

# Naming Convention

This document explains the way we name the exported Arnold nodes after the corresponding Softimage objects.

We try, as much as we can, to have a unique syntax for all the nodes (in older SltoA versions, we had several different ways to compose the names, and different characters to separate the name fields).

This also give a scripiter/developer the ability to retrieve a given node out of the native Softimage object, and vice-versa. For instance, after an .ass file has been exported, a user may want to edit one of its nodes having as a reference the Softimage scene.

As a general rule, the names have the following format:

*Softimage Object Full Name* + ".SltoA." + *something*

where "*something*" is a string depending on the node type and described in more details below.

So, if you need to retrieve the Softimage object out of the Arnold node name, you can simply trim the name string up to the last occurrence of the ".SltoA." substring.

Viceversa, if you need to retrieve an exported node name out of a Softimage object, here are the details, depending on the node type:

Softimage Object	Arnold Node(s)	Arnold Node(s) Name
Mesh	<i>polymesh</i>	Mesh FullName + ".SltoA." + time
Light	<i>point_light, distant_light, etc.</i>	Light FullName + ".SltoA." + time
Camera	<i>camera</i>	Camera FullName + ".SltoA." + time (2.6, was Camera plain Name in 2.5)
Shader	<i>standard, utility, etc.</i>	Shader FullName + ".SltoA." + time + "." + Uniqueld
Standin	<i>procedural</i>	Standin FullName + ".SltoA." + time
Instance	<i>ginstance</i>	Instance Model FullName + " " + master node name
Instance of light	<i>light (cloned)</i>	Instance Model FullName + " " + master node name
Hair	one <i>curves</i> per chunk	Hair FullName + ".SltoA." + time + "." + chunkId
Shape on hair	one <i>polymesh</i> per strand	Hair FullName + ".SltoA." + time + "." + strandId + " " + master node name

ICE:

PointCloud Shape	Arnold Node(s)	Arnold Node(s) Name
Points	<i>points (mode=disk)</i>	PointCloud FullName + ".SltoA.Disk." + time
Sphere	<i>points (mode=sphere)</i>	PointCloud FullName + ".SltoA.Sphere." + time
Disc	one <i>disk</i> per point	PointCloud FullName + ".SltoA.Disc." + time + "." + discId
Box	one <i>polymesh</i> with six quads per point	PointCloud FullName + ".SltoA.Box." + time
Rectangle	one <i>polymesh</i> with one quad per point	PointCloud FullName + ".SltoA.Rectangle." + time
Cylinder	one <i>cylinder</i> per point	PointCloud FullName + ".SltoA.Cylinder." + time + "." + cylinderId
Cone	one <i>cone</i> per point	PointCloud FullName + ".SltoA.Cone." + time + "." + coneId
Instance	one <i>ginstance</i> per point	PointCloud FullName + ".SltoA.Instance." + time + "." + instanceId + " " + master node name
Lights	one <i>light (cloned)</i> per point	PointCloud FullName + ".SltoA.Instance." + time + "." + instanceId + " " + master node name
Strands	<i>curves</i>	PointCloud FullName + ".SltoA.Strands." + time
Shape on strands	one <i>polymesh</i> per strand	PointCloud FullName + ".SltoA.InstanceStrands." + time + "." + strandId

where *time* is the Softimage frame multiplied by 1000 and cast to integer. In cpp that's:

```
(LONG)floor(in_frame * 1000.0f + 0.5f);
```

We use spaces in just one case, that is instances, for IPR reasons. For example, if you have a cube under a Model at frame 35, the cube is exported as:

```
polymesh
{
  name Model.cube.SItoA.35000
  ...
}
```

If this model is instanced by Model\_Instance, then the instanced cube is exported as

```
ginstance
{
  name "Model_Instance Model.cube.SItoA.35000"
  node "Model.cube.SItoA.35000"
  ...
}
```

So, also for ginstance nodes, it is easy to retrieve the master node and its Softimage object.

The various ID's (chunkId, disclId, etc.) are unique integers, needed when several Arnold nodes derive from a unique Softimage object. For instance, say you have a pointcloud setting the particle shape to a cone. A unique Softimage object (the point cloud) is exported as a collection of Arnold cone nodes, and each must have a unique name, as required by Arnold. This is what the .ass file looks like at frame 35:

```
cone
{
  name pointcloud.SItoA.Cone.35000.0
  ...
  shader "Sources.Materials.DefaultLib.Scene_Material.Phong.SItoA.35000.1"
  id 949
}

cone
{
  name pointcloud.SItoA.Cone.35000.1
  ...
  shader "Sources.Materials.DefaultLib.Scene_Material.Phong.SItoA.35000.1"
  id 949
}

cone
{
  name pointcloud.SItoA.Cone.35000.2
  ...
  shader "Sources.Materials.DefaultLib.Scene_Material.Phong.SItoA.35000.1"
  id 949
}

...
```