word. This coordinate is useful when there is more than one possible feature to be found (such as several holes in close proximity, or the front and back edge of a corner). This coordinate is used only as a guide, so it doesn't need to point exactly to the desired feature, just closer to that feature than any others. If either X or Y is zero the assist point will be ignored. NOTE: that the image coordinates start at [1,1] at the top, left of the image.

Example: ScanFeature, 1, 2, 8192, 1,PNT,1.0,1.0,3.0,4.0,5.0,6.0,0.0,1.0,EOP
        This will start displaying the scan box with [1,2,3] as [3.4.5] as its corners
.
Example: ScanFeature, 1, 2, 0, 1,PNT,1.1,2.2,3.3,4.1,5.2,6.3,0.0,0.0,1.000,EOP
        This will move the scan box to a new position, defined by the object coordinates [1.2, 2.2, 3.3],[4.1, 5.2, 6.3].

Example: "ScanFeature, 1, 2, 4096, 1,PNT,1.1,2.2,3.3,4.1,5.2,6.3,0.0,1.0,EOP
        This will scan in the specified region and process the results, looking for a 'dot' feature. If successful, the result will be 3 [XYZ,IJK] rays being sent back. These rays will point to the center and two points on the circumference of the ellipse found (major and minor axes).

Response:

The ScanFeature command has two forms of response, depending on whether it is manipulating the scan-box (on, off, repositioning, resizing), or performing an actual scan.

If a scan (and feature recognition processing) isn't being done, the response will take the following form:
>RET:<proj #>,DoScanBoxXYZ, 1,OK
where the only change would be the projector number (typically '1').

If a scan is being performed, the response will consist of a header, a list of rays (points on the feature), and an /EOR to tag the end of the response.
>RET:<status>,ScanFeature, <proj #>, <feature type>, <confidence>, <# rays>, No File
<list of rays, one per line>
/EOR

where:
<status> 0 = failure
        1 = success of some command that doesn't have a result
        2 = success of a scan with some number of points as a result
<proj #> the ID number of the projector (position in the laser group)
<feature type> as sent in: 1=intersection, 2=dot, 3=edge, 4=raw contour
<confidence> percent certainty of the result -- not implemented, always 100.
<# rays> the number of lines of data following
<list of rays> each ray consists of 6 comma-separated floating pt numbers: X,Y,Z,I,J,K
/EOR = End Of Response -- a tag put at the end showing that there's no more data.

RayTracer Remote API Manual Rev. 2

**ReprocessFeature**: This command will use the previous scan data and reprocess it, using the specified information. This is particularly useful when the first scan found the wrong feature and an assist point has to be specified.

Syntax: "ReprocessFeature, <projector #>,<flag>,<x,y>"

<projector#>: index of the projector in the laser group (usually 1, unless you're using a multi-projector setup)

<flag>: not currently implemented. Use "0"

<x,y>: assist point in image coordinates (pixels). This coordinate is useful when there is more than one possible feature to be found (such as several holes in close proximity, or the front and back edge of a corner). This coordinate is used only as a guide, so it doesn't need to point exactly to the desired feature, just closer to that feature than any others. If either X or Y is zero, the assist point will be ignored. NOTE: that the image coordinates start at [1,1] at the top, left of the image

Example: ReprocessFeature, 1, 0, 54, 46
        This will result in the previous scan data being processed using [54,46] as an assist point to identify the feature.

The response for this command is the same as the response for ScanFeature, except that it would contain the term "ReprocessFeature" instead of "ScanFeature" after the status.