

Autodesk®

Command Line Utilities

AliasStudio 2008



Command Line Utilities
Copyright and trademarks

AliasStudio 2008

© Copyright 2002-2007 Autodesk, Inc. All rights reserved.

This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

AUTODESK, INC., MAKES NO WARRANTY, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS, AND MAKES SUCH MATERIALS AVAILABLE SOLELY ON AN "AS-IS" BASIS. IN NO EVENT SHALL AUTODESK, INC., BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF ACQUISITION OR USE OF THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO AUTODESK, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE, IF ANY, OF THE MATERIALS DESCRIBED HEREIN.

Autodesk, Inc., reserves the right to revise and improve its products as it sees fit. This publication describes the state of this product at the time of its publication, and may not reflect the product at all times in the future.

Autodesk Trademarks

The following are registered trademarks or trademarks of Autodesk, Inc., in the USA and other countries: 3DEC (design/logo), 3December, 3December.com, 3ds Max, ActiveShapes, Actrix, ADI, Alias, Alias (swirl design/logo), AliasStudio, Alias|Wavefront (design/logo), ATC, AUGI, AutoCAD, AutoCAD Learning Assistance, AutoCAD LT, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk, Autodesk Envision, Autodesk Insight, Autodesk Intent, Autodesk Inventor, Autodesk Map, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSnap, AutoSketch, AutoTrack, Backdraft, Built with ObjectARX (logo), Burn, Buzzsaw, CAiCE, Can You Imagine, Character Studio, Cinestream, Civil 3D, Cleaner, Cleaner Central, ClearScale, Colour Warper, Combustion, Communication Specification, Constructware, Content Explorer, Create>what's>Next> (design/logo), Dancing Baby (image), DesignCenter, Design Doctor, Designer's Toolkit, DesignKids, DesignProf, DesignServer, DesignStudio, Design|Studio (design/logo), Design Your World, Design Your World (design/logo), DWF, DWG, DWG (logo), DWG TrueConvert, DWG TrueView, DXF, EditDV, Education by Design, Extending the Design Team, FBX, Filmbox, FMDesktop, GDX Driver, Gmax, Heads-up Design, Heidi, HOOPS, HumanIK, i-drop, iMOUT, Incinerator, IntroDV, Kaydara, Kaydara (design/logo), LocationLogic, Lustre, Maya, Mechanical Desktop, MotionBuilder, ObjectARX, ObjectDBX, Open Reality, PolarSnap, PortfolioWall, Powered with Autodesk Technology, Productstream, ProjectPoint, Reactor, RealDWG, Real-time Roto, Render Queue, Revit, Showcase, SketchBook, StudioTools, Topobase, Toxik, Visual, Visual Bridge, Visual Construction, Visual Drainage, Visual Hydro, Visual Landscape, Visual Roads, Visual Survey, Visual Syllabus, Visual Toolbox, Visual Tugboat, Visual LISP, Voice Reality, Volo, and Wiretap.

The following are registered trademarks or trademarks of Autodesk Canada Co. in the USA and/or Canada and other countries: Backburner, Discreet, Fire, Flame, Flint, Frost, Inferno, Multi-Master Editing, River, Smoke, Sparks, Stone, Wire.

All other brand names, product names or trademarks belong to their respective holders.

Third-Party Copyright Notices

This product includes software developed by the Apache Software Foundation.

Macromedia Shockwave™ Player and Macromedia Flash™ Player software by Macromedia, Inc., Copyright © 1995-2000 Adobe Systems Incorporated. All rights reserved.

Portions relating to JPEG Copyright © 1991-1998 Thomas G. Lane. All rights reserved. This software is based in part on the work of the Independent JPEG Group.

Portions relating to TIFF Copyright © 1997-1998 Sam Leffler. Copyright © 1991-1997 Silicon Graphics, Inc. All rights reserved.

GOVERNMENT USE

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer Software-Restricted Rights) and DFAR 227.7202 (Rights in Technical Data and Computer Software), as applicable.

Published By: Autodesk, Inc.
111 McInnis Parkway
San Rafael, CA 94903, USA

Documentation build date: April 5, 2007

Contents

Command line utilities 1

Platforms 2

Background utilities 4

Command reference 5

- AliasBatch 5
- AlToCa 6
- AlToC5 6
- AlToDWG 7
- AlToGR3 7
- AlToIv 8
- AlToObj 10
- AlToSt 11
- AlToUG, AlToUG19, AlToUG20, AlToUG21, AlToUG22 13
- BakeOrientation 14
- bsdl 14
- CaToAl 14
- C5ToAl 15
- DWGToAl 16
- FieldAssembler 16
- from100to97 17
- GrToAl 19
- hp_glplotf, hp_gl2plotf 19
- IGESToAl 21
- imgcvt 21
- IvToAl 25
- Imgrd 25
- lmutil 27
- ObjToAl 27
- PRenderToAlias 28
- print_wire_header 28
- psplotf 29
- renderer/raytracer/powercaster/powertracer 30

StToAI 34

SwToAI 35

UGToAI, UGToAI19, UGToAI20, UGToAI21, UGToAI22 35

Index 37

COMMAND LINE UTILITIES

Describes various command line programs included with AliasStudio.

PLATFORMS

Lists which platforms each utility is available on.

The functions are available in the following operating environments.

Function	Platforms	Purpose
AliasBatch	Windows	Batch File Translator does unattended translation of data files. This replaces Alias-B .
AItoDWG	Windows	Translates Alias wire files to DWG file format.
AItoCa	Windows	Translates Alias wire files to CAI file format.
AItoC5	Windows	Translates Alias wire files to CATIA V5 file format.
AItoGR3	Windows	Translates Alias wire files into Granite 3 format.
AItoIv	Windows	Translates Alias wire files to Inventor file format.
AItoObj	Windows	Translate Alias wire files to OBJ file format.
AItoSt	Windows	Translates Alias wire files to ISO10303 (STEP) format.
AItoUG, AItoUG19, AItoUG20, AItoUG21, ALtoUG22	Windows	Translates Alias wire files to Unigraphics 19, 20 or 21 format.
BakeOrientation	Windows	Bakes Alias wire file normals orientation.
bsdl	Windows	Converts a binary SDL file into ASCII text
CaToAI	Windows	Translates CAI files to Alias wire files.
C5ToAI	Windows	Translates CATIA V5 files to Alias wire files.
DWGToAI	Windows	Translates DWG or DXF files to Alias wire format.
FieldAssembler	Windows	Interlaces scan-lines of images rendered on fields into a single image; also pastes together incompletely rendered images. See the online documentation that comes with FieldAssembler.
from100to97	, Windows	Converts V10 and later wire files to V9.7
GrToAI	Windows	PTC file to Alias Wire file translator (using DirectConnect).
hp_glplotf, hp_gl2plotf	Windows	Translates Alias plot files into HP-GL (IBM-GL) and HP-GL2 formats.
IGESToAI	Windows	IGES file to Alias Wire file translator (using DirectConnect).
imgcvt	Windows	Converts between a wide variety of image file formats.
IvToAI	Windows	Translates Inventor files to Alias wire files.

Function	Platforms	Purpose
Imgrd	Windows	Looks for the license file, which contains the information about vendors and features.
Imutil	Windows	Is a licence management program from Globetrotter Software.
ObjToAI	Windows	Converts OBJ files to Alias wire file format.
powercaster	Windows	Is a command line parallel renderer for AliasStudio SDL files. See the section <i>renderer/raytracer/powercaster/powertracer</i> .
powertracer	Windows	Is a command line parallel raytracer for AliasStudio SDL files. See the section <i>renderer/raytracer/powercaster/powertracer</i> .
PRenderToAlias	Windows	Converts Pro/Engineer render (.slp) files to Alias wire file format.
print_wire_header	Windows	Views the headers of wire files.
psplotf	Windows	Translates Alias plot files into PostScript.
raytracer	Windows	Is a command line raytracer for AliasStudio SDL files. See the section <i>renderer/raytracer/powercaster/powertracer</i> .
renderer	Windows	Is a command line renderer for AliasStudio SDL files. See the section <i>renderer/raytracer/powercaster/powertracer</i> .
StToAI	Windows	Converts STEP format files to Alias wire format.
SwToAI	Windows	Converts SolidWorks files to Alias wire format.
UGToAI, UGToAI19, UGToAI20, UGToAI21, UGToAI22	Windows	Converts Unigraphics 19, 20, or 21 files to Alias wire file format.

BACKGROUND UTILITIES

The following utilities are called by other programs, are used in the background, or are documented elsewhere.

Function	Platforms	Purpose
Alias.exe	Windows	Is the executable for the Studio interactive package
AliasWarn	Windows	a background utility to display error messages
Imutil	Windows	component of the Licence Manager
Imgrd	Windows	component of the Licence Manager. See the description of Imutil
Imtools	Windows	component of the Licence Manager
sgiawd	Windows	component of the Licence Manager
toimg	Windows	works with imgcvt during conversion of image files

Using command line utilities on Windows

- 1 Choose **Programs > Accessories > Command Prompt** from the Start menu.
- 2 Click the cursor in the window.

You can see what directory you are in by typing

```
dir
```

A list of the directory contents will be provided; at the top of the list is your current location.

- 3 Navigate to the directory containing the files to be converted. If you have installed AliasStudio in its default location, the path will be similar to

```
C:\Documents and Settings\[UserID]\My  
Documents\AliasStudio\user_data\[Project]
```

where [UserID] is replaced by your login userid (and no square brackets) and [Project] is replaced by the name you have assigned to the project: the default project is demo (again, no square brackets). So to navigate to the directory, on a single line, type

```
cd C:\Documents and Settings\[UserID]\My  
Documents\AliasStudio\user_data\[Project]\wire
```

- 4 To use a stand-alone utility to convert a file named “headlight.[fileextension]” to an AliasStudio wirefile named “headlight.wire, choose the translator and type

```
[translator] headlight.[fileextension] [flags] headlight.wire
```

where [translator] is the name of the stand-alone utility being used, [fileextension] is the three-letter extension after the dot, and [flags] are the stand-alone utility’s options.

COMMAND REFERENCE

AliasBatch

Platforms

Windows

Description

The Alias Batch File Translator is a command-line invocation of AliasStudio that allows unattended data file translation. You can use this interface to AliasStudio in a UNIX shell script to run multiple file translations unattended, perhaps in an overnight batch job. All file formats supported by AliasStudio retrieve/save features are supported, with the exception of Filter.

The standalone utility Alias -B has been replaced by the application AliasBatch, which uses the OpenModel API. You will find that AliasBatch is more robust. For help, enter `AliasBatch -h`.

Note: Alias -B is no longer supported.

> Command Line Options

AliasBatch [-h] [-w -sdl -r1 -r2] [-o...] -s RetFile...

AliasBatch	Launches the Alias Batch Processor, reads in the RetFile, translates it according to settings in an options file, writes it out as SavFile, and exits.
-h	Displays help information.
-o[OptFile]	Loads the options file OptFile. (Its full path is required.)
-s[SavFile]	Writes output to SavFile. This option is compulsory.
-w	Overwrites the SavFile if it exists.
-sdl	Sets the format to SDL; overrides the save format in the options file.
-r1	Optimizes redundant pre-V5.0 trimmed surfaces by converting them to surfaces.
RetFile	Retrieves files at start up (wire, IGES, etc.). If more than one file is retrieved, the SavFile is a merged stage or stage set.

AlToCa

Platforms

Windows

Purpose

AlToCa translates Alias wire file to CAI format files. CAI is CATIA/Alias Interoperability format.

Description

The usage statement for AlToCa is as follows:

```
AlToCa :
```

```
AlToCa [ options ] [-i<input Wire file>] -o<output CAI file>
```

-h	outputs online help, then exits.
-b	displays a log file on screen during execution
-x	produces an extended log file
-i	name of input wire file; if not specified, stdin (generally the keyboard) is used
-o	name of output CAI file; must be specified

AlToC5

AlToC5 - Convert Alias Wire files to CATIA Part (.CATPart) files.

Platforms:

Windows

Usage:

```
AlToC5 [<options>] -i <input Wire file> -o <output CATIA file>
```

where

Argument	Value
-i	input Wire filename (.wire)
-o	output CATIA filename (.CATPart)
-l	logfile option: 0=No logfile output (default) 1=Output a logfile (-l1) 2=Output an extended logfile (-l2)
-b	Processes symmetry information from Alias layers; the geometric objects are converted.

Argument	Value
-a	Convert layer to open body/geometrical set or hybrid body feature
-v	do not convert invisible geometric entities
-h	display this help information, then exit
Surface (divide/split) options:	
-c	Do not divide closed/periodic surface in multiple surfaces. By default, AIToC5 will split any closed/periodic surface and trimmed surface in two pieces if this option is not specified.
-d	divide surface with multi-knots in multiple surfaces

By default, AIToC5 will split any closed/periodic surface and trimmed surface in two pieces.

If -c option is specified, we don't split them.

AIToDWG

Option	Value
-i	input Wire file
-o	output DwG file

AIToDWG is an Autodesk DirectConnect utility.

AIToGR3

Converts Alias wire files to Granite 3 file format.

Arguments: -i <infile> -o [<outfile>]

If no output filename is provided, the input file's base name will be used as the outfile.

Option	Value
-h	Show help
-n	Save PTC Neutral file instead of Granite file.

Option	Value
-i <infile>	an Alias Wire file.
-o <outfile>	a Granite part file that will be appended with .g

AIToIv

Platforms

Windows

Purpose

AIToIv translates Alias wire file to SGI OpenInventor format files.

Description

AIToIv [options] [-i <infile> [-o <outfile>]]

-h	outputs online help
-binary	switches the output to a more compact Inventor binary format. The default is Inventor ASCII format.
-ascii	outputs the ASCII inventor
-verbose	displays all messages.
-quiet	operates with no feedback
-notransforms	outputs world space objects
-alltransforms	outputs a full hierachy with transforms at each DAG node and each object space object
-transforms	outputs required transforms at each DAG node and at each object space object
-tri	tesselates all NURBS surfaces to triangles using settings from the Render Stats window
-quad	tesselates all NURBS surfaces to quadrangles wherever possible, using settings from thje Render Stats window
-nurb	outputs NURBS surfaces
-cameras	outputs cameras
-nocameras	does not output cameras
-instances	converts AliasStudio instances to Inventor instances
-noinstances	converts AliasStudio instances to copies

-inline	creates inline texture data instead of references
-noinline	creates referenced textures where possible (for file textures only)
-inventory	creates a single Inventor material or texture for each AliasStudio material or texture, and an instance for each object
-noinventory	creates an Inventor material or texture for each object
-units <i>name</i>	outputs units with a specific name to indicate the type of the units. <i>Name</i> can be: MICRONS MILLIMETERS CENTIMETERS METERS KILOMETERS INCHES FEET MILES
-nounits	don't output units
-scale <i>scale</i>	scale the Inventor file by a specified amount
-xres <i>resolution</i>	Non-file textures sample to this X resolution
-yres <i>resolution</i>	Non-file textures sample to this Y resolution
-noinstances	does not convert AliasStudio instances to Inventor instances
Where:	
-i <infile>	specifies an Alias wire file to use as input. If not specified, input comes from stdin.
-o <outfile>	specifies an Inventor file to write output to. If it is not specified, the output is written to stdout.

Limitations

- The Inventor/VRML translator does not support annotation information.
- If you have extended **SoNodeKit** classes in your Inventor files, AliasStudio can only read these nodes if there is a Dynamically Shared Object (DSO), containing the extensions, in the path of `LD_LIBRARY_PATH`.

For more information, see *The Inventor Mentor: Programming Object Oriented 3D Graphics with Open Inventor*, Release 2, Chapter 11, "File Format," Page 17: *Reading in Extender Nodes and Engines*.

The rest of this section lists the Inventor nodes that are partially supported or not supported by the Inventor import and export translators (IvToAI and AItoIv).

- The analytical geometry nodes `SoCube` and `SoCubeDetail` are not supported directly. However, an Inventor file containing these nodes can be imported if optimization is on.
- These text and camera nodes are not supported: `SoAsciiText`, `SoText2`, `SoText3`, `SoTextDetail`, `SoLabel`, `SoFont`, `SoFontStyle`, `SoDrawStyle`, `SoOrthographicCamera`, `SoTranslation` and all their subclasses.
- These animation nodes are not supported: `SoSensor` and `SoEngine` and their subclasses.
- These nodes are implemented as `SoGroup` and do not fully support the concept of level of detail: `SoLOD` and `SoLevelOfDetail`.

AItoObj

Platforms

Windows

Purpose

Converts Alias wire files to OBJ files.

Description

```
AItoObj [-h] -i <infile> [-o <outfile>] [-h]
```

-h	displays help information, then exits.
infile	name of input Alias wire file;. If this is not specified, standard input (generally the keyboard) is used.
outfile	name of output OBJ format file; this must be specified.
-c	convert faces to curves; default is trimmed surfaces
-g	do not create a group for each geometric object

-u	<units> where 1 is inches 2 is millimeters 3 is feet 4 is miles 5 is meters 6 is kilometers 7 is millimeters 8 is microns 9 is centimeters 10 is microinches
-s	<scale> geometry scale factor
-t	<tol> tolerance value for tessellation

AIToSt

Platforms

Windows

Purpose

Converts Alias wire files to ISO10303 format, specifically these:

- application protocols ISO10303-203, or Configuration Controlled Design, conformance classes 1 to 4, and
- ISO10303-214, or Core Data for Automotive Mechanical Design Process, conformance classes 1 and 2.

The import and export of this data is supported via ISO10303-21 Physical File exchange. You might need a separate licence to run this utility.

Application protocol support

The geometric data formats in ISO10303-203 and ISO10303-214 are identical. This comprises the core of the implementation of the translator. The following table shows the mappings made.

Step Entity	AliasStudio Entity
Cartesian Point	Point
Line	B-Spline Curve
Circle	B-Spline Curve
Ellipse	B-Spline Curve
Parabola	B-Spline Curve
Hyperbola	B-Spline Curve

Step Entity	AliasStudio Entity
PolyLine	B-Spline Curve
Composite Curve	B-Spline Curve (Grouped)
Trimmed Curve	B-Spline Curve
B-Spline Curve	B-Spline Curve
Plane	B-Spline Surface
Cylindrical Surface	B-Spline Surface
Conical Surface	B-Spline Surface
Spherical Surface	B-Spline Surface
Toroidal Surface	B-Spline Surface
Surface of Linear Extrusion	B-Spline Surface
Surface of Revolution	B-Spline Surface
B-Spline Surface	B-Spline Surface
Rectangular Trimmed Surface	Trimmed Surface
Curve Bounded Surface	Trimmed Surface
Offset Surface	B-Spline Surface
Manifold Solid Brep	Shell (Closed)
Shell Based Surface Model	Shell (Open/Closed)

STEP logfile

When retrieval `finishes` without errors, this message is displayed:

```
STEP files retrieved successfully.
```

If there were errors in the retrieval, you see:

```
Problem Reading Step File, refer to log file for details.
```

The STEP logfile contain an error message for each problem entity that is encountered. Each error contains the STEP entity ID and entity type.

> Description

AIToSt [<options>] -i<infile> -o<outfile>	
-p	specifies which AP to output: either 203 or 214. The default is AP214
-m <i>value</i>	specifies which model representation to output. <i>Value</i> can be one of: <ul style="list-style-type: none">• 1 = wireframes• 2 = surface models• 3 = manifold surface models (shells)• 5 = brep solids• 6 = hybrid models (the default)
-t <i>value</i>	specifies the type of trimming that is done. This option is valid for surface models only. <i>Value</i> can be one of: <ul style="list-style-type: none">• 1 = parameter space trimming (the default)• 2= world space trimming
-g <i>value</i>	specifies the type of geometry to output. This option is valid only for shells. <i>Value</i> can be one of: <ul style="list-style-type: none">• 1 = shells only (the default)• 2 = all geometry
-c	outputs presentation data. This option is valid only for AP214 files.
-l	outputs layer data. This option is valid only for AP214 files.
-h	Displays help on usage
Where:	
<infile>	specifies an AliasStudio wire file to use as input.
<outfile>	specifies the STEP file to write output to.

AIToUG, AIToUG19, AIToUG20, AIToUG21, ALToUG22

AIToUG

Purpose

Convert AliasStudio wire files to Unigraphics files.

- AIToUG detects the required output format.
- AIToUG19, AIToUG20, AIToUG21, and AIToUG22 convert AliasStudio wire files to the respective Unigraphics format (NX1, NX2, NX3, or NX4).

Description

AIToUG [options] <infile> <outfile>	
infile	The name of the input AliasStudio wire file.
outfile	The name of the output Unigraphics file.
Options:	
-i	Input Wire file.
-o	Output Unigraphics file.
-u	Units to be output 1 = millimeters 2 = inches.
-b	Ignore surface continuity breaks (multiknots). Default is to split.
-g	Convert categories. Default is not to convert.
-c	Support collaboration mode.
-d<tol>	Unigraphics distance value to use in a Unigraphics part.
-f	Force to overwrite an existing outfile.
-e	Echo logfile to console.
-l	Create extended logfile.

BakeOrientation

Bake Alias Wire file orientation. This utility zeroes transforms, swaps UV's and deletes history as necessary.

Option	Value
-i	input Alias Wire filename; if not specified, <i>stdin</i> is used.
-o	output Alias Wire filename; must be specified.
-h	display this help information, then exit.

bsdl

Platforms

Windows.

Overview

This command-line utility extracts ASCII SDL from a binary SDL file, and inserts ASCII SDL back into a binary SDL file. It is useful when you need to hand-edit scene description language files.

Binary files must be used in AliasStudio 12 for rendering, so any file that is converted to ASCII must be converted back to binary format before rendering.

Description

```
bsdl extract [-f] <text> <binary>
```

Extract ASCII SDL to <text> from a binary SDL file called <binary>. The file <text> will not be overwritten unless the option -f is specified.

```
bsdl replace <text> <binary>
```

Insert the ASCII SDL file <text> into the binary SDL file <binary>, replacing the ASCII SDL data.

To see the help statement for the utility, type

```
bsdl -h
```



On Windows operating systems, this stand-alone utility is called bsd.exe.

CaToAI

Platforms

Windows

Purpose

Translates CAI files to Alias wire file format. CAI is CATIA/Alias Interoperability format.

Description

The usage statement for CaToAI is as follows:

```
CaToAI
```

```
CaToAI [ options ] [-i<input Wire file>] -o<output CAI file>
```

-h	displays help information, then exits
----	---------------------------------------

-b	displays a log file on screen during execution
----	--

-x	produces an extended log file
----	-------------------------------

-i	name of input CAI file; must be specified
-o	output CAI filename; if not specified. stdout is used.

C5ToAI

Alias provides a stand-alone utility, C5ToAI, to enable you to convert CATIA Part (.CATPart) and CATIA product (.CATProduct) document files to Alias Wire files without running the AliasStudio application. You may find this useful for creating a batch script if you have many files that need to be translated from the CATIA format.

Usage:

```
C5ToAI -i <CATIA V5 file> -o <Wire file> [<options>]
```

where

Argument	Value
-i	input CATIA filename (either .CATPart or .CATProduct)
-o	output Wire filename (.wire)
-l	logfile option: 0=No logfile output (default) 1=Output a logfile (-l1) 2=Output an extended logfile (-l2)
-a	import geometrical sets and hybrid bodies as layers
-wantInf bool	Want the infinite (unbounded) geometry imported. Default is false
-wantInv bool	Want the invisible (noshow) geometry imported. Default is false
-wantUnstitch bool	Want solids and open sheet bodies as surfaces. Default is false

Using stand-alone utilities on Windows

- 1 Choose **Programs > Accessories > Command Prompt** from the Start menu.
- 2 Click the cursor in the window.

You can see what directory you are in by typing

```
dir
```

A list of the directory contents will be provided; at the top of the list is your current location.

- 3 Navigate to the directory containing the files to be converted. If you have installed AliasStudio in its default location, the path will be similar to
C:\aw\[UserID]\Alias\user_data\[Project]\wire
where [UserID] is replaced by your login userid (and no square brackets) and [Project] is replaced by the name you have assigned to the project: the default project is demo (again, no square brackets). So to navigate to the directory, type

```
cd C:\aw\{UserID}\Alias\user_data\[Project]\wire
```

- 4 To use the stand-alone utility to convert a CATIA file named “headlight.CATPart” to an AliasStudio wirefile named “headlight.wire” without autostitching, and with a logfile created, type

```
C5ToAI -i headlight.CATPart -o headlight.wire
```

DWGToAI

DWG to Alias Wire file translator (using DirectConnect).

Argument	Description
-i	input DWG/DXF file
-o	output Wire file

FieldAssembler

Platforms

Windows

Purpose

Interlaces scan-lines of images rendered on fields into a single interlaced image; also pastes together incompletely rendered images.

Description

The FieldAssembler utility interlaces field-rendered images. Field-rendered images are created in a sequence for video playback. Television draws every other line on the screen and then fills in the alternate lines. Each field is one-half of a frame (the even lines or the odd lines).

NTSC and PAL video systems both use interlaced fields.

- NTSC video systems, used in North America, display 30 frames or 60 fields per second.
- PAL video systems, used in Europe, display 25 frames or 50 fields per second.

Each field is stored in a separate file. For animations that will be played back on television, you can render images into fields. However, to view them on a computer screen or to transfer them to certain other computer programs, you need to reassemble, or interlace, them into full images. The result is half as many files that each have twice as much data.

For information on FieldAssembler, from the FieldAssembler window select **Help > Help**.



If you will be using Composer to composite images rendered as fields, you must either interlace the fields together before importing them into Composer, or render the images in Alias PIX or RLA format. You can only import fields into Composer if they are in Alias PIX or RLA format. (In addition, RLA format fields must be named name.1, name.2, name.3, name.4, and so on, not name.1o, name.1e, name.2o, name.2e, and so on.)

from100to97

Platforms

Windows

Purpose

The standalone utility `from100to97` lets you convert a wire file from AliasStudio 10 or later to a AliasStudio 9.7 wire file.

Description

The following table outlines the details of this function:

```
from2008to97 -s [saved_file] [retrieved_file]
```

-s	saves the result
saved_file	is the name of the new, V9.7 file
retrieved_file	is the name of the old, V10 file
-w	overwrite savFile if it exists



The `from100to97` utility will convert fillets created with the new **Fillet surface** tool and save only the surface information. It will *not* save construction history, so you will not be able to modify the fillet after conversion. See New Surface Fillet tool section in the What's New manual.

The `from100to97` utility will convert round surfaces created with the **Round** tool and save only the surface information. It will *not* save construction history, so you will not be able to modify the round after conversion.

> Limitations

- Object editor commands are discarded (because they were not available in Version 9.7)

- Offset commands are discarded (because they were not available in Version 9.7)
- Fillet surface commands are discarded. Because of the great number of improvements to fillets that were made in Version 10, these commands aren't readable in previous versions.
- Because sketching is only available on the Windows platforms, any sketch data that is contained in an AliasStudio 10 file is only converted to Version 9.7 if the from100to97 utility is run on a Windows platform. If run on any other platform, this data is removed.

To use the from100to95 utility on windows

- 1 Choose Start Menu -> Programs -> Accessories and open the Command Prompt window



In this example, we are assuming that AliasStudio is installed in the default location, and that the wire file directory is also in the default location. Let's say your wirefile is called "bunchaspheres.wire"

- 2 Type in the Command Prompt window:

```
cd C:\Program Files\Alias\StudioTools10.0\bin
and press Enter.
```

This puts you in the directory where from100to97.exe is located.

- 3 If your original wire file is called "bunchaspheres.wire", you need to choose a new file name for the 9.7 file. Let's call it "bunchaOLDSpheres.wire"

You'll also need to use your full path to where the wire file is located; in this case, let's assume it's

```
C:\aw\username\Alias\user_data\wire\
```

- 4 To convert the file to 9.7, type:

```
from100to97 -s
C:\aw\username\Alias\user_data\wire\bunchaOLDSpheres.wire
C:\aw\username\Alias\user_data\wire\bunchaspheres.wire
```

To use the from100to95 utility on UNIX

In this example, we are assuming that AliasStudio is installed in the default location, and that the wire file directory is also in the default location. Let's say your wirefile is called "bunchaspheres.wire"

- 1 Open a shell and type:

```
cd /usr/aw/alias10.0/bin
and press Enter.
```

This puts you in the directory where from100to97.exe is located.

- 2 If your original wire file is called "bunchaspheres.wire", you need to choose a new file name for the 9.7 file. Let's call it "bunchaOLDSpheres.wire"

You'll also need to use your full path to where the wire file is located; in this case, let's assume it's

```
/aw/username/Alias/user_data/wire/
```

- 3 To convert the file to 9.7, type:

```
from100to97 -s
C:\aw\username\Alias\user_data\wire\bunchaOLDspheres.wire
C:\aw\username\Alias\user_data\wire\bunchaspheres.wire
```

GrToAI

PTC file to Alias Wire file translator (using DirectConnect)

Argument	Description
-i	input ProE/Granite file (.prt, osm.g)
-o	output Wire file

hp_glplotf, hp_gl2plotf

Platforms

Windows

Purpose

hp_glplotf translates Alias plot files into HP-GL (IBM GL) format files that are suitable for plotting on any supported plotter.

hp_gl2plotf creates similar output, but includes initialization commands required for HP-GL2 plotters.

Overview

hp_glplotf and hp_gl2plotf are filters which take input from stdin or <in_file> and send output to stdout or <out_file>, which would normally be sent to lp.

Description

```
hp_glplotf [-a] [-c<x>[<y>]] [-h] [-i] [-m<model>] [-n] [-p<paper>] [-r<angle>] [-s] [t # #] [<in_file>]
[<out_file>]]
```

-a	specifies automatic feed is to be used.
----	---

-c<x>[<y>]	specifies scale correction factors.
------------	-------------------------------------

-h	displays the on-line help.
----	----------------------------

-i	displays version information.
----	-------------------------------

-m<model>	specifies the plotter model type.
-n	separates HPGL plot commands on new lines.
-p<paper>	specifies the paper type.
-r<angle>	specifies the force plot rotation angle.
-s	sizes the plot produced automatically. The plotter's default size is used.
-t # #	specifies the text scaling where # and # are the width and height.
<in_file>	reads input from the file <in_file>. The default is to read from stdin.
<out_file>	writes output to the file <out_file>. The default is to write to stdout.



Any errors encountered by the HP-GL and IBM-GL device drivers are written to a file `/tmp/plot_err.<number>` as well as `/usr/adm/syslog`. Look there for any clues as to why the plotter is not working. The `<number>` is determined by the operating system.

> Automatic page advance

Some plotters can be fitted with a spool feed or have an automatic page advance option. This also applies to printers like the IBM-4019 and IBM-4029, which can automatically advance the page. To tell the plotter driver that you are using the spool feed option the following option should be specified:

`-fa`

> Correction factor

When large plots are plotted, some inaccuracies can creep in over large distances. As an interim fix, it is possible to compensate for this error by specifying x and y correction factors. The default x and y correction factor is 1.0, which is no correction at all. The correction factors are used internally as multipliers.

> Example

If you plotted something that should have come out 30" and it came out to 30.125", the plot was 0.125" too large. Assuming that the same error occurs in both the x and y directions, you would need a correction factor of $30/30.125 = 0.9959$. The following should compensate for the error:

`-fc0.9959`



Setting the correction values in the **Plotting Interface** window opened by **File > Plot**, overrides the correction values in the **LP queue options** for plot output section of the **Alias Preferences** window.

IGESToAI

IGES file to Alias Wire file translator (using DirectConnect).

Platforms:

Windows

Description:

```
IgesToAI -i <IGES file> -o <Wire file> [<options>]
```

Argument	Value
-i	input IGES (.igs, iges) file
-o	output Wire (.wire) file
-u	integer
-useTrimmingCurves	integer. specifies the trim curves that the translator will use: 0: use 3D model space curves 1: use 2D parameter space curves 2: use the preference flag in the IGES file By default, 3D model space curves are used.
-s (boolean)	True or False
-shrinkSurfaces	specifies if base surfaces are to be shrunk to trim regions or not. By default, shrink surfaces.

imgcvt

Platforms

Windows

Purpose

The `Imgcvt` utility converts images or a sequence of images from one image format to another.

Description

In any shell window, enter `imgcvt` followed by the name of the image that you want to convert. The input and output image formats are usually determined by the filename extension or image content. However, when the input or output image has no extension, or cannot be identified, the `-f` and `-t` options can be used to indicate the desired formats. You can use the following options.

```
imgcvt <options> input_image output_image
```

> Example 1

```
imgcvt -f sgi -t tiff input_image output_image
```

This line converts the image named `input_image` that is in the Silicon Graphics format to the TIFF format and saves the image out to a different name, `output_image`.

> **Example 2**

```
imgcvt -f tiff -t iff input_image output_image
```

This line converts a TIFF image to the IFF file format.

The following table lists the `imgcvt` options.

Command line options	Description
<code>-f input_image_extension</code>	Identifies the specific image format that you are converting from. For example, <code>-f sgi</code> means that only files matching the Silicon Graphics image format are accepted as input.
<code>-t output_image_extension</code>	Identifies the specific image format that you are converting to. For example <code>-t tiff</code> means that files will be converted to the TIFF image format only. For the list of possible extensions, see the section on <i>Using image filename extensions</i> .
<code>-n start end step</code>	Is the start, end and step of an input image sequence (only whole numbers are accepted).
<code>-N start end step</code>	Is the start, end and step of an output image sequence (only whole numbers are accepted).

Command line options	Description
-r range	<p>In an input image sequence, selects the range of images to be converted from. For example:</p> <pre>imgcvt -r 1-5 ... imgcvt -r 1-5x2 ... imgcvt -r 1-5,10-20x2,50 ...</pre> <p>A sequence specifier is a list of single frames or ranges separated by a comma or a slash. In this example, they are individual, numbered files. For each range, an optional step can be specified after an <i>x</i>.</p> <p>Sequences can be reversed and offsets can be added to the input or output frame number:</p> <pre>imgcvt -r 100-1 -R 1-100 in.#.rgb out.#.rla imgcvt -r 1-100 in.#.rgb out.#+100.rla</pre>
-R range	<p>In an output image sequence, selects the range of images to be converted to.</p>
-h	<p>Provides a brief description of each option.</p>
-s	<p>Silent mode.</p>
-v	<p>Verbose mode.</p>
-V pal/ntsc	<p>Video mode for yuv files.</p>
-C compress	<p>TIFF compression types: LZW (the default) or NONE.</p>
-q N	<p>JPEG quality (0...100; default is 75).</p>
-m	<p>Explore map format.</p>

When you are working on a sequence of images, the # and @ symbols refer to the current frame number:

- The hash (#) indicates a four-digit padded number,
- whereas the @ symbol indicates a non-padded number.

The following converts a sequence of 100 frames:

```
imgcvt -n 1 100 1 image_in.iff.@ image_out.#.rla
```

For more precise control of the frame number format, use either multiple @ symbols to set a specific padding or the standard printf(3S) notation:

```
imgcvt -n 1 100 1 image_in.@@@.rgb image_out.#.rla
imgcvt -n 1 100 1 image_in.%03d.rgb image_out.#.rla
imgcvt -n 1 100 1 image_in.%.2f.rgb image_out.#.rla
```

Using image filename extensions

Use the following filename extensions to convert images from one image format to another.

- In the `-f` option, you specify the extension of the image format that you are converting from.
- In the `-t` option, you specify the extension of the image format that you are converting to.

Image Format	Extension
Abekas NTSC or PAL	.yuv
Alias	.als
Explore	
GIF	.gif
JPEG	.jpg
Kodak Cineon	.cin
Lucas Film	.lff
Maya IFF	.iff
Pixibox PXB	.pxb
PXN	
PPM raw/ascii	.ppm
Prisms	.pri
Quantel	.qtl
SGI	.rgb, .sgi or .bw
Softimage	.pic
Targa RGB/BW	.tga
TIFF 6.0	.tif or .tiff
Vista	.vst
Wavefront RLA	.rla
Windows bitmap	.bmp

Platforms

Windows

Purpose

IvToAl translates SGI OpenInventor files into Alias wire files.

Description

IvToAl [<options>] [-i <infile> [-o <outfile>]]

-o	Do not optimize the Inventor file
-s <i>scale</i>	Use an input scale factor of <i>scale</i> (for example, 2.0)
-g	Do not group the geometry
<infile>	specifies an Inventor file to use as input. If infile is not specified, input comes from stdin.
<outfile>	specifies the Alias wire file to write output to. If outfile is not specified, output is written to stdout.

For information on this utility's limitations, see *AIToIv* (page 8).

Imgrd

When you invoke Imgrd, it looks for the license file which contains the information about vendors and features. Usage is:

```
Imgrd [ -app ] [ -c license_file ] [ -t timeout_interval ] [ -l logfile ]  
[ -s timestamp_interval ] [ -2 -p ] [ -v ] [ -x lmdown ] [ -x lmremove ]
```

Option	Purpose
-app	Required for Windows when run as a command, but not used when run as a service.
-c license_file	Use the license file specified by license_file.
-t timeout_interval	Sets a timeout interval, in seconds, during which redundant daemons must complete their connections to each other. The default value is 10 seconds. A larger value may be desirable if the daemons are run on busy systems or a very heavily loaded network.
-l logfile	Write the debug log to the specified logfile.

Option	Purpose
-s timestamp_interval	Specifies the logfile timestamp interval, in minutes. The default is 360 minutes.
-2 -p	Restricts usage of lmdown, lmread, and lmremove to a FLEXlm administrator who is by default the root account. If there is a Unix group called 'lmadmin', then use is restricted to members of that group. If root is not a member of that group, then root can not use any of the above utilities. The '-p' option is available in FLEXlm v2.4 and later.
-v	Prints lmgrd's version number and copyright, and then exits.
-x lmdown	Disallows the lmdown command (no user can run lmdown). If lmdown is disabled, you must stop lmgrd via 'kill pid' (Unix) or CTRL-ALT-DEL and stop the lmgrd and vendor daemon processes (Windows 95). On Unix, be sure that the kill command does not have a -9 argument. The -x lmdown option is available in FLEXlm v4.0 and later.
-x lmremove	Disallows the lmremove command (no user can run lmremove). The -x lmremove option is available in FLEXlm v4.0 and later.

lmutil

Platforms

Windows

Purpose

lmutil is the first among FLEXlm license manager utilities. FLEXlm is the network license manager used by Autodesk to control the use of their software products.

If you are a UNIX systems administrator or user, it is likely that one or more of the products on your network is licensed by FLEXlm. Details about installing with FLEXlm are in your Installation Notes.

The AliasStudio installation program creates a FLEXlm directory, which contains a FLEXlm control panel and its required DLL, plus a number of related utilities

- Imutil
- Imtools
- sgiawd (if you install the licensing component of the software)
- Imgrd.exe (if you install the licensing component of the software)

While you do not call all the utilities directly, they are needed and must not be deleted.

ObjToAl

Platforms

Windows

Purpose

Converts OBJ files to Alias wire files.

Description

```
ObjToAl -i <infile> [-o <outfile>] [-h]
```

infile	the name of the input OBJ file; this must be specified.
outfile	the name of the output Alias wire file. If this is not specified, standard output (generally the screen) is used.
-h	displays help information, then exits.

PRenderToAlias

Pro/Engineer Render (.slp) file to Alias Wire file converter.

Option	Value
-s scale	Input scale factor (e.g. -s2.0)
-u units	Input units [MI,FT,IN,MIL,UIN,KM,M,CM,MM,UM] (default = -uIN)
-g	Group the geometry
-r	Recalculate vertex normals
-n	Do not merge vertices
-p	Merge vertices according to xyz position only
-t pos_tol	Specify merge vertices position tolerance value in input units (default = 0.0001)
-a nrm_tol	Specify merge vertices normal tolerance value in degrees (default = 1 degree).
<infile>	Name of Pro/Engineer Render file. If <infile> is absent, input comes from stdin.
<outfile>	Name of Alias wire file to be created. If <outfile> is absent, output goes to stdout.

By default, all vertices are merged according to their positions and normals.

print_wire_header

Platforms

Windows

Purpose

print_wire_header prints the header and can be used to verify the named wire files.

Description

```
print_wire_header [-v] file [file...]
```

-v sets the verify mode. If **-v** is included, the entire wire file is read and if the trailer record is not found, the file is reported as invalid. This option indicates that a wire file is not truncated. It is useful for verifying the success of a file transfer operation, such as a copy to or from tape or across NFS.

file [file...] is a list of files to be processed. There must be at least one file listed. Wildcard listing of files is supported.

> Example

The following is an example of output produced using the command `print_wire_header -v *` in a directory.

```
Filename:                SkiBoot
Alias Product Name:      Alias Studio
Alias Product Version:   2008
Alias Software Date      2001/07/18 13:08
Alias File Version:      V2008-12
File Creation Date:      Thu Jul 29 17:09:25
                          2001
File Type:               WIRE
Coordinate System        Z Up
Graphics Display Size:   830 x 840
SkiBoot_strap.iges is not an Alias Wire File.
Warning: Wire file truncated.als is corrupted.
Trailer record is missing!
```

psplotf

Platforms

Windows

Purpose

Translates Alias plot files into PostScript for printing on any PostScript printer.

The plotter driver program `psplotf` is a filter that takes input from `stdin` or `<in_file>` and sends output to `stdout` or `<out_file>` instead of `lp`.

Description

`psplotf [-f fontname] [-h] [-i] [-r #] [-s #,#,#,#] [-t #] [-u] [-w #] [<in_file> [<out_file>]]`

<code>-f fontname</code>	specifies the font name to use.
<code>-h</code>	displays the on-line help.
<code>-i</code>	displays the version information.
<code>-r #</code>	specifies the rotation angle.
<code>-s #, #, #, #</code>	specifies the paper size. The # values represent x min, x max, y min, y max.
<code>-t #</code>	specifies the font size.
<code>-u</code>	specifies that the program generated initialization string is not to be used.
<code>-w #</code>	specifies the line width.
<code><in_file></code>	reads input from the file <code><in_file></code> . The default is to read from <code>stdin</code> .
<code><out_file></code>	writes output to the file <code><out_file></code> . The default is to write to <code>stdout</code> .

You can add more initialization commands to the file by defining the environment variables `ALIAS_PSLOT_INIT` and `ALIAS_PSLOT_INIT2`.

renderer/raytracer/powercaster/powertracer

Platforms

Windows

Purpose

The renderer creates raycast or raytraced image files from AliasStudio Scene Description Language files while in a command line. The AliasStudio render programs create image files as follows:

- `renderer`: raycasts images
- `powercaster`: multi-threaded version of the renderer images (purchaseable)
- `raytracer`: raytraces images
- `powertracer`: multi-threaded version of the raytracer images (purchaseable)

AliasStudio Renderer for Windows is a command-line tool for rendering Alias files. The rendered files produce a sequence of images that can be played back as an animation.

The functions powercaster and powertracer might not be present if they have not been purchased for your system.

Description

All four render programs use the same arguments. The usage statement for **renderer**, **raytracer**, **powercaster**, or **powertracer** is described below.

```
<command> [-a#] [-b#] [-B#] [-c <quantized_output_file>]
[-C <color_map_filename>] [-d <filename>] [-e#] [-E#]
[-f <script>] [-h#] [-H] [-J] [-k] [-K#] [-m <filename>] [-p <filename>] [-P] [-q#] [-Q#] [-r#]
[-R#] [-s#] [-S#] [-t#] [-T#] [-w#] [-W#] [-x#] [-y#] [-Y#] [<sdl_filename>]
```

<command>	is renderer or raytracer or powercaster or powertracer
-a#	sets the anti-aliasing level (aalevel) to the integer #. aalevel is the maximum anti-aliasing level per pixel.
-b#	sets the by frame number for animation sequences to the floating point number #.
-B#	sets the by extension for animation sequences to the integer #.
-c <quantized_output_file>	outputs the quantized image to the file <quantized_output_file> after each frame.
-C color_map_filename	uses the SGI image format file <color_map_filename> as the color map to refer to for quantizing after each frame.*
-d <filename>	uses <filename> as the depth filename.
-e#	sets the ending frame number for animation sequences to the floating point number #.
-E#	sets the size extension for animation sequences to the integer # where # indicates the number of 0 padding before the extension number. For example, -E 4 produces file extensions such as <file>.0001 indicating frame 1.
-f <script>	invokes the program <script> after each frame.
-h#	sets the image height for the partial image to be rendered to the integer # without changing the view port. However, this sub-region to be rendered always originates from the lower left hand corner of the image. The integer # moves the origin of this window around the view port.
-H	displays the on-line help.
-J	creates a depth file called 'timing' representing time per pixel.
-k	keeps depth maps in memory after reading them once.**
-K#	turns depth maps on disk usage to #. 0 is OFF. Any number other than zero is ON. ***
-m <filename>	produces a matte file and uses <filename> as the filename.
-n#	sets, to the integer #, the number of processors to render on. This option is only available with powertracer and powercaster.

-p <filename>	uses <filename> as the pix filename.
-P	preserves the non-glowed image unless DOF or Quantize are on. This option allows you to save both a glowed and non-glowed image on disk. The non-glowed image will have the same name as the glowed image, but will have the suffix .ng.
-q#	sets the quiet flag to #. # can be 0, 1, or 2.
-Q#	sets the resolution in the X direction and the view port to the integer #. This option is useful for overriding the resolution specified in a given SDL file. For example, it is useful for switching between rendering NTSC and 1/4 NTSC for a quick preview render.
-r#	sets the aspect ratio to the floating point number #.
-R#	sets the resolution in the Y direction and the view port to the integer #. This option is useful for overriding the resolution specified in a given SDL file. For example, it is useful for switching between rendering NTSC and 1/4 NTSC for a quick preview render.
-s#	sets the starting frame number for animation sequences to the floating point number #.
-S#	sets the start extension for animation sequences to the integer #.
-t#	sets the aathreshold to the integer #. aathreshold is the anti-aliasing threshold value that adaptively super-samples pixels based on color difference. The higher the value, the more sensitive the super-sampling is to color difference.
-T#	sets the number of Y pixels in a tile to the integer #. A tile is a row of pixels to be rendered together. The main reason to use this option is to reduce the amount of memory used by lowering the Y value. This controls the tile size the image is broken up into for rendering and has no effect on the final image or its resolution.
-v	render normally outside of viewport region
-V	render image with hidden lines
-w#	sets the image width for the partial image to be rendered to the integer # without changing the view port. However, this sub-region to be rendered always originates from the lower left hand corner of the image. The integer # moves the origin of this window around the view port.
-W#	sets the ylow for backgrounds to #. The ylow and yhigh define the region of the rendering, specified in pixels, where the background should appear.
-x#	sets the xleft to the integer #. xleft is the left corner of the partial image to be rendered.
-y#	sets the ylow to the integer #. ylow is the left corner of the partial image to be rendered.

`-Y#` sets the yhigh for backgrounds to #. The ylow and yhigh define the region of the rendering, specified in pixels, where the background should appear.

`sdl_filename` sets the SDL filename to a specific filename. If no filename is specified, standard input is used.

* If required, the renderer quantizes images after rendering them. The option `-C` allows a previously generated color table to be used for quantizing images. However, `aquant` quantizes images much more quickly.

** Generally, when a `depth_input` file is specified for a shadowing spotlight, it is read every frame. `-k` forces the renderer to read the shadow map only once during the first frame of rendering.

*** Rather than editing your SDL files to add `depth_input` or `depth_output` commands to shadowing spotlights, this command line option may be used. When set to a non-zero value (for example `-K 1`), the renderer automatically creates depth output files named after the spotlights in your current directory, or if these files already exist, they are used.

Using any of the above options overrides any equivalent SDL keyword settings in the SDL file.

> Examples

The following are examples showing the uses of the rendering utilities.

This command:

```
renderer testframe.sdl
```

renders an SDL file called `testframe.sdl`. All parameters and keywords set within the SDL file will be used. If any of the keywords is missing and requires a value, the default value for that parameter is used.

This command:

```
renderer -a0 -s1 -b2 -e20 -p testpix -q0 -h512 -w512 -x0 -y0  
scene.sdl
```

renders an SDL file called `scene.sdl`. The scene to be rendered is an animation.

Anti-aliasing is turned OFF by setting the `aalevelmax` to 0. The animation specified in the SDL file is overridden by the specification of `-s1 -b2 -e20`, which renders with a frame step of 2 the first twenty frames of animation. Normal messages are output, and the test image size is 512x512 pixels square. The animation sequence is output to a series of files starting with the name `testpix`.

This command:

```
renderer -s 1.5 -e 3.5 -b .25 -S 1 -B 2 -E 3 sdl/foo
```

produces the following:

```
foo.001 (a snap shot of the animation at time 1.5)  
foo.003 (a snap shot of the animation at time 1.75)  
foo.005 (a snap shot of the animation at time 2.0)  
...  
foo.019 (a snap shot of the animation at time 3.25)  
foo.021 (a snap shot of the animation at time 3.5)
```

Assuming a 512x512 image, the following command:

```
renderer -h 255 -w 255 -x 255 -y 255 sdl/foo
```

produces an image of the top right 255 x 255 pixel region of the original image.

> **Important notes**

Saved geometry is not compatible with motion blur.

You cannot stop and start a render process that uses saved geometry and retain the saved geometry.

Saved geometry uses a significant amount of memory. You should have at least 100 megabytes of swap space available before attempting to use it.

Alias backgrounds interpolate between the top and bottom of the current rendering pixel span in Y. If you are rendering only a sub-region, this might not be what you intend, because the background is repeated many times in Y. Therefore, use the options `-W` and `-Y` to specify a final resolution from which AliasStudio can calculate background positioning.

File formats

SDL files are binary, and are compatible across platforms. To edit an SDL file, you must first convert it to text using the standalone utility *bsdl* (page 15). The TIFF file format is an image display format that is also compatible across platforms. All other image formats are automatically read and saved in SGI image format.

You must uncompress any texture-map files before you render.

StToAI

Platforms

Windows

Purpose

Converts ISO10303 files to Alias wire files, specifically:

- application protocols ISO10303-203, or Configuration Controlled Design, conformance classes 1 to 4, and
- ISO10303-214, or Core Data for Automotive Mechanical Design Process, conformance classes 1 and 2.

The import and export of this data is supported via ISO10303-21 Physical File exchange.

Description

StToAI [-s] <infile> <outfile>

<infile> specifies the STEP file to use as input. If not specified, input comes from stdin.

<outfile> specifies an Alias wire file to write output to. If not specified, output goes to stdout.

SwToAI

PTC SolidWorks file to Alias Wire file translator (using DirectConnect).

Option	Value
-i input	Solidworks part or assembly file
-o	output Wire file

UGToAI, UGToAI19, UGToAI20, UGToAI21, UGToAI22

UGToAI

Platforms

- UGToAI19, UGToAI20, UGToAI21, UGToAI22: Use directly for the Windows platform.

Purpose

Convert Unigraphics files to AliasStudio wire files.

Description

UGToAI [options] <infile> <outfile>

(For Windows, substitute UGToAI with UGToAI19, UGToAI20, or UGToAI21.)

infile	The name of the input Unigraphics file.
--------	---

outfile	The name of the output AliasStudio wire file.
---------	---

Options:

-i	Input Unigraphics file.
----	-------------------------

-o	Output Unigraphics file.
----	--------------------------

-s	Do not stitch model on input. Default is to stitch.
----	---

-e	Echo logfile to console.
-l	Create extended logfile.
-g	Convert categories. Default is not to convert.

Index

A

- Abekas 24
- Alias batch file translator 5
- Alias plot files 29
- Alias to DWG format 7
- Alias to Granite3 file format 7
- AliasBatch 5
- AIToC5 6
- AIToCa 6
- AIToDWG 7
- AIToGR3 7
- AIToIv 8
- AIToObj 10
- AIToSt 12
- AIToUG 13
- AIToUG utility 13
- ASCII SDL 14

B

- bake normal orientation 14
- BakeOrientation 14
- binary SDL 14
- bsdl 14

C

- C5ToAI 15
- CAI format 6, 15
- CATIA Part (.CATPart)
format 6, 15
- CATIA product (.CATProduct)
document format 15
- CATIA/Alias Interoperability
format 6
- CaToAI 15
- command line programs 1

- command line utilities
 - on IRIX 5
 - on Windows 4
- converting
 - image formats 22

D

- DWG to Alias Wire format 16
- DWGTToAI 16

E

- Explore image format 24

F

- FieldAssembler 16
- FLEXIm 27
- from100to97 utility 17

G

- GIF files 24
- GrToAI 19

H

- hp_gl2plotf 20
- hp_gplotf 19
- HP-GL2 plotters 20

I

- image formats
 - conversion 22
- imgcvt 22
- interlaced images 16

- ISO10303 format 11, 34
- IvToAI 25

J

- JPEG files
 - extension 24

K

- Kodak Cineon 24

L

- license file 25
- Imgrd 25
- Imutil 27
- Lucas Film format 24

M

- Maya IFF 24

N

- normals
 - baking 14

O

- OBJ format 10, 27
- ObjToAI 27
- orientation
 - normals 14

P

- Pixibox PXB 24

PostScript 29
powercaster 30
powertracer 30
PPM raw/ascii 24
PRenderToAlias 28
print_wire_header 28
Prisms 24
Pro/Engineer Render (.slp) file
to Alias wire file 28
psplotf 30
PTC Granite file to Alias Wire
file 19
PTC SolidWorks file to Alias
wire file 35
PXN 24

Q

Quantel 24

R

raycaster 30
raytracer 30
renderer 30

S

setting normals
stand-alone utility 14
SGI 24
SGI Inventor format 8, 25
Softimage 24
StToAI 35
StudioTools 9.7 wire file,
converting 17
SwToAI 35

T

Targa 24
TIFF 24

U

UGToAI utility 35
Unigraphics format 13, 35
using command line utilities
on IRIX 5
on Windows 4

V

Vista 24

W

Wavefront RLA 24
Windows bitmap 24