

# Primitive Variables



These may also be referred to as "prototype attributes" for clarity versus [Instance Attributes](#) which may be varied per instance.

## Common

Name	Type	Default	Description
identifier:object	string	empty	Object shape name.
stats:prototypeidentifier	string	empty	Identifier string used by the stats system when referring to this prototype.

## Shading

Name	Type	Default	Description
derivatives:extrapolate	int	1	Extrapolated smooth normals across grid boundaries. Value range: true/false.
displacement:ignorereferenceinstance	int	0	Indicates if displacement shading should ignore properties of the reference instance.
displacementbound:CoordinateSystem	string	object	The name of the coordinate system that the displacement bound is measured in.
displacementbound:offscreen	int	0	Apply displacementbound to offscreen geometry when dicing. Geometry that is entirely offscreen, but displaces into view may be prone to undertessellation because of the offscreen dicing strategy. This can be corrected by enabling this control and setting a reasonable displacementbound. Doing so will incur a performance penalty commensurate with the size of the displacement bound.
displacementbound:sphere	float	0	Maximum displacement distance used to compute object bounds. This value should be as tight as possible.
Ri:Orientation	string	outside	Geometry winding order that determines surface normal. This attribute effects lighting and displacement. Supported values: inside, outside.
trace:autobias	int	1	Enable automatic raytrace bias. Value range: true/false.
trace:bias	float	0.01	Manual raytrace bias in object space. Small values should be used to avoid artifacts such as shadow acne.
trace:displacements	int	1	Enable displacement shading. Value range: true/false.
trace:sssautobias	int	1	Enable automatic raytrace bias for subsurface scattering rays. Value range: true/false.
trace:sssbias	float	1e-4f	Manual raytrace bias (in object space) for subsurface scattering rays. Small values should be used to avoid artifacts such as surface acne.

## Dicing

Name	Type	Default	Description
dice:multipolygonlength	float	1	Micropolygon distance in raster space for "instanceprojection" dicing. Values are expressed in pixel size.
dice:offscreenmultiplier	float	-1	Dicing rate multiplier for objects outside the viewing frustum. If a value less than zero is specified, the value from the Option with the same name will be used instead. Value range: -1 or more.
dice:offscreenstrategy	string	viewfrustumdistance	Dicing method of objects outside the viewing frustum. Supported values: viewfrustumdistance, worlddistance, objectdistance, planarprojection, sphericalprojection.
dice:rasterorient	int	1	Turning this off enables non-oriented dicing, a mode of dicing that computes micropolygon sizes using non-oriented raster space rather than screen-aligned raster space. Non-oriented dicing is useful for displacement around edges where projected micropolygon distance can be arbitrarily small. Value range: true/false.

dice: referencecamera	<i>string</i>	<i>empty</i>	Specify the camera used for dicing. If no reference camera is specified, RenderMan will use the primary camera.
dice: referenceinstance	<i>string</i>	<i>empty</i>	Specify the reference instance used for dicing and displacement shading. The reference instance is used to drive dicing and displacement based on its position, scale, user attributes, and scoped coordinate systems. Reference instances are specified by its identifier:name attribute (instance name). If no reference instance is specified, RenderMan will automatically pick the nearest instance inside the primary camera's view frustum.
dice: strategy	<i>string</i>	instanceprojection	Dicing method of objects within the viewing frustum. Supported values: instanceprojection, worlddistance, objectdistance, sphericalprojection.
dice: worlddistancelength	<i>float</i>	-1	Micropolygon distance in world space for "worlddistance" dicing or object space for "objectdistance" dicing.
Ri: GeometricApproximationFocusFactor	<i>float</i>	0	Allows the renderer to use more coarse dicing for blurry objects due to depth of field.

## Points

Name	Type	Default	Description
falloffpower	<i>float</i>	0	For use with points, if not supplied, or set to zero, the points will have a hard edge. A value of 1 is a "reasonable" value that emulates the usual cosine based falloff; this will likely be the goto value for most people doing volumetric particle effects. Values between 0 and 1 makes the falloff faster, eroding the point faster - point has "less presence". Values higher than 1 (up to infinity) makes the falloff slower to the point of being non-existent.

## Curves

Name	Type	Default	Description
curve: opacitysamples	<i>int</i>	0	number of opacity/presence shading samples along a curve for opacity caching, if equal to zero then the value is determined using the dicing oracle and the length of each curve.
curve: widthaffectscurvature	<i>int</i>	1	When true the curve width of round curves is taken into account in the computation of the tube curvature, otherwise only the curvature along the curve is. This control does not affect curve ribbons. Value range: true/false.

## Volume

Name	Type	Default	Description
dice: minlength	<i>float</i>	-1	Volume minimum dice length. Negative indicates to automatically compute this value.
dice: minlengthspace	<i>string</i>	<i>empty</i>	Coordinate space of dice:minlength.
Ri: Bound	<i>float[6]</i>	0 0 0 0 0 0	Volume bounds.
volume: aggregate	<i>string</i>	globalVolumeAggregate	The name of the volume aggregate to which a volume belongs. If non-empty, the volume will only be rendered if the integrator explicitly uses that named aggregate, or if a surface shader such as PxrSurface binds the named aggregate to its interior.
volume: aggregaterespectvisibility	<i>int</i>	<i>empty</i>	If set to 1, standard visibility attributes will be respected by the volume if it is part of an aggregate. If set to 0, the visibility attributes will be ignored and only the integrator visibility settings will apply. Value range: true/false.
volume: dsominmax	<i>int</i>	<i>empty</i>	Currently only used for aggregate volumes, and only for volumes that use an ImplicitField DSO. If set to 1, the DSO may be able to use stored information from the file directly to compute the minimum and maximum values within the volume. This may allow the renderer to avoid shading the volume up front, greatly decreasing time to first pixel. This can only be enabled if the density signal from the volume is used directly, or if the density signal is modulated only by the DSO itself. Any shading modifications of the density signal requires setting this parameter to off. Value range: true/false.
volume: dsosvelocity	<i>int</i>	0	Used only for aggregate volumes that use an ImplicitField DSO. If set to 1, the DSO can provide velocity bounds to the renderer directly via the BoxMotion call. This can allow the renderer to avoid upfront shading of the volume's velocity data when using Eulerian velocity, improving render time and time to first pixel greatly. This can only be enabled if the velocity signal from the DSO is used directly. Value range: true/false.
volume: fps	<i>float</i>	1	The frames per second for volumetric velocity data. The velocity data will be divided by this quantity to derive the velocity for the frame.

volume: shutteroffset	<i>float</i>	1	The shutter offset used to interpret volumetric velocity. A value of 1 will use the current position of the object and the position of the object on the next frame as the time interval to use for motion blur. A value of -1 will use the position of the object on the previous frame and the current position of the object as the time. A value of 0 will generate an interval which begins halfway through the previous frame and ends halfway into the next frame.
volume: velocityshuttercorrection	<i>int</i>	0	If set to 1, volumetric velocity data for Eulerian motion blur will be assumed to be relative to the entire frame, and will be corrected to match the current shutter interval. If set to 0, no correction will be made. Value range: true/false.

## SubdivisionMesh

Name	Type	Default	Description
dice:pre tessellate	<i>int</i>	1	Pre-tessellate subdivision geometry to polygons. Value range: true/false.
dice:watertight	<i>int</i>	0	Tessellate geometry with no holes. Watertight geometry requires less raytrace bias. Value range: true/false.
shade:faceset	<i>int[n]</i>	<i>empty</i>	Active geometry face indexes.
stitchbound:CoordinateSystem	<i>string</i>	<i>empty</i>	
stitchbound:sphere	<i>float</i>	0	

## NuPatch

Name	Type	Default	Description
trimcurve:sense	<i>string</i>	inside	Supported values: inside, outside.

## PolygonMesh

Name	Type	Default	Description
polygon:concave	<i>int</i>	1	Allow concave polygons. Value range: true/false.
polygon:smoothdisplacement	<i>int</i>	0	Output smoothed (per-vertex) normals as Ndsp primvar, if polygon:smoothnormals hasn't already inserted smooth normals. Value range: true/false.
polygon:smoothnormals	<i>int</i>	0	Smooth (per-vertex) normals if not provided. Value range: true/false.

## Procedural

Name	Type	Default	Description
procedural: immediate subdivide	<i>int</i>	0	Used by RiProcedural to compel a procedural to immediately subdivide (behavior supported via the __immediatesubdivide meta-parameter in RiProcedural2). This attribute is "special" in that it is not inherited into sub-procedural contexts - meaning parent immediate-subdivide procedurals don't accidentally trigger immediate-subdivide children. A value of 0 (the default) disables, 1 enables, and 2 forces all child procedurals to crack immediately.
procedural: reentrant	<i>int</i>	0	Enabling this attribute will cause any subsequent procedural to not lock the global mutex. Note that this requires that the procedural be thread safe. This attribute is disabled by default, i.e. all procedurals remain locked. Value range: true/false.