SOFTIMAGE|XSI Mod Tool includes software developed by the OpenSSL Project for use in the
OpenSSL Toolkit (http://www.openssl.org/). The OpenSSL toolkit is:
Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: [1.] Redistributions of source code must retain the
above copyright notice, this list of conditions and the following disclaimer. [2.] Redistributions in
binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution. [3.] All
advertising materials mentioning features or use of this software must display the following
acknowledgment: "This product includes software developed by the OpenSSL Project for use in the
OpenSSL Toolkit. (http://www.openssl.org/)" [4.] The names "OpenSSL Toolkit" and "OpenSSL
Project" must not be used to endorse or promote products derived from this software without
prior written permission. For written permission, please contact openssl-core@openssl.org. [5.] Products
derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in
their names without prior written permission of the OpenSSL Project. [6.] Redistributions of any
form whatsoever must retain the following acknowledgment: "This product includes software
developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)".

This software is provided by the OpenSSL Project "as is" and any expressed or implied
warranties, including, but not limited to, the implied warranties of merchantability and fitness
for a particular purpose are disclaimed. In no event shall the OpenSSL Project or its contributors
be liable for any direct, indirect, incidental, special, exemplary, or consequential
damages (including, but not limited to, procurement of substitute goods or services, loss of use, data,
or profits; or business interruption) however caused and on any theory of liability, whether
in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of
the use of this software, even if advised of the possibility of such damage.

The SSLeay library is:
Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.
This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The
implementation was written so as to conform with Netscape SSL.
This library is free for commercial and non-commercial use as long as the following conditions are
observed. The following conditions apply to all code found in this distribution, be it the RCA,
RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this
distribution is covered by the same copyright terms except that the holder is Tim Hudson
(tjh@cryptsoft.com).
Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be
removed. If this package is used in a product, Eric Young should be given attribution as the author
of the parts of the library used. This can be in the form of a textual message at program startup or
in documentation (online or textual) provided with the package.
Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met:
[1.] Redistributions of source code must retain the copyright notice, this list of conditions and the
following disclaimer. [2.] Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the documentation and/or other
materials provided with the distribution. [3.] All advertising materials mentioning features or use
of this software must display the following acknowledgment: "This product includes
cryptographic software written by Eric Young (eay@cryptsoft.com)". The word 'cryptographic' can
be left out of the routines from the library being used are not cryptographic related. [4.] If you
include any Windows specific code (or a derivative thereof) from the apps directory (application
code) you must include an acknowledgement: "This product includes software written by Tim
Hudson (tjh@cryptsoft.com)"

This software is provided by Eric Young "as is" and any express or implied
warranties, including, but not limited to, the implied warranties of
merchantability and fitness for a particular purpose are disclaimed. In
no event shall the author or contributors be liable for any direct,
indirect, incidental, special, exemplary, or consequential damages
(including, but not limited to, procurement of substitute goods or
services; loss of use, data, or profits; or business interruption) however
caused and on any theory of liability, whether in contract, strict
liability, or tort (including negligence or otherwise) arising in any way
out of the use of this software, even if advised of the possibility of such
damage.
The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

SOFTIMAGE|XSI includes software developed by the Apache Software Foundation (http://www.apache.org/). The Xerces software is: Copyright (c) 1999-2003 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
[1.] Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. [2.] Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution; [3.] The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. [4.] The names 'Xerces' and 'Apache Software Foundation' must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. [5.] Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, International Business Machines, Inc.. http://www.ibm.com. For more information on the Apache Software Foundation, please see http://www.apache.org/.

This document is protected under copyright law. The contents of this document may not be copied or duplicated in any form, in whole or in part, without the express written permission of Avid Technology, Inc. This document is supplied as a guide for the Softimage product. Reasonable care has been taken in preparing the information it contains. However, this document may contain omissions, technical inaccuracies, or typographical errors. Avid Technology, Inc. does not accept responsibility of any kind for customers’ losses due to the use of this document. Product specifications are subject to change without notice.

Documentation Team
Judy Bayne, Stephen Blair, Grahame Fuller, Amy Green, Edna Kruger, Luc Langevin, and Jamal Rahal.

09 2005
Contents

Copyright and Disclaimer ............................................. 2

Chapter 1  
About Guide Hairs and Render Hairs ................................. 17  
What Makes Up Hair? .................................................. 17  
The Hair Object .......................................................... 18  
Useful Tools for Hair ................................................... 19  
Overview of Creating Hair ............................................. 21

Chapter 2  
Creating Hair ............................................................... 23  
Choosing the Hair Emitting Surface .................................. 24  
Hair is Constrained to its Emitter ..................................... 24  
Modifying the Hair Emitter .............................................. 25  
Scaling the Hair Emitter ................................................ 25  
Deleting Hair .............................................................. 25  
Emitting Hair from Objects and Clusters ............................. 26  
Applying Hair to Objects and Clusters ................................ 29  
Emitting Hair from Subdivision Surfaces ............................. 30  
Creating Hair between Curves ......................................... 32  
Animating Hair from Curves .......................................... 33  
Viewing the Hairs .......................................................... 34  
Hiding Hair ................................................................. 36

Chapter 3  
Styling and Animating Guide Hairs .................................... 37  
About the Styling Operators ........................................... 38  
Styling and Dynamics .................................................... 38  
Using Deformations and Obstacles as Styling Tools ............... 39  
Scripted Operators and Hair .......................................... 39  
Progressive Styling by Freezing the Stack ........................... 40  
Selecting Guide Hairs .................................................... 42  
Creating Clusters of Hair Points ...................................... 43  
Brushing and Combing Hair ........................................... 44  
Brushing Hair .............................................................. 44  
Combing Hair .............................................................. 46  
Recombing Hair ........................................................... 48  
Changing the Hair Length .............................................. 50  
Scaling the Hairs .......................................................... 50  
Removing Hair ............................................................ 51  
Attenuating the Length .................................................. 51  
Cutting the Hairs ......................................................... 51  
Resetting the Hairs ....................................................... 52  
Shaping the Hair with Transformations ............................... 53  
Translating Hairs .......................................................... 53  
Rotating Hairs .............................................................. 53
Contents

Scaling Hairs .................................................. 54
Puffing and Straightening Hair ............................. 56
Clumping Hairs Together or Fanning Them Apart ...... 57
Interpolating Between Guide Hairs ....................... 58
Splitting Hairs ................................................... 58
Merging Hairs .................................................... 59
Separating (Shattering) Hairs .............................. 59
Viewing the Hair Group IDs ................................. 60
Locking Selected Points on Strands ...................... 61
Making Hair Move with Dynamics ....................... 63
Applying Dynamics to Hair ................................ 63
Caching Files for Playback ................................. 64
Adjusting the Hair’s Reaction to Dynamics ............. 65
Muting the Dynamics Operator for Styling .............. 67
Setting Up Hair Collisions .................................. 68
Tips for Setting Up Obstacles for Hair .................. 68
Setting Up Objects as Obstacles for Hair ............... 69
Stretching Hair Segments ..................................... 71
Styling Hair in Stretchy Mode ............................. 71
Animating Hair with Deformations in Stretchy Mode .. 73
Setting Up Forces for Hair ................................... 74
Copying Hair Styles ............................................. 75
Copying Styles ..................................................... 76

Chapter 4  Getting the Look with Render Hairs ......... 77
  Setting Render Hair Properties ......................... 78
  Viewing Render Hairs ......................................... 78
  Connecting Maps to Render Hair Parameters .......... 79
  Setting the Number of Hairs Rendered ................. 81
  Setting the Total Number of Hairs ..................... 81
  Rendering a Percentage of Hairs ....................... 82
  Increasing the Number of Render Hairs ............... 83
  Tips for Using the Strand Multiplier .................. 84
  Removing Render Hairs (Density) ....................... 85
  Setting the Render Hair Thickness ..................... 86
  Setting the Thickness Type ............................... 86
  Setting the Root and Tip Thickness .................... 87
  Setting the Render Hair Resolution (Segments) ....... 88
  Changing the Length of Render Hairs ................. 89
  Randomly Scaling the Render Hairs .................... 89
  Cutting the Render Hairs .................................. 89
  Replacing Render Hairs with Objects (Instances) ... 92
  Adding Frizz and Waves (Kinks) ....................... 94
  Frizzling Out ................................................ 94
  Kinks and Waves ............................................ 95
Contents
Contents
Hair

This guide covers the topics and tools that about hair and fur in XSI, generically referred to as hair.

• Hair and Fur Basics on page 16
  Basic concepts and tools for working with hair.

• Creating Hair on page 23
  Creating hair on objects.

• Styling and Animating Guide Hairs on page 37
  Styling guide hairs and using dynamics on hair.

• Getting the Look with Render Hairs on page 77
  Working with render hairs.

• Rendering Hair on page 97
  Hair shaders and issues regarding rendering hair.
Chapter 1

Hair and Fur Basics

Give me a head with hair—long beautiful hair. You can have it long, straight, curly, fuzzy, snaggy, shaggy, ratty, matty ... you get the idea. Yes, in XSI you can make all sorts of hairy and furry things—from Lady Godiva to rabbits, bears, and caterpillars. There are few limits to the hairy possibilities.

Hair in XSI is a fully integrated hair generator that interacts with elements of the scene's environment, such as lighting, other objects, and natural forces. As a result, you can create shadows for hair, create collisions with objects, and apply dynamics and forces to make the hair move.

Hair comes with a set of styling tools that allow you to groom and style the hair, almost as easily as if it was on your head! You can control the styling hairs one at a time, or grab hundreds and style as you please.

To control the rendered look, you can use special hair shaders designed for hair, or you can use any other XSI shader. And as with all things rendered in XSI, you can use the render region to preview accurate results.
About Guide Hairs and Render Hairs

Hair is actually represented by two types of hairs: guide hairs and render hairs. Guide hairs are actually segmented curves that act like inverse kinematics (IK) chains and are used for styling, while render hairs are the “filler” hairs that are generated from and interpolated between the guide hairs. As their name implies, render hairs are the only hairs that are actually rendered.

For information on styling guide hairs, see Chapter 3: Styling and Animating Guide Hairs on page 37.

For information on setting render hair properties, see Chapter 2: Creating Hair on page 23.

What Makes Up Hair?

Hair is actually an assembly of different parts that work together: the hair emitter, the hair object (geometry), the emitter object, the dynamics engine, and the hair shader.

- The hair emitter is the object from which the hair is emitted. This is any object that you select and apply hair to.
- The hair generator is the operator that generates the hair strands.
- The hair object is the actual object whose geometry is displayed as guide hairs. Styling operators you apply define the look of the guide hairs.
- By default, the Hair Renderer shader is a surface shader that is attached to hair to define the look of the render hairs. It lets you set the color on the hair strands, add transparency, self-shadowing, and optimize the rendering in different ways. You can also connect the Hair Geo shader to hair, as well as standard XSI shaders.
- If you apply dynamics to the hair, the dynamics operator is the engine that calculates the movement of the hair according to the velocity of the emitter object and any natural forces applied to the hair object.
Finding Hair Elements in the Explorer

In the explorer, you can see the relationship among all the parts that make up hair.

Applying hair creates a hair primitive object (appropriately called Hair) and its hair generator for each element to which you applied hair. The hair object is a child of the hair emitter object, with its local transformation parameters muted.

You can make hair a part of a character, especially within the structure of a model, and then transform that character as you like; for example, branch-select and transform the emitter object and the hair strands stay with it.

The hair object’s centre, however, stays at the global origin which matters for deformations.
Useful Tools for Hair

The Hair Toolbar

The Hair toolbar is your one-stop shopping center for hair styling tools. The commands on the Hair toolbar let you create hair itself, then perform a myriad of styling tasks on guide hairs, such as lengthening and shortening, combing, clumping, cutting, and many others.

Create commands
Create hair, natural forces, and apply dynamics.

Modify commands
Modify the hair's shader, guide hair style (all the hair styling tools), transfer maps for mappable parameters, and copy hair styles.

Display commands
Toggles styling and dynamics, toggles display of render and guide hairs, sets percentage of render hairs displayed, and activates stretchy mode.

To display the Hair toolbar
Click the toolbar title and choose Hair.

or

Press Ctrl+2 at the top of the keyboard.

If the Palette or Weight panel is currently displayed, first click the Toolbar icon or press Ctrl+1.

The Scale, Rotate, and Translate tools on the Transform panel also help you style the guide hairs.
The Hair Property Editor

When you create hair, the Hair property editor opens, allowing you to set viewing and render hair properties. See Chapter 2: Creating Hair on page 23 for more information.

**To open the Hair property editor**

- Select the hair object and press Enter.
  
  or

- Click the Hair node's icon in the explorer.

Snapping Filter for Hair

To make elements in the scene snap to guide hairs when styling, use the Hair filter in the Snap menu in the main command panel.
Overview of Creating Hair

This visual overview shows how to set up a hair simulation, including the chapter name in which you can find more information for each step.

For an example of creating and styling hair, see the Hair simulation tutorial in the Tutorials guide.

1. Apply hair to an object or cluster—see Creating Hair on page 23.
2. Style the guide hairs—see Styling and Animating Guide Hairs on page 37.
3. View and set up the render hairs—see Getting the Look with Render Hairs on page 77.
4. Apply dynamics and forces for hair movement—see Styling and Animating Guide Hairs on page 37 and Natural Forces on page 23.
5. Select obstacles for collisions—see Setting Up Hair Collisions on
6. Adjust the default hair shader or apply another one—see Rendering Hair on page 97.
Chapter 2  Creating Hair

You can create hair from a number of different objects, such as polygons, curves, and NURBS surfaces. After you create hair, you see the guide hairs appear. Hair is actually represented by two types of hairs: guide hairs (curves) for styling, and render hairs for viewing and rendering.

You can create, edit, and render as many instances of hair as you like in a scene. When you create hair, a hair primitive object and its hair generator are created for each object or component to which you have applied hair.
Chapter 2 • Creating Hair

Choosing the Hair Emitting Surface

The first step for creating hair is to set up the objects or components that will emit hair. Hair can be emitted from polygon meshes, selected polygons, polygon clusters, geometry-approximated subdivision surfaces, NURBS surfaces and subsurfaces, and curves.

- You cannot emit hair from point clusters, either on polygon or NURBS objects.
- Once you have generated the hair, you cannot change the number of guide hairs on the object.

Once you have set up the object, cluster, or curves as you like, you can emit hair from it:

- To emit hair from NURBS surfaces and subsurfaces, polygon meshes, polygon clusters, or selected polygons, see Emitting Hair from Objects and Clusters on page 26.
- To emit hair from polygon meshes with geometry-approximated subdivision surfaces, see Emitting Hair from Subdivision Surfaces on page 30.
- To emit hair from curves, see Creating Hair between Curves on page 32.

You can apply hair to the same object, cluster, or polygon more than once. This lets you, for example, create fuzz on a caterpillar for its undercoat and long stiff hairs for its spikes.

Hiding the Hair Emitter

You can emit hairs from an object and then hide that object. This allows you to create a low-resolution hair emission object that closely resembles the high-resolution model that is seen.

Hair is Constrained to its Emitter

When you create hair, it is pose-constrained to the hair emitter. This means that the hair object’s global transformation values (scaling, rotation, translation) are constrained to the emitter’s transformation. As well, the hair object’s local transformation parameter are muted. For information on the pose constraint, see Pose Constraints on page 300 in the Animation guide.

Having the hair pose-constrained to its emitter makes it easier to animate hair as part of character rigs, such as having hair enveloped or animated using deform operators.

You can make hair a part of a character, especially within the structure of a model, and then transform that character as you like; for example, branch-select and transform the emitter object and the hair strands stay with it.

The hair object’s centre, however, stays at the global origin which matters for deformations.
Modifying the Hair Emitter

After you apply hair to the emitter object, you can change its topology, such as changing its subdivisions, or adding and deleting polygons or points.

Before you make any modifications, however, you should freeze the hair’s operating stack (see Progressive Styling by Freezing the Stack on page 40) to keep the current hair style and turn off dynamics (see Making Hair Move with Dynamics on page 63). This way, the hair looks for the guide hair at the closest vertex and bases the creation of new guide hairs on that one instead of using the default guide hair state.

Scaling the Hair Emitter

When you scale the emitter, the hair scales with it because the hair object’s transformation is pose-constrained (which constrains scaling, rotation, and translation) to the emitter’s transformation.

If you are scaling the emitter in non-uniform scaling mode, you may find that the hair styling will look distorted. For example, if the emitter is squashed down on the Y axis, the hair appears to stretch as you pull it horizontally and shrink when you pull it vertically. You should probably avoid using this scaling mode with hair.

Deleting Hair

When things get too hairy, you can delete the hair object by selecting either it or its hair generator operator and pressing Delete. This does not remove the hair emitter object.
Emitting Hair from Objects and Clusters

You can emit hair from polygon meshes, selected polygons, polygon clusters, NURBS surfaces and subsurfaces.

You cannot emit hair from point clusters, either on polygon or NURBS objects.

Polygon Mesh Objects

For polygon meshes, the resolution of the object determines how many guide hairs are created—one per vertex. This has an impact on the final look so make sure that the resolution is what you want.

One guide (styling) hair is created at each polygon vertex.

As a result, there are more guide hairs around the face and paws where there are more polygons.
Emitting Hair from Objects and Clusters

**Individual Polygons or Clusters of Polygons**

With individual polygons or clusters, you can emit hairs from only selected areas of the object. If you select individual polygons, a cluster containing them is automatically created when you apply hair. As with complete polygon mesh objects, one guide hair is created at each polygon vertex.

The rendered hair starts abruptly at the edges of selected polygons, so you may want to design your model with a hairline created from many smaller polygons. You can also use a density or cut map (see *Removing Render Hairs (Density)* on page 85 and *Changing the Length of Render Hairs* on page 89) to attenuate the density of the hair growth or the hair length near the hairline.
Whole Object versus Clusters

Emitting hair from a smaller area on an object (using clusters) creates thinner and shorter hairs than emitting hair from the whole object. This is because the bounding box of the cluster is smaller than the bounding box of the entire object.

By default, the hair thickness and length is proportional to the size of the emitting surface's bounding box. This means that a large object with a scaling of 1,1,1 will have bigger hairs than a small object with a scaling of 1,1,1.

However, you can change the hair thickness to remain as an absolute value (in Softimage units), regardless of the size of the emitter. This may be useful if you’re emitting from smaller clusters on an object but you want them to match the hair thickness of the whole object’s hair. For more information on this, see Setting the Render Hair Thickness on page 86.

Extracted Polygons

You can extract polygons from an object to make a new object from which to emit hair. This may increase the performance of hair because the number of polygons in relation to the overall object is fewer. However, because the extracted polygons are a new object, you’ll have to add a texture to them: they do not inherit the texture from the object from which they were extracted.

NURBS Surfaces and Subsurfaces

You can emit hair from any NURBS surfaces or subsurfaces, as well as from clusters of NURBS subsurfaces. As with polygon mesh objects, one guide hair is created per vertex. Fewer guide hairs mean easier styling and better dynamics performance.
Applying Hair to Objects and Clusters

You can create, edit and render as many instances of hair as you like in a scene. When you create hair, a hair primitive object and its hair generator are created for each object or component to which you have applied hair.

To apply hair to an object or cluster

1. Select the object, cluster, or polygons from which you want to emit hair.

   If you branch-select an object, hair is applied only to the parent, not to the whole branch.

2. Choose Create > Hair > from Selection from the Hair toolbar.

   Guide hairs appear on the emitter object or cluster, sticking out in the direction of the object’s normals. One guide hair is created at each vertex.

   Guide hairs are curves that you use to style the hair—styling operations are described in Styling and Animating Guide Hairs on page 37.

Once you have generated the hair, you cannot change the number of guide hairs on the object.

The Hair property editor also opens as soon as you create hair. This lets you view the render hairs and set their properties—see Viewing the Hairs on page 34 for more information.
Emitting Hair from Subdivision Surfaces

If you created hair on a polygon mesh that has a geometry-approximated subdivision surface, you can have the guide hairs emitted from this surface instead of the outer “hull.” The positions of the guide hairs on the subdivision surface correspond to the same point location on the hull.

When you apply a weight map or texture map to a subdivision surface (see Connecting Maps to Render Hair Parameters on page 79), the maps are evaluated on its surface at the guide hair locations, as with other polygon or NURBS objects.

If you want to use the hair-emitting subdivision surface as an obstacle for its own hair, see Tips for Setting Up Obstacles for Hair on page 68.

To emit hair from a subdivision surface

1. Select a polygon mesh and create a geometry approximation property for it by choosing Get > Property > Geometry Approximation.

   For more information about subdivision surfaces, see Applying Geometry Approximation on page 171 in the Polygon Modeling guide.

2. In the object’s Geometry Approximation property editor, click the Polygon Mesh tab and set the Subdivision > OGL and/or Render Level.

3. Apply hair to the polygon mesh, or to selected polygons or clusters on the mesh (choose Create > Hair > from Selection from the Hair toolbar).

4. In the Hair Generator Operator property page (appears at the bottom of the Hair property editor), set the level of the Emitter Subd Level slider.
For example, if you want the hairs to be directly upon the subdivision surface, set the level value here to match the number of subdivisions of the geometry approximation. Notice that the number of guide hairs does not change.

When you change the subdivision level, it is considered a topology change and may slightly modify the shape of the guide hairs.

From the Geometry Approximation property editor, drag+drop the Render Level (or OGL Level) parameter’s animation icon to the Emitter Subd Level parameter’s animation icon to create an “equal to” expression. Then whenever you change the Render Level (or OGL Level) value, the hair will be emitted from that same level of subdivision.

For more information, see Creating Expressions on page 353 in the Animation guide.
Creating Hair between Curves

 Emitting hair from curves is a little different than from other objects because there isn’t a surface from which they’re emitted. Instead, the hairs are created between three or more curves along their length, resulting in “ribbons” of hair. No guide hairs are created when you create hair from curves: only render hairs.

 While you can’t make a volume of hair as you do from other objects, hair from curves is useful for creating things like hair strands, pony tails, eye lashes, or seaweed.

 When you generate hair from curves, the number of vertices on the curve determines how smooth the resulting hair strands are. One hair “guide” point is created for each vertex on the curve. If there are too few segments, the hair may look too “chunky.” In this case, you may need to add more vertices to the curves before generating hair from them.

 Parameters that use weight and texture maps (see Connecting Maps to Render Hair Parameters on page 79), such as Cut Map and Density Map, are not directly supported for hair from curves because there is no surface upon which to project a map. However, you can create an interpolated surface for texturing hair from curves by attaching some shaders to the hair’s render tree. See Creating a Surface for Texturing Curve-Based Hair on page 112 for more information.

 To create hair from curves

 1. Create at least three curves, each with a minimum of three points.

 2. Select the curves in a consecutive order. This is important because the hair is interpolated between each adjacent pair of curves in the order in which you select them.

    To help select the curves, use the free-form selection tool (press F9) or any selection tool that lets you pick in a definite order. Using a rectangle selection tool selects many curves at once and may not produce the correct interpolation.

 3. With the curves selected, choose Create > Hair > from Curves from the Hair panel.
Creating Hair between Curves

The render hairs are created between the curves. The curves themselves are the guides that you modify to style the hair.

Animating Hair from Curves

No guide hairs are created when you create hair from curves: only render hairs. As a result, you can’t manipulate or style the hairs using the standard styling operators for guide hairs, but you can change the hair by deforming the curves themselves (moving points, using deformers, etc.).

When you animate the curves, they are reparameterized on the fly, so if there’s any kind of stretching on the source curve, the hairs may jump around. Using more subdivisions on the curve minimizes this problem.

If you animate hair created from curves using a deformation (such as a lattice, envelope, etc.), the render hairs may appear to shift or “pop,” especially if they have kink or frizz. To resolve this, set the Stiffness parameter value to 1 (on the Dynamics page in the Hair property editor). This means you can’t use dynamics on the hair, but the shifting will be resolved.

Hair created from curves cannot collide with obstacles.
Chapter 2 • Creating Hair

Viewing the Hairs

Hair is actually represented by two types of hairs: *guide* hairs (geometry) for styling and *render* hairs for viewing and rendering. You can display either the guide or render hairs (or both) in a viewport.

Render hairs are the “filler” hairs that are generated from and interpolated between the guide hairs. And as their name implies, render hairs are the hairs that are actually rendered. Color information is consistent with what you set in the shaders that are connected to the hair, and much tweaking of hair properties can be done without needing to use a render region.

Guide hairs are displayed when you select the hair object, regardless of the display type that is set.

For information on styling the guide hairs, see Chapter 3: Styling and Animating Guide Hairs on page 37; for information about render hairs, see Chapter 4: Getting the Look with Render Hairs on page 77.

To view guide or render hairs

- Select Guide Hairs as the Display Type in the Hair property editor to display only the guide hairs. This speeds up the display and is useful when you’re styling the hair.

- Select Render Hairs as the Display Type to display only the render hairs.

Then set the Render hairs % value to view only a certain percentage of the render hairs (the default is 10%) in the viewport.
This value is a percentage of the **Total Hairs** value (6500 by default) in the Render Settings section of the Hair property editor (see *Setting the Number of Hairs Rendered* on page 81). Increasing the percentage gives you a better sense of what the final results will be, but also slows down the interaction speed.

Remember that the **Render hairs %** value in the Display Settings is the value only for viewing the render hairs in the viewport. To determine the percentage of hairs rendered, you must set the value for the **Render Hairs%** parameter in the Render Settings area as described on *Rendering a Percentage of Hairs* on page 82.

Whatever percentage and display option you set for the render hairs in the Hair property editor is reflected in the **Render Hairs** menu on the Hair toolbar, as shown here.

- Select **Use Display Mode** to view the hairs using whatever display type is set in each viewport. Guide hairs appear in the Wireframe, Depth Cue, and Hidden Line Removal type, and render hairs appear in the Constant, Shaded, Textured, Textured Decal, and Realtime Shader type (the OpenGL display types).
This can speed up the interaction and reduce visual clutter since you won’t have render hairs in every view.

You can view the render hairs in the OpenGL display types using geometry approximation. This can speed up displaying hair if you set the hair’s subdivision level to a low value. See Setting the Geometry Approximation on page 127 for more information.

**Hiding Hair**

To hide or display hair in a viewport

- Toggle the Hair option in the viewport’s view menu (click the eye icon).
  
  or

- Select Hair on the Objects page in the Camera Visibility property editor (press Shift+s or choose Visibility Options in the viewport’s view menu).
Chapter 3  

Styling and Animating Guide Hairs

When you’re styling, you always work with the guide hairs: these are the styling hairs that are similar to and behave like segmented skeleton chains. In fact, the most intuitive way to style hair is to grab a tip and position it the same way you would the end effector on an IK (inverse kinematics) chain.

Guide hairs are curves (geometrical objects), which means that you can perform many of the standard XSI operations on them as you can with any other geometry (such as transforming or deforming). Guide hairs are composed of 15 segments, regardless of their length. Even when you “cut” the guide hairs (see Cutting the Hairs on page 51), the hairs are actually scaled and keep the same number of segments.
About the Styling Operators

Each styling tool is an operator that gets added to the hair object’s stack, as displayed in the explorer. This includes the Scale, Rotate, and Translate operators on the Transform panel (indicated by a MoveComponent node). When you select and use a styling tool, it is combined “on top of” the previous results.

As with other operators in XSI, you can mute them, delete and undo operations, as well as freeze the hair’s operator stack (see Progressive Styling by Freezing the Stack on page 40). In fact, to work more efficiently, you will probably want to freeze the hair’s stack frequently after styling.

Editing Styling Properties

When you use a styling tool, its property editor is added as a page to the Hair property editor—you won’t see the editor automatically appear on its own as with many XSI operators. The parameters in these property editors let you mute the operator, as well as further refine some of the operations. Of course, you can also open the operator’s property editor by clicking on the operator’s icon in the explorer.

Styling and Dynamics

Dynamics allows the hair to be affected by the movement of the hair emitter object and natural forces. You can also use dynamics as a styling tool by freezing the hair when it’s at a state that you like. For example, apply dynamics, apply some wind to the hair, then freeze the hair when it has the right wind-swept look.

If you have applied dynamics to the hair, you cannot use the style operators on the hair unless you mute the dynamics operator, as described in Muting the Dynamics Operator for Styling on page 67.

If you have animated the parameters of certain styling operators (Comb, Clump, Puff, Rotate, and Scale), you cannot then use dynamics on the hair.
Using Deformations and Obstacles as Styling Tools

Because guide hairs are actual geometry, you can use all of the standard Deform operators on them (select all, specific strands, tips, or specific points along the strand) to come up with some groovy hairdos!

Lattices, envelopes, deform by cluster center, randomize, and deform by volume usually produce the most favorable results. However, if you animate the deformations, you cannot then use dynamics on the hair.

As well, you can use obstacle collisions (with or without having dynamics active) to mold the hair around the obstacle to style it. See Setting Up Hair Collisions on page 68 for more information.

Scripted Operators and Hair

When using a scripted operator on hair geometry, you can change point positions but you cannot change the distance between points on a strand or scale a strand. The condition that successive points on a strand are equidistant takes precedence over any manipulations you perform in the scripted operator.

To have smoother animation of deformed hair, you can activate Stretchy mode to let the hair segment lengths adjust to the deformation. See Stretching Hair Segments on page 71 for more information.
Progressive Styling by Freezing the Stack

When you use a hair styling tool, its operator gets added to the hair object’s stack, as you can see in the explorer. You can freeze the hair object’s operator stack as you would any object in XSI (see Freezing the Operator Stack on page 54 in the Modeling and Deformation Basics guide for more information).

Freezing collapses all the styling operators (including any Scale, Rotate, or Translate operators from the Transform panel) into the hair generator operator and keeps the hair’s state as the base state.

You can also use the Immediate mode when styling hair, which doesn’t keep any operators in the stack. See Immediate Mode on page 64 in the Modeling and Deformation Basics guide.

Freezing the stack lets you style the hair progressively, taking advantage of the dynamics at each new hair state. For example, do some styling, add dynamics and move the hair around to enhance the styling, and then freeze the stack. The results, taking into account the dynamics and any styling operators that are present, are “baked” into a new hair state that becomes the current state. With this new state, you can continue building the style, and dynamics are applied relative to the new hair state.
To freeze the hair's stack

- Select the hair object and click the Freeze button in the Edit panel of the main command panel.

  All the styling operators are deleted, including any Scale, Rotate, or Translate operators from the Transform panel as contained in the MoveComponent operator node. If dynamics is applied to the hair, its operator is also removed unless it’s muted.

  The hair generator operator remains because it is required to generate hair (it is freeze-resistant!).

  Also, any points that were locked remain locked even though their Lock operator is removed. This is because the locked state was baked into the base state.
Selecting Guide Hairs

Before you style, you must select the guide hairs. Using the buttons on the Select panel, you can select hair in different ways: strands, tips, or points along the strand. For many of the hair styling tools (such as Comb, Puff, or Scale), it doesn’t matter which hair components you have selected. However, using the Transformation tools (Scale, Rotate, and Translate) and the Brush tool, the styling results can be very different depending on what is selected.

With the hair object selected, use any of these options on the Select panel:

- **Strand**—Select the hair strands (all points on the strand) by dragging over any part of the strand. This tool simply selects all the points on the hair strand as a convenience—there is no such thing as a strand “component.”

- **Strand By Root**—Is your hair a tangled mess and you want to select a strand without trying to untangle? Select Strand By Root, drag near the surface of the emitter object, and the whole strand is selected.

- **Tip**—Select just the tips of the strands. You can also press the ‘ (single apostrophe) key to activate this tool. Drag over any part of the hair strand, and only its tip is selected. This tool is simply a convenient way for selecting the end points of the hair: tips are no different than any other point on a hair strand.

   When you use a styling operator after selecting Tip, press Alt+spacebar to return to the Tip selection tool. This is the same as clicking the big green selection arrow on the Select panel.
Selecting Guide Hairs

• **Point**—Select points on the hair strand as on any other object in XSI. With this option, you can see the segments on the hair strand (always 15).

To move the point selection up and down the strand one segment at a time, press Alt+left or right arrow. This is handy, for example, to select the point on the strand before cutting it.

When you select points on the hair, the strand and point information is shown in the **Selection** text box. For example: `pnt [ (2, 3), (2, 4) ]` means that points 3 and 4 on strand 2 are selected.

---

**Creating Clusters of Hair Points**

You can create clusters of different points, tips, or strands of hair like any other geometry component in XSI. To do so, select the components you want in the cluster and click the **Cluster** button in the Edit panel. When you create the clusters, they appear in the Clusters folder under the Hair object. For more information on clusters, see *Clusters* on page 139 in the Scene Elements guide.

Once you have a cluster, you can use it as you would any other cluster, such as by constraining objects to it using the **Object to Cluster** constraint (see *Object to Cluster Constraints* on page 306 in the Animation guide).
Brushing and Combing Hair

As in real life, brushing and combing hair in the virtual world is a basic requirement for good grooming. Brushing uses a proportional modeling brush that lets you easily sculpt hair in any direction you like. You can comb the guide hairs in different directions according to the emitter object’s normals, the three axes, or away from the camera in a viewport. As well, you can recomb hair to “tidy up” hairs while keeping the guide hairs in the same approximate direction in which they are already going.

Brushing Hair

Brushing grooms the hair with a proportional modeling brush, allowing you to intuitively sculpt the hair in an overall manner. With proportional modeling, hair points closest to the brush’s cursor (the crosshairs at the center of the brush radius) are affected the most while those farther away are affected less, according to the distance falloff.

The brush tool uses the standard proportional modeling settings to control things such as the distance limit for affected points and the shape of the falloff profile. To modify these settings, open the Proportional property editor by right-clicking the Prop button on the Transform panel.

The brush can be used on other types of geometry besides hair. For information, see Using Proportional Modeling on page 117 in the Modeling and Deformation Basics guide.

To brush hair

1. Set the selection filter according to the parts of the hair you want to brush: tips, points, or strands. You can also select the whole hair object itself to brush all hairs.

2. Select the hair components you want to brush: only the selected hairs are affected by the brush. For example, select the tips on one side of the object to groom only those hairs while leaving the hairs on the other side unaffected.

If nothing is selected, then all components inside the brush radius are modified.
3. Click the Brush button on the Hair toolbar: a brush pointer with a radius around it appears.

![Brush tool center with radius around it.]

Scroll the mouse wheel or press R and drag to the left or right to resize the radius.

![When you click, the points that are influenced the most are displayed in bright red. The points' colors darken to show the falloff from the brush's center.]

4. To change the brush’s radius, scroll the mouse wheel, or press R and drag to the right to make the radius larger or to the left to make it smaller. The size you set is persisted so that it’s the same the next time you activate the brush tool.

5. Position the center of the brush where you want the strongest brushing effect to be, then click and drag to brush the hair:
   - Left-click and drag to translate the hair in the direction you want.
   - Middle-click and drag to rotate the hair in a clockwise or counterclockwise direction. This is especially useful with tips selected.
   - Right-click and drag to scale the hair toward or away from the brush’s center: drag to the right to splay (fan) apart the hairs, or to the left to bring (clump) the hairs together.
Chapter 3 • Styling and Animating Guide Hairs

6. Click and drag in another area of the hair to progressively sculpt the shape of the hair as you like.

Combing Hair

You can comb the guide hairs in different directions according to the emitter object’s normals, the three axes, or away from the camera in a viewport.

To comb hair

1. Select at least one point on each of the guide hairs, or select the hair object itself to comb all hairs.

2. Choose one of these commands from the Comb menu in the Hair toolbar:
   - Along Normals: Makes the hairs stand on end in the direction of the emitter object’s normals, as is the case with the initial groom state when you create hair. This command uses the Hair Puff operator (see Puffing and Straightening Hair on page 56).
- **Along X, Y, Z or Negative X, Y, Z Axis**: Combs in the positive or negative X, Y, and Z axes directions in global space.

- **Away from Cameras**: These combing patterns are based on the four viewports (labelled A, B, C, D in their upper-left corners). They each comb away from the camera currently assigned to that viewport. The most common way to use this is to orbit the camera and then comb; for example, orbit the camera in the B viewport and choose Comb > Away from Camera View B.

3. To comb more accurately, the Comb property editor contains two parameters of use:

   - **Comb Amount** lets you specify the amount of combing you want, where 1 is fully combed and 0 is the previous state without any combing, thus allowing you to blend between the previous and combed states.

   You can animate the Comb Amount to create movement on the hair; however, if you do this, you cannot use dynamics on the hair.
- The X, Y, Z options let you select a vector that defines the combing direction.

**Tips for Combing**

- After combing, you may need to use the Puff > At Root command to lift the hairs away from the hair emitter’s surface (see Puffing and Straightening Hair on page 56).
- To “comb” a part between the hairs, use the Interp > Split command (see Interpolating Between Guide Hairs on page 58).
- Translating strands of hair on different axes also “combs” the hair in different directions (see Shaping the Hair with Transformations on page 53).

**Recombing Hair**

You can adjust the hair styling by “recombing” the guide hairs. Recombing combs each hair along the emitter’s surface but keeps the guide hair in the approximate direction in which it is already going (using the root-to-tip vector).

For example, if you've carefully styled the guide hairs so that they are all pointing in various directions (such as when styling fur or short, messy hair), you can recomb the hairs so that they lie along the emitter, but the general shape of the hair style is unchanged.

To recomb hair

1. Select at least one point on each of the guide hairs, or select the hair object itself to comb all hairs.
2. Choose the Modify > Comb > Recomb command in the Hair toolbar.
- This command applies a Hair Recomb Operator but does not open its property editor.

This property editor contains the **Comb Amount** parameter. This lets you specify the amount of combing you want, where 1 is fully combed and 0 is the previous state without any combing, thus allowing you to blend between the previous and combed states.

You can animate the Comb Amount to create movement on the hair; however, if you do this, you cannot use dynamics on the hair.
Chapter 3 • Styling and Animating Guide Hairs

Changing the Hair Length

There are different ways in which you can change the length of the guide hairs: using the Hair Scale tool, attenuating, and cutting. You can also reset the hair length.

- You can drag selected hair points or tips in Stretchy mode (see Stretching Hair Segments on page 71) to quickly change the guide hair length.

- You can randomly scale the render hair length by using the Random Scale parameter in the Hair property editor (see Changing the Length of Render Hairs on page 89). This doesn’t affect the length of the guide hairs.

Scaling the Hairs

You can change the length of guide hairs using the hair Scale tool.

To change the hair length

1. Select the hair strands, tips, or points on the hair object (make sure the hair object itself isn’t selected). The points on the guide hairs are always equidistant, so you really only need to select one point on a strand to scale the whole thing.

2. Click the Scale button on the Hair toolbar, then click and drag in the viewport. This creates a Scale operator for the hair and a property editor.

3. Do one of the following to change the hair length:
   - Drag the mouse to the right to lengthen the hairs and to the left to shorten them.
   
   or
   
   - Open the Hair Scale Operator property editor and drag the Scale Factor slider to set the length precisely.

   Each time you click and drag, a new Scale operator is applied. The results are cumulative so that an operator that scales by a factor of 2 followed by one that scales by 0.5 negates the scale.

   You can animate the Scale factor to create movement on the hair; however, if you do this, you cannot use dynamics on the hair.

   If you scale down too much and can’t see the strands anymore, select the hair object and choose the Pop Zero Hairs command (see Resetting the Hairs on page 52).
Changing the Hair Length

Removing Hair

There are two different ways to remove hair, such as for creating bald spots, shaved patterns in hair, or mowed patterns in grass. You can either:

- Scale the selected guide hair strands down to 0.

or

- Connect texture or weight maps to the Density Map (see Removing Render Hairs (Density) on page 85) or Cut Map parameters (Changing the Length of Render Hairs on page 89).

Either of these parameters are especially useful for quickly creating patterns of hair removal. Density Map determines the number of render hairs based on the map that’s connected to it (it essentially prevents render hairs from being displayed), and Cut Map determines the render hair length.

Attenuating the Length

You can also scale the guide hairs proportionately to the size of corresponding polygon on the emitter object: smaller polygons produce shorter guide hairs.

Attenuation is good for fur because you usually need the hairs to be shorter in the areas of high detail, and longer in areas of low detail. Choose the Attenuate command a few times in a row to initially set the length, then adjust by scaling the guide hairs to the exact length you want.

To attenuate the hair length

1. Select at least one point on each of the strands you want to attenuate, or select the hair object to attenuate all strands.

2. Choose Modify > Length > Attenuate on the Hair toolbar. Choose this command repeatedly to iteratively alter the hairs (middle-click on the Length button to do this easily).

Cutting the Hairs

You can cut (shorten) the guide hairs similar to what you would do in real life: pick a point and snip! Although it appears that the guide hairs are cut, no segments are actually removed or harmed during this operation; instead, they are scaled to the length you’ve specified and then refitted to that portion. This makes it possible to undo a hair cut that was too short, something we would have all liked to have done at some point in our lives!

You can also alter the render hair length by attaching a texture or weight map to the Cut Map parameter in the Hair property editor (see Changing the Length of Render Hairs on page 89).

To cut guide hairs

1. Select a point on the hair strand where you want to cut, or select the hair object itself.

Press Alt+left or right arrow key to move up or down the strand to select a point before cutting.
Chapter 3 • Styling and Animating Guide Hairs

2. Choose Modify > Cut. The hair is cut at the selected point, and that point remains selected relative to the length of the strand so that you can keep on cutting. If you selected the hair object, one hair segment is cut from each strands’ end.

[Selected point in red where guide hair is to be cut.]

[Guide hair after cutting.]

3. If the cut wasn’t exactly what you had in mind, you can change it. In the Hair Cut property editor, you can set the Offset parameter to add a relative offset to the cut, allowing you to change the hair point that was selected and cut. For example, an offset of -1 cuts at the point before the one you selected (resulting in shorter hair), and a offset of 1 cuts at the point after.

Resetting the Hairs

To reset hairs along the normals

• Select hair strands and choose Modify > Length > Pop Selected. This resets the hair, returning the hairs to their default state (sticking out in the direction of the emitter object’s normals) at the default length.

• Select the hair object and choose Modify > Length > Pop Zero Hairs to lengthen hairs that have been scaled down to zero and can’t be selected (useful if you’ve gotten a little overzealous with the cutting!). This also makes the hairs stick out in the direction of the emitter object’s normals.

In the Hair Pop Operator property editor, you can set the Zero Threshold value to choose at what point (in Softimage units) that “zero hairs” will pop out. This makes it possible for hairs that are almost zero to be popped.
Shaping the Hair with Transformations

As with other geometry in XSI, guide hairs can be scaled, translated, and rotated using the standard tools found in the Transform panel. For rotating, you have additional options using the commands in the Rotate menu on the Hair toolbar.

Translating Hairs

1. Select the hair strands, tips, or points on the hair object (don’t select the hair object itself).
2. Select the appropriate Translate tool (X, Y, Z) on the Transform panel.
3. Drag the mouse along the axis you want to translate.

The selected elements move in the direction you drag. When you move the tips, the results are similar to moving the effector on a multi-segmented IK chain.

Rotating Hairs

You can rotate hairs in two different ways: with the regular Rotate tool on the Transform panel or with the Rotate tools on the Hair toolbar. Note that points on the hair may not always rotate in a perfect arc because of the way the fixed-length segments on the guide hairs are connected.

1. Select the hair strands, tips, or points, then select the appropriate Rotate tool (X, Y, Z) on the Transform panel and drag the mouse.
2. Or
• Select the hair tips or points, then choose a command from the **Rotate** menu on the Hair toolbar:

  - **Around Root** rotates strands, tips, or points around their roots perpendicular to the emitter object’s normal. Drag the mouse to the left or right: the farther you drag, the more the points are rotated.

  - **Around Tip** rotates points around a vector from the root to the tip. This is useful for setting the orientation of a strand. Drag the mouse to the left or right: the farther you drag, the more the points are rotated.

  - **Around Cursor** rotates tips or points around the initial mouse pointer position in the viewing plane. In a viewport, click at the point around which you want to rotate the hairs, and then drag the mouse to the left or right. The farther you drag, the more the points are rotated.

With any of the Rotate operators, you can have more control by setting the **Rotation Angle** value (in degrees) in the Hair Rotate property editor. You can also animate this option to create movement on the hair, but then you can’t use dynamics on the hair.

**Scaling Hairs**

Scaling hairs using the Scale tool on the Transform panel works on hair as it does on any geometry component, such as points. However, if you want to change the length of the hair, use the Hair Scale tool on the Hair toolbar (see *Scaling the Hairs* on page 50).

**To scale hairs**

1. Select the hair strands, tips, or points on the hair object (don’t select the hair object itself).

2. Select the appropriate Scale tool (X, Y, Z) on the Transform panel.

3. Drag the mouse in the direction you want to scale.
The selected hair components move in the same way that scaling moves other object components, such as points on a polygon mesh. The hair strand (and each segment of the hair) does not change length.
Puffing and Straightening Hair

There are times when you just need a little more volume in your hair, or you need to straighten it out. For example, after you comb the hair in a downward direction (negative Y), you may need to give a little lift (puff) to the roots at the scalp.

The two commands in the Puff menu on the Hair toolbar can help you with these styling issues.

To puff hair at the roots

- Select some hair strands or points and choose Puff > At Root on the Hair panel.

As you drag the mouse, the hair straightens (puffs) out starting at the root, moving it away from the emitter object’s surface. The farther you drag, the more the points are straightened.

To straighten hairs

- Select some hair strands or points and choose Puff > Stand on End on the Hair panel.

As you drag the mouse, the hair straightens out, starting at the tip. The further you drag, the more the points are straightened.

For either of these Puff operators, you can change the amount of straightening that happens by setting the Puff Factor in the Hair Puff property editor. A value of 0 results in no change, and 1 results in completely straight hairs.

You can animate the Puff Factor to create movement on the hair; however, if you do this, you cannot use dynamics on the hair.
Clumping Hairs Together or Fanning Them Apart

There are certain hair styles that require a good gel or mousse to make the hair stick together in clumps. You may also need hair to clump together for making eyelashes or animal whiskers. Likewise, you may need to do the reverse: to fan the hairs apart.

Using the Clump command on the Hair toolbar, you can do either operation, depending on which direction you drag the mouse. Clumping moves points on guide hairs either toward or away from their average center. For this reason, you’ll have the best results when points are fairly close together.

You can create the wet look on render hairs by using Multiplicity > Splay by Tips in the Hair property editor (see Increasing the Number of Render Hairs on page 83).

To clump or fan hair

1. Select some points on hair strands (such as the tips on several strands that you want to clump together).

2. Choose the Clump command on the Hair toolbar. Drag the mouse to the right to clump the points together, or to the left to fan the points apart. The farther you drag the mouse, the more the points are displaced.

You can also set the Clump Factor in the Hair Clump property editor to determine the way in which the points are moved. Positive values move the points toward their average center, and negative values move them away from their average center. You can animate the Clump Factor to create movement on the hair; however, if you do this, you cannot use dynamics on the hair.
Interpolating Between Guide Hairs

There are hair styles that are easier to create when you can use groups of hairs. The commands in the Interp menu on the Hair toolbar let you decide the relationship between groups of guide hairs: you can split groups of them apart (such as for creating parts in the hair), merge them together, or have them all completely separate with no interpolation.

When you set the guide hair interpolation, the render hairs associated with each guide hair are also interpolated accordingly. For example, if you split one group of guide hairs from another, half the render hairs between the guides follow one guide and the other half follow the other guide.

To see the results of these commands, make sure that you have the render hairs displayed.

Splitting Hairs

To split hairs apart

1. Select a group of guide hairs (or at least one point on each of them) that you want to split off from the rest, such as for creating a part.

2. Choose Interp > Split. This creates an interpolation group of these hairs.

You’ll see that this group’s render hairs do not interpolate with the others.
Merging Hairs

To merge hairs together

1. Select all the guide hairs (or at least one point on each of them) that you want to bring together.

For example, you may want to do this after you’ve created a part with some guide hairs on the top of the head, but you want the hairs on the back of the head to be together.

2. Choose Interp > Merge. This activates the interpolation of render hairs between guide hairs that have been previously split or shattered.

When guide hairs are merged, an asterisk is displayed next to the interpolation group ID, when the Group IDs are visible (see Viewing the Hair Group IDs on page 60).

Separating (Shattering) Hairs

To keep all render hairs separate

1. Select all the guide hairs (or at least one point on each of them) that you want to keep as islands unto themselves with no interpolation. For example, this is useful for creating porcupine quills or spiky gelled hair.

2. Choose Interp > Shatter. This separates each selected guide hair into its own interpolation group.
You'll see that the render hairs associated with each of their guide hairs do not interpolate with the others.

Viewing the Hair Group IDs

To see the guide hair assignment

1. Press Shift+s or click the eye icon in a viewport and choose Visibility Options.
2. In the Camera Visibility property editor, select the Hair Interpolation Group IDs option on the Attribute page.

This shows which guide hairs belong to which interpolation group. The group's ID is shown at the base of each hair with an asterisk beside the number stating that the hair has been merged.
Locking Selected Points on Strands

You can lock selected points of the hair strands to the emitter object’s surface, letting all the unselected ones roam free. For example, you can use this to create ponytails by fixing the points on the hair strands where the hair would be held together by a rubber band. You may also want to temporarily lock some hair points (such as tips) when styling to change the effect of the style operator on the hair. Then unlock the points after you’ve finished styling.

The locked points move with the emitter object, and you can move the locked points themselves, but the points stay locked down in relation to the strand when you apply dynamics or styling operators. The selected points are locked in space relative to the closest polygon on the hair emitter object.

To lock points on hair

1. Select some points on hair strands and choose Modify > Surf. Lock > Lock on the Hair toolbar. By default, locked points are displayed with a little blue padlock icon.

2. Add dynamics to the hair or apply a styling operator. The locked points are not affected, but all other points on the hair strand can move freely.

For a ponytail, you may also want to lock down all points for hair that are near the head so that they’re not affected by dynamics.
To unlock hair points

1. Select either the hair or just the locked points that you want to unlock.
2. Choose Modify > Surf. Lock > Unlock. This makes them available to be affected by dynamics and styling.

If you scale hair tips with locked points, the hair snaps back. Make sure to unlock any points before scaling, then relock the points, if necessary.

To hide or display the hair lock icons

1. Press Shift+s or click the eye icon in a viewport and choose Visibility Options.
2. In the Camera Visibility property editor, toggle the Locked Hair Points option on the Attribute page.
Making Hair Move with Dynamics

By applying dynamics to hair, you can have it respond to the movement of the hair emitter object, as well as any natural forces or obstacles you set up for it. You can play back the dynamics live while you’re tweaking to get just the right motion, or write the simulation to cache files and then read them for faster playback and consistent rendering, even when skipping frames.

When you apply dynamics to hair, you make it possible for the hair to move according to the velocity of the hair emitter object, like long hair whipping around as a character turns her head quickly. The dynamics calculations also take into account any natural forces applied to hair such as gravity or wind (see Chapter 2: Natural Forces on page 23), as well as any collisions of the hair with obstacles (see Chapter 3: Collisions with Obstacles on page 37).

- You need to apply dynamics to hair before you can apply a natural force to it.
- When the dynamics operator is active, you cannot do any styling on the hair. To get back into styling mode, see Muting the Dynamics Operator for Styling on page 67.
- If you have animated the parameters of certain style operators (Clump, Comb, Puff, Rotate, or Scale) or have animated deformations on the hair, you cannot also use dynamics.

Applying Dynamics to Hair

To apply dynamics to hair

1. Select the hair object.
   
   If you’re applying dynamics to hair created from curves, make sure to select the “hair curves,” not the source curves from which you generated the hair.

2. Choose Create > Dynamics on the Hair toolbar.
   
   When you choose this command, a hair dynamics operator is added to the hair object’s stack (above the hair generator operator), and the Dynamics Operator property editor appears in which you can cache files for faster playback (see the next section).

   For information on the Sensitivity to Forces parameters in this property editor, see Setting Up Forces for Hair on page 74.

3. On the timeline, select the Loop icon and click Play. By default, the dynamics are played back in live mode.

4. Move the hair emitter object around or animate its transformations, apply a natural force to the hair, or have hair collide with obstacles.

Hair • 63
You can also adjust the Dynamics parameters to determine how the guide hairs react to the effect of dynamics. See *Adjusting the Hair’s Reaction to Dynamics* on page 65 for information.

**Caching Files for Playback**

You can cache dynamic simulation files for faster playback using the options in the Hair Dynamics Operator property editor. Caching the simulation to file means you can move to any frame and get an update, as well as play the simulation backward. It also provides for consistent rendering results and motion blur.

By default, the *Cache Mode* is set to *Live* (no caching) so that you can work with the time looped and have the hair update constantly based on any changes you make to the hair parameters.

- If you keep the cache files, make sure to check the storage folder you have defined for them and delete any unused files. These can build up and take up lots of space if you don’t occasionally do some housecleaning!
- To increase the speed of the playback, mute any viewport that you’re not using by middle-clicking its letter (A, B, C, or D).

**To write to and read from cache files**

1. Select *Read & Write* as the *Cache Mode*. When you change a hair parameter, the simulation is reset and the cache is cleaned, but nothing is recomputed until you change frames.

2. Enter a location and name for the cache file. By default, it writes to the Simulation folder of the current project. If you select *Usr*, the path is displayed as you entered it; if you select *Res*, the resolved path is displayed.

3. Play the simulation to cache the files.

4. Click **Clear Cache** to delete the current cache files in the folder; that is, it won’t clear cache files with names different from what is currently set for the *Filename*. The dynamics are not reset when you delete the cache.

5. Click **Reset Dynamics** to reset the hair to its state at the time that dynamics was applied to the style.
Making Hair Move with Dynamics

To have the dynamics read from cache files

- Select Read only if you want the dynamics to read data from the current cache file, but not write new cache files, even when you make a change to the hair.

Adjusting the Hair’s Reaction to Dynamics

Once you have dynamics applied to the hair, you can set the parameters on the Dynamics page in the Hair property editor to determine how the hair reacts to the dynamics. These parameters are applied to the guide hairs.

The cache file is not automatically copied with the scene if you save it in a new project. Make sure to copy the cache files to the Simulation folder (or wherever you want them) in the new project so that you don’t lose valuable data!

The Dynamics parameters (except Wiggle) can have weight and texture maps connected to them, as identified by a little connection icon beside them. This allows you to create either specific patterns or random values for the hair, depending on the effect you want. For example, you can use a weight map on the Stiffness parameter so that the hair is stiffer in specific areas, such as where the hair is shorter.

Because these parameters only affect guide hairs, subtle value variations in the maps be lost, so you may find weight maps to be generally more effective than texture maps.

- For general information about mappable parameters, see Connecting Maps to Render Hair Parameters on page 79.
- For an example on how to connect a weight map to a hair parameter, see Cutting the Render Hairs on page 89.

Stiffness

Stiffness is the tendency of the hair to hold its rest shape and resist the dynamics that are affecting it, rather like a good hair spray! This is the stiffness of the guide hair starting from the tip and going down to the root. A value of 1 means that hair will always keep its styled shape and resist the effect of dynamics on it, while a value of 0 means no resistance to dynamics.
Root Stiffness is the same as Stiffness except that it determines the stiffness at the root of each guide hair. The values are the same as for Stiffness, with the default value of 1 meaning that the root always keeps itself aligned in the direction of the style.

You can use Stiffness and Root Stiffness together to have varying levels of stiffness along the length of the guide hair. For example, with a low Stiffness value and high Root Stiffness value, you can have the hair affected by dynamics only toward the tips.

Dampening

Dampening determines how quickly the hair recovers from the effect of dynamics. With a value is 0, the hair does not return to its position; with higher values (maximum value is 1), the hair bounces back into its styled position more quickly.

Wiggle (Random Displacement)

Wiggle simulates the effect of a light breeze, moving randomly about the points of the guide hairs. Its Amplitude controls the strength or amount of the point displacement, and its Speed controls how quickly or slowly the displacement occurs, like wheat stubble on a field quickly quivering in the wind or seaweed slowly swaying with the waves.

You can also animate the Frizz settings (also on the Effects page), which can give similar results as the Wiggle parameters except that it affects only the render hairs. Generally, this is more desirable because you don’t see any interpolation between the guide hairs. See Frizzling Out on page 94 for more information.
**Mutating the Dynamics Operator for Styling**

Once you've applied dynamics on the hair object, you cannot do any hair styling using the style operators on the Hair toolbar unless you mute the dynamics operator. This is because dynamics need to be calculated on a specific styling “state” that cannot change during playback.

If you style the hair with dynamics active, nothing will happen because no styling operator is added to the stack (and you'll see an error logged in the script editor).

**To mute the dynamics operator**

1. Do one of the following:
   - Click the **Style** button at the bottom of the Hair toolbar.
   - Select the **Mute** option in the Hair Dynamics Operator property editor.
   - Right-click on the Hair Dynamics Operator in the explorer and choose **Mute**.

2. Style the hair into a new state.

3. Unmute the dynamics operator.

Now when you play back the simulation, the dynamics operator uses this new “rest state” to update the dynamics.

- If you freeze the hair operating stack (see *Progressive Styling by Freezing the Stack* on page 40) with the dynamics operator is muted, the dynamics operator is not removed.
- When you reapply styling after dynamics, the hairs may have shifted a little. To get back to the “real” current hair state, click the Style button, click Reset Dynamics in the Dynamics Operator property editor, or move the hair emitter object.
Setting Up Hair Collisions

If you're creating medium or long hair, you will most likely need to set up obstacles with which the hair will collide. For short hair, fur, or short grass, you may not need any obstacle objects, depending on what your hair-emitter object is doing.

If you're creating a human head with long hair, the shoulders and neck need to be selected as obstacles; or if you have an animal run through a field of wheat, the animal would be considered the obstacle for the wheat.

Tips for Setting Up Obstacles for Hair

Here are some tips for setting up the objects that will be used as obstacles:

- You can use polygon or NURBS surface objects as obstacles, but you cannot use implicit objects or hair objects (that is, the actual hair strands) as obstacles. Obstacle objects can be either stationary or animated.

- You can use the hair emitter object itself as the obstacle for hair. For example, select the head and shoulders of a long-haired character to be the obstacles for the hair.

- If you're emitting hair from a geometry-approximated subdivided surface (see Emitting Hair from Subdivision Surfaces on page 30), you cannot also use it directly as the obstacle for its own hair.

To have a subdivided surface hair obstacle, create a subdivided polygon mesh (choose Create > Poly. Mesh > Subdivision from the Model toolbar) and set its subdivision level to what you need for the obstacle. Then set this subdivided polygon mesh as the obstacle (you may want to hide it).
• You may want to use polygons as obstacles to create more accurate
  collisions: a polygon's points lie directly on its surface unlike those of a
  NURBS surface object.

• For best performance, make the object have as few polygons as possible
  and still have the hair conform to its shape.

• Use an obstacle object that is large enough and closed so that the hairs do
  not intersect it. The obstacle should be bigger than the longest guide hair's
  segment length (that is, one of the hair's 15 segments).

For example, if the entire guide hair is 10 units long, but the obstacle is only
0.5 units in diameter, it won't work well, such as with the brim of a hat.

• Keep the obstacle object simple. For example, if you have a complex
  character with which you want the hair to collide, use only the body
  geometry of where the hair will actually collide as the obstacle (that is, you
  don't usually need to include the lower body and arms).

Otherwise, all these elements are calculated as part of the collision, which
slows things down considerably. You can, for example, tag only the
polygons from the back and neck area, extract the polygons, and use this
result as your collision obstacle instead.

• Use simple proxy objects to stand in for complex obstacle objects. For
  example, use a simple sphere scaled to the proportions of your character's
  torso as the obstacle object. This proxy can be hidden or set to non-
  rendering after being constrained to the real obstacle object.

• If an animated obstacle moves too much from one frame to the next, the
  hair may pass through it. Again, having an object large enough and closed
  can help avoid this problem.

Setting Up Objects as
Obstacles for Hair

To have hair collide with obstacles

1. You normally apply dynamics to the hair (see Making Hair Move with
   Dynamics on page 63), but you can also use obstacles without dynamics to
   use them as styling tools (the hairs stay outside the obstacles when it
   comes close to the hair).

2. Set the Hair Collision options on the General page the Hair property
   editor, depending on the type of collision you want to create:

   - Average Sphere (fast) creates a sphere around the obstacle object for the
     collision detection, giving fast calculations. This disregards the settings
     in the Obstacle property editor.

   - Exact (slow) uses the actual shape of the obstacle object for the collision
     detection, which is preferable for most collisions. This gives accurate
     results but may have a slower calculation time. This also disregards the
     settings in the Obstacle property editor.
- **XSI (fast, rough)** is the only method that detects obstacle collisions based on the settings in the Obstacle property editor (see *Setting Up Obstacles* on page 39). For example, if you have Actual Shape as the Obstacle Type and a Push Length value set for the obstacle, this is taken into consideration for the collision. This method is fairly fast and as accurate as the Actual Shape that you choose.

  - The Friction and Elasticity parameters in the Obstacle property editor are not used in calculating obstacle collisions with hair, even if you have XSI set as the hair’s Collision Type.
  - If you want to change the hair’s Collision Type after you’ve picked the obstacles, you must first freeze the hair. Changing the collision type is considered to be a topology change for hair.

3. From the Simulate toolbar or Hair toolbar, choose **Modify > Environment > Set Obstacle**.

4. Pick one or more objects in the scene that will act as obstacles for the hair. Right-click to end the picking session.

5. Set the general and physical parameters in the Obstacle property editor that appears (see *Setting Up Obstacles* on page 39) to control the collision’s behavior.

   An Obstacle property is created for the obstacle object. By deleting this property, you can disconnect the obstacle from the hair.
Stretching Hair Segments

Although hair segments are normally fixed in length, you can temporarily change their behavior in “stretchy” mode to allow them to change length. When you style or animate hair in stretchy mode, the hair strands behave more like curves than an IK chain with fixed segment lengths.

In stretchy mode, you can style hair using various deformations or transformations. Then when you resample the hairs, each hair segment is approximated in a uniform way over the length of the hair strands to smooth out the shape. After you resample, you can apply dynamics to the hair to animate it (see the following section, Styling Hair in Stretchy Mode).

If you apply dynamics on hair before you resample it, the “stretched” segments of the hair will return to their closest constrained position (that is, IK will be solved for the hair strands).

Stretchy mode allows each hair segment to change length in response to how an animated deformation (such as a lattice or envelope) changes the shape of the hair. Animating hair in stretchy mode can alleviate the problem of hair sometimes getting kinked when being deformed (see Animating Hair with Deformations in Stretchy Mode on page 73).

Styling Hair in Stretchy Mode

You can style hair in stretchy mode using any deformation or transformation. Combined with proportional modeling or the Brush tool, using stretchy mode for styling can be a great way to style hair in an organic fashion. As well, it’s a quick way to change the hair length.

After the style is the way you want it, you need to resample the hair so that the hair segments are uniform again. You can then turn off stretchy mode and apply dynamics to animate the hair.
To style hair in stretchy mode

1. Select the hair points or tips you want to style.

2. Do one of the following to activate stretchy mode:
   - Click the Stretchy button at the bottom of the Hair toolbar.
   - In the Hair property editor, select Allow Stretching of Hair Segments in the Hair Generation section. This activates the Stretchy button.

3. Style the hair using any of the deform operators or transformations.
   Notice that the hair strands act as if they are linear curves, with no IK behavior.

4. After the hair is styled, resample the hair by choosing Length > Resample Stretched Hairs.
   This resamples the guide hairs, approximating the shape as much as possible so that the 15 hair segments are evenly distributed along the length of the guide hair. This allows for better deformation for animation and accurate dynamics simulation.
   Resampling works based on selection, so only the selected strands that you have styled in stretchy mode are resampled.

5. To deactivate stretchy mode, click the Stretchy button again or click the Fix button beside it. When you click Fix, the hair is also automatically resampled.
When you animate hair with a deformation operator (such as an envelope or lattice), the deformation can cause the hair to get kinked because it tries to compress the hair but can’t because the hair segments are fixed in length. When the hair is animated in stretchy mode, however, it allows the hair segments to stretch and compress without kinking.

When you activate stretchy mode, the deformation operators receive a \texttt{<stretchy>} flag to indicate that they are to run in this mode. This is displayed in the hair’s operator stack in the explorer, as shown on the left.

![Hair deformed by lattice.](image1)

![Hair deformed by lattice in stretchy mode.](image2)

Keep stretchy mode active for the deformation operator so that the hair deforms properly when animated.

To animate hair in stretchy mode for new deformations

1. Click the Stretchy button at the bottom of the Hair toolbar.
2. Apply a deformation (such as a lattice or envelope) to the hair.

When the deformation is animated, the hair stretches and compresses to accommodate the deformation.

To toggle stretchy mode for existing deformations

If the deformation is already applied to the hair:

1. Select the deformation operator in the explorer.
2. Click the Stretchy button.

This toggles the state of the deformation operator between stretchy and non-stretchy modes.
Chapter 3 • Styling and Animating Guide Hairs

Setting Up Forces for Hair

After you’ve created a force, you need to set up the sensitivity of the hair to the force. The values you set for the parameters here specify the degree to which the natural force acts upon the hair. This means that you can leave a force’s values the same (useful if the force is affecting multiple objects in a scene) while changing only the hair’s reaction to it.

To set up forces for hair

1. Make sure that dynamics is set on the hair objects (see Making Hair Move with Dynamics on page 63).
2. Apply forces to the hair as described in Applying a Force on page 33.
3. Open the hair’s Hair Dynamics property editor.
4. In the Sensitivity to Forces group, modify the values for each parameter that correspond to the forces acting on the hair (that is, if you have wind affecting the hair, set the Wind parameter’s value here).

The settings you use here represent a percentage of the “global” effect that the force has.

The value for any parameter here must be greater than 0 (zero) for its corresponding force to have an effect. By default, each parameter’s value is set to 1 so that the effect works immediately at 100% of its strength, but you can change it to any value you like. You can also enter negative values for the gravity, attractor, and vortex forces.
Copying Hair Styles

You’ve just spent days perfecting your character’s hair style. Great. And now your client likes it so much that she wants all the other characters to have the same style ... by tomorrow! Don’t panic—you can do it easily by using the Copy Style command on Hair toolbar. This command copies the styling done on the guide hairs from one hair object to another. This makes it easy to share the same style among a number of objects, or groom the hairs on a low-resolution version of your model, then copy the finished style to the hair on the high-resolution version.

You can copy styles between objects that have similar but not necessarily identical topologies. You can even copy styles between hair on a polygon object and hair on a NURBS object: the geometry doesn't matter.

Tips for Copying Hair Styles

Here are some tips to consider before copying hair:

• Size does matter: copying styles between objects of vastly different sizes won’t give you the best results. This may give you a fairly similar style, but the guide hairs are not the same scale.

• Copying between very different topologies is not advisable, and orientations of the guide hairs may be different.

• Both hair objects must be of the same type meaning that you can’t copy a style between hair generated from curves and hair generated from any other object. The hair objects created for each method are different.

• Make sure that dynamics is off for both hair objects involved in the copying process. You can either mute the dynamics operator or delete it.

If you import a scene with hair from a version of XSI previous to version 3.5 and copy its style to hair created in a later version of XSI, the later version of hair will be offset from the emitter. This is because in versions of XSI previous to version 3.5, the hair object was not pose-constrained to the hair emitter (see Hair is Constrained to its Emitter on page 24).

To copy the hair style, you must position the source (previous version) and destination (later version) objects at the scene's global origin and then use the Copy Style command on the Hair toolbar.
Copying Styles

To copy hair styles between objects

1. Select the hair object (not just tips, strands, or points) to which you want to copy the new style.

2. Choose Modify > Copy Style on the Hair toolbar.

3. Pick the hair from which you want to copy the style.

The hair style is copied to the hair you had initially selected.
Hair is actually represented by two types of hairs: guide hairs (curves) for styling, and render hairs for viewing and rendering.

The render hairs are the “filler” hairs that are generated from and interpolated between the guide hairs. And as their name implies, render hairs are the hairs that are actually rendered.

You can determine what the final rendered hair will look like by setting the many options for the render hairs. You can determine the number of hairs rendered, their number of segments (which sets their resolution), and their thickness. You can even determine where render hairs will be generated and what their length will be using mappable parameters. Although you don’t style render hairs in the same way as you do guide hairs, you can add frizz, waves, and kink to them.
Chapter 4 • Getting the Look with Render Hairs

Setting Render Hair Properties

When render hairs are generated from the guide hairs, they take into account any parameters set in the Hair property editor. Many of these parameters let you control how the render hairs look, such as adding frizz or applying maps to parameters to temper their values. As well, dynamics parameters applied to the guide hairs have a subsequent effect on the render hairs.

Getting Started with Render Hair Presets

On the Presets page in the Hair property editor, there are several presets that you can select to set the appearance of the render hairs, such as for an afro, curly hair, wavy hair, etc.

To apply a preset

• Select the hair and click one of the buttons on the Preset page of the Hair property editor.

Viewing Render Hairs

Render hairs are the “filler” hairs that are generated from and interpolated between the guide hairs. And as their name implies, render hairs are the hairs that are actually rendered. Color information is consistent with what you set in the shaders that are connected to the hair, and much tweaking of hair properties can be done without needing to use a render region.

To view render hairs

• Select Render Hairs as the Display Type to display only the render hairs. Then set the Render hairs % value to view only a certain percentage of the render hairs (the default is 10%).

You can view the render hairs in the OpenGL display types (see above) using geometry approximation. This can speed up displaying hair if you set the hair’s subdivision level to a low value. See Setting the Geometry Approximation on page 127 for more information.
Connecting Maps to Render Hair Parameters

Many parameters in the Hair property editor can have weight and texture maps connected to them. These parameters are identified with a little connection icon beside them.

- Weight maps let you paint parameter values across the surface of an object. For example, you can use weight maps to vary the length or density of render hairs.
  
  For an example on how to connect a weight map to a hair parameter, see Changing the Length of Render Hairs on page 89.

- A texture map is the combination of a texture projection plus an image. Instead of one value being applied over the surface as with a weight map, a texture map applies a color. You create a texture map in which you select the texture projection method, then link up an image file whose pattern of colors you want to map.

  For an example on how to connect a texture map to a hair parameter, see Connecting a Texture Map to Hair Color Parameters on page 109.

For general information about mappable parameters, see Parameter Maps on page 117 in the Scene Elements guide.

Unlike other geometry in XSI, hair is not a typical surface so you can't apply maps directly to it. Instead, you need to first create a weight or texture map property for the hair emitter object, and then transfer it to the hair object itself.

**To connect a map to a parameter for hair**

1. Create a weight map or texture map for the hair emitter object.
2. Open an explorer so that you can pick the weight or texture map from it.
3. Select both the hair and emitter objects and click the Transfer Map button on the Hair toolbar. This transfers the map from the emitter object to the hair object and prompts you to pick the map.
4. In the explorer, pick the map that you created for the emitter object.

Under the hair object’s Cluster folder, you’ll see a new map that matches the one you transferred from the emitter object.
5. In the Hair property editor, click a parameter’s connection icon and choose Connect.
6. In the pop-up explorer, pick the map that you transferred to the hair object from the emitter object (\textit{wm\_cut\_hair} in this example).

The connection icon shows that a map is connected (turns red). You can click on this icon to open the weight or texture map’s property editor.

After you connect the map, you can go back and adjust the weight or texture map on the emitter object’s original map, and see that the connected map (and the results on the hair) is updated.
Setting the Number of Hairs Rendered

There are different ways to determine the number of hairs that are rendered. The basis for determining the number of hairs rendered is the **Total Hairs** parameter. Once you’ve set this, you can choose which percentage of this to display (see Viewing Render Hairs on page 78) or render (see Rendering a Percentage of Hairs on page 82). As well, you can increase the number of render hairs created by using a **Strand Multiplier** value (see Increasing the Number of Render Hairs on page 83) or decrease the number of render hairs created with a **Density Map** (see Removing Render Hairs (Density) on page 85).

**Setting the Total Number of Hairs**

The Total Hairs parameter in the Render Settings area of the Hair property editor is the total number of render hairs shown for the hair object. The higher the value, the more hair, but with slower interaction and render time. Depending on the look you need to achieve, you could reduce the overall number of hairs and instead increase the root and/or tip thickness. Using a slight kink or frizz can also help in giving the illusion of more hair.

Also, the size of the hair emitter object makes a difference: 50,000 hairs over a small object is quite densely hairy, but the same amount over a large model looks much sparser.

The slider goes only to 50,000, but you can type in values higher than this.

The default value of 6500 hairs gives you enough hair for a decent preview while styling without slowing down the interaction too much. However, for final previewing and rendering, you will probably want to use a value much higher than this for a full head of hair (such as 100,000 or more).
Chapter 4 • Getting the Look with Render Hairs

Rendering a Percentage of Hairs

The Render Hairs % parameter for the Render Settings looks at the Total Hairs value and then uses a percentage of that to determine the number of hairs to draw when rendering, either in the render region or to file. Typically you would use 100% for the final render, but you can decrease this value to get a rough idea of the final result for quicker render tests.

The Render Hairs % parameter in the Display Settings (see Viewing the Hairs on page 34) does the same thing as this parameter, but is only for viewing: this is not what is rendered in the region or to file.
Increasing the Number of Render Hairs

If you want to add more render hairs, you can use the Hair Multiplicity parameters in the Hair property editor. These parameters include a strand multiplier to increase the number of hairs for each render hair strand, and parameters to offset (splay) the new hairs starting at either the render hair’s root or tip.

The Hair Multiplicity parameters can have weight and texture maps connected to them, as indicated by the connection icon beside them. This allows you to create either patterns or random areas of increased amounts of hair. For general information about mappable parameters, see Connecting Maps to Render Hair Parameters on page 79.

To increase the number of hairs

1. Set the Strand multiplier to a value above 0. For example, if you use a value of 5, a total of five render hairs appear at the root or tip of each render hair strand.

2. Set the Splay at root and/or Splay at tip value to be above 0 to offset the new render hairs from the original render hair strand:
   - Splay at root offsets the new hairs (based on the value set for Strand multiplier) around the root area of the original render hair strand.
     
     Its value determines how far apart in space the copied strands are located in relation to the size of the underlying polygon: short hairs are placed closer to the original strand (splayed in less space) while long hair is splayed over more space around the original strand.
     
     - Splay at tip offsets the new hairs (based on the value set for Strand multiplier) starting from the tip, creating the effect of sprouts—or hair plugs!

    Decimal numbers (for example, 1.5 or 2.3) get truncated internally to the largest whole number less than the value entered (for example, 1 or 2). Decimal values are useful as a multiplier when you are using a weight map or texture map.
Chapter 4 • Getting the Look with Render Hairs

As with Splay at root, its value is the amount of splaying in space as determined by the underlying polygon which initially determines the length of the hair strands.

Tips for Using the Strand Multiplier

In some cases, using the strand multiplier can be faster to render than simply setting the number of hairs (see Setting the Total Number of Hairs on page 81) to a certain value. For example, let’s say you want to have a total of 150,000 hairs on an animal. Setting the number of hairs to be 50,000 and then using a strand multiplier value of 3 would be faster than simply setting the number of hairs to 150,000. This is because there are fewer original hair strands that need to be considered for the calculation of dynamics and render hair attributes. This requires less memory and therefore less time.

However, keep in mind that using a large strand multiplier value can slow down interaction significantly because it’s adding \( n \) hairs to each render hair strand. For example, if the total number of hairs on an animal is 50,000 and you use a strand multiplier of 20, that makes 1 million (50,000 x 20) hairs that are being rendered.
Removing Render Hairs (Density)

If you want to remove render hairs, you can connect a weight or texture map to the Density Map parameter (on the Effects page in the Hair property editor). You can determine the density of the render hairs in different areas on the hair emitter object. This parameter changes the total number of render hairs because it prevents hair from being displayed (“emitted”) that would have otherwise been there.

When you connect a map, Density Map blocks out hair where the map has values lower than 1. For example, a value of 1 (such as pure white in a texture map) allows all the render hairs (as determined by the value for the Total Hairs parameter) to be displayed, while a value of 0 (such as pure black in a texture map) prevents any render hairs from being displayed. Any value in between is the probability of the render hair not being displayed.

- For an example of connecting a texture map, see Connecting a Texture Map to Hair Color Parameters on page 109.
- For an example of connecting a weight map, see Example: Connecting a Weight Map for Cutting Hairs on page 90.

![Weight map created for hair emitting object (grid).](image1)

![Weight map connected to the Density Map parameter. Values less than 1 on the map prevent render hairs from being displayed.](image2)
Chapter 4 • Getting the Look with Render Hairs

Setting the Render Hair Thickness

The render hairs’ thickness has a great impact on how the hair looks, as well as the render time required to calculate the hair. Having thicker hair strands with fewer render hairs is faster for previewing the hair style. When you’re ready for a final preview or render, use lower thickness values and increase the number of render hairs.

You can set the thickness of the render hairs using the Thickness parameters on the Effects page in the Hair property editor.

The Thickness Root and Tip parameters can have weight and texture maps connected to them when the Thickness Type is set to Proportional. This allows you to create patterns or random thickness for the hair. For general information about mappable parameters, see Connecting Maps to Render Hair Parameters on page 79.

Setting the Thickness Type

To select a thickness type

1. Open the Hair property editor and click the Effects tab.
2. Select one of the methods for the Thickness Type to determine each render hair’s thickness:
   - **Proportional** (the default) determines the hair thickness relative to the size of the hair emitting area. Basically, the larger the emitting area, the thicker (and longer) the render hairs (see the illustration in Modifying the Hair Emitter on page 25).
   - **Absolute** lets you set the hair thickness in an absolute size using the Root and Tip thickness parameter values. The values are in 100ths of a Softimage unit. For example, a Root thickness value of 50 would make the roots 50/100 or 0.5 Softimage units in diameter.

   Scaling the hair emitter object also changes the thickness of the hair accordingly. See Scaling the Hair Emitter on page 25 for more information.
Setting the Root and Tip Thickness

Having separate root and tip thickness parameters allows you to change the shape of the render hairs. For example, by setting a Tip value equal to or higher than the Root value, you can create blunt bristles as found on a toothbrush. Setting the transparency at the root and tip also helps change the apparent shape of the hair when rendering (see Setting the Hair's Transparency on page 114).

To set the root and tip thickness

1. Open the Hair property editor and click the Effects tab.
2. Select a thickness Type.
3. Set either or both of these to determine the size of the render hairs:
   - Set the Root to determine the thickness of the render hairs at their base.
   - Set the Tip to determine the thickness of the render hairs at their tips.

Depending on the thickness Type that you select, the Root and Tip thickness values will have different results:

- With Proportional set as the Type, the Root and Tip values are relative to the size of the hair emitting area. For example, Root and Tip values of 1 for hair emitted from a large area make hair appear thicker than the same values for hair emitted from a small area.

- With Absolute set as the Type, the Root and Tips values are the actual diameter of the hair (at the root and tip, respectively) in 100ths of Softimage units.

Render hairs with Absolute Type. Root value is 1 and Tip value is 2.5.

Render hairs with Absolute Type. Root value is 4 and Tip value is 0.5.
Setting the Render Hair Resolution (Segments)

The render hairs’ thickness and resolution (number of segments) have a great impact on how the hair looks, as well as the render time required to calculate the hair.

Each guide hair is always composed of 15 segments, regardless of the hair length. However, you can affect the hair’s resolution by setting the number of Hair Segments used to calculate the render hairs, found on the General page in the Hair property editor.

The default is 10 segments over the length of each strand. You can reduce the segments to a low value, such as 5 or even 2 (the lowest value), for fur or very short hair to speed up the processing time. For long hair or styling that is very curly or frizzy, you may want to increase this to a value around 50 or more for long hair. Of course, nothing is for free: higher values equal more processing time. Sorry.

One way to work around having a high number of segments is to animate the Hair Segments parameter. Keep the value high for good resolution when the camera is closer to the hair, then lower the value when the camera is farther away from the hair.

[Image: Render hairs with 5 hair segments. Render hairs with 20 hair segments.]
Changing the Length of Render Hairs

You can change the length of the render hairs in two ways: apply a random scale to the hair or use a cut map to determine exactly where the hairs will be shorter. Changing the length of the render hairs does not affect the guide hairs, but the opposite is true. For information on scaling the guide hairs, see Changing the Hair Length on page 50.

Randomly Scaling the Render Hairs

You can set the value of the Random Scale parameter on the Effects page of the Hair property editor to do as advertised. This parameter randomly scales the render hairs down: hair is never scaled to be longer than it is. A value of 1 is totally random, while a value of 0 does nothing.

Cutting the Render Hairs

Using the Cut Map parameter on the Effects page of the Hair property editor, you can determine the length of the render hairs in different areas on the hair emitter object based on a weight map or texture map connected to it. The length of the guide hairs is first taken into account, then the map is applied. Cut Map makes the hair shorter where its connected map has values lower than 1. For example, a value of 1 (such as pure white in a texture map) makes all the render hairs the full length of the guide hairs, while a value of 0 (such as pure black in a texture map) cuts the render hairs at the root so that you won’t see them. Any value in between is a portion of the guide hair length starting at the root.

If you’re painting a weight map on a detailed model, always start out with the base map as 0 (black) and then add value to it.

If you start the other way around, you won’t be able to “erase” the values for the vertices in places like the mouth and deep creases. And you don’t want hair in your mouth!

For an example procedure of connecting a texture map, see Connecting a Texture Map to Hair Color Parameters on page 109.
Example: Connecting a Weight Map for Cutting Hairs

Here’s how to connect a weight map to the Cut Map parameter:

1. Create a weight map and paint the map values onto the surface of the hair emitter object. For information in general on weight maps, see Parameter Maps on page 117 in the Scene Elements guide.

2. Open an explorer so that you can pick the weight map from it.

3. Select both the hair and hair emitter objects and click the Transfer Map button on the Hair toolbar. This transfers the map from the emitter to the hair object and prompts you to pick the weight map.

4. In the explorer, pick the emitter object’s weight map that you created. Under the hair object’s Cluster folder, you’ll see a new map that matches the one you transferred from the emitter object.

5. In the Hair property editor, click the Cut Map parameter’s connection icon and choose Connect.

6. In the pop-up explorer, pick the map that you transferred to the hair object (wm_cut_hair in this example).

The connection icon shows that a map is connected (turns red). You can click on this icon to open the weight map’s property editor.
7. The hair is cut according to the values on the weight map. Notice that the hair shape always remains the same, no matter how short the hair.

After you connect the map, you can go back and paint some more on the emitter object’s original weight map, and see that the connected map (and the results on the hair) is updated.

Hair is cut according to weight map. Hair that corresponds to 0 on the map is cut to the root, and hair that corresponds to values over 0 is cut according to its value, up to 1 which is full length.
Reverting to hair rendering with objects (Instances)

Replacing hair with objects allows you to use any type of geometry in a hair simulation. You can replace hair with one or more geometric objects (referred to as instances) to create many different effects. Simply put the objects you want to instance into a group, then each object in the group is assigned randomly to a guide hair.

For example, you could instance a feather object for a bird or instance a leaf object to create a jungle of lush vegetation. The instanced geometry can even be animated, such as its local rotation or scaling, or animated with deformations. This allows you to animate the hair without needing to use dynamics, such as instancing wriggling snakes on a head to transform an ordinary character into Medusa!

The instanced geometry is calculated at render time so you'll only see the effect in a render region or when you render the frames of your scene.

When the instanced geometry replaces the hair, its local Y axis is aligned along the guide hair strand, and it is scaled to fit the length of the strand. Objects that aren't generally the same shape and length as the hair strand can get distorted, so be sure to choose an object that is proportionally appropriate.

The instanced geometry takes on the hair's grooming state and dynamics, deforming with the shape of the guide hair. To have the instanced object deform smoothly with the hair, make sure that it has enough subdivisions, especially along its Y axis.

To replace hairs with instances:

1. Select one or more objects you want to instance and create a group that includes them (press Ctrl+g or click the Group button on the Edit panel).
2. Select the hair object to which you want to attach the instances and open its Hair property editor.
3. On the General page in the Hair property editor, select the **Instancing > Enabled** option to activate instanced geometry as the render type instead of the hair geometry.

4. In the Instancing area, click the **Pick** button beside the **Instance Group** text box. In the explorer, pick the group of objects you want to instance.

5. Draw a render region around the hair or render the scene to see the instanced object geometry on the hair.

   To make sure that the instance source objects aren’t rendered, select the **Instance Master Hidden** option on the Rendering tab in the Visibility property editor of the instance source object.

   Notice how the two objects in the group are applied at random on the hair object, and that the instanced objects keep their own material and texture properties.

   If you make changes to an instanced object (such as changing its topology or material attributes), it is updated immediately on the hair.

6. You may want to reduce the **Render Hairs %** value in the Render Settings part of the Hair property editor (see *Setting the Number of Hairs Rendered* on page 81) because the instanced geometry usually takes up more space than do regular render hairs.

   **To remove instancing from hair**
   
   - Click the **Remove** button in the Instancing area in the Hair property editor.
Chapter 4 • Getting the Look with Render Hairs

Adding Frizz and Waves (Kinks)

You can apply overall effects to the render hairs using the Frizz and Kink parameters on the Effects page of the Hair property editor.

Many of these parameters can have weight and texture maps connected to them, as identified by a little connection icon beside them. This allows you to create either specific patterns or random areas of frizz and kinks for the hair, depending on the effect you want. For general information about mappable parameters, see Connecting Maps to Render Hair Parameters on page 79.

Frizzing Out

You can make hair appear frizzy or affected by static by using the appropriately-named Frizz parameters. These parameters work by adding a rotation with noise to the hair.

- **Frizz Frequency** is the frequency of the rotation noise pattern and is relative to the hair emitter object’s size. This means that the same value produces the same number of “bends” on small objects as on large objects.

  You can set the frizz along each of the X, Y, and Z axes. Select the Link XYZ axes option to change all XYZ values to the same value at once.

- **Frizz at root** and **Frizz at tip** add the noisy rotation to the strand using a value in the range of 0 to 180 (or more). This value is scaled by three different harmonic noise fields, whose frequency is set by the Frizz Frequency value.

  **Frizz at root** starts the rotation from the root and interpolates along the length of the strand to the tip, while **Frizz at tip** starts the rotation at the tip and interpolates along the strand to the root.

  If **Frizz at root** is relatively high and **Frizz at tip** is relatively low, the effect is like using mousse to scrunch together hair at the tips while preserving the chaos at the roots.

![Frizz at root at 40 and Frizz at tip at 5 with Frequency set to 30.](image1)
![Frizz at root at 10 and Frizz at tip at 80 with Frequency set to 30.](image2)
![Frizz at root at 1 and Frizz at tip at 50 with Frequency set to 80.](image3)
Adding Frizz and Waves (Kinks)

**Animating the Frizz**

To create effects similar to the Wiggle parameters used when the hair has dynamics applied (see *Adjusting the Hair’s Reaction to Dynamics* on page 65), you can animate the frizz for extra hair motion. You can use this in place of the Wiggle parameters for creating certain effects, such as wavy grass.

- Set the Frizz anim value to control the amount that the frizz is animated (similar to amplitude).
- Set the Anim speed to control how quickly the frizz is animated.
- Set the Direction of the frizz animation along the X, Y, and Z axes.

**Kinks and Waves**

The Kink parameters also apply noise to the hair, but in this case it’s a displacement of each point along the hair strand. The effect you see is kinked or wavy hair.

- **Kink Frequency** is the frequency of the noise pattern and is relative to the hair emitter object’s size. This means that the same value produces the same number of “bends” on small objects as on large objects.
  
  You can set the kink along each of the X, Y, and Z axes. Select the Link XYZ axes option to change all XYZ values to the same value at once.

- **Kink at root** and **Kink at tip** add the noise to the strand. This value is scaled by three different harmonic noise fields, whose frequency is set by the Kink Frequency value.

  **Kink at root** starts the noise from the root and interpolates along the length of the strand to the tip, while **Kink at tip** starts the noise at the tip and interpolates along the strand to the root.

- **Kink at root** at 1 and **Kink at tip** at 10 with Frequency set to 10.
- **Kink at root** at 1 and **Kink at tip** at 20 with Frequency set to 20.
Kink at root at 1 and Kink at tip at 30 with Frequency set to 40.

Kink at root at 1 and Kink at tip at 50 with Frequency set to 25.
Rendering Hair

Rendering hair is similar to rendering any other object in XSI. You can use all standard lighting techniques (including final gathering and global illumination), set shadows, apply motion blur, and do field rendering.

While you can use any type of XSI shader on hair, there are two special hair shaders that give you the most control and options for making hair look the way you want. With them, you can control the tip and root colors, the transparency, and the way the illumination values are mixed, as well as setting render optimizations.
General Rendering Information

Although rendering hair is similar to rendering anything else in XSI, here are a few topics that you should know about.

For some tips on how to optimize hair for rendering, see Tips for Rendering Hair on page 131.

Motion Blur

You can have motion blur on hair whether or not it has dynamics applied to it; however, without dynamics (live mode), the results are unpredictable. With dynamics on hair, make sure to select either Read & Write or Read only as the Cache Mode in the Dynamics Operator property editor (see Making Hair Move with Dynamics on page 63).

For more information on motion blur, see Motion Blur on page 155 in the Lights and Cameras guide.

Field Rendering

You can use field rendering with hair as you would any other object in XSI.

If you have dynamics applied to hair, make sure to select either Read & Write or Read only as the Cache Mode in the Dynamics Operator property editor (see Making Hair Move with Dynamics on page 63).

For more information on field rendering, see Field Rendering on page 94 in the Rendering guide.

Segmented and Distributed Rendering

If you want to render hair over a number of machines, it’s faster to use segmented rendering (rendering a scene as a separate, non-overlapping sequence of frames on each render machine) instead of distributed rendering. This is because the amount of data generated for the hair rendering exceeds what’s stored in the scene.

For example, if you were using segmented rendering, you could load a 500K scene that has a character with 50,000 hairs on it. If you use distributed rendering with 10 render slaves, you would need to spread about 160 Mb of data (50,000 x 10 x 32) around your network. The time to copy that amount of data over the network usually exceeds the time needed to render it.

For more information, see Batch Rendering Basics (XSI -R) on page 143 and Distributed Rendering on page 163 in the Rendering guide.

To use segmented rendering for hair

1. Bake the hair into a cache file.
2. Set the hair to Read only as the Cache Mode (see Making Hair Move with Dynamics on page 63).
3. Save the hair scene.
4. Split the sequence into $n$ equal length segments, where $n$ is number of machines on which you want to render.

5. Run this command on each of the render machines:

   \texttt{xsi -r -startframe \textit{start} -endframe \textit{end} <scene>}


Chapter 5 • Rendering Hair

Using Shaders on Hair

To create a rendered image of hair for either the render region or rendering to file, you apply a shader to it, just as you would for any other object in a scene. The default shader that you use to render hair is the Hair Renderer shader. This is a surface shader that allows you to set the color, transparency, etc. of the hair.

You can also use the Hair Geo shader. This shader lets you set the color shading and transparency on the hair using gradient sliders which allow you lots of control on where the shading occurs along the hair strand. See Using the Geometry Hair Shader on page 119 for information.

You can also use the standard XSI shaders on hair because it creates real geometry (for example, you can use the Phong shader or a Toon shader). You can use the shaders by plugging them directly in to the hair's Material node (see Connecting Shaders to Hair on page 102).

About the Hair Shaders

The Hair Renderer shader gives you control over accurate strand coloring, transparency, and shadows than do other XSI shaders. You can optimize the render and take advantage of final gathering.

The Hair Geo shader lets you set the color shading and transparency on the hair using gradient sliders which allow you lots of control on where the shading occurs along the hair strand.

Many of the parameters of the Hair shaders are mappable (any one with a little connection icon), meaning that you can connect texture maps (not weight maps) to them to create specific effects.

For an example of this, see Connecting a Texture Map to Hair Color Parameters on page 109.

To modify the parameters for either of the Hair shaders

• Open their property editors in one of the following ways:
- Select the hair and choose Modify > Shader from the Hair toolbar.

or

- Double-click the Hair Renderer or Hair Geo Shading node in the render tree.

or

- Click on the Hair Renderer or Hair Geo Shading shader’s icon in the explorer.

• For information on setting up all parameters in the Hair Geo shader, see Using the Geometry Hair Shader on page 119 for information.

• For information on lighting and shadows with the Hair Renderer shader, including shading models for hair length, see Lighting and Shadows for Hair on page 115.

• For information on coloring with the Hair Renderer shader, see Coloring Your Hair on page 104.

• For information on setting transparency with the Hair Renderer shader, see Setting the Hair’s Transparency on page 114.

**Geometry Render Type**

The Hair Renderer and Hair Geo shaders use a geometry render type for rendering hair. This type is used by default unless you activate the Instancing option to render instanced geometry instead of hair—see Replacing Render Hairs with Objects (Instances) on page 92 for more information.

With geometry rendering, the same shading algorithm is used as for any XSI surface shader. Hair is rendered as a special hair primitive by the mental ray renderer.

If you want to quickly render a scene without rendering hair but don’t want to put the hair in a separate layer, see Deactivating Hair for Rendering on page 130. The hair is still generated, just not rendered.
Connecting Shaders to Hair

The Hair Renderer shader is the default shader applied to hair. It is a surface shader that you can connect to the Surface and Shadow inputs of the hair object’s Material node. You can connect the Hair Geo shader to hair in the same way as the Hair Renderer shader.

As well, you can connect other standard XSI shaders, such as the Phong shader or a Toon shader, to the hair object’s Material node.

For general information about the render tree, see The Render Tree on page 87 in the Material and Shader Basics guide.

To reconnect the Hair Renderer shader

The Hair Renderer shader is automatically connected when you create hair. However, if you need to reconnect it, here’s how:

1. Open a Render Tree and choose Nodes > Hair > Hair Renderer.
2. Attach the shader’s output to the Surface and Shadow inputs on the hair object’s Material node.

To connect the Hair Geo shader

1. Open a Render Tree and choose Nodes > Hair > Hair Geometry Shading.
2. Attach the shader’s output to the Surface and Shadow inputs on the hair object’s Material node.
To connect other XSI shaders

- Remove the Hair Renderer shader and connect the XSI shaders directly to the Surface input, and optionally to the Shadow input, of the hair’s Material node.

  For example, you can attach a Phong shader to the Surface input of the hair’s Material node to change the hair’s color.

You can also connect shaders like the Toon shaders directly to the hair Material node’s Surface and Shadow inputs as well to create some interesting results.

For information on the Toon shaders, see Toon Shading on page 13.
Coloring Your Hair

Trying to cover up those dark roots? Leaving just a touch of gray? You can do these coloring activities and more with the various illumination parameters in the Hair Renderer shader. While you can apply any XSI shader to the hair object to define color, the Hair Renderer shader allows you to specify different root and tip colors, as well as where the root/tip color starts and ends.

Getting Started with Color Presets

Better than a shade card from your hairdresser or private colorist, you can have drag-and-drop hair color! There are many presets to choose from for hair colors, just like having your own virtual salon. The presets are simply variations based on the Hair Renderer shader. You can use them “as is” or as a starting point to creating your own custom hair color.

To apply the presets

1. From the main menu, choose View > Toolbars > Shader Presets, then click the Hair tab.

2. Drag the presets and plop them directly on the hair object in a viewport or on the hair object’s Material node in the render tree.

3. Create a render region and admire.
Overview of Setting the Hair Color

To color the hair strands

1. Open the Hair Renderer property editor.

2. On the Illumination page, specify the color information for the hair strands using the parameters outlined here:

- Defines the hair's **ambient color**—see Ambient and Diffuse Colors on page 106.
- Defines the hair's **root color**. You can define how this color is blended with the tip colors—see Ambient and Diffuse Colors on page 106.
- Defines the hair's **tip colors** and balance—see Setting the Tip and Root Colors on page 106.
- Defines the **balance** between the root and tip colors—see Balancing Tip and Root Colors on page 108.
- Mixes in **random colors** of the same value—see Adding Color Variation (Random Color Mixing) on page 107.
- Defines the **highlights** on the hair—see Specular Color on page 106.

All of the parameters on this page are mappable, meaning that you can connect texture maps to them to create specific effects. For an example of using a texture map for colors, see Connecting a Texture Map to Hair Color Parameters on page 109.
Setting the Tip and Root Colors

Ambient and Diffuse Colors

Ambient Color is the hair surface’s underlying ambient (shadow) color. This color gets modified by the scene’s ambience. To get luminescent hair, use a high value for this parameter.

The hair’s Diffuse (main) colors are divided between a tip and root color. You can easily set the root and tip colors and then choose how you want to blend them (see the images starting in Balancing Tip A and B Colors on page 108).

- Tip Color A and B combine to define the diffuse color of the tip of each hair strand.
  - Tip Color A is the main tip color.
  - Tip Color B provides an alternate tip color that gets added to random hair strands.
- The Root Color does what you expect: it defines the hair strands’ diffuse root color. Making the root color darker than the tip helps to give more depth and realism to the hair strands.

Specular Color

You can set the color and area (decay) of Specular highlight on the hair. Lower values for Specular Decay give a larger highlight area indicating less decay. The specular color value (not the decay) should be very high for dark hair and very low for light hair.

Although you can’t directly set the Specular values of the hair’s root and tip separately, you can do this by connecting other shaders to the Hair Renderer shader.

To set the root and tip specularity separately
1. Select the hair and open a render tree (press 7).
2. Choose Nodes > Mixers > Gradient and plug it into the Specular input on the Hair Renderer shader.
3. Choose Nodes > State > Scalar State and plug it into the input of the Gradient node.
4. Open the Scalar State property editor and change the State Parameter to Barycentric B/Lengthwise Hair.
   - This creates a white/black gradient running along the length of each hair. You can reverse the gradient direction by plugging an image processing/invert node into the tree.

See About the Barycentric Hair Parameters on page 113 for an explanation of the barycentric parameters.
5. Open the Gradient property editor and change the values on the gradient slider to define the hair’s root and tip specularity.

Setting the Colors

To set the ambient, diffuse, or specular colors

- Drag the color sliders, enter values, or click the color box to open the color editor. For information on these controls, see Defining Color Properties on page 67 in the Interface and Tools guide.
- Copy a color by clicking its color box and dragging its “color chip” to another color box (on the same property page or from another one).

Adding Color Variation (Random Color Mixing)

In addition to these basic colors, you can set the Color variation value which adds some hue jittering on the hair’s final diffuse color. For example, this may be useful to use when you want subtle variations in white or light-colored hair.

The original color is converted to HSV, the saturation is set to 1, and a random hue is picked. Then this color is mixed with the original color. A value of 0 is no mixing, while 1 is 10% mixing (you can type in values up to 10 for 100% mix). At 100%, you will see only chromatically random colored hairs.
Balancing Tip A and B Colors

The A/B Balance slider lets you set the balance between which tip color is used: a value of 0 is only tip color A, and 1 is only tip color B.

Tip Colors A and B used for the following images.

Balancing Tip and Root Colors

The root and tip colors are blended along the length of the entire hair. The Root/Tip Crossover Center slider sets the point along the strand at which there is an equal percentage mix of root and tip colors. A value of 0 is mostly tip color and 1 is mostly root color.
Blending Tip and Root Colors

The Root/Tip Crossover Range slider defines the way in which the root and tip colors blend together (that is, how much of a contrast between the types of colors). A value of 0 is sharp contrast and 1 is full blending.

Connecting a Texture Map to Hair Color Parameters

This is an example of how to change the hair colors on a moth using the colors from a texture-mapped image.

A texture map is the combination of a texture projection plus an image. Instead of one value being applied over the surface as with a weight map, a texture map applies a color. You create a texture map in which you select the texture projection method, then link up an image file whose pattern of colors you want to map.

Unlike other geometry in XSI, hair is not a typical surface so you can’t apply projections directly to it. Instead, you need to create a texture map property for the hair emitter object first, and then transfer it to the hair itself.

When mapping a texture to the hair, the color of the individual strands are derived from the texture color found at the root of the hair, so make sure your map is painted accordingly.

1. Apply a texture map to the hair object by doing either of the following:
   - Select the hair emitter object and choose Get > Property > Texture Map > Texture Map.
     This creates a texture map property for the hair emitter and opens the blank texture map property editor in which you need to set the texture projection and select an image to be used as the map (see next step).
   - Select the hair emitter object and choose Get > Property > Texture Map > projection type (such as Cylindrical, Spherical, UV, XY for a plane, etc.) that is appropriate for the shape of the object.
This creates a texture map property and a texture projection but doesn’t open the Texture Map property editor in which you can select an image.

Now you must associate the image to this projection to use as the map, as described in the next step.

2. In the Texture Map property editor, select an image file for the map.

3. In the UV Property area beneath the image, create a New texture projection (if there isn’t already one) that is appropriate to the shape of the hair emitter object or how you want to project the mapped image.

4. Give the texture map a descriptive name by clicking the Rename button at the top of the property editor.

   This opens a property editor in which you can enter a New Name for the texture map (projection). The other options in this property editor control how references to the projection are updated. For more information, see Renaming Texture Projections on page 140 in the Texturing guide.

5. Open an explorer and expand the hair emitter object’s tree so that you see the Texture Map icon.

6. Select both the hair emitter object and the hair and click the Transfer Map button on the Hair toolbar. This transfers the texture projection coordinates from the emitter to the hair and prompts you to pick the texture map.

   Remember that it’s the texture map itself that you must transfer, not the image or the texture projection.

7. In the explorer, pick the emitter object’s texture map that you created, as shown here.
8. Do one of the following:

- In the Hair Renderer shader’s property editor, click the connection icon for one of the mappable parameters (such as the Tip Color A/B or Root Color) and choose Image. Pick the texture map you transferred, or any other image.

or

- In the render tree, choose Nodes > Texture > Image to create an image node and attach it to the parameters you want in the Hair Renderer shader (such as the Tip A color, as shown below).

Make sure to select the correct projection type in the Image node’s property editor.

You will now see the fuzz on the moth rendered using the colors from the image you mapped.
Chapter 5 • Rendering Hair

Mapping Only the Tip or Root Color

The following shows the results when you map an image to only the root color parameter, but not the tip color. Mapping selectively to different parameters allows you a lot of flexibility in the effects you can achieve.

Creating a Surface for Texturing Curve-Based Hair

Hair that is created from curves does not have the same geometry base as hair created from other objects, such as polygons. As a result, you cannot project texture maps onto the hair directly. However, you can extract the surface interpolation value of the hair between curves to create a surface for mapping.

To create a surface for hair from curves

1. Select the hair created from curves and open a render tree.
2. Choose Nodes > State > Scalar State twice to create two Scalar State nodes.
3. Open the property editor for each of these nodes:
   - Change the State Parameter for one of the nodes to Barycentric B/Lengthwise Hair.
   - Change the State Parameter for the other node to Barycentric C/Spline Hair Interpolant.

   See About the Barycentric Hair Parameters on page 113 for an explanation of the barycentric parameters.
4. Choose Nodes > Conversion > Scalars to Vector. Plug both of the Scalar State nodes to the inputx and inputy inputs of the Scalars2Vector node.

   It depends on the orientation of the image you’re using to determine which Scalar State node plugs into which input. For example, you can plug the Barycentric C (interpolant) node into inputx and the Barycentric B (lengthwise) node into inputy.
5. Choose Nodes > Map Lookup > Image Lookup. Open the Image Lookup property editor and get the image you want to map.
6. Plug the Scalars2Vector output to the coord input of the Image Lookup node.
7. Plug the Image Lookup output into any of the mappable color parameters in the Hair Renderer shader.

**About the Barycentric Hair Parameters**

Barycentric hair parameters in the Scalar State shader let you define more accurate specular color along a hair strand (see *Specular Color* on page 106), as well as map custom values over hair or create a surface for curve hairs upon which you can apply textures.

Barycentric coordinates give the coordinates of a point relative to the triangle it is on as a weight of the three corner vertices, normalized so that the sum is 1. That means that from the barycentric coordinates of a point, you can get the "real" coordinates of the point.

- The **Barycentric A / Crosswise Hair** parameter returns values from 0 to 1 along the width of the hair (from one edge to the other: $0 > 1 > 0$).
- The **Barycentric B / Lengthwise Hair** parameter returns values from 0 to 1 along the length of the hair, with 0 being the root and 1 being the tip.
- The **Barycentric C / Spline Hair Interpolant** parameter assigns a number to each hair created from curves. This number is based on where it originates within the collection of guide hairs (curves). If you have three curves, a hair that is generated between curve 1 and 2 gets assigned a number between 0 and 0.5 depending on its proximity to either curve.
Setting the Hair’s Transparency

The Hair Renderer shader lets you set the transparency level of the tips and roots of hair strands, as well as where their levels meet and blend.

- For a more realistic hair look (especially for white or light-colored hair), use lots of hair strands with a fairly high transparency value. This also decreases shadowing because it allows light to pass through the hair volume. Having a very low transparency value for the roots can make the hairs opaque and look like hair plugs.

- To save time while previewing, deactivate the transparency in the Hair Renderer shader.

To set the hair strands’ transparency

1. Open the Hair Renderer property editor and click the Transparency tab.

2. Set the Tip and Root levels to have the hair completely opaque at the tip/root (a value of 0) or completely transparent (a value of 1).

3. Set the Root/Tip Crossover Center to define the point along the hair strands at which the transparency switches from the root to tip transparency value. A value of 0 is mostly tip and 1 is mostly root.

4. You can also control the blending between the root and tip transparency values by setting the Root/Tip Crossover Range. A value of 0 provides a sharp transition (no blending) and 1 is full blending.

All of the parameters on this page are mappable, meaning that you can connect texture maps to them to create specific effects. For an example of mapping, see Connecting a Texture Map to Hair Color Parameters on page 109.
Lighting and Shadows for Hair

As with any other object in XSI, the rendering of hair is affected by the lighting in a scene. However, there are a few special lighting issues to consider with hair.

Lighting for Hair

While the type of light doesn’t matter with hair, the number of lights does. It’s best to have as few lights as possible on the hair. If you have a number of lights in the scene but you don’t want to use them all on hair, make the lights that you don’t want the hair to consider exclusive of the hair.

Light Shading Models for Hair Length

A hair pointing toward a light will receive less light than a hair that is perpendicular to the light (more surface area of the hair facing the light). You’ll notice this when you first apply hair on any round shape: the hairs don’t have much surface area on which the light can land. If you comb this hair up or down, you’ll see that the light falls mainly on the hair’s center.

Likewise, hairs that are perpendicular to the light (usually longer hairs) receive the most light contribution because of the large amount of surface area facing the light.

To help render the hair appropriately, you can choose between two different Shading Models on the Render Settings page in the Hair Renderer shader’s property editor: one optimized for short hair and one for long hair.

The Diffuse Models are used for calculating an artificial normal that is used for the diffuse color results:

- The Short Hair model is optimized for short, spiky hair or fur. This method uses the normal of the hair emitter’s surface to determine the normal of the hair. This means that if the normal of the surface is pointing away from the light, no light contribution applies to that hair (as with the emitter’s surface).

- The Long Hair model is appropriately optimized for longer hairs. This method tries to aim the hair’s normals toward the light currently being calculated for contribution. There will always be some light on the hair even if the hair emitter’s surface is pointing away from the light.
The **Normal Blend** slider controls the mixing between the normal (as dictated by the Short or Long Hair model) and the actual underlying geometry of the hair object.

**Final Gathering and Hair**

If you have final gathering set in your scene, you can also have it affect hair. To determine final gathering’s effect on hair, set the **Final Gathering - Color** value on the Indirect Illumination page in the Hair Renderer shader’s property editor.

This value controls the color and intensity (strength) of the final gathering effect over the hair’s surface. The color that is received on the hair from final gathering is multiplied by the color value you set here and then added on top. Be aware when you’re setting this value that hair is very sensitive to final gathering because it’s like thousands of tiny objects blasting color at each other in a short distance.

If you keep this Color value low for the hair, you can then associate an inclusive light for only the hair for shadows. This light should be quite dark so that it doesn’t change the final gathering look. You can then exaggerate the specular of the hair so that it shines more from the dark light. You may also need to use negative umbra values to get dark enough shadows because of the dark light.

Also, if you connect a texture map to this parameter (see *Connecting a Texture Map to Hair Color Parameters* on page 109), you can reveal a final gathering on a specific location on the hair.

For information in general about final gathering, see *Global Illumination and Caustics* on page 47 in the Lights and Cameras guide.
Shadows and Hair

You can have hair cast shadows on other objects in the scene and the individual hairs cast shadows on each other. In addition, the transparency values set for the hair are taken into consideration when shadows are calculated.

As with anything else in XSI, casting shadows with hair depends on the number and types of lights you’re using. Make sure you have shadows activated for the lights you want to use.

- Set the light’s Umbra values to over 0.2 to give the best results.
- Using segment shadows can help reduce render times as well as improve the quality of the shadows.

You can have the hair cast shadows with light sources inside the hair’s bounding volume; however, if the eye ray hits a non-hair object before it hits hair, no hairs are displayed or shadows created for that segment.

For information on shadows in general, see Creating Shadows on page 33 in the Lights and Cameras guide.

Self-Shadows

You can have hair cast shadows on themselves by setting the Self-Shadowing Factor value in the Hair Renderer shader. The shadows cast by the hairs are attenuated by this factor (zero is no self-shadowing).
For best results, make sure shadows are turned on for both the lights and self-shadowing for the hair. However, with very short hair, you may not need self-shadowing, which speeds up the processing time.

**Shadow-mapped Shadows**

Shadow-mapped shadows are usually the best option for hair. The mental ray renderer uses the actual raw geometry from XSI to create the depth maps for shadow mapping: the shaders are not considered in this calculation.

When you’re creating the shadow maps, be sure to eliminate all unnecessary objects from the shadow calculations. This can speed up the calculation process considerably.

As well, shadow map size can be important, especially when there’s lots of detail involved. Spot lights often work well with shadow maps to help keep the coverage low and the sampling and size down.

To eliminate flickering, you can increase the shadow map’s sampling which, of course, increases the render time correspondingly. To reduce the rendering time, you can render the shadow map element in a separate pass with a lower sampling resolution, then blur that and composite it with the hair pass.

You can also use volumic shadow maps with geometry hair to reveal more of it details and depth. Volumic shadow maps store the changes in color or intensity along with the depth at which the change occurred during the raymarching process. For information on these, see *Creating Volumic Shadow Maps* on page 38 in the Lights and Cameras guide.
**Using the Geometry Hair Shader**

With the Hair Geo shader, you can set the diffuse, ambient, and specular color shading and transparency for hair. All parameters in this shader have the option of using a gradient slider so that you can have more accurate control of how the color shading and transparency occurs along the hair strand.

You can also add incandescence (glow) to the inner and outer hair edges. This creates a glowing appearance or makes the hair appear more “tubular,” which can improve the hair’s look or add the type of effect you want.

To connect the Hair Geo Shading shader

1. Open a Render Tree and choose Nodes > Hair > Hair Geometry Shading.
2. Attach the shader’s output to the Surface and Shadow inputs on the hair object’s Material node.

To edit the Hair Geo Shading shader

- Double-click the Hair Geo Shading node to open its property editor.

Just one of the types of looks you can get with the Hair Geo shader. Notice how the hairs look slightly tubular because of the transparency and incandescence options.
Setting the Hair’s Diffuse and Ambient Colors

The hair’s Diffuse color is its main color while the Ambient Color is the hair surface’s underlying ambient (shadow) color. This color gets modified by the scene’s ambience. To get luminescent hair, use a high value for this parameter.

To set the diffuse and ambient colors

- With both the Diffuse and Ambient colors, you can select the Use Diffuse/Ambient Gradient option to use a gradient slider. With both these color types, the gradient slider determines the diffuse color value along the hair strand’s length. The left end of the slider represents the hair strand root and the right end represents the hair strand tip.

- If the Use Diffuse/Ambient Gradient option is off, you can use the regular color controls found with the color type.

To blend the diffuse and ambient colors

- Select Use Blend Gradient in the Diffuse/Ambient Blending area to define the mixture of diffuse and ambient along the hair strand using a gradient slider.

  The left end of the slider represents the hair strand root and the right end represents the hair strand tip. A value of 1 means that both Diffuse and Ambient are visible, and a value of 0 means that only Ambient is visible.

For information on using the gradient controls for all parameters, see Using the Gradient Controls on page 125.

Parameters in this property editor that have a connect icon beside them, as shown on the right, are mappable, meaning that you can connect texture maps to them to create specific effects.

For an example of using a texture map for colors, see Connecting a Texture Map to Hair Color Parameters on page 109.
Using the Geometry Hair Shader

Setting the Hair’s Specular Color and Decay

With the Specular parameters on the Illumination page, you can set the color and area of falloff (decay) of Specular highlights on the hair. The specular color value should be very high for dark hair and very low for light hair.

**To set the specular color**

- Select Use Specular Gradient in the Specular area to define the specular color along the hair strand using a gradient slider.

  The gradient slider determines the diffuse color value along the hair strand’s length. The left end of the slider represents the hair strand root and the right end represents the hair strand tip.

  - If the Use Specular Gradient option is off, you can use the regular color controls found with the color type.

**To set the specular decay**

- If the Use Specular Gradient option is off, you can set the Specular Decay to control the size and falloff of the specular highlights.
Chapter 5  •  Rendering Hair

• When you select Use Specular Gradient, you can set the location and value of the specular decay along the length of the hair strand using the gradient slider for Specular Decay Root/Tip. The left end of the slider represents the hair strand root and the right end represents the hair strand tip.

Setting the Hair’s Transparency

By tapering the transparency along the hair strand, you can improve the hair’s overall look and reduce hair flickering. Reduced flickering gives a much smoother look, which is required for some furry animals, and also renders faster because you don’t need to have high antialiasing settings to get rid of artifacts. Another benefit of controlling the hair’s transparency tapering in this shader is that you can leave the hair width value as a constant in the Hair property editor.

To set the basic transparency

• On the Transparency page, activate the calculation of transparency values by selecting Enable.

• Set the Overall Transparency, which is the transparency value of the whole hair strand. A value of 0 is no transparency (completely opaque) and 1 is completely transparent when Length and Cross are not active.

When either Length or Cross are active, a value of 0 is no transparency (completely opaque) and 1 is 100% of either Length of Cross (whichever is active).

To set the transparency along the hair strand’s length or across its width

• Set the transparency along the hair strand’s length with the Length gradient controls. The left end of the slider represents the hair strand root and the right end represents the hair strand tip. A value of 0 is completely opaque and 1 is completely transparent.
• Set the transparency across the hair strand using the Cross gradient controls. The left end of the slider represents the hair strand edges (rim) and the right end represents the hair strand center (core). A value of 0 is completely opaque and 1 is completely transparent.

Hair with higher Cross transparency values at the edges. Hair with lower Cross transparency values at the edges.

**Setting the Hair’s Translucency**

Translucency allows light to partially or diffusely pass through the hair strands. The Hair Geo shader computes the illumination from both sides of the hair and adds the result. This lets you create a glow or rim lighting effect on hair, which can increase the realism of hair, especially when the hair is white or light-colored or in a brightly lit environment.

You can set the overall translucency for the whole hair strand or you can use the gradient slider to have varying translucency values along the hair strand. For example, hair is usually more translucent at the tips but less so at the roots.

**To set the overall translucency**

1. On the Indirect Illumination page of the Hair Geo shader, activate the calculation of translucency values by selecting Enable in the Translucency area.

2. Set the value for the Translucency slider, which is the translucency value of the whole hair strand. A value of 0 is no translucency (completely opaque) and 1 is full translucency.

**To set the translucency along the hair strand’s length**

1. Select Translucency Gradient in the Translucency area. This activates translucency using the gradient controls that appear.

2. Set the values with the gradient slider, which determines the translucency value along the hair strand’s length.

   The left end of the slider represents the hair strand root and the right end represents the hair strand tip. A value of 0 is completely opaque and 1 is completely translucent.
For information on using the gradient controls, see Using the Gradient Controls on page 125.

Setting the Effect of Indirect Illumination on the Hair

The parameters on the Indirect Illumination page let you control how global illumination, caustics, and final gathering affect the hair. You can also add incandescence (glow) to the inner and outer hair edges.

To control the effect of indirect illumination on the hair

- Set the value for the Radiance parameter in the Global Illumination / Caustics / Final Gathering area controls the color and intensity (strength) of any indirect lighting effect over the hair’s surface.

  The color that is received on the hair from final gathering is multiplied by the color value you set here and then added on top. Be aware when you’re setting this value that hair is very sensitive to final gathering because it’s like thousands of tiny objects blasting color at each other in a short distance.

  If you keep this value low for the hair, you can then associate an inclusive light for only the hair for shadows. This light should be quite dark so that it doesn’t change the final gathering look. You can then exaggerate the specular of the hair so that it shines more from the dark light. You may also need to use negative umbra values to get dark enough shadows because of the dark light.

  You can also texture map this parameter to reveal indirect illumination on a specific location of the hair.
To set incandescence on the hair

The parameters in the Incandescence area add an emissive color to the hair strand on top of the hair shading. This lets you create glow effects and allows the hair to appear more tubular when a light color is used on the inner core of the hair strand and a darker color is used on the edges (rim) of the strand. You can use these parameters to improve the hair’s look or add a special type of effect.

- Enable the Inner parameter and set its color values to add this color to the inner core of the hair strand. You can also control the Intensity of the incandescent color values.
- Rim is the same as Inner except that it applies the incandescent color settings to the outer edges of the hair strand.

Using the Gradient Controls

The different gradient sliders in the Hair Geo Shading shader’s property editor let you adjust the parameter’s value using a gradient.

To select color markers

- Click a square marker below the gradient slider. You can then set its color using the color sliders or key its position to animate the color (see below).

To change the gradient composition

- Drag a color marker anywhere along the gradient.

To insert a color marker

- Click at any point along the gradient slider.

By default, the new marker assumes the color at that point in the gradient. You can use up to eight color markers, each with its own color.
To delete a color marker

- Right-click it and choose Delete marker or press Delete.

To change the blending between two colors

- Drag a round marker above the gradient slider closer to either square color marker.

  Round markers appear between each pair of color markers, indicating the mid-point in the blend between those two colors.

To set the gradient’s interpolation

- Click the Cubic/Linear button to toggle between linear and cubic interpolation of the gradient.

  Cubic interpolation results in a smoother transition between alpha values, while linear interpolation results in sharper transitions.

To animate the markers

1. Go to the frame at which you want to set keys.

2. Select a square color marker and click the Pos parameter’s animation icon.

To remove the animation on the markers

1. At any frame, select a square color marker and go to a frame where there is a key.

2. Click the Pos parameter’s animation icon to remove the key.
Setting Up for Rendering Hair

Before you render hair, either in a render region or to file, there are some settings that can make a big difference to the quality of the render. This includes setting the quality of the render, setting the geometry approximation, and setting up a BSP tree for hair.

Selecting the Render Quality

You can choose to render hair with a low quality for quick rendering, or with a higher quality rendering that takes longer to process. This setting affects only pre-processing time (hair generation). If the hair doesn't have a lot of kink or frizz, and has few segments (such as short hair or fur), use the Low setting for faster processing; otherwise, use the High setting for better quality.

1. Attach a shader to the hair object as described in Connecting Shaders to Hair on page 102.
2. Open the Hair property editor and select either Low or High from the Render Quality list.

Setting the Geometry Approximation

When either viewing or rendering hair, you can set its geometry approximation. Geometry approximation specifies how objects and curves should be tessellated (divided into triangles and line segments, respectively, at rendering time).

When you set the geometry approximation for hair, a general Bézier curve is created to represent the hair. You can reduce the number of segments in the hair curves and still render a very smooth surface.

For general information about geometry approximation, see Rendering Options on page 55 in the Rendering guide.

To set the geometry approximation for hair

1. If the hair does not yet have a local Geometry Approximation property assigned to it, assign one by choosing Get > Property > Geometry Approximation from any toolbar.
2. Click the Selection button on the Select panel to display the selected object's nodes in a pop-up explorer.
3. Click the Geometry Approximation icon to open its property editor.
4. On the Hair tab, specify the degree of curve approximation for the Subdivision - Curve Type:
   - Linear creates a linear curve approximation (straight segments). If you select this option, the OGL Level and Render Level options are unavailable because you cannot approximate straight lines with more lines.
- **Quadratic** creates a quadratic curve approximation (smooth). This option doesn’t guarantee a continuous curve throughout the entire length of the hair. If the original line segments are at too steep of an angle, there may be a cusp where they meet.

If you select this option, additional points required to construct the curves are generated. This results in one extra control point for each generated hair point, less one.

- **Cubic** creates a cubic curve approximation (smoothest), guaranteeing continuity between hair segments. If you select this option, additional points required to construct the curves are generated. This results in two new control points.

5. Set the values for the **OGL Level** and/or **Render Level** of subdivisions. These parameters are independent of each other.

- **OGL Level** is the subdivision level for approximating the curves when viewing hair in the OpenGL views in the viewports.

- **Render Level** is the subdivision level for approximating the curves when rendering hair (in the render region or to file).

Either OGL or Render Level generates $2^{\text{Level}}$ line segments for approximation. For example, if this value is 0, the curves are approximated by 1 line segment; for a value of 1, by 2 segments; for a value of 2, by 4 segments, etc.

The Level sliders display a range of 0 to 4, but you can enter a value up to 7 in the text boxes.

You can also press the + and - shortcut keys to respectively increase or decrease the Level values.

### Setting the BSP Tree for Hair

You can set up a BSP tree exclusively for rendering hair. These settings work in the same way as do the standard BSP tree settings for the scene, but they are calculated separately. This allows you to deal with any special considerations that hair requires for rendering, apart from the rest of the scene.

You should spend a good amount of time tweaking the hair BSP settings to speed up rendering. Draw small render regions on the hair as you go so that previews are fast.

For information on BSP tree settings in general, see *BSP Raytracing Acceleration* on page 83 in the Rendering guide.

For some general tips about rendering hair, see *Raytracing and Antialiasing* on page 133.
Setting Up for Rendering Hair

To set up the BSP tree for hair

1. Open the Hair property editor and make sure that Instancing is not enabled so that you are rendering the hair geometry, not instanced geometry.

2. Set the values for the Hair Geometry parameters on the same page:

   - **Maximum Depth** is the maximum number of branches down the tree can go until it stops splitting into any further branches (1 to 50).
   - **Maximum Size** is the maximum number of hair segments each leaf on the tree can hold before converting that leaf into a branch to allow for further splitting (1 to 30).

Setting the Hair Chunk Size for Rendering

When you render hair, it is split into separate mental ray objects in approximately 400,000-strand chunks. This results in approximately 40,000 hairs per object given the default 10-segment hair.

Splitting the hair objects into smaller hair objects allows you to speed up the rendering. This means that you can render more hair, which increases the overall realism. As well, making smaller hair chunks can help render hair with motion blur because memory requirements increase when motion blur is used.

To set the hair chunk size for rendering

- Enter SI_HAIR_CHUNK_SIZE as an environment variable in the setenv.bat file (in the XSI Application\bin folder) and enter a value for the maximum hair strands you want per hair object.

  For more information about setting environment variables, see the XSI Setup guide.

Currently the default is set to 40,000 strands, so say you have 1000 hairs with 80 segments each, this give you a total of 80,000 strands. When you render this we’ll generate two hair objects of 40,000 strands each at render time to optimize it.
**Deactivating Hair for Rendering**

Hair can often slow down the rendering of a scene, and you may want to render the scene without rendering the hair. Instead of hiding the hair or putting it on a separate layer and making that layer unrenderable, you can easily toggle an option in the Render Options property editor. This applies only to hair that uses the geometry render type, not instanced geometry.

**To deactivate geometry hair for rendering**

1. Choose *Render > Render > Options* on the Render toolbar.

2. In the Render Options property editor, click the Active Effects tab and deselect the *Geometry Hair* option. The hair is still generated, but not rendered.

The same command is also available for batch rendering.

**XSIBatch Command:**

-geo_hair [on|off]

- For more information about all of the rendering (and render region) options, see *Rendering Options* on page 55 in the Rendering guide.

- For more information about batch rendering options, see *Rendering Methods* on page 131 in the Rendering guide.
Tips for Rendering Hair

While rendering hair is similar to rendering anything else in XSI, there are a few things to keep in mind. These are some general tips, but all suggestions are relative to the look you're creating for your hair or fur.

Setting the Look

- Making the root color darker than the tip helps to give more depth and realism to the hair strands. This is generally used to create or enhance a shadow effect on hair.
- Generally, the specular color value (not its decay) should be very high for dark hair and very low for light hair. Of course, this depends on the specific effect you're trying to achieve.
- For a more realistic hair look (especially for white or light-colored hair), use lots of hair strands with a fairly high transparency value. This also decreases shadowing because it allows light to pass through the hair volume. Having a very low transparency value for the roots can make the hairs opaque and look like hair plugs. However, to save time while previewing, deactivate the transparency in the shader.
- Make sure you're rendering with shadows on for the final render. Set the light's Umbra values to over 0.2 to give the best results. Use shadow maps with geometry hair for the best performance. You can also use volumic shadow maps to accurately render transparent or colored shadows (see Shadow-mapped Shadows on page 118).
- While you can use thicker hair (see Setting the Render Hair Thickness on page 86) and fewer of them to render quickly, for a final render you'll probably want to reduce the hair thickness and increase the number of hairs (Total Hairs value—see Setting the Number of Hairs Rendered on page 81). Very thick hair strands can penetrate each other and cause flickering.
- The overall number of hairs has a big impact on the render time. Depending on the look you need to achieve, reduce the overall number of hairs and instead increase the root and/or tip thickness. Using a slight kink or frizz can also help in giving the illusion of more hair.
- You can texture the underlying hair-emitting geometry with hair so that you can use fewer real hairs but still have a full look.
- Use the Strand Multiplier to increase the number of render hairs for good coverage (see Increasing the Number of Render Hairs on page 83). Splay at Root covers well and gives a slight clumping effect, if that's the look you're trying to achieve.
- Use weight maps to control exactly where hair appears on your character. For example, you can create a density map (see Removing Render Hairs (Density) on page 85) so that there are more hairs only where you need them, such as for doing close-ups on an animal's head.
Comb the hairs by a small amount in one direction. If hairs lie on top of each other, you don’t need as many hairs to cover up any bald spots.

**Optimizing Rendering Time**

Many production houses use two different shaders for hair:
- One is a surface shader that simulates the hair color and texture at a distance.
- The other is the real hair shader that defines the render hairs’ look.

Having the underlying skin of the object textured with an image that looks like hair can help tremendously because you can use a lot less hair and still have the same visual complexity.

Mixing different proportions of these two shaders in the right shot can give you the required look of hair without rendering time being unreasonable.

- Use as low a number of hair segments as is visually acceptable for the final render. You may find that higher values (such as 20 or more) are needed when the hair style is very long, frizzy, or curly. Short hair/fur can go as low as 8 (or less) for the final render.

  However, the price for precision is rendering time, as high segment values can slow down a scene. Use lower values of 2 or 5 for quick previewing and interaction while you’re styling the hair. See *Setting the Render Hair Resolution (Segments)* on page 88 for more information.

- Motion blur can increase render times by a large amount, so you should consider adding it as a post process.

- Shadows also have a high rendering price, but not as much as motion blur. If possible, don’t use self-shadowing.

  You can often decrease render times as well as improve the quality of shadows by using segment shadows as opposed to regular ones. If you’re rendering geometry hair, shadow maps are probably the most efficient (see *Shadows and Hair* on page 117).

- Use as few lights as possible for the hair, such as one shadow-casting light and up to three other lights in the scene.
Raytracing and Antialiasing

- Have raytracing selected as the rendering method when you are rendering hair with regular shadows. If you're using shadow maps, you can use the scan line method which is faster than raytracing.

For more information, see Selecting a Rendering Mode on page 80 in the Rendering guide. As well, see Controlling Aliasing on page 71 in the same guide.

- Use antialiasing for reducing any jagged edges on the hair. Start with settings of Min -1/Max 1 and progressively reduce the threshold until you reach an acceptable quality. Settings of 0/2 often give good results.

- Using a filter helps to avoid excessively sampling the hair while still getting a good quality image. Try using the Mitchell filter with a 4/4 size for a good speed while retaining much of the image’s sharpness.

Since the hairs are usually less than a pixel in diameter, the probability of a ray hitting a hair increases when you bump up the minimum aliasing level.

- Also you might want to set your sampling threshold to something like RGBA = 0.06. Increasing the number of samples per pixel will give you different results since, statistically, more hairs should get hit by sampling rays.

- For a good render, hair requires a higher sampling level than most geometry. While there is no sampling setting just for hair, you could render the hair in a separate pass so that the other objects in the scene can use lower sampling levels.

- Spend a good amount of time tweaking the hair BSP settings (see Setting the BSP Tree for Hair on page 128) to speed up rendering. Draw small render regions on the hair as you go so that previews are fast.

Rendering White Fur or Hair

Realistic white fur or hair can be fairly difficult to achieve, so here are some things to consider:

- White hair is not white: it’s chromatic gray. There are lots of tints of “off white” in white, but if you have the color turned to pure white (1), you won’t have much variation. You can use the Color Variation option (see Adding Color Variation (Random Color Mixing) on page 107) in the Hair Renderer shader to achieve subtle color changes.

- You don’t want to fill up the whole color spectrum range with the base color—you need to allow a little room for the specular color (which is additive). The white should be brighter than anything else in the frame, so make sure your subject and background don’t compete.

- White hair is self-illuminating in that light bounces all over the place inside. There is very little diffuse range, which means that the hair’s shading is mostly the base color—anything else comes from shadows.
You can set the Use Blend Gradient parameter (see Setting the Hair’s Diffuse and Ambient Colors on page 120) in the Hair Geo shader so that it’s mostly Ambient (about 85%) to diminish the effect of the diffuse shading. In other words, the shaded hair (pre-shadow) is just the base hair color with no dark-to-light variation.

The rest of the shading comes from the shadows and specular value. It’s not ambient in the traditional sense because the setting doesn’t affect the shadow density.

- Use lots of hair strands (Total Hairs value—see Setting the Number of Hairs Rendered on page 81) that are very transparent. This allows lots of light to pass through the volume, giving a nice fluffy appearance (if you’re trying to create fluffy fur).

- Try applying multiple instances of hair to the same object, each with different density settings (if it’s fur): one that’s long and loose (low stiffness, higher frizz), and one that’s short, dense, and stiff.
Index

A
antialiasing
  for hair 133
attenuating hair length 51

B
barycentric coordinates for hair 113
brushing
  hair 44
brushing hair See styling hair
BSP tree
  for hair 128

C
caching
  hair files for playback 64
clumping hair 57
clusters
  on hair 43
colors
  hair 104
combing hair 44
curves
  emitting hair from 32
cutting hair
  cut map 89
  cut operator 51

D
dampening
  of hair to dynamics 66
deformations
  on hair 39
  on hair with Stretchy mode 73
density
  of hair 85
dynamic behavior
  and styling on hair 67
  locking and unlocking points on
  hair 61
muting operator for hair 67
on hair 63
reaction of hair to 65

E
elasticity
  of hair segments 71
emitter objects
  for hair 24
emitting
  hair from objects 24
explorer
  hair elements in 18

F
field rendering
  hair 98
files
  caching for hair playback 64
final gathering
  and hair 116, 124
freezing
  hair 40
frizzy hair 94
fur See hair

G
geometry approximation
  hair 127
grooming hair See styling hair
guide hairs
  about 17
dampening effect of dynamics 66
displaying 34, 78
resistance to dynamics 65
stiffness 65
wiggling with dynamics 66
See also hair, render hairs, styling

H
hair
  about 16
antialiasing 133
barycentric coordinates in scalar
  state shader 113
brush tool 44
changing topology of emitter
  object 25
clusters of 43

collisions with obstacles 68
color 104
copying styles between objects 75
creating 23
deactivating geometry hair for
  rendering 130
deforming 39
deforming with Stretchy mode 73
deleting 25
density of 85
displaying guide and render
  hairs 34, 78
dynamics 63
dynamics and styling 38
emitting from curves 32
emitting from NURBS surfaces 28
emitting from polygons 26
emitting from subdivision
  surfaces 30
geometry approximation for
  viewing or rendering 127
Hair Geo shader 119
Hair Renderer shader 100
hiding 36
incandescence 125
instances 92
interpolation group IDs 60
length 50
light and shadows 115
matted or wet (clumping) 57
motion blur 98
natural forces on 74
number of 81
operator stack 38, 40
optimizing rendering 131
overview 21
playing simulation 64
primitive object (Hair) 18
property editor 20
removing 51
render type (geometry or
  volume) 101
rendering 97
replacing hair with geometry 92
resistance to dynamics 65
Index

scaling the emitter object 25
segmented rendering 98
segments (resolution) 88
segments, stretching 71
selecting components 42
shadow maps 118
shadows 117
texture maps on color parameters 109
thickness 86
translucency 123
transparency 122
white hair 133
See also guide hairs, Hair Geo shader, Hair Renderer shader, render hairs, styling hair
Hair Geo shader 119
ambient and diffuse colors 120
connecting in render tree 102
final gathering and global illumination 124
gradient controls 125
incandescence 125
specular color and decay 121
translucency 123
transparency 122
Hair Renderer shader
about 100
BSP tree 128
color variation 107
colors 104
connecting in render tree 102
final gathering 116
geometry and volume rendering 101
lighting for hair length 115
optimizing rendering 131
root and tip colors 106
self-shadows 117
specular color 106
transparency 114
Hair toolbar 19
I
incandescence
hair 125
instances
hair 92
interpolation
between guide hairs 58
K
kinky hair 95
L
locking
points on hair 61
M
matted hair 57
merging
hair interpolation 59
motion blur
for hair 98
muting
dynamics operator for hair 67
N
natural forces
on hair 74
O
obstacles
and hair 68
operator stack
hair 38, 40
optimizing
render time for hair 131
P
playing
hair simulations 64
points
selecting on hair objects 42
popping hair 52
presets
render hairs 78
puffing hair 56
R
recombing hair 48
render hairs
about 17
adding 83
cutting (cut map) 89
density map 85
displaying 34, 78
frizz 94
kinks and waves 95
number of 81
percentage displayed 34, 78
percentage rendered 82
presets 78
randomly scaling 89
segments (resolution) 88
splaying 83
thickness 86
See also guide hairs, hair, and styling hair
render options
geometry hair toggle 130
render tree
connecting hair shaders in 102
rendering
hair 97
number of hairs 82
segmented for hair 98
rotations
hair 53
S
scaling
hair 54
hair emitter object 25
random for hair 89
segmented rendering
for hair 98
segments of hair 88
selecting
hair components 42
self-shadows
for hair 117
shaders
connecting hair in render tree 102
shadow maps
for hair 118
shadows
  for hair 117
  hair self-shadows 117
shattering hair interpolation 59
snapping
  hair 20
splitting
  hair interpolation 58
stiffness
  of hair with dynamics 65
strands of hair, selecting 42
stretching
  hair segments for deformations and styling 71
Stretchy mode for hair 71
styling hair
  about 37
  brushing 44
  clumping 57
  combing 44
  copying styles between objects 75
deformations 39
freezing 40
locking points 61
merging interpolation 59
operator stack 38, 40
puffing 56
recombing 48
resetting 52
rotating 53
scaling 54
setting length 50
shattering interpolation 59
splitting interpolation 58
spreading 57
straightening 56
Stretchy mode 71
styling operators 38
translating points 53
unlocking points 61
with dynamics 38
subdivision surfaces
  emitting hair from 30

T
texture maps
  on hair colors 109
tips of hair, selecting 42
toolbars
  Hair 19
translations
  hair 53
translucency
  hair 123
transparency
  hair 114, 122
U
unlocking
  points on hair 61
W
wavy hair 95
weight maps
  cutting render hairs 90
wet hair 57
white hair 133
wigging
  of hair with dynamics 66
Index