

Electric Image

Copyright © 2001 Electric Image, Inc. All Rights Reserved

Warranty & License

Copyrights

Copyright ©1987-2001 Electric Image Inc. All rights reserved.

The ElectricImage Universe software and documentation are Copyright ©1987-2001 Electric Image Inc. All rights reserved. This document may not, in whole or in any part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable format without prior consent, in writing, from Electric Image Inc.

Trademarks

Electric Image Universe, Universe Modeler, Universe Animator and their logos are trademarks of Electric Image Inc. Windows is a trademark of Microsoft Corporation. Macintosh is a trademark of Apple Computer, Inc. Solaris is a trademark of Sun Microsystems. All other trademarks are the property of their respective holders.

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE INSTALLING THE ELECTRICIMAGE UNIVERSE SOFTWARE. THE OPENING OF THE SOFTWARE PACKAGES AND/OR INSTALLATION OF THE SOFTWARE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD RETURN THE ENTIRE PRODUCT IN NEW CONDITION TO THE PLACE OF PURCHASE, AND YOUR MONEY WILL BE REFUNDED.

Except as otherwise specifically provided, Electric Image Inc. is the “Licensor” of the ElectricImage Universe software and accompanying documentation. The ElectricImage Universe software program is referred to in this License Agreement as the “Program”. The Licensor grants to you, the original end user licensee, a nonexclusive license to use the Program for your own personal use and not for sublicense, subject to the terms and condi-

tions stated in this License Agreement. You assume responsibility for the selection of the Program to achieve your intended results.

LICENSE

You may: (i) use the Program on a single machine; (ii) copy the Program into any machine-readable form only for necessary backup or modification purposes in support of your use of the Program on a single machine; and (iii) transfer the Program and license to another party if the other party agrees to accept all terms and conditions of this Agreement. If you transfer the Program, You must at the same time either transfer all copies, whether in printed or machine-readable form, to the same party or destroy any copies not transferred. You may not sublicense or lease the Program or accompanying documentation. You may not assign or transfer the license to the Program or accompanying documentation except as expressly provided in this Agreement. Except as expressly permitted above, any attempt to lease, sublicense, assign or transfer any of the rights, duties or obligations hereunder is void.

Upgrades and Updates

If this Software is being licensed to you as an upgrade or update to software previously licensed to you, you must destroy the software previously licensed to you, including all copies resident on your hard-disk drive, within sixty (60) days of the purchase of the license to use the upgrade or update.

RESTRICTIONS

You may not distribute copies of the Program to others or electronically transfer the Program or accompanying documentation from one computer to another over a network. The Program contains trade secrets, and in order to protect them, You may not decompile, reverse engineer or disassemble the Program, or otherwise reduce the Program to a human perceivable form. You may not modify, adapt, translate, rent, lease, loan, resell for profit, distribute, network or create derivative works based on all or any

part of the Program or accompanying documentation. You may not circumvent operation of the accompanying hardware key device or “dongle” or any other form of copy protection employed by the Program. You may not copy, clone or emulate or otherwise duplicate the functions of the hardware key in any manner, or allow the Program or hardware key to be so copied.

Disclaimer

Computer-Aided Design Software and other technical software are not substitutes for your professional judgement. Computer-Aided Design Software and other technical software are intended to assist with product design and are not substitutes for independent testing of product stress, safety, and utility. due to the large variety of potential applications for the software, the software has not been tested in all situations under which it may be used. Electric Image Inc. shall not be liable in any manner whatsoever for the results obtained through the use of the software. Persons using the software are responsible for the supervision, management, and control of the software. This responsibility includes, but is not limited to, the determination of appropriate uses for the software and the selection of the software and other programs to achieve intended results. Persons using the software are also responsible for establishing the adequacy of independent procedures for testing the reliability and accuracy of any program output, including all items designed by using the software.

TITLE

The original and any copies of the Program and accompanying documentation, in whole and in part, including translations, compilations, partial copies, modifications, and updates are the property of Licensor or its suppliers. You have only the limited rights granted by this license. You are not an

owner of a copy of the Program, and therefore 17 U.S.C. Section 117 does not apply. You must reproduce and include the proprietary rights notices on any copy of the Program and accompanying documentation.

LIMITATIONS

This license does not include any rights to reproduce, copy, commercialize or otherwise use or make use of the sounds, music, pictures, effects or other content contained in the Program except in connection with your personal use of the Program as permitted hereunder.

TERMINATION

You may terminate your license to the Program at any time by destroying the Program and accompanying documentation together with all copies, modifications and merged portions in any form. The license also will terminate as set forth elsewhere in this Agreement if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the Program and accompanying documentation, together with all copies, modifications and merged portions, in any form.

LIMITED WARRANTY.

Electric Image Inc. ("Licensor") warrants to the purchaser of ElectricImage Universe from an authorized dealer ("End User") that the CD-ROM(s) media on which the ElectricImage Universe software program ("Program") are furnished are free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery to End User as evidenced by a copy of End User's receipt ("Program Warranty"). Licensor does not warrant that the documentation or the functions contained in the Program will meet End User's requirements or that the operation of the Program will be uninterrupted or error free. Licensor's entire liability and End User's exclusive remedy with respect to breach of the Program Warranty will be the replacement of any CD-ROM media not meeting the Program Warranty if returned to the Licensor for the particular nonconforming

Program within the Program's Warranty period with a copy of the applicable receipt(s). End User is responsible for any costs, expenses, customs, duties, taxes, and other similar charges arising with regard to return shipment of the Program to the place of purchase and Licensor's return of the Program under the Software Warranty. End user must pay all such costs upon invoice.

NO OTHER WARRANTY.

EXCEPT FOR THE LIMITED WARRANTY STATED ABOVE, THE ELECTRICIMAGE UNIVERSE PROGRAM AND ACCOMPANYING DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OR CONDITION OF ANY KIND, EITHER EXPRESSED OR IMPLIED AND THE IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, TITLE AND FITNESS FOR A PARTICULAR PURPOSE ARE SPECIFICALLY DISCLAIMED. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE ELECTRICIMAGE UNIVERSE PROGRAM AND ACCOMPANYING DOCUMENTATION IS WITH END USER. SHOULD THE PROGRAM PROVE DEFECTIVE, END USER (AND NOT THE LICENSOR OR ANY AUTHORIZED DEALER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. No oral or written information or advice given by Licensor, its dealers, distributors, agents or employees will create a warranty or in any way increase the scope of the limited warranties set forth herein, and End User may not rely on any such information or advice. SOME STATES MAY NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY. THIS WARRANTY PROVIDES SPECIFIC LEGAL RIGHTS AND THE END USER MAY ALSO HAVE OTHER RIGHTS THAT VARY FROM STATE TO STATE.

Remedies

The entire liability of Electric Image Inc. and your exclusive remedy under the warranty provided herein will be, at the option of Electric Image Inc. to attempt to correct or work around errors, to replace the media, or to refund the purchase price and terminate this Agreement. This remedy is subject to return of the Software to Electric Image Inc. or to the Authorized Electric Image Inc. Dealer from whom it was obtained with a copy of your receipt.

LIMITATION OF LIABILITY.

IN NO EVENT WILL LICENSOR OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION OR DELIVERY OF THE PROGRAM, OR ANY OR ALL OF THE ELECTRICIMAGE UNIVERSE CONTENT BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM, ACCOMPANYING DOCUMENTATION OR CONTENT, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. The foregoing limitations of liability shall apply notwithstanding any failure of essential purpose of any limited remedy or any claim by any other party. Licensor's liability to the End User or any third party arising out of or related to this Agreement however caused and on any theory of liability, whether in contract, tort (including negligence), or otherwise will not exceed the fee paid by the End User for the Program, as applicable. The Program was not developed and is not licensed for use in any nuclear, aviation, mass transit, life support or medical application, or any other inherently dangerous applications. SOME STATES MAY NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO ALL OR A PORTION OF THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY.

General

This Agreement shall not be governed by the UN Convention on Contracts for the Sale of Goods; rather this Agreement shall be governed by the laws of the State of California, including its Uniform Commercial Code without reference to conflict-of-laws principles. This Agreement is the entire agreement between us and supersedes any other communications or advertising with respect to the Software and Documentation. If you have any questions, please contact in writing: Electric Image Inc., Customer Service, 3 Imperial Promenade, Suite 400, Santa Ana, California, 92707.

Table of Contents

Chapter 1 Constraints	1
Introduction	1
Applying Constraints	2
Constraint Types	4
Circular Dependencies	9
 Chapter 2 Inverse Kinematics	 11
Introduction	11
Creating an IK Handle	12
IK Handle Info Window	13
X-Form Tab	14
Properties Tab	15
Chain Tab	19
Using IK on Models	19
 Chapter 3 Skin and Bones	 21
Introduction	21
Bones	22
Binding the Skin	24
The Skin Editor	25
Strength Maps	27
Painting	30
 Chapter 4 Illuminators	 33
Introduction	33
Illuminator Light Info Window	34
Using Illuminators	36

Chapter 5 Blur Preview.....	37
Introduction.....	37
Previewing the Scene.....	37
Chapter 6 Animator Extras.....	39
Cubic Reflection Maps.....	39
Dragging Behavior.....	39
Drawing Performance.....	40
Draft Mode.....	40
Linking to Animated Groups.....	41
Tool Tips.....	41
OS X.....	42
Setting Camera's Memory Allocation.....	42
Scrubbing.....	43
Show Illumination.....	43
Shortcut Keys.....	44
Chapter 7 Modeler 4.0.....	45
Introduction.....	45
Shelling.....	46
Boundary-Constrained Surface Rebuilding.....	48
Example.....	49
Skinning and Surface Blending to a Point.....	51
Enhancements.....	52
Image Templates and Background Images.....	53
Interruptible Operations.....	55
Layer View Enhancements.....	55
Mac OS X.....	56
Index.....	57

Constraints

1.0 Introduction

The constraint system in EIU allows you to animate position, rotation, scale, or the pole-vector (IK handles) using the position, rotation, or scale of other objects. The object that is constrained is called the “constraint object”. The objects doing the constraining are called, “targets”. One constraint object may be constrained by many targets. The influence of each target is controlled by a floating-point value called, “weight”. This value is used to compute a weighted average of the desired constraint. This final value is called the “target point”. If all targets have equal weight, the target point is a simple average between them. A higher weight value relative to the other weights increases the influence of a target, while a lower weight value relative to the other weights decreases the influence of a target. A value of zero effectively removes the target's influence from the constraint. All weight values can be animated, which allows you to change the constraint over time, effectively creating animations without changing the position, rotation, or scale channels of the constraint object.

The new constraint system replaces the Look-At and Auto-Bank features that were included in the older versions of the ElectricImage™ Animation System. Unlike the older method, which required pre-computation of Auto-Frames, the new system is real-time and interactive. For example, if your camera is constrained to look at an object, you will see the camera rotate in the View windows as you drag the object around the scene.

1.1 Applying Constraints

Constraints are added to an object via the Constraint menu. Select the Constraint Object and then choose one of the available constraint types from the Constraint menu. A dialog box will come up letting you know that you are about to replace any animation data with the constraint. Click away this reminder and you are then prompted to choose the target(s). Targets are selected by clicking on them in the Project or View windows. The selection process is terminated by hitting the Escape key or Command-. (Macintosh). If the constraint type is inappropriate for the object, that type will be grayed out in the menu. If the constraint conflicts with another constraint, for example, aim conflicts with either Auto Look or Rotate, you will be notified and no constraint will be added.

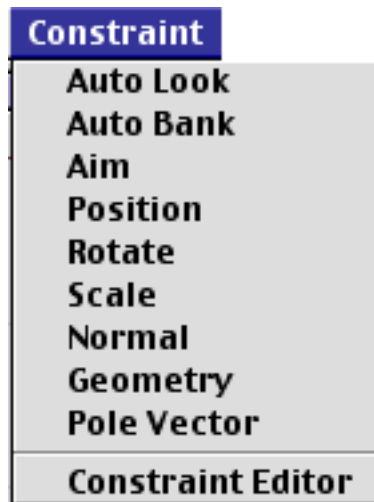


Figure 1.0 — Constraint Menu

Once constraints have been applied to an object, the Constraint Editor dialog box may be brought up by first selecting the constraint object and then choosing Constraint Editor from the Constraint menu. The dialog has a “Type” pull-down menu with a list of constraint types that have been applied to the object. Each type has an associated list of targets in the “Target” area of the dialog box. All the constraints allow you to change the Weight of each target (select the target from the list and then change the Weight value) and to delete the target. You may also disable the constraint engine for the constraint type by turning off the Enable Constraint Engine check-box.

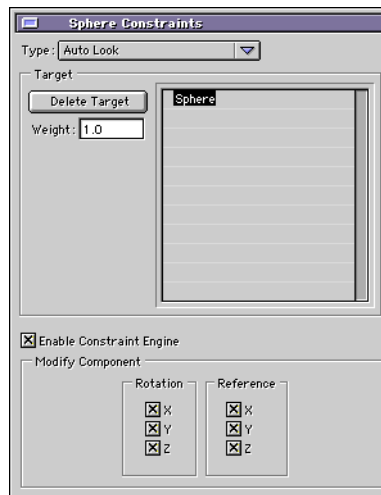


Figure 1.1 — Constraint Editor

The Constraint Editor also has a Modify region that lets you disable the constraint’s effect on individual axes. Targets may be deleted using the Delete button. If all targets for a particular constraint are deleted then the constraint itself is deleted.

1.2 Constraint Types

Auto Look The Auto Look constraint causes the constraint object to point toward the targets and can blend in rotation/reference data from the constraint object itself. This constraint replaces the look-at feature in 3.0 Universe and earlier versions of EIAS. To enable Auto Look, select a constraint object, and then select the targets to look at. One of the targets can include the constraint object itself, which indicates to the engine to use the constraint object's rotation/reference animation channels as part of the weighted solution.

While the constraint is in force, the rotation/reference channel values in the Group Info window of the constraint object cannot be altered. It is necessary to turn off the engine temporarily to allow the editing of the constrained channels. The “Enable Constraint Engine” button in the Constraint Editor dialog box can be used to do this. Turning it off will revert the object's rotation/reference back to its animation channel values.

As with all targets, the rotation/reference channel has a weight associated with it, allowing you to smoothly blend the various look-at points and animation values for the desired rotation/reference orientation.

Auto Bank Auto Bank allows an object's orientation to be animated while the object is moving along a curved path. This constraint replaces the auto bank feature in Universe 3.0 and earlier versions of EIAS.

To enable Auto Bank, select the object, and then select itself as the target. It actually doesn't matter what object you choose since the target isn't used by the constraint system. The weight of the selected target is used to blend the Auto Bank rotation with the rotation in the animation channel of the constraint object.

The “Enable Auto Roll” button adds a “roll” into the banking based on the gravity vector length. The tighter the curve, the more pronounced the roll. The larger the gravity length, the less pronounced the roll. The gravity direction has no effect.

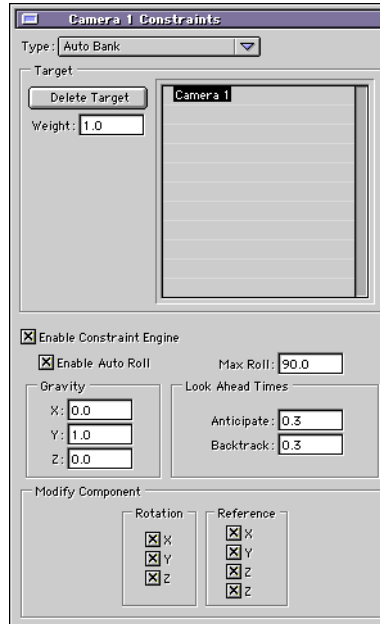


Figure 1.2 — Constraint Editor-Auto Bank

The orientation is computed by looking back and ahead a certain amount of time. Looking ahead in time allows you to “anticipate” an orientation change on the curvature of the path. Looking back in time allows you to “ride out” an orientation change. Looking ahead in time is called Anticipation and is entered in seconds. Looking back in time is called Backtrack and is also entered in seconds.

Aim Aim is a multi-purpose aiming constraint that allows you to specify which direction to aim at and to set an up-direction to keep the object upright. The aim vector is the main vector to aim the constraint object toward. It is possible to aim the object in directions other than the main three axis x, y, or z. The local up-vector keeps the object upright in its local space, while the global up-vector keeps the object upright in its global space.

The global up-vector is a reference vector for the local up-vector. When the aim vector aligns with the target, the local up vector tries to align itself as close to the global up-vector as possible.

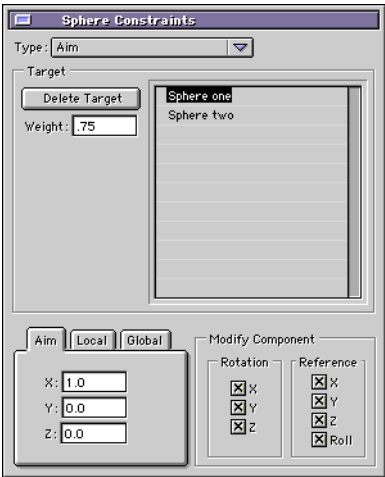


Figure 1.3 — Constraint Editor-Aim Constraint

Position The Position constraint moves the constraint object to the target point. The target point is the weighted average of the target centers. An optional X,Y,Z offset can be included. This offset is applied to the computed target point.

- Rotate** The Rotate constraint rotates the constraint object in global space as the result of the weighted average of the global rotations of its target(s).
- Scale** The Scale constraint scales the constraint object based on the weighted average of the scales of the targets. Either local or global scales are used based on the check-box in the Constraint Editor.
- Normal** The Normal constraint keeps the constraint object oriented normal (perpendicular) to the target. This constraint is different than the others in that the Constraint Object must be linked to another group and that group is animated, not the Constraint Object. The simplest way of achieving this is to link the Constraint Object to an effector. Create a Normal Constraint between the Constraint Object and the target. As you drag the effector over the target, the Constraint Object will move with the effector and rotate to remain normal to the target.

While computing the constraint, there is a choice of point search methods. The “Every Vertex” method is the slowest and will use all vertices in the target to find the closest point between the constraint object and the target. Use it if the target has non-shared vertices. The “Closest Vertex” method is the fastest and will use the closest vertex to find the closest point on the target. Use it if the target has all vertices shared.

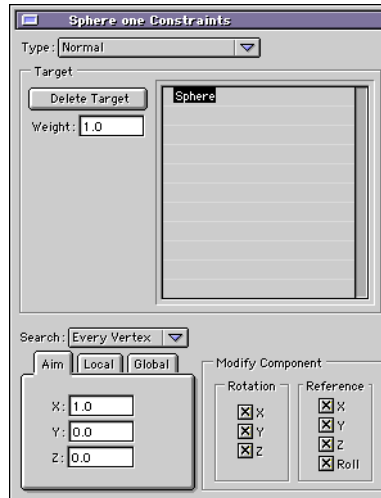


Figure 1.4 — Constraint Editor-Normal Constraint

Geometry The Geometry constraint keeps the constraint object at the position of the target point and normal (perpendicular) to the target point. The target point is the point on the target that is closest to the constraint object. The target must be a geometric mesh (a group). The constraint system will only use one target at a time. If you have added multiple targets, the one with the largest strength will be used.

This constraint is different than the others in that the Constraint Object must be linked to another group and that group is animated, not the Constraint Object. The simplest way of achieving this is to link the Constraint Object to an effector. Create a Geometry Constraint between the Constraint Object and the target. As you drag the effector over the target, the Constraint Object will move to stay fixed to the target.

While the Normal constraint will only use the target's computed surface normal to affect the orientation of the constraint object, the Geometry constraint will use the computed surface normal AND surface point to compute the orientation and position of the constraint object. Since both will clash with the rotation/reference channel, use the Geometry constraint if you desire both normal and position constraint.

The surface normal at the target point can be the target orientation so that the aim vector of the constraint object aligns with the normal vector.

- Pole** The Pole Vector constraint rotates the Pole Vector toward the target point.
- Vector** The target point is the weighted average of the position of all of the targets.

1.3 Circular Dependencies

Universe Animator will prevent you from creating conditions that allow circular dependencies. For instance, create a new project and then add a sphere and a cube. Now constrain the sphere to the cube's position (sphere is constraint object, cube is target). Animator will no longer allow you to link the cube to the sphere in a parent/child hierarchy. Why? Because the sphere is getting its position from the cube via the constraint. If the cube were a child of the sphere, then, through inheritance, the cube would be getting its position from the sphere. A cycle would then be created and Animator wouldn't know how to resolve the correct position of the objects.

Inverse Kinematics

2.0 Introduction

Inverse Kinematics (IK for short) is the process of computing the rotations and positions of the parents in a hierarchy given the position and rotation of the children. The term “Inverse” comes from this method of solving the hierarchy from the bottom-up. Traditional, forward kinematics, computes the position and rotation of the children by inheritance from the parents (top-down).

IK is a more direct method of animation for characters when you want to position the end of a hierarchical branch (called a “chain”) and have the other parts of the branch follow naturally. With IK, you can drag a character’s hand to the door knob and the forearm and upper arm will follow.

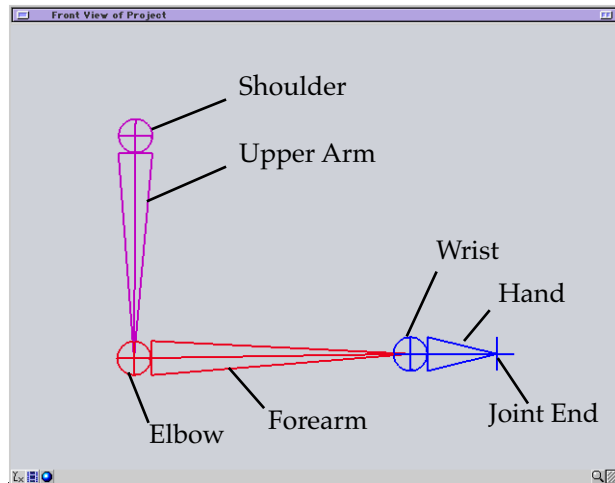


Figure 2.0 — Arm Chain

Universe Animator allows you to apply a controller, called an “IK Handle”, to chains. Once applied, the handle is used to manipulate the chain. Moving the handle causes the IK engine to solve for the position and rotation of the members of the chain.

2.1 Creating an IK Handle

An IK Handle is created by selecting the bones to be controlled by the Handle in the Project window or one of the View windows and then choosing Add IK Handle from the Character menu. Instead of choosing all the bones in the chain, you may choose only the last one. In this case, the IK Handle created will control all the bones from the end of the chain up to the top of the branch (the chain must contain at least two elements or no IK Handle will be added).

Once added, the IK Handle will appear in the Project window and in the View windows.

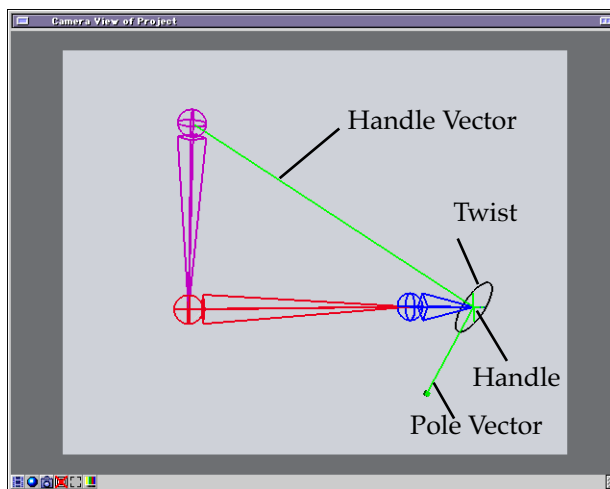


Figure 2.1 — Arm Chain with IK Handle

In the View windows, the IK Handle is represented by a green plus sign. The first and last bone being controlled by the Handle is indicated by a Handle Vector. Optionally, two other IK Handle sub-controllers can be shown, the Twist (oval shape) and the Pole Vector (line ending in small cube). These sub-controllers are discussed below.



Figure 2.2 — IK Handle in Project Window

The IK Handle is added to the bottom of the Project window and given a unique number to differentiate it from existing handles.

Multiple IK Handles can be added to a chain but the Handle Vectors may not overlap. In the arm example, Figure 2.1, you could add a Handle from the shoulder to the wrist and then a second handle from the wrist to the joint end. But you could not add a second handle from the elbow to the joint end. Overlapping control is not permitted. If you attempt to do so, Animator will not add the Handle.

2.2 IK Handle Info Window

Double-clicking on the IK Handle in either the Project window or View windows brings up the IK Handle Info window. Like all other object classes in Universe Animator, the IK Handle has an info window that controls basic attributes such as position, rotation, etc. The IK Handle Info window is divided into three tabs:

- X-Form
- Properties
- Chain

On the left side of the window you will see three check-boxes.

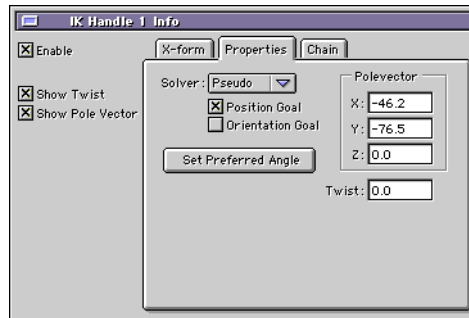


Figure 2.3 — IK Handle Info Window

Enable

This check box (defaults on) allows you to disable the IK Handle. The Handle won't disappear from the View windows, but the IK engine will no longer be called when the handle is moved.

Show Twist

This check box (defaults off) causes the circular -shaped twist controller to be displayed in the View windows.

Show Pole Vector

This check box (defaults off) causes the Pole Vector controller to be displayed in the View windows.

X-Form Tab

The X-Form tab contains the edit boxes that control position, rotation, and scale for the IK Handle.

Properties Tab

As depicted in Figure 2.3, this tab has all the controls for the IK engine. Including those for the type of IK engine (called a solver) and the current settings for the Pole Vector and twist.

Set Preferred Angle

There is an angle for each joint that the solver tends to move toward. This is called the Preferred Angle. Initially this angle is set equal to the rotation values in your chain at the time the IK Handle is applied. Pressing the Set Preferred Angle button resets the preferred angles to the current rotation values in your chain.

Why is a preferred angle needed? Imagine a chain made up of four bones in the shape of a “W” with the IK Handle at the top-right end of the chain. Pulling the handle to the far right would stretch the “W” shape out into a straight line. As you drag the Handle back to the left, it would be nice if the chain slowly took on its original “W” shape again. The IK engine is, in fact, able to do this by remembering the preferred angle for each of the joints in the chain.

The preferred angle determines the preferred configuration of the IK chain and makes the IK solution predictable. Choosing the right preferred angle can also avoid singularity conditions. These conditions occur when you pick the fully extended position as the preferred angle for a chain.

Note *When creating a chain (building the arms of a character for instance), it is best to add the bones such that there is a angle less than 180 degrees between the joints. Creating bones in a perfectly “straight line” can lead to singularity conditions because the preferred angle is initially set to the angle of bone creation.*

Solver

There are three different mathematical engines (solvers) you can choose from to determine the optimal behavior of each IK Handle in your project.

Pseudo The Pseudo solver solves an approximated linear system in a small interval and then integrates the results. This solver is more accurate than the Minimizer, but slower.

If you are using joints with rotation limits, the Pseudo solver may never reach the goal after it runs out of its time interval. This makes the Pseudo solver slow when using rotation limits. Therefore, we suggest using the Minimizer solver when rotation limits are involved.

Minimizer The Minimizer is implemented using a conjugate gradient method. It uses a recurrence formula to generate a series of mutually conjugate search directions and moves along the search directions to minimize the measure function of the chain configuration.

The Minimizer solver efficiently handles joints with rotation limits and is superior to the Pseudo solver when rotation limits are used.

2-Bone The 2-Bone solver is the fastest of the three solvers but only works on the first two bones in the chain. It is ideal for arms, legs, and other bone pairs. Do not use it with chains that contain more than 2 bones. It does not offer high-accuracy when rotation limits are used.

Position Goal

This check-box tells the solver that its goal is to move the chain end to the same position as the IK Handle. This is the default.

Orientation Goal

This check-box tells the solver to match the orientation of the chain end to the orientation of the IK Handle. Only the Minimizer solver uses this option. Orientation can be confusing since the solver may rightly decide that the solution has been reached before the chain has been dragged to the position of the handle. The Minimize solver creates a balance between the position goal and the orientation goal when both check-boxes are on.

The solver will not do anything if both Position Goal and Orientation Goal are turned off.

Chain Plane

There is an imaginary plane that runs through the joints in your chain. This plane is used by the engine to compute the orientation of the chain. Specifically, the Plane controls the twist of the chain. The IK engine has two controllers, the Pole Vector and Twist, that allow you to control the overall orientation of the Chain Plane, and thus, the orientation of the chain. In Universe, the IK engine uses the first two bones in the chain to create the plane.

Pole Vector

The Pole Vector is shown in the View windows as a green line that begins at the location of the IK Handle and terminates in a tiny box. You can drag this vector in the View windows to alter the overall orientation of the IK Plane.

Note that if the Pole Vector crosses the Handle Vector or becomes opposite to it during posing or animation, the entire chain can suddenly flip 180 degrees around the Handle Vector. Take care to orient the Pole Vector to prevent this flipping behavior.

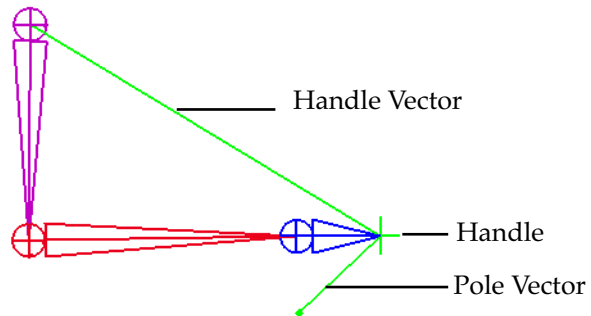


Figure 2.4 — Pole Vector

Twist

The Twist controller allows you to rotate the entire chain about the Handle vector. The controller defaults to 0.0 degrees and indicates that by its black color. As you drag your mouse around the controller, the twist value will increase and the controller will become green to indicate the twist value.

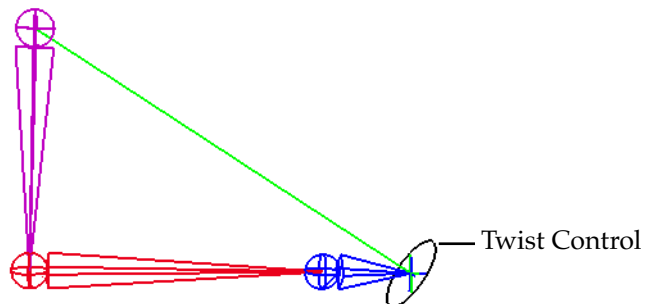


Figure 2.5 — Twist

Chain Tab

The Chain tab contains the list of bones that are under the control of the IK Handle. The list is not interactive, it is for reference only.

2.3 Using IK on Models

The sections above discuss the use of IK Handles with bone chains. Electric Image Universe 4.0 allows you to use IK Handles on model groups directly. This is useful if your character is mechanically jointed. A good example of this is a robot. The geometry data for a robot is not a single skin but is instead a series of separate, jointed model groups.

Once your groups have been organized into a character hierarchy, you can select model chains and apply IK Handles to them just as you would with bone chains.

You will need to use the Joint Editor (**Hierarchy > Joint Editor**) to set the link points for each of the groups in the chain (found in the X-Form tab). When creating a bone chain, this is done automatically. For model chains, it must be done manually.

Skin and Bones

3.0 Introduction

Universe 4.0 includes a new skinning engine that automatically deforms the geometry of your character (the “skin”) as you move your character’s bones.

The new skinning engine improves on the old method in both speed and ease of use. Restrictions such as the bones being the children of the skin and adding bones-style deformations have been eliminated.

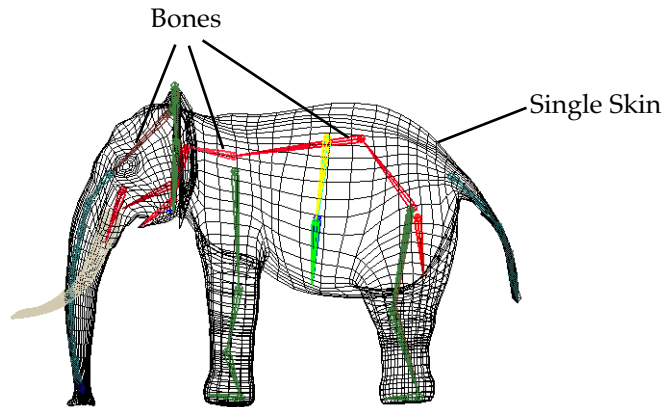


Figure 3.0 — Skin With Bones

Skinning is now a dependency. That is, a relationship outside the hierarchy is created between the bones and the skin (geometry). This means that the bone chains driving the skin cannot be dependent on the geometry in any other way. Therefore, BONES CANNOT BE CHILDREN of the skin, as was the case in previous versions of Universe and EIAS.

There is one exception to this rule. Bones used with the old-style bones deformation system can and must still be children of the geometry.

3.1 Bones

Bones are a type of effector designed to give geometry motion and deformation. They are linked together in short hierarchies, called bone chains or skeleton chains.

For information on adding bones to your scene, see *The Animator 3.0 Manual, Chapter 13, “Using Effectors”*.

When using bones with the skinning engine, the options in the Bone Info window are valid with a few exceptions. Bone length, influence region, and strength are not animated values. At the time of the skin binding, the current values will be used and further changes will be ignored. Also note that “Rest” Position has no meaning anymore. The minimum and maximum influence values for bones will still have effect and may be useful in influencing the “folding” of the skin.

Note Animator version 4.0 has removed the “skeleton effector”. This was an alternate effector type that could be used to manipulate the position of geometry but not alter its shape. Bones fulfill both of these purposes, and in addition, IK Handles can now work directly on model chains. See *“Using IK on Models” on page 19*

Split Bone

A new command has been added to the Character Menu, “Split Bone” (**Character > Split Bone**). This command divides the selected bone into two separate bones, maintaining the hierarchy. The new bone is inserted as a child of the original bone and children of the original bone become children of the new bone. Split Bone can be used after you have created your skeleton to give a particular chain more definition and control. For example, let us say your character’s spine has 3 bones and you decide that the character needs to bend in a very flexible way. Splitting the bones would now give you 6 bones in the spine without having to go to the trouble of manually rebuilding the skeleton.

Bone Strength Tab

The Bone Strength tab has been added to the Bone Info window in version 4.0 of Universe Animator. This tab lists the strength maps that the bone is associated with. See *“Strength Maps” on page 27* for more information. The list is in the form of map name, followed by the skin name in parenthesis. The button next to the list entry may be used to temporarily remove the bone from its association with a particular strength map. The “Del” button permanently deletes entries from the list.

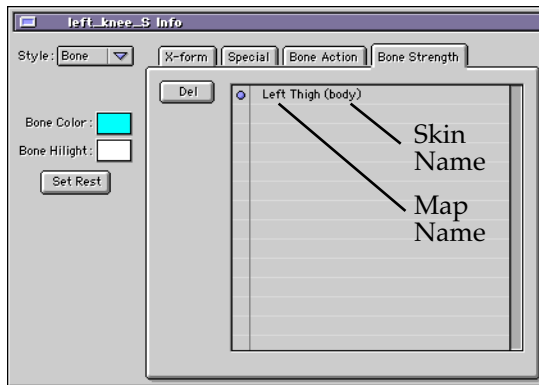


Figure 3.1 — Bone Strength Tab

3.2 Binding the Skin

Binding is the process of creating a dependency relationship between a group of bones, the “skeleton”, and the skin (geometry).

When building the skeleton, place it in the center of your skin. When you are ready bind the bones to the skin, these spatial relationships will be preserved throughout the animation. Changing the initial relationship will involve detaching the skin from the bones and then re-binding the skin geometry to the bones.

Skin binding is accomplished in one of two ways:

1. Select any bone in a skeleton chain and use the “Bind Skin To Skeleton” in the Character menu (**Character > Bind Skin to Skeleton**) to bind the ENTIRE bone chain to the geometry group.

2. Select one or more bones and use the “Bind Skin To Selected” in the Character menu (**Character > Bind Skin To Selected**) to bind only the **SELECTED** bones to the geometry group.

You can add more bones to the skin at any time by selecting the bone(s) and then using **Character > Bind Skin To Selected**. Also, a bone or a chain may be bound to more than one skin at a time. This is important when your character is made up of multiple geometry groups.

Once bones have been bound to a skin, the skin can no longer be translated, rotated, or scaled directly. These functions must be performed on the bones (or the controllers and IK handles you have added to assist in moving the bones). The bones give rise to both the movement and deformation of the skin. Morphing, deformations, materials, and any other surface-modifying features can still be applied directly to the skin.

Note *Skinning will work several times faster on FACT geometry than on geometry created by plug-ins. This is because the skinning engine caches data to prevent recomputation at every frame. This isn't possible with plug-ins because they can change the number and order of vertices every frame.*

3.3 The Skin Editor

The behavior of the skinning engine can be adjusted for each skin in your project using the Skin Editor. Select the skin with skinning options you wish to examine and choose “Skin Editor” from the Character menu (**Character > Skin Editor**).

The bottom half of the Skin Editor contains the bone list. The “All” tab shows the list of the bones bound to the skin. The “Del” button can be used to unbind selected bones from the skin.

The “Active” tab shows the list of the bones that actually influence the skin. Even though a bone is bound to the skin, it can be deactivated by turning off the “Activate” button in the Bone Info window. Bones with this button turned off will not be shown in this list.

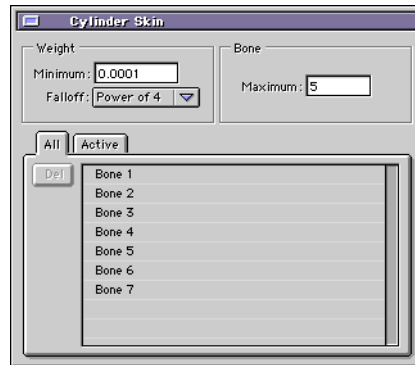


Figure 3.2 — Skin Editor

Weight Minimum

The Minimum weight allows you to specify the minimum influence a bone has on a vertex. If the computed weight falls below this threshold, the bone will not influence the vertex. This is a time versus accuracy control. Smaller numbers will improve the accuracy of the skin engine but at the cost of slower performance.

Falloff

The distance Falloff Power allows you to “tighten” or “loosen” the influence of the bones based on the distance between the skin’s vertices and the bones. The lower the falloff power, the “softer” the influence becomes. A “linear” falloff allows the vertex to pull away from the bone while a “Power of 32” falloff grips the vertex very tightly to the bone.

Bone Maximum

The Bone Maximum edit box contains the maximum number of bones that can influence each vertex in the skin. A value of 0 does not allow any influence, hence no skinning deformation at all. A value of 1 only allows the closest bone to influence the vertex, thus producing folds which are very angular and tend to penetrate the skin, while higher values produce more pleasing folds but at a somewhat slower performance.

Note *If you notice that vertices in your skin are being left behind (separating) as you move the bones, try reducing the Weight Minimum and increasing the Bone Maximum.*

3.4 Strength Maps

Strength maps allow you to fine tune the effect of a bone's influence on the skin. Strength is a relationship between a particular vertex in the skin and any number of bones. The skin may have many of these strength maps, each of which controls how much influence one bone has over another at a

particular point on the skin. Each bone can be associated with a single map per skin (but a bone may be bound to more than one skin).

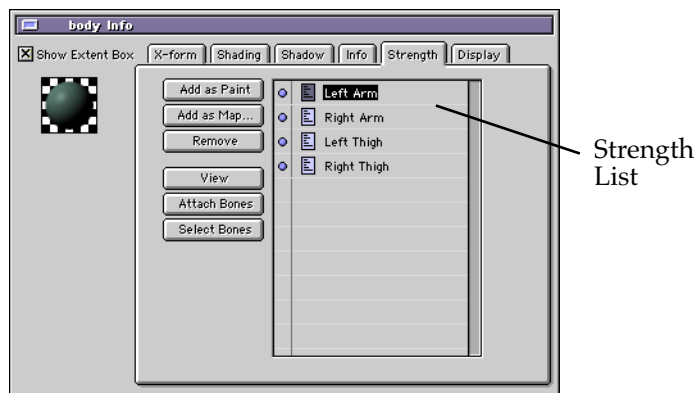


Figure 3.3 — Group Info Window—Strength Tab

Strength maps are a property of the skin and are added, removed, and controlled using the “Strength” tab in the Group Info window. There are two kinds of strength maps; Painted maps that you paint directly on the skin, and Texture maps that are applied using the normal material tools.

Add as Paint

Clicking on the “Add as Paint” button creates a new vertex Paint map. You are asked to give the paint map a name. The named map appears in the Strength List. Double-clicking on the paint map in the Strength List allows you to rename the map.

Add as Map

This brings up a texture get-file dialog box. You choose the texture map and it appears in the Strength List. Double-Clicking on the texture map in the Strength List brings up the Texture Info window. This window is used to position the map on the skin. The Texture map will automatically set the strength at each vertex based on the luminance value of the texture pixels.

Note *Only the first frame of an animated texture map will be used.*

Remove

The Remove button deletes the highlighted map from the Strength List.

View

Places the View windows into the strength map display mode. For texture strength maps the color of maximum strength may be chosen. The View palette pops up that allows the display color of maximum influence (100%) to be changed. Minimum influence (0%) is always shown as black. The color is displayed on the skin in the view windows as long as the palette is open. For painted maps, the Paint palette pops up and interactive painting on the skin may take place.

Note *The View windows must be in OpenGL mode (the hardware engine must be in use) and set to Flat, Gouraud, or Phong to see the strength.*

Attach Bones

Associates the bones selected in the Project window or View windows with the map highlighted in the Strength List.

Select Bones

Selects the bones in the Project window and View windows that are associated with the map highlighted in the Strength List. This is a quick way to see which bones are associated with a particular strength map.

3.5 Painting

When using a Paint map, you can interactively paint the skin using the built-in paint tool. To bring up the tool, select the Paint map from the Strength List and click on the “View” button. The Paint palette pops up and the skin’s color is changed to show the strength at each vertex. The color will be shown on the skin in the View windows until the Paint palette is closed.

Shortcut The W key can be used to temporarily turn off the paint colors while you are in paint mode.

Note *The View windows must be in OpenGL mode and set to Flat, Gouraud, or Phong to see the strength.*

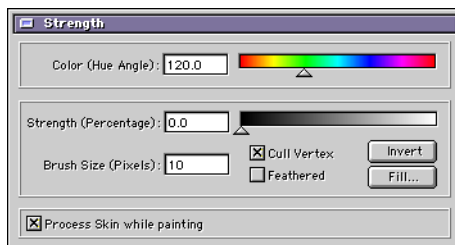


Figure 3.4 — Paint Palette

Color (Hue Angle)

Changes the color of maximum influence (100%). Minimum influence (0%) is always shown as black. This doesn't alter the behavior of the skin it is just a color preference.

Strength (Percentage)

Sets the value for the paint brush (0 to 100 percent). 0 percent means that the bones associated with the skin do not affect the skin at all. 100 percent means that the bones have full influence on the skin. Keep in mind that the final computed strength for any vertex on the skin is the weighted average of all the bones.

Shortcut Hold down the Control key and right mouse button (Control and Command keys on Mac), then drag the mouse right or left.

Brush Size (Pixels)

The Brush Size sets the radius of the brush in pixels. The brush cursor is shown as a circle in the View windows.

Shortcut Hold down the right mouse button (Control Key on Mac) and drag the mouse right or left.

Cull Vertex

The Cull Vertex check-box, when on, causes the brush to modify only the vertices visible in the View window in which you paint. When off, the brush paints straight through the skin to the polygons behind. If you are painting the left ear in the Side View window, the right ear will also be painted.

Feathered

Feathered modifies the vertex strength by the distance of the vertex to the brush center (cursor) and its value. This causes the strength to fall off as it moves from the center of the brush outward.

Invert

The “Invert” button inverts all of the strength values in the strength map. Setting 100% to 0% and vice-versa.

Fill

Brings up a dialog box that allows you to set every vertex in the map to a specified value.

Process Skin While Painting

When on, the skinning engine is called as you paint. This gives you constant feedback on the effect your painting is having on the skin deformation. The multitude of calculations required to provide this feedback is significant. You may wish to turn this option off to increase the speed of painting.

Note *If you see whole polygons turning a different color as you brush instead of nicely blended vertex colors, your OpenGL card or driver does not support the standard Vertex Blending feature. See your OpenGL card’s specification for more information on this feature.*

Illuminators

4.0 Introduction

A new lighting object has been added to Animator in Version 4.0. It is called an Illuminator. An Illuminator is an array of lights in a Dome-like or Box-like pattern. It can be used to create natural, day-lit lighting effects.

Illuminators are added by selecting Add Illuminator from the Object menu (**Object > Add Illuminator**). Click in one of the World View windows to define the center and then drag out the desired stage radius. Release the mouse button to complete the operation. A Dome Illuminator is added to the Project window and the World View windows.



Figure 4.0 — Illuminator in Project Window

Note *Illuminators are drawn using the light color parameter. The default light color is white. If the background color of the World View windows is also white, you won't see the Illuminators in the windows. We suggest setting the background color of your World View windows to a light gray.*

4.1 Illuminator Light Info Window

Double clicking on an Illuminator to open up its Info window. Comparing the Light Info window for an Illuminator to one for a regular light source (*The Animator 3.0 Manual, Chapter 4, “Lights in Universe Animator”*), you’ll notice that the Flare, Glow, Fog, and Projection tabs are missing. Illuminators do not support these options.

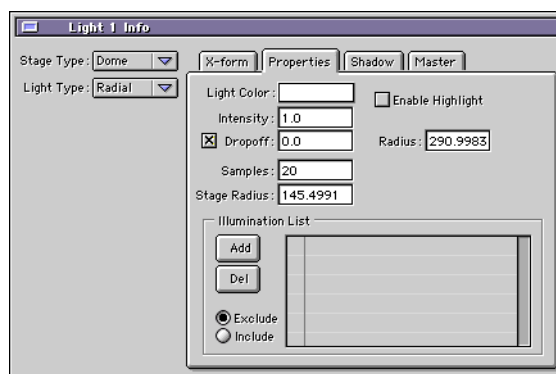


Figure 4.1 — Illuminator Light Info Window

Samples Illuminators are arrays of radial lights. The lights are distributed on the dome or the rectangle (depending on the Stage Type) in a random pattern. The number of individual lights that are distributed is controlled using the Samples parameter in the Properties tab of the Info window.

Stage Radius The Stage Radius controls the overall size of the Z-buffer shadow that is optionally cast by the Illuminator. You can go into the Shadow tab and set the usual shadow parameters. We suggest softening the overall effect of the shadows. This makes for a more pleasing effect.

You can also set the Illuminator to cast raytraced shadows. Please keep in mind that, for 20 samples, you would be adding 20 raytraced shadows to your scene. This will have an adverse effect on render times.

Each shadow cast by a sampled light falls on the center of the Illuminator stage because each of the sampled lights is pointed from their positions on the dome or box toward the center of the stage.

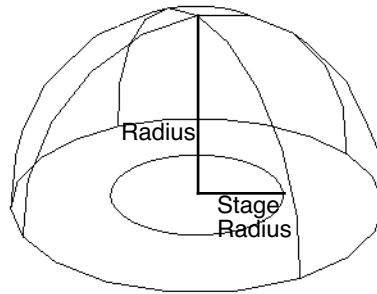


Figure 4.2 — Dome Illuminator—As Shown in World View Windows

Light Type You can choose to have either Radial lights or Parallel lights distributed across the surface of the Illuminator

Stage Type The Stage Type: pull-down menu to changes the Illuminator type from Dome to Box. Box lights have Height, Width, and Length dimensions that define the outer shell that the lights are distributed on. They behave in all other respects just like the Dome Illuminators.

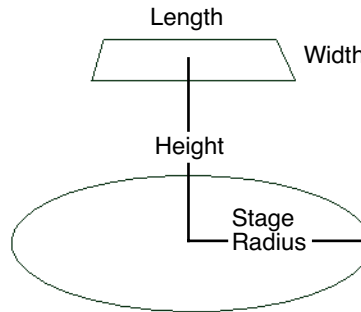


Figure 4.3 — Box Illuminator— As Shown in World View Windows

4.2 Using Illuminators

Because Illuminators focus their attention on a single point in your scene, the center of the stage, they are best suited for lighting a discrete object or region. One rewarding technique has been to use two Illuminators, one pointing down from above and one underneath pointing upwards. The Illuminator underneath acts to 'reflect' light back up from the stage. This Illuminator is given a lower light intensity value and a color in keeping with the object on the stage.

Blur Preview

5.0 Introduction

The blur preview gives a good approximation of the final rendered blurring effects (from motion blur and depth of field blur) using the accumulation buffer feature of OpenGL and the internal software renderer. It does so by rendering multiple frames into a buffer and averaging the result.

5.1 Previewing the Scene

Hold down the Alt key (Option key on Macintosh) and click on the preview button in any of the View windows. A dialog will appear allowing you to set the options and execute the preview.

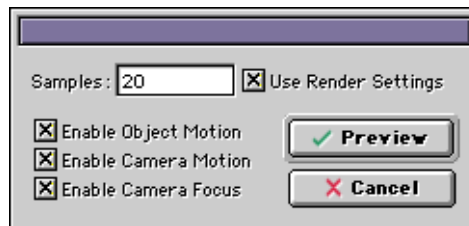


Figure 5.0 — Blur Preview Control

Samples The number of frames to render before displaying. The higher the number, the more accurate the blur, but the longer it will take to compute. This value is ignored if the “Use Render Settings” check-box is on.

Use When this check-box is on, the number of samples is taken from the multi-frame blur edit box in the render dialog (Motion Blur tab). If the check-box is off, the value in the Samples edit-box is used.

Enable This option adds the motion of the objects in your scene into the calculation
Object for the motion blur preview.
Motion

Enable This option adds the Camera's motion into the calculation for the motion
Camera blur preview.
Motion

Enable This option enables the calculation of the out-of-focus effect of the Cam-
Camera era's F-Stop. To maximize the effect, leave the grid on, use a long focal
Focus length lens (large focal length value) and a small F-Stop. Remember, the F-Stop works just like a real camera's F-stop. Large scenes will show very little out-of-focus effect, while small scenes create a larger effect. Keep your scene within 10 x 10 unit range and the focal length to 100 mm or more while the F-Stop is less than 1.0.

Clicking the Preview button starts the blur calculations. When completed, the View window will display the calculated blur. Clicking in the View window again clears the blur and returns the window to normal operation.

Animator Extras

6.0 Cubic Reflection Maps

Universe 4.0 supports Cubic Reflection Maps. These are six-frame animations that get applied to your group as a single, six-sided reflection map. One frame is applied to each side of your group. The frames must be stored within the animation in the following order:

1. Front
2. Back
3. Left
4. Right
5. Top
6. Bottom

Cubic Reflections are a bitmap reflection mode and can be chosen in the Texture Info window, Projection tab, Map Type pull-down. They are available for both the Reflection and Reflectivity channels.

6.1 Dragging Behavior

The dragging behavior in the View windows has been changed for objects with reference vectors (Cameras, Spotlights, etc.). When you drag on the object's control widget, the reference vector will stay in place. Formerly, the object *and* its reference vector moved. This older behavior is still available if you hold down the control key when you drag on the widget. And, if you hold down the control key and drag the object, but not on its widget, the object will move along its reference vector but the vector will stay put.

6.2 Drawing Performance

Drawing is faster in 4.0 and more interactive for larger projects (For large projects, in the Preferences, Drawing tab, use “Large “Project” mode, set “Mouse Drag” to 0.1 and “Draw Update” to 0.5). All drawing is now interruptible when “Incremental Update” is turned on. *Incremental Update is recommended when working on large projects.*

A new preference has been added to the Drawing tab in the Preferences, “Update while dragging objects”. This preference defaults on. If you turn it off, the Camera View window won’t update as you drag objects in the World View windows. This greatly speeds up the interactivity of the drawing system.

6.3 Draft Mode

A new pull-down menu has been added to the Resolution tab of the Render Control window, “Draft”. It provides an easy way to render out to fractional resolutions. The default is “Full” which provides the resolution displayed in the X and Y edit boxes. Changing Draft to a fractional setting won’t alter the resolutions displayed in the edit box, but Camera will render to the fractional size.

The setting is ignored by snapshot renderings.

Note *It’s easy to forget the setting of Draft. Make sure you set it back to Full before you begin your final rendering.*

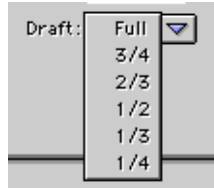


Figure 6.0 — Draft Pull-Down Menu

6.4 Linking to Animated Groups

Linking a child to an animated parent can cause the child to animate in undesirable ways. This occurs because a rotation is pre-calculated for the child. The motion path of the child is modified which can cause it to ‘fly off into outer space’.

You can avoid this behavior by turning off Rotation Inheritance in the Joint Editor for the child prior to linking it to its new parent. No pre-rotation calculation will be performed and the original motion path will be kept. After the linking has been done, the Rotation Inheritance can be turned back on

6.5 Tool Tips

Tool tips have been added to all icons within the Animator. Leave your mouse pointer over an icon for a second or two and a description of the icon will pop up in yellow.

This feature can be turned off using the Preferences, Viewer tab, Display Tool Tips check-box.

6.6 OS X

Universe 4.0 runs natively under Mac OS X—it also runs under Mac OS 9. Running in the Classic environment is not recommended.

Warning *Do not run Universe with the Classic environment running in the background. Classic, even when idle, takes a lot of CPU cycles. It also has its own driver for the dongle. This can cause corruption of your projects. A warning dialog box appears if you attempt to launch Animator or Modeler when Classic is running. Shut down Classic using the Classic Control panel in the Mac OS X System Preferences.*

After installing the applications under OS X, launch the Camera and set its memory allocation as described below.

Under OS X, the Quit and Preferences menu items have been moved to the “Universe Animator” menu.

6.7 Setting Camera’s Memory Allocation

Under all operating systems (except Mac OS 9), you must manually set the memory allocation for Universe Camera within the application itself. This is done in Camera using the Set Memory Usage in the File menu (**File > Set Memory Usage**).

Make sure there is no Camera.ccn file in the same folder with Camera (or Camera will begin rendering), launch Camera, and choose **File > Set Memory Usage**. A dialog box will open, allowing you to set Camera’s memory partition. Set it to a number between 40 (40MB) and 1000 (1 GB). Future

versions of Mac OS X may allow more than 1GB to be allocated to Camera. If, during rendering, Camera launches and immediately quits, you have assigned more memory than your operating system will provide. Throw away the Camera.ccn file, launch Camera and lower the memory allocation.

6.8 Scrubbing

Scrubbing the view windows with the time thumb has been added to version 4.0. Hold down the Control key while dragging the time thumb to see all View windows update as you drag. Hold down the Control key and Shift keys while dragging the time thumb to see just the Camera view window update as you drag.

This feature works with the time thumbs found in the Project window, Time palette, and F-Curve editor. It does not work in the Morph window.

6.9 Show Illumination

A new check-box has been added to the column of check-boxes on the left side of the Light Info window (also to the Illuminator Info window). Show Illumination (defaults on) can be used to control which lights in your scene are used during OpenGL drawing.

When this check-box is turned off, the light will no longer be used to shade in the View window. However, Camera will still use the light when it renders.

You can create light sets and turn their Show Illumination check-boxes on and off in groups to manage your drawing speed. OpenGL can only use

eight lights at a time. It renders the scene with the first eight lights, if there are more than eight, the scene must be re-rendered and added to the first rendering. Each of these rendering passes eats up a lot of time. The Show Illumination check-box can be used to determine which lights and how many lights get used by OpenGL. Please keep in mind that one light is always reserved by the drawing engine for lighting the light and camera widgets. Therefore, the first OpenGL pass really consists of the first seven lights in your scene.

6.10 Shortcut Keys

The following shortcut keys work in 3D windows:

T for Translation (dragging)

R for Rotate

S for Scale

] for Zooming in

[for Zooming out

= for Fit to view

Also, in any active orthographic window (that is, not the Camera View), you can switch from that specific view to any other orthographic projection.

1 for Top

2 for Side

3 for Front

This is useful if you operate with only one orthographic window to improve drawing speed.

Modeler 4.0

7.0 Introduction

Welcome to Universe Modeler, version 4.0!

Universe Modeler now runs natively under Apple's MacOS X, allowing you to take advantage of this modern operating system's stability, performance, and feature enhancements.

Modeler now uses OpenGL to perform all drawing. This allows you to use today's fast graphics hardware to its fullest capacity, and to be able to gain graphic performance by upgrading to newer technology. Working in smooth-shaded mode is no longer a penalty; it's blazing fast!

Now you can convert your solid objects or surfaces to objects that have a thickness, like sheet metals using the new Shelling tool. You can create thin-walled solids or thick surface sheets.

Universe 3 introduced the NURBS Surface Reduce Knots tool for rebuilding complex surfaces to simple ones within a tolerance. Boundary-constrained surface rebuilding takes this tool one step further by constraining the boundary of the new surface to remain close to the original, so that it can stay connected to neighboring surfaces.

Two tools in the 3D Tools palette have been enhanced. The Skinning and Blend Surface From Edges can, in addition to curves, accept a point as the final pick for the operation. This allows you to create shapes that end perfectly at a point.

While building on the stability of Universe Modeler 3.1, several enhancements have been made to improve your modeling experience.

- Modeler 4.0 uses less RAM than 3.1 for comparable projects, which will translate into more stability for larger projects, and snappier performance in general
- Expanded list of image formats that can be used as image templates for constructing objects
- Improved perspective views
- The layer window now supports marquee selection for manipulation of objects
- Several time consuming operations are now interruptible

7.1 Shelling

Modeling parts often requires surfaces of a certain thickness, or a solid object to be hollow inside. The Body Edit > Shell tool is an essential link between surface and solid modeling.

Figure 7.0 shows a cube that was beveled **(a)** and then shelled with a positive distance — shelled outward **(b)**, and then a negative distance —

inward (c). In both cases, the object has been cut to make the shelling visible. Note that in the case where the shelling was inward, the topology has been modified to remove the blend as appropriate.

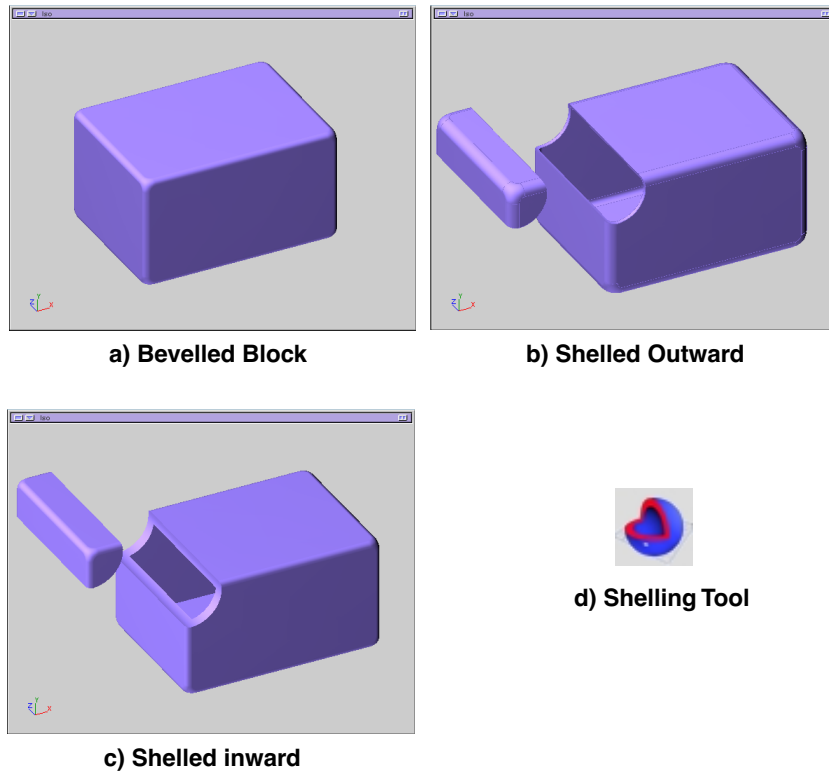


Figure 7.0 — Shelling Tool

There are limitations to this tool:

- The sheet supplied must be manifold (i.e. physically resiable), and single lump and single shell

- Shelling operations may fail on surfaces that are not tangent continuous everywhere (run Body Edit > Check Body tool on the object; it should not return errors about curves/ surfaces being non-G1, i.e. not tangent continuous)
- Shelling must fall within the radius of curvature (can't be self-intersecting)

7.2 Boundary-Constrained Surface Rebuilding

Universe 3 introduced the NURBS Surface Reduce Knots tool for rebuilding complex surfaces to simple ones within a tolerance. This new option takes this tool one step further by constraining the boundary of the new surface to remain close to the original, so that it can stay connected to neighboring surfaces.

The implications of this are that less knots are removable from the surface, and you will see that the surface may not be simple compared to the previous version. So the default behavior of the tool remains the same, and to avail the new functionality where the boundary of the surface is constrained, please press and hold down the SHIFT key while selecting the NURBS Surface Edit > Reduce Knots tool.

The usefulness of this tool lies in the operations after this step. Consider for example that you have created a blend surface between two sculpted NURBS surfaces. The blend surface is generally much denser, and if you use the reduce knots tool the boundary may change enough that the three surfaces may not stitch (Booleans > Stitch) back together. To avoid this, hold down the SHIFT key when you use the tool, and you have a better chance of having the three surfaces stitch together.

Example

Figure 7.1 (a) and (b) show a wheel rim with a center piece that connects the top and bottom piece. In edit mode, the center piece shows many CVs. When the surface was rebuilt using the Reduce Knots tool, the number of CVs was reduced significantly (c), but on committing the changes, we see that the boundary of the center piece does not match the center hole of the rim cover (d). While using the same tolerance value, we hold down the

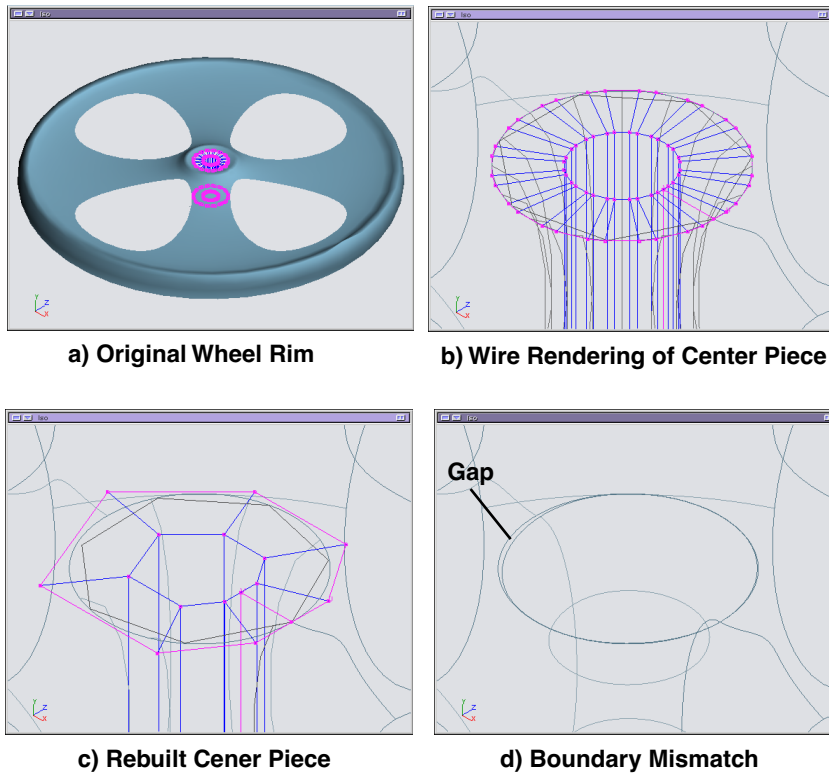


Figure 7.1 — Rebuild with Boundary Constraints

Model Courtesy Paul Sherstobitoff

SHIFT key when choosing the NURBS Surface > Reduce Knots tool, and we see a similar reduction in knots Figure 7.2(a). We also see, after committing the changes (b), that the boundaries still match and we can stitch the objects together.

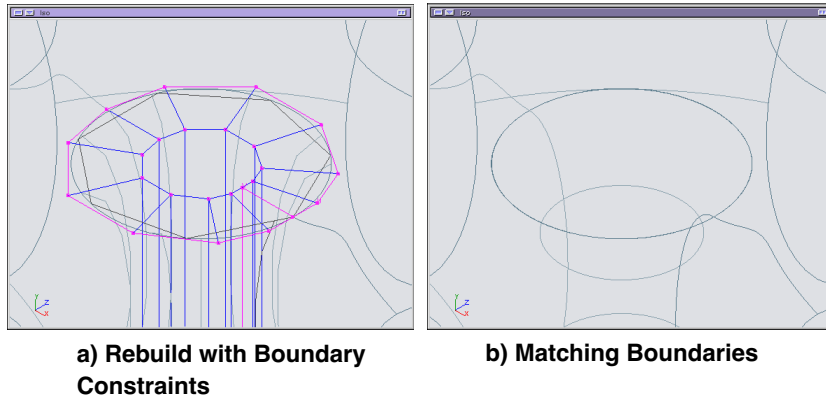


Figure 7.2 — Rebuild with Boundary Constraints Continued

7.3 Skinning and Surface Blending to a Point

Two tools in the 3D Tools palette have been enhanced. The Skin and Blend Surface from Edges tools can, in addition to curves, accept a point as the final pick for the operation. This allows you to create shapes that end perfectly at a point.

The tools haven't changed, so you can pick all the wires as usual, but when you are ready to pick a point as your last point, set the Pick Filter to Point mode, and pick a point from any existing object — polyline, curve, solid object, surface, etc. The skin or the surface blend will be created without requiring any further action.

Figure 7.3 shows a series of closed curves that were skinned with the last pick being the end point of the line.

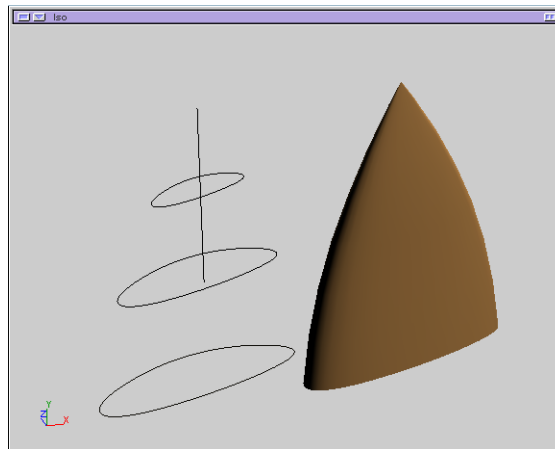


Figure 7.3 — Skin to a Point

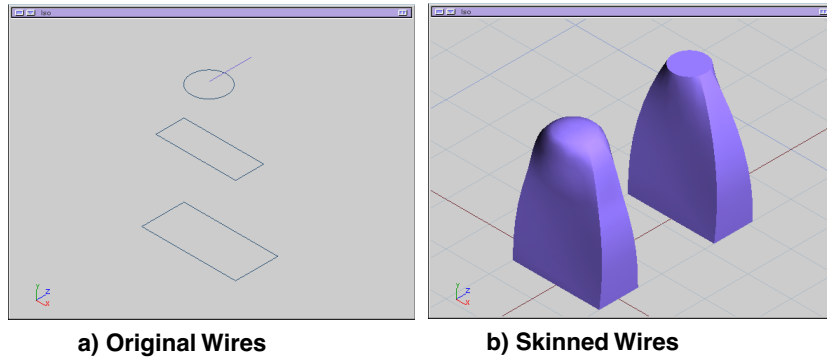


Figure 7.4 — Skin to a Point Example

In the example shown in Figure 7.4, note the use of the circle as the penultimate wire which gives the skin a smoother approach into the point. In (b) the object in the background was skinned without selecting a point. The object in foreground was skinned to a point. This gives it a rounded top.

7.4 Enhancements

Several areas of Modeler have been enhanced to increase your productivity. An important consequence of this is that the Modeler project file format has changed. UM4 files are incompatible with UM3.x releases. However, you can continue to open all your current projects from UM3.x in UM4. If you find the need to take a UM4 project to UM3.x, you can export objects in the SAT file format and import them into UM3.x.

In general Modeler uses less RAM making it more responsive and less prone to memory issues seen in the past. Larger projects are easier to work with in this version. Combined with several bug-fixes, this version is very reliable.

Image Templates and Background Images

Modeler now distinguishes between images brought into parallel views and perspective views. Invoking the context menu in any of the parallel views will reveal a menu item labeled “Load Image Template...” whereas in a perspective view the corresponding item is labeled “Load Background Image...”, as shown in Figure 7.5 and Figure 7.6.

In parallel views the image templates behave nearly like geometry. The image can be moved around in the world to accommodate model creation. You can have multiple image templates in a project and they will be visible in every view, including perspective views. Images templates can be selected in the layer view and deleted when necessary. You can also invoke the information window for each image template.

Hint If you are creating curves based on the image template which lies in the same plane, the curves may not be completely visible. Move the image template back a bit and now your curves will be fully visible.

In perspective views, the background images do not behave like objects; they do not appear in the layer view, and cannot be transformed. Only one background image can be loaded per view, and this background image is not visible in any other view. To delete this background image, invoke the context menu and choose “Remove Background Image”.

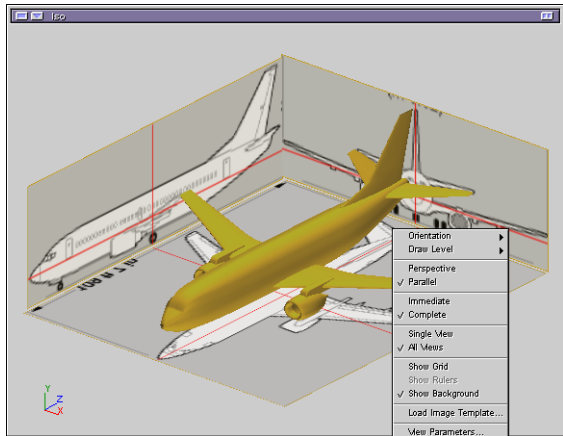


Figure 7.5 — Image Template



Figure 7.6 — Background Image

Both image templates and background images can now be not only in the EI Image format, but also GIF, JPEG, PICT, PNG and TIFF formats.

Interruptible Operations

Several complex and time-consuming operations are now interruptible by pressing the ESC key. These tasks can be interrupted only when you see the busy cursor. Specifically these tasks can be interrupted:

- Booleans > Union, Intersection, subtraction
- Bevels > Rounding, Chamfering
- Body Edit > Check Body
- Face Edit > Remove Face
- 3D Tools > Skinning, Profile Skin, Birail, Face-to-Face Blend Surface from Edges

Layer View Enhancements

A much requested feature has been added to the Modeler - the ability to marquee select objects from the layer view. This follows standard behavior, including the use of the SHIFT key to extend selections. For added feedback, objects being edited are ghosted in the layer view.

Mac OS X

The modeler now runs natively under Mac OS X. When running under OS X, the Quit and System Preferences (renamed 'Preferences') are located under the "Universe Modeler" menu item. Document Preferences remain under the Edit menu.

Warning *Do not run Universe Modeler with the Classic environment running in the background. Classic, even when idle, takes a lot of CPU cycles. It also has its own driver for the dongle. This can cause corruption of your projects. A warning dialog box appears if you attempt to launch Modeler when Classic is running. Shut down Classic using the Classic Control panel in the Mac OS X System Preferences.*

Changes to Check Body

Body Edit > Check Body tool's output file in case of errors, is named check-Body.txt which most OS's will recognize as a text file and open appropriately, or give you a choice of applications to open with.

Instead of sending an error message to the diagnostics file when you select UN/UM for check body, Modeler will now log a message in the status window indicating that the body can not be checked for errors unless it is fully converted.

Fact Export

When Modeler exports a FACT file on MacOS, it will set the creator to Animator, so that opening it will launch Animator. Under Windows, FACT files are already associated with Animator through the Windows registry.

Index

A

- Add as Map 29
- Add as Paint 28
- Anticipate 5
- Attach Bones 29
- Auto Roll 5

B

- Background Images 53
- Backtrack 5
- Bind Skin To Selected 25
- Bind Skin to Skeleton 24
- Binding 24
- Blur Preview 37
 - Enable Camera Focus 38
 - Enable Camera Motion 38
 - Enable Object Motion 38
 - Samples 37
 - Use Render Settings 38
- Bone List 26
- Bone Maximum 27
- Bone Strength 23
- Bones 12, 22
- Boundary-Constrained 48
- Brush Size 31

C

- Camera 42
- Chain Plane 17
- Circular Dependencies 9

- Classic environment 42, 56
- Closest Vertex 7
- Color 31
- Constraint
 - Geometry 8
 - Pole Vector 9
- Constraint Editor 3
- Constraints 1
 - Aim 6
 - Auto Bank 4
 - Auto Look 4
 - Normal 7
 - Position 6
 - Rotate 7
 - Scale 7
- Cubic Reflection Maps 39
- Cull Vertex 31

D

- Dome Illuminator 33
- Draft 40
- Dragging 39
- Drawing 40

E

- Every Vertex 7

F

- Falloff 27
- Feathered 32

Fill 32

Forward Kinematics 11

G

GIF 55

Gravity 5

H

Hue Angle 31

I

IK Handle 12

Illuminator 33

Image format 55

Image Templates 53

Interruptible Operations 55

Inverse Kinematics 11

Invert 32

J

JPEG 55

L

Layer View 55

Linking 41

M

Mac OS X 55

Memory 42

Modeler 45

N

NURBS 48

O

Orientation Goal 17

OS X 42, 56

P

Paint Brush 31

Paint Palette 30

Painting 30

Perspective views 53

PICT 55

PNG 55

Pole Vector 13, 17

Position Goal 16

Preferred Angle 15

Process Skin While Painting 32

S

Samples 34

Scrubbing 43

Select Bones 30

Set Memory Usage 42

Set Preferred Angle 15

Shelling 46–47

Shortcut Keys 44

Show Illumination 43

Show Pole Vector 14

Show Twist 14

Skeleton Effector 22

Skin Editor 25

Skin to a Point 51

Skinning 51

Skinning Engine 21

Solver

2-Bone 16

Minimizer 16

Pseudo 16

Split Bone 23

Stage Radius 34

Strength 31

Strength List 30

Strength Maps 27

Surface Blending 51

Surface Blending to a Point 51

T

TIFF 55

Tool Tips 41

Twist 13, 18

U

Up-Vector 6

W

Weight Minimum 26