**IVREAD   
3D Graphics File Conversion**

**IVREAD** is a FORTRAN90 program which reads a 3D graphics file and converts the information to another format.

**IVREAD** was originally written to read SGI Inventor 3D graphics files and convert them to Digistar VLA format. Since then, the program has evolved to allow the input and output of:

* ASE - [AutoCAD](http://www.autodesk.com/) ASCII export files;
* BYU - [Movie.BYU surface geometry files](https://people.sc.fsu.edu/~jburkardt/data/byu/byu.html)(output only);
* DXF - [AutoCAD](http://www.autodesk.com/) DXF files;
* HRC - [SoftImage](http://www.softimage.com/) hierarchy files;
* IV - [SGI Inventor](http://www.sgi.com/) files;
* OBJ - a file format from ALIAS ( <http://www.alias.com/eng/index.shtml>)
* OFF - Object file format for [GeomView](http://www.geomview.org/)
* OOGL - CMESH files from [GeomView](http://www.geomview.org/) (input only);
* POV - [Persistence of Vision](http://www.povray.org/) files (output only);
* SMF - Michael Garland's format for [QSLIM](http://www.cs.cmu.edu/afs/cs/user/garland/www/quadrics/qslim.html);
* STL - [ASCII Stereolithography files](https://people.sc.fsu.edu/~jburkardt/data/stla/stla.html);
* TEC - [TECPLOT](http://www.amtec.com/) files (output only);
* TS - Mathematica [ThreeScript](http://documents.wolfram.com/v4/AddOns/Grph_ThreeScript-.html);
* TXT - text files (output only);
* TRI/TRIA - [a simple ASCII triangle format](https://people.sc.fsu.edu/~jburkardt/data/tri/tri.html) requested by Greg Hood;
* UCD - [Advanced Visual Systems (AVS)](http://www.avs.com/) Unstructured Cell Data (output only);
* VLA - [Digistar VLA files;](https://people.sc.fsu.edu/~jburkardt/data/vla/vla.html)
* WRL - VRML (Virtual Reality Modeling Language) files (output only).
* XYZ - [XYZ files](https://people.sc.fsu.edu/~jburkardt/data/xyz/xyz.html) (inefficient but simple, point and line format);

There is also a simple-minded facility to project any 3D image onto a 2D plane, and make a [PostScript image](https://people.sc.fsu.edu/~jburkardt/data/ps/ps.html) of the result.

Graphics file formats are often poorly documented; hence this program was written primarily by examining typical files and struggling to interpret them. Particularly in the case of Inventor and DXF files, the program is not able to handle all the subtleties of the file format, and may mangle data, or even crash.

**Usage:**

**ivread** *input.ext* *output.ext*

converts *input.ext* to *output.ext*, where the input and output formats are determined by the file name extensions.

**ivread**

if no files are specified, an interactive dialog is set up, and the user can specify the files to be converted.

**Licensing:**

The computer code and data files described and made available on this web page are distributed under [the GNU LGPL license.](https://people.sc.fsu.edu/~jburkardt/txt/gnu_lgpl.txt)

**Languages:**

**IVREAD** is available in [a FORTRAN90 version](https://people.sc.fsu.edu/~jburkardt/f_src/ivread/ivread.html).

**Related Data and Programs:**

[GRF\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/grf_io/grf_io.html), a FORTRAN90 library which can read or write a GRF file.

[IVCON](https://people.sc.fsu.edu/~jburkardt/cpp_src/ivcon/ivcon.html), a C++ program which reads 3D graphics file information in a variety of formats, and can output the same information in a different format.

[OBJ\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/obj_io/obj_io.html), a FORTRAN90 library which can read or write an OBJ file.

[PBMA\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/pbma_io/pbma_io.html), a FORTRAN90 library which can read or write an ASCII Portable Bit Map (PBM) file.

[PGMA\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/pgma_io/pgma_io.html), a FORTRAN90 library which can read or write an ASCII Portable Gray Map (PGM) file.

[PLOT3D\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/plot3d_io/plot3d_io.html), a FORTRAN90 library which can read or write a PLOT3D file.

[PPMA\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/ppma_io/ppma_io.html), a FORTRAN90 library which can read or write an ASCII Portable Pixel Map (PPM) file.

[STLA\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/stla_io/stla_io.html), a FORTRAN90 library which can read or write an ASCII Stereolithography file.

[TEC\_TO\_FEM](https://people.sc.fsu.edu/~jburkardt/f_src/tec_to_fem/tec_to_fem.html), a FORTRAN90 program which can read a TECPLOT file describing a surface in 3D composed of triangles, and write a set of FEM files.

[TEC\_TO\_OBJ](https://people.sc.fsu.edu/~jburkardt/f_src/tec_to_obj/tec_to_obj.html), a FORTRAN90 program which can read a TECPLOT file describing a surface in 3D composed of triangles or quadrilaterals, and write an OBJ file.

[TEC\_WRITE](https://people.sc.fsu.edu/~jburkardt/f_src/tec_write/tec_write.html), a FORTRAN90 library which can write a TECPLOT file.

[XYZ\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/xyz_io/xyz_io.html), a FORTRAN90 library which reads and writes XYZ files.

**Reference:**

1. Adrian Bowyer, John Woodwark,  
   A Programmer's Geometry,  
   Butterworths, 1983.
2. James Foley, Andries van Dam, Steven Feiner, John Hughes,  
   Computer Graphics, Principles and Practice,  
   Addison Wesley, Second Edition, 1990.
3. Albert Nijenhuis, Herbert Wilf,  
   Combinatorial Algorithms,  
   Academic Press, 1978, second edition,  
   ISBN 0-12-519260-6.
4. 3D Systems, Inc,  
   Stereolithography Interface Specification,  
   October 1989.

**Source Code:**

* [ivread.f90](https://people.sc.fsu.edu/~jburkardt/f_src/ivread/ivread.f90), the source code;

**Tests and Examples:**

* [cube.ase](https://people.sc.fsu.edu/~jburkardt/data/ase/cube.ase);
* [cube.byu](https://people.sc.fsu.edu/~jburkardt/data/byu/cube.byu);
* [diamond.dxf](https://people.sc.fsu.edu/~jburkardt/data/dxf/diamond.dxf);
* [cube.hrc](https://people.sc.fsu.edu/~jburkardt/data/hrc/cube.hrc);
* [materials.iv](https://people.sc.fsu.edu/~jburkardt/data/iv/materials.iv);
* [cube.obj](https://people.sc.fsu.edu/~jburkardt/data/obj/cube.obj);
* [cube.off](https://people.sc.fsu.edu/~jburkardt/data/off/cube.off);
* [surface.oogl](https://people.sc.fsu.edu/~jburkardt/data/oogl/surface.oogl);
* [cube.pov](https://people.sc.fsu.edu/~jburkardt/data/pov/cube.pov);
* [cube\_face.smf](https://people.sc.fsu.edu/~jburkardt/data/smf/cube_face.smf);
* [magnolia.stl](https://people.sc.fsu.edu/~jburkardt/data/stl/magnolia.stl);
* [cube.tec](https://people.sc.fsu.edu/~jburkardt/data/tec/cube.tec);
* [cube.tria](https://people.sc.fsu.edu/~jburkardt/data/tri/cube.tria);
* [diamond.vla](https://people.sc.fsu.edu/~jburkardt/data/vla/diamond.vla);
* [materials.wrl](https://people.sc.fsu.edu/~jburkardt/data/wrl/materials.wrl);
* [spiralpoints.xyz](https://people.sc.fsu.edu/~jburkardt/data/xyz/spiralpoints.xyz);

**List of Routines:**

* **MAIN** is the main program for IVREAD.
* **ANGLE\_RAD\_3D** returns the angle in radians between two vectors in 3D.
* **ASE\_READ** reads graphics information from an ASE file.
* **ASE\_WRITE** writes graphics information to an ASE file.
* **BYU\_READ** reads graphics data from a Movie.BYU surface geometry file.
* **BYU\_WRITE** writes out the graphics data as a Movie.BYU surface geometry file.
* **CH\_CAP** capitalizes a single character.
* **CH\_EQI** is a case insensitive comparison of two characters for equality.
* **CH\_IS\_CONTROL** reports whether a character is a control character or not.
* **CH\_TO\_DIGIT** returns the integer value of a base 10 digit.
* **COMMAND\_LINE** works with command line parameters.
* **COR3\_NORMAL\_SET** recomputes zero node normal vectors.
* **COR3\_RANGE** computes and prints the coordinate minima and maxima.
* **CROSS0\_3D** computes the cross product of (P1-P0) and (P2-P0) in 3D.
* **DATA\_CHECK** checks the input data, and enforces limits.
* **DATA\_INIT** initializes internal graphics data.
* **DATA\_READ** reads a file into internal graphics data.
* **DATA\_REPORT** gives a summary of the contents of the data file.
* **DATA\_WRITE** writes the internal graphics data to a file.
* **DEGREES\_TO\_RADIANS** converts an angle from degrees to radians.
* **DIGIT\_TO\_CH** returns the character representation of a decimal digit.
* **DOT0\_3D** computes the dot product of (P1-P0) and (P2-P0) in 3D.
* **DXF\_READ** reads graphics information from an AutoCAD DXF file.
* **DXF\_WRITE** writes graphics data to an AutoCAD DXF file.
* **EDGE\_ADD\_NODES** adds the edge defined by two nodes to the edge list.
* **EDGE\_BOUND** reports the edges which are part of the boundary.
* **EDGE\_COUNT** determines the number of edges in a graph.
* **EDGE\_MATCH\_FACE** seeks an edge common to a face and the edge list.
* **EDGE\_MATCH\_NODES** seeks an edge of the form (N1,N2) or (N2,N1) in EDGE.
* **EDGE\_NULL\_DELETE** deletes face edges with zero length.
* **ENORM\_ND** computes the Euclidean norm of a vector in ND.
* **ENORM0\_3D** computes the Euclidean norm of (P1-P0) in 3D.
* **FACE\_AREA\_SET** computes the area of the faces.
* **FACE\_CHECK** checks and analyzes a set of faces.
* **FACE\_FLIP** flips faces to achieve a consistent orientation.
* **FACE\_NORMAL\_AVE** sets face normals as average of face vertex normals.
* **FACE\_NULL\_DELETE** deletes faces of order less than 3.
* **FACE\_PRINT** prints out information about a face.
* **FACE\_REVERSE\_ORDER** reverses the order of the nodes in each face.
* **FACE\_SORT** renumbers the faces in order of object and tier.
* **FACE\_SUBSET** selects a subset of the current faces as the new object.
* **FACE\_TO\_EDGE** converts face data to edge data.
* **FACE\_TO\_LINE** converts face information to line information.
* **FACE\_TOUCH** reports whether two polygonal faces touch.
* **FILE\_GET\_NEXT\_WORD** returns the next word and trailing context from a file.
* **FILE\_NAME\_EXT\_GET** determines the "extension" of a file name.
* **GET\_UNIT** returns a free FORTRAN unit number.
* **HELLO** prints out a message about the program.
* **HELP** prints out a help message about the interactive commands.
* **HRC\_READ** reads graphics information from a SoftImage HRC file.
* **HRC\_WRITE** writes graphics data to an HRC SoftImage file.
* **I4\_MODP** returns the nonnegative remainder of integer division.
* **I4\_SWAP** switches two integer values.
* **I4\_TO\_S\_ZERO** converts an integer to a string, with zero padding.
* **I4\_WRAP** forces an integer to lie between given limits by wrapping.
* **INFILE** determines the input filename and type.
* **INTERACT** interacts with the user to specify input and output files.
* **INTNEX** "reads" integers from a string, one at a time.
* **IV\_POINT\_WRITE** writes point and line data to an Inventor file.
* **IV\_READ** reads graphics information from an Inventor file.
* **IV\_WRITE** writes graphics data to an Inventor file.
* **I4VEC\_MAX** computes the maximum element of an integer array.
* **I4VEC\_REVERSE** reverses the elements of an I4VEC.
* **I4VEC\_ROTATE** rotates an object in place.
* **LCON** reports whether a character is a control character or not.
* **MESH\_T3** produces a grid of pairs of 3 node triangles.
* **NEWS** prints out news (old and new) about the program.
* **NODE\_RELAX** smooths a shape by an averaging operation on the node positions.
* **NODE\_TO\_VERTEX\_MATERIAL** extends node material definitions to vertices.
* **OBJ\_READ** reads graphics information from a Wavefront OBJ file.
* **OBJ\_WRITE** writes graphics information to a WaveFront OBJ file.
* **OBJECT\_BUILD** builds edge-connected "objects" out of polygonal faces.
* **OBJECT\_INVERT** makes an inverted duplicate of the object.
* **OFF\_READ** reads graphics information from a GEOMVIEW OFF file.
* **OFF\_WRITE** writes graphics information to a GEOMVIEW OFF file.
* **OOGL\_READ** reads graphics information from a OOGL file.
* **OOGL\_GRID** adds a grid to an OOGL data file.
* **OUTFILE** determines the output filename and type.
* **PI** returns the value of pi.
* **PLANE\_EXP2IMP\_3D** converts an explicit plane to implicit form in 3D.
* **PLANE\_IMP\_POINT\_NEAREST\_3D:** nearest point on a implicit plane to a point in 3D.
* **POINTS\_DISTANCE\_3D** finds the distance between two points in 3D.
* **POLY\_2\_TRI** converts a collection of polygons into a collection of triangles.
* **POV\_WRITE** writes graphics information to a POV file.
* **PROJECT\_2D** projects 3D data to 2D based on user choices.
* **PROJECT\_ANGLE** converts 3D data to 2D using a presentation angle.
* **PROJECT\_OPLANE** projects 3D points onto an orthographic plane.
* **PROJECT\_PPLANE** projects a point through a focus point onto a perspective plane.
* **PS\_WRITE** writes 2D face and line information to a PostScript file.
* **R4\_RANDOM** returns a random real in a given range.
* **R4\_SWAP** switches two R4's.
* **R4COL\_FIND** seeks a table column equal to a real vector.
* **RGB\_TO\_HUE** converts (R,G,B) colors to a hue value between 0 and 1.
* **RELNEX** "reads" real numbers from a string, one at a time.
* **R4VEC\_TO\_S** "writes" an R4VEC into a string.
* **S\_BLANK\_DELETE** removes blanks from a string, left justifying the remainder.
* **S\_BLANKS\_DELETE** replaces consecutive blanks by one blank.
* **S\_CAP** replaces any lowercase letters by uppercase ones in a string.
* **S\_CAT** concatenates two strings to make a third string.
* **S\_CONTROL\_BLANK** replaces control characters with blanks.
* **S\_EQI** is a case insensitive comparison of two strings for equality.
* **S\_INDEX\_LAST** finds the LAST occurrence of a given substring.
* **S\_IS\_I4** returns .TRUE. if STRING represents an integer.
* **S\_IS\_R4** returns .TRUE. if STRING represents a real number.
* **S\_TO\_I4** reads an I4 from a string.
* **S\_TO\_R4** reads an R4 from a string.
* **S\_TRIM\_ZEROS** removes trailing zeros from a string.
* **SMF\_READ** reads graphics information from an SMF file.
* **SMF\_WRITE** writes graphics information to an SMF file.
* **SORT\_HEAP\_EXTERNAL** externally sorts a list of items into linear order.
* **STLA\_READ** reads graphics information from an ASCII StereoLithography file.
* **STLA\_WRITE** writes graphics information to an ASCII StereoLithography file.
* **TEC\_WRITE** writes graphics information to a TECPLOT file.
* **TIMESTAMP** prints the current YMDHMS date as a time stamp.
* **TMAT\_INIT** initializes the geometric transformation matrix.
* **TMAT\_MXM** multiplies two geometric transformation matrices.
* **TMAT\_MXP** multiplies a geometric transformation matrix times a point.
* **TMAT\_MXP2** multiplies a geometric transformation matrix times N points.
* **TMAT\_MXV** multiplies a geometric transformation matrix times a vector.
* **TMAT\_ROT\_AXIS** applies a coordinate axis rotation to the geometric transformation matrix.
* **TMAT\_ROT\_VECTOR:** arbitrary axis rotation to geometric transformation matrix.
* **TMAT\_SCALE** applies a scaling to the geometric transformation matrix.
* **TMAT\_SHEAR** applies a shear to the geometric transformation matrix.
* **TMAT\_TRANS** applies a translation to the geometric transformation matrix.
* **TRIA\_READ** reads graphics information from an ASCII triangle file.
* **TRIA\_WRITE** writes the graphics data to an ASCII "triangle" file.
* **TS\_READ** reads graphics information from a Mathematica TS file.
* **TS\_WRITE** writes graphics information to a Mathematica TS file.
* **TXT\_WRITE** writes the graphics data to a text file.
* **UCD\_WRITE** writes graphics data to an AVS UCD file.
* **VECTOR\_UNIT\_ND** normalizes a vector in ND.
* **VERTEX\_NORMAL\_SET** recomputes the face vertex normal vectors.
* **VERTEX\_TO\_NODE\_MATERIAL** extends vertex material definitions to nodes.
* **VLA\_READ** reads graphics information from a VLA file.
* **VLA\_WRITE** writes graphics data to a VLA file.
* **VRML\_READ** reads graphics information from a VRML file.
* **VRML\_WRITE** writes graphics data to a VRML file.
* **WORD\_NEXT\_READ** "reads" words from a string, one at a time.
* **XGL\_WRITE** writes graphics data to an XGL file.
* **XYZ\_READ** reads graphics information from an XYZ file.
* **XYZ\_WRITE** writes graphics data to an XYZ file.

You can go up one level to [the FORTRAN90 source codes](https://people.sc.fsu.edu/~jburkardt/f_src/f_src.html).

*Last revised on 15 November 2006.*

**IVREAD   
3D图形文件转换**

**IVREAD** 是一个FORTRAN90程序，它读取3D图形文件并将信息转换为另一种格式。

**IVREAD**最初编写用于阅读SGI Inventor 3D图形文件并将其转换为Digistar VLA格式。从那时起，该计划已经发展到允许输入和输出：

* ASE - [AutoCAD](http://www.autodesk.com/) ASCII导出文件;
* BYU - [Movie.BYU曲面几何文件](https://people.sc.fsu.edu/~jburkardt/data/byu/byu.html)（仅输出）;
* DXF - [AutoCAD](http://www.autodesk.com/) DXF文件;
* HRC - [SoftImage](http://www.softimage.com/) 层次结构文件;
* IV - [SGI Inventor](http://www.sgi.com/)文件;
* OBJ - 来自ALIAS的文件格式（ <http://www.alias.com/eng/index.shtml>）
* OFF - [GeomView的](http://www.geomview.org/)目标文件格式
* OOGL - 来自[GeomView的](http://www.geomview.org/) CMESH文件 （仅限输入）;
* POV - [视觉](http://www.povray.org/)文件的[持久性](http://www.povray.org/)（仅输出）;
* SMF - Michael Garland的[QSLIM](http://www.cs.cmu.edu/afs/cs/user/garland/www/quadrics/qslim.html)格式 ;
* STL - [ASCII Stereolithography文件](https://people.sc.fsu.edu/~jburkardt/data/stla/stla.html) ;
* TEC - [TECPLOT](http://www.amtec.com/)文件（仅输出）;
* TS - Mathematica [ThreeScript](http://documents.wolfram.com/v4/AddOns/Grph_ThreeScript-.html) ;
* TXT - 文本文件（仅输出）;
* TRI / TRIA - Greg Hood要求[的简单ASCII三角形格式](https://people.sc.fsu.edu/~jburkardt/data/tri/tri.html) ;
* UCD - [高级视觉系统（AVS）](http://www.avs.com/) 非结构化单元数据（仅输出）;
* VLA - [Digistar VLA文件;](https://people.sc.fsu.edu/~jburkardt/data/vla/vla.html)
* WRL - VRML（虚拟现实建模语言）文件（仅输出）。
* XYZ - [XYZ文件](https://people.sc.fsu.edu/~jburkardt/data/xyz/xyz.html) （低效但简单，点和线格式）;

还有一个简单的工具，可以将任何3D图像投影到2D平面上，并制作结果的 [PostScript图像](https://people.sc.fsu.edu/~jburkardt/data/ps/ps.html)。

图形文件格式通常记录不清; 因此，这个程序主要是通过检查典型文件并努力解释它们来编写的。特别是在Inventor和DXF文件的情况下，程序无法处理文件格式的所有细微之处，并且可能会破坏数据，甚至崩溃。

**用法：**

**ivread***input.ext output.ext*

将*input.ext*转换为*output.ext*，其中输入和输出格式由文件扩展名确定。

**ivread**

如果未指定文件，则设置交互式对话框，用户可以指定要转换的文件。

**许可：**

在此网页上描述和提供的计算机代码和数据文件是在[GNU LGPL许可](https://people.sc.fsu.edu/~jburkardt/txt/gnu_lgpl.txt)下分发 [的。](https://people.sc.fsu.edu/~jburkardt/txt/gnu_lgpl.txt)

**语言：**

**IVREAD**是提供 [一个FORTRAN90版本](https://people.sc.fsu.edu/~jburkardt/f_src/ivread/ivread.html)。

**相关数据和程序：**

[GRF\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/grf_io/grf_io.html)，一个可以读取或写入GRF文件的FORTRAN90库。

[IVCON](https://people.sc.fsu.edu/~jburkardt/cpp_src/ivcon/ivcon.html)，一种C ++程序，可以读取各种格式的3D图形文件信息，并可以以不同的格式输出相同的信息。

[OBJ\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/obj_io/obj_io.html)，一个可以读取或写入OBJ文件的FORTRAN90库。

[PBMA\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/pbma_io/pbma_io.html)，一个FORTRAN90库，可以读取或写入ASCII便携式位图（PBM）文件。

[PGMA\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/pgma_io/pgma_io.html)，一个FORTRAN90库，可以读取或写入ASCII便携式[灰度图](https://people.sc.fsu.edu/~jburkardt/f_src/pgma_io/pgma_io.html)（PGM）文件。

[PLOT3D\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/plot3d_io/plot3d_io.html)，一个可以读取或写入PLOT3D文件的FORTRAN90库。

[PPMA\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/ppma_io/ppma_io.html)，一个FORTRAN90库，可以读取或写入ASCII便携式像素图（PPM）文件。

[STLA\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/stla_io/stla_io.html)，一个FORTRAN90库，可以读取或写入ASCII Stereolithography文件。

[TEC\_TO\_FEM](https://people.sc.fsu.edu/~jburkardt/f_src/tec_to_fem/tec_to_fem.html)，FORTRAN90程序，可以读取TECPLOT文件，描述由三角形组成的3D表面，并编写一组FEM文件。

[TEC\_TO\_OBJ](https://people.sc.fsu.edu/~jburkardt/f_src/tec_to_obj/tec_to_obj.html)，FORTRAN90程序，可以读取TECPLOT文件，描述由三角形或四边形组成的3D表面，并写入OBJ文件。

[TEC\_WRITE](https://people.sc.fsu.edu/~jburkardt/f_src/tec_write/tec_write.html)，FORTRAN90库，可以写入TECPLOT文件。

[XYZ\_IO](https://people.sc.fsu.edu/~jburkardt/f_src/xyz_io/xyz_io.html)，一个读取和写入XYZ文件的FORTRAN90库。

**参考：**

1. Adrian Bowyer，John Woodwark，  
   A Programmer's Geometry，  
   Butterworths，1983。
2. James Foley，Andries van Dam，Steven Feiner，John Hughes，  
   Computer Graphics，Principles and Practice，  
   Addison Wesley，Second Edition，1990。
3. Albert Nijenhuis，Herbert Wilf，  
   Combinatorial Algorithms，  
   Academic Press，1978，第二版，  
   ISBN 0-12-519260-6。
4. 3D Systems，Inc，  
   Stereolithography Interface Specification，  
   1989年10月。

**源代码：**

* [ivread.f90](https://people.sc.fsu.edu/~jburkardt/f_src/ivread/ivread.f90)，源代码;

**测试和示例：**

* [cube.ase](https://people.sc.fsu.edu/~jburkardt/data/ase/cube.ase) ;
* [cube.byu](https://people.sc.fsu.edu/~jburkardt/data/byu/cube.byu) ;
* [diamond.dxf](https://people.sc.fsu.edu/~jburkardt/data/dxf/diamond.dxf) ;
* [cube.hrc](https://people.sc.fsu.edu/~jburkardt/data/hrc/cube.hrc) ;
* [materials.iv](https://people.sc.fsu.edu/~jburkardt/data/iv/materials.iv) ;
* [cube.obj](https://people.sc.fsu.edu/~jburkardt/data/obj/cube.obj) ;
* [cube.off](https://people.sc.fsu.edu/~jburkardt/data/off/cube.off) ;
* [surface.oogl](https://people.sc.fsu.edu/~jburkardt/data/oogl/surface.oogl) ;
* [cube.pov](https://people.sc.fsu.edu/~jburkardt/data/pov/cube.pov) ;
* [cube\_face.smf](https://people.sc.fsu.edu/~jburkardt/data/smf/cube_face.smf) ;
* [magnolia.stl](https://people.sc.fsu.edu/~jburkardt/data/stl/magnolia.stl) ;
* [cube.tec](https://people.sc.fsu.edu/~jburkardt/data/tec/cube.tec) ;
* [cube.tria](https://people.sc.fsu.edu/~jburkardt/data/tri/cube.tria) ;
* [diamond.vla](https://people.sc.fsu.edu/~jburkardt/data/vla/diamond.vla) ;
* [materials.wrl](https://people.sc.fsu.edu/~jburkardt/data/wrl/materials.wrl) ;
* [spiralpoints.xyz](https://people.sc.fsu.edu/~jburkardt/data/xyz/spiralpoints.xyz) ;

**例程列表：**

* **MAIN**是IVREAD的主要程序。
* **ANGLE\_RAD\_3D**返回3D中两个矢量之间的弧度角度。
* **ASE\_READ**从ASE文件中读取图形信息。
* **ASE\_WRITE**将图形信息写入ASE文件。
* **BYU\_READ**从Movie.BYU曲面几何文件中读取图形数据。
* **BYU\_WRITE**将图形数据写出为Movie.BYU曲面几何文件。
* **CH\_CAP**将单个字符大写。
* **CH\_EQI**是对两个字符进行不区分大小写的比较。
* **CH\_IS\_CONTROL**报告字符是否是控制字符。
* **CH\_TO\_DIGIT**返回基数为10的整数值。
* **COMMAND\_LINE**使用命令行参数。
* **COR3\_NORMAL\_SET**重新计算零节点法向量。
* **COR3\_RANGE**计算并打印坐标最小值和最大值。
* **CROSS0\_3D**计算3D中的（P1-P0）和（P2-P0）的叉积。
* **DATA\_CHECK**检查输入数据，并强制执行限制。
* **DATA\_INIT**初始化内部图形数据。
* **DATA\_READ**将文件读入内部图形数据。
* **DATA\_REPORT**给出了数据文件内容的摘要。
* **DATA\_WRITE**将内部图形数据写入文件。
* **DEGREES\_TO\_RADIANS**将角度从度数转换为弧度。
* **DIGIT\_TO\_CH**返回十进制数字的字符表示。
* **DOT0\_3D**计算3D中的（P1-P0）和（P2-P0）的点积。
* **DXF\_READ**从AutoCAD DXF文件中读取图形信息。
* **DXF\_WRITE**将图形数据写入AutoCAD DXF文件。
* **EDGE\_ADD\_NODES**将由两个节点定义的边添加到边列表。
* **EDGE\_BOUND**报告作为边界一部分的边。
* **EDGE\_COUNT**确定图表中的边数。
* **EDGE\_MATCH\_FACE**寻找面和边列表**共有**的边。
* **EDGE\_MATCH\_NODES**在EDGE中寻找形式（N1，N2）或（N2，N1）的边缘。
* **EDGE\_NULL\_DELETE**删除长度为零的面边。
* **ENORM\_ND**计算ND中向量的欧几里德范数。
* **ENORM0\_3D**在3D中计算（P1-P0）的欧几里德范数。
* **FACE\_AREA\_SET**计算面部区域。
* **FACE\_CHECK**检查并分析一组面。
* **FACE\_FLIP**翻转面以实现一致的方向。
* **FACE\_NORMAL\_AVE**将面法线设置为面顶点法线的平均值。
* **FACE\_NULL\_DELETE**删除小于3的面。
* **FACE\_PRINT**打印出有关面部的信息。
* **FACE\_REVERSE\_ORDER**反转每个面中节点的顺序。
* **FACE\_SORT**按对象和层的顺序重新编号面。
* **FACE\_SUBSET**选择当前面的子集作为新对象。
* **FACE\_TO\_EDGE**将面数据转换为边数据。
* **FACE\_TO\_LINE**将面部信息转换为线路信息。
* **FACE\_TOUCH**报告两个多边形面是否接触。
* **FILE\_GET\_NEXT\_WORD**返回文件中的下一个单词和尾随上下文。
* **FILE\_NAME\_EXT\_GET**确定文件名的“扩展名”。
* **GET\_UNIT**返回一个免费的FORTRAN单元号。
* **HELLO**打印出有关该程序的消息。
* **HELP**打印出有关交互式命令的帮助消息。
* **HRC\_READ**从SoftImage HRC文件中读取图形信息。
* **HRC\_WRITE**将图形数据写入HRC SoftImage文件。
* **I4\_MODP**返回整数除法的非负余数。
* **I4\_SWAP**切换两个整数值。
* **I4\_TO\_S\_ZERO**将整数转换为字符串，填充为零。
* **I4\_WRAP**通过换行强制整数位于给定限制之间。
* **INFILE**确定输入文件名和类型。
* **INTERACT**与用户交互以指定输入和输出文件。
* **INTNEX**从一个字符串“读取”整数，一次一个。
* **IV\_POINT\_WRITE**将点和线数据写入Inventor文件。
* **IV\_READ**从Inventor文件中读取图形信息。
* **IV\_WRITE**将图形数据写入Inventor文件。
* **I4VEC\_MAX**计算整数数组的最大元素。
* **I4VEC\_REVERSE**反转**I4VEC**的元素。
* **I4VEC\_ROTATE**将对象旋转到位。
* **LCON**报告角色是否是控制角色。
* **MESH\_T3**生成一对3节点三角形的网格。
* **NEWS**打印出有关该计划的新闻（新旧）。
* **NODE\_RELAX**通过节点位置上的平均操作来平滑形状。
* **NODE\_TO\_VERTEX\_MATERIAL**将节点材质定义扩展到顶点。
* **OBJ\_READ**从Wavefront OBJ文件中读取图形信息。
* **OBJ\_WRITE**将图形信息写入WaveFront OBJ文件。
* **OBJECT\_BUILD**从多边形面构建边缘连接的“对象”。
* **OBJECT\_INVERT**生成对象的反转副本。
* **OFF\_READ**从GEOMVIEW OFF文件中读取图形信息。
* **OFF\_WRITE**将图形信息写入GEOMVIEW OFF文件。
* **OOGL\_READ**从OOGL文件中读取图形信息。
* **OOGL\_GRID**将网格添加到OOGL数据文件。
* **OUTFILE**确定输出文件名和类型。
* **PI**返回pi的值。
* **PLANE\_EXP2IMP\_3D**将显式平面转换为3D中的隐式形式。
* **PLANE\_IMP\_POINT\_NEAREST\_3D：**隐式平面上距离3D点的最近点。
* **POINTS\_DISTANCE\_3D**在3D中找到两点之间的距离。
* **POLY\_2\_TRI**将多边形集合转换为三角形集合。
* **POV\_WRITE**将图形信息写入POV文件。
* **PROJECT\_2D**根据用户选择将3D数据**投影**到2D。
* **PROJECT\_ANGLE**使用展示角度将3D数据转换为2D。
* **PROJECT\_OPLANE**将3D点投影到正交平面上。
* **PROJECT\_PPLANE**将焦点通过焦点投影到透视平面上。
* **PS\_WRITE**将2D面部和线条信息写入PostScript文件。
* **R4\_RANDOM**在给定范围内返回随机实数。
* **R4\_SWAP**切换两个R4。
* **R4COL\_FIND**寻找一个等于实数向量的表列。
* **RGB\_TO\_HUE**将（R，G，B）颜色转换为0到1之间的色调值。
* **RELNEX**从一个字符串中“读取”实数，一次一个。
* **R4VEC\_TO\_S**将**R4VEC** “写入”字符串。
* **S\_BLANK\_DELETE**从字符串中删除空格，左**对齐**余数。
* **S\_BLANKS\_DELETE**将连续空格替换为一个空格。
* **S\_CAP**用字符串中的大写字母替换任何小写字母。
* **S\_CAT**连接两个字符串以生成第三个字符串。
* **S\_CONTROL\_BLANK**用空格替换控制字符。
* **S\_EQI**是两个字符串的不区分大小写，用于求相等。
* **S\_INDEX\_LAST**查找给定子字符串的最后一次出现。
* **S\_IS\_I4**返回**.TRUE**。如果STRING表示整数。
* **S\_IS\_R4**返回**.TRUE**。如果STRING代表实数。
* **S\_TO\_I4**从字符串中读取I4。
* **S\_TO\_R4**从字符串中读取R4。
* **S\_TRIM\_ZEROS**从字符串中删除尾随零。
* **SMF\_READ**从SMF文件中读取图形信息。
* **SMF\_WRITE**将图形信息写入SMF文件。
* **SORT\_HEAP\_EXTERNAL从**外部将项目列表按线性顺序排序。
* **STLA\_READ**从ASCII StereoLithography文件中读取图形信息。
* **STLA\_WRITE**将图形信息写入ASCII StereoLithography文件。
* **TEC\_WRITE**将图形信息写入TECPLOT文件。
* **TIMESTAMP**将当前YMDHMS日期打印为时间戳。
* **TMAT\_INIT**初始化几何变换矩阵。
* **TMAT\_MXM**将两个几何变换矩阵相乘。
* **TMAT\_MXP**将几何变换矩阵乘以一个点。
* **TMAT\_MXP2**将几何变换矩阵乘以N个点。
* **TMAT\_MXV**将几何变换矩阵乘以矢量。
* **TMAT\_ROT\_AXIS**将坐标轴旋转应用于几何变换矩阵。
* **TMAT\_ROT\_VECTOR：**任意轴旋转到几何变换矩阵。
* **TMAT\_SCALE**将缩放应用于几何变换矩阵。
* **TMAT\_SHEAR**将剪切应用于几何变换矩阵。
* **TMAT\_TRANS**将转换应用于几何变换矩阵。
* **TRIA\_READ**从ASCII三角形文件中读取图形信息。
* **TRIA\_WRITE**将图形数据写入ASCII“三角形”文件。
* **TS\_READ**从Mathematica TS文件中读取图形信息。
* **TS\_WRITE**将图形信息写入Mathematica TS文件。
* **TXT\_WRITE**将图形数据写入文本文件。
* **UCD\_WRITE**将图形数据写入AVS UCD文件。
* **VECTOR\_UNIT\_ND**规范化ND中的矢量。
* **VERTEX\_NORMAL\_SET**重新计算面顶点法线向量。
* **VERTEX\_TO\_NODE\_MATERIAL**将顶点材质定义扩展到节点。
* **VLA\_READ**从VLA文件中读取图形信息。
* **VLA\_WRITE**将图形数据写入VLA文件。
* **VRML\_READ**从VRML文件中读取图形信息。
* **VRML\_WRITE**将图形数据写入VRML文件。
* **WORD\_NEXT\_READ** “读取”字符串中的单词，一次一个。
* **XGL\_WRITE**将图形数据写入XGL文件。
* **XYZ\_READ**从XYZ文件中读取图形信息。
* **XYZ\_WRITE**将图形数据写入XYZ文件。

您可以上一级到[FORTRAN90源代码](https://people.sc.fsu.edu/~jburkardt/f_src/f_src.html)。

*最后修订于2006年11月15日。*