tInitializes with 0 frame size
tFrame:(const NXRect *)fRectInitializes with specified frame size
eFrees the N3DCamera

OOL)lockFocusYES if PostScript and RenderMan drawing lock on camera
ockFocusUnlocks PostScript and RenderMan drawing
wSelf:(NXRect *)rects :(int)nRectsPerforms all RIB and PostScript drawing

derRenders the camera and any content shapes
derSelf:(RtToken)contextOverride to perform custom rendering in the camera
FlushRIB:(BOOL)flagSets the receiver to invoke flushRIB within render
esFlushRIBTests whether the receiver invokes flushRIB within render
shRIBWaits until all RIB code has been rendered

wPS:(NXRect *)rects :(int)nRectsOverride to perform custom PostScript drawing

BackgroundColor:(NXColor)colorSets the NXColor filled behind all drawing
XColor)backgroundColorThe NXColor filled behind all drawing
DrawBackgroundColor:(BOOL)flagIf flag is YES, fills color behind all drawing
OOL)doesDrawBackgroundColorYES if camera fills color behind all drawing

Frame:(const NXRect *)fRectSets frame for both PostScript and RenderMan coordinate systems
veTo:(NXCoord)x :(NXCoord)yMoves both PostScript and RenderMan coordinate systems
veBy:(NXCoord)deltaX :(NXCoord)deltaYMoves both PostScript and RenderMan coordinate systems
eTo:(NXCoord)width :(NXCoord)heightResizes both PostScript and RenderMan coordinate systems
eBy:(NXCoord)deltaWidth Resizes both PostScript and RenderMan coordinate
:(NXCoord)deltaHeightsystems
ateTo:(NXCoord)anglePrevents rotation of PostScript coordinate system
ateBy:(NXCoord)deltaAnglePrevents rotation of PostScript coordinate system

WorldShape:a3DShapeSets world shape
rldShapeReturns world shape

ProjectionRectangle:(float)left Sets the 3D coordinate system projection rectangle
:(float)right
:(float)top
:(float)bottom
ProjectionRectangle:(float *)left Returns the 3D coordinate system's projection
:(float *)right rectangle
:(float *)top
:(float *)bottom

Projection:(N3DProjectionType)aProjectionSets the projection type
3DProjectionType)projectionTypeReturns the projection type

PreTransformMatrix:(RtMatrix)theMatrixSets the camera's pretransformation matrix
PreTransformMatrix:(RtMatrix)theMatrixReturns the camera's pretransformation matrix
UsePreTransformMatrix:(BOOL)flagSets the camera to use its pretransformation matrix
OOL)usesPreTransformMatrixYES if camera uses its pretransformation matrix

EyeAt:(RtPoint)fromPoint Sets the eye-to-viewpoint vector and roll
toward:(RtPoint)toPoint
roll:(float)aRollAngle
EyeAt:(RtPoint *)fromPoint Gets the eye-to-viewpoint vector and roll
toward:(RtPoint *)toPoint
roll:(float *)aRollAngle
veEyeBy:(float)sDistance Moves camera in its own coordinate system
:(float)tDistance
:(float)uDistance
ateEyeBy:(float)dElev :(float)dAzim Rotates the camera in its own coordinates
about:(RtPoint)pivotPtr

ClipPlanesNear:(float)aNearPlane Sets the camera's near and far clipping planes
far:(float)aFarPlane
ClipPlanesNear:(float *)aNearPlane Returns the camera's near and far clipping planes
far:(float *)aFarPlane

FieldOfViewByAngle:(float)viewAngleSets the viewing angle of the camera
FieldOfViewByFocalLength:(float)aLengthConverts a focal length into a viewing angle for the camera
oat)fieldOfViewReturns the viewing angle of the camera

nvertPoints:(NXPoint *)mcoords Converts PostScript points to world coordinates
count:(int)pointCount
toWorld:(RtPoint *)wcoords

t)numCropWindowsCount of rectangle divisions for photoreal rendering
pInRects:(NXRect *)theRects Returns rectangles representing horizontal strips of
nRects:(int)rectCountcamera image

t)frameNumberReturns 1

OOL)canPrintRIBReturns YES

pyRIBCode:(NXStream *)streamCopies RIB code generated by the receiver

rldBegin:(RtToken)theContextCalls RiWorldBegin()
rldEnd:(RtToken)theContextCalls RiWorldEnd()

Delegate:cameraDelegateSets the receiver's delegate
egateReturns the receiver's delegate

3DHider)hiderReturns the receiver's N3DHider
Hider:(N3DHider)cameraHiderSets the receiver's N3DHider
SurfaceTypeForAll:(N3DSurfaceType)surface
chooseHider:(BOOL)flagSets surface type for shapes in world shape hierarchy

t)renderAsEPSBegins rendering, returns identifier for rendering session
t)renderAsTIFFBegins rendering, returns identifier for rendering session

d:(NXTypedStream *)theStreamReads the camera from the stream
te:(NXTypedStream *)theStreamWrites the camera to the stream
akePerforms additional initialization after unarchiving

eDestroys all contexts, frees the receiver

Token)mainContextReturns (creating if necessary) the application's main context

Token)createContext:(const char *)nameCreates a named context
Token)createContext:(const char *)nameCreates a named context for a specific renderer
withRenderer:(RtToken)renderer
Token)createContext:(const char *)cameCreates a named context on a file
toFile:(const char *)ribFile
Token)createContext:(const char *)nameDoes nothing, returns NULL
toStream:(NXStream *)stream

Token)currentContextThe application's current context
Token)setCurrentContext:(RtToken)contextSets the current context, returns previous context
Token)setCurrentContextByName:(const char *)name
Sets the current context by name, returns previous context

id)destroyContext:(RtToken)contextDestroys the context
id)destroyContextByName:(const char *)name
Destroys a context with name

akePerforms additional initialization after unarchiving

Initializes the receiver as an N3D_AmbientLight

Type:(N3DLightType)aTypeSets the receiver's light type
3DLightType)typeReturns the receiver's light type
keAmbientWithIntensity:(RtFloat)intensitySets type N3D_AmbientLight with appropriate parameter
kePointFrom:(RtPoint)from Sets type N3D_PointLight with appropriate
intensity:(RtFloat)intensityparameters
keDistantFrom:(RtPoint)fromPointSets type N3D_PointLight with appropriate

From:(RtPoint)fromPointSets the from point
From:(RtPoint)fromPoint Sets the from and to points
to:(RtPoint)toPoint
From:(RtPoint *)fromPoint Returns the from and to points
to:(RtPoint *)toPoint
ConeAngle:(RtFloat)coneAngle Sets the cone angle, cone delta, and beam distribution
coneDelta:(RtFloat)coneDelta
beamDistribution:(RtFloat)distribution
ConeAngle:(RtFloat *)coneAngle Gets the cone angle, cone delta, and beam distribution
coneDelta:(RtFloat *)coneDelta
beamDistribution:(RtFloat *)distribution
Intensity:(RtFloat)intensitySets the intensity
Float)intensityReturns the intensity

derSelf:(N3DCamera *)theCameraRenders the light as a local light
derGlobal:(N3DCamera *)theCameraRenders the light as a global light

Global:(BOOL)flagOverride to add behavior on becoming/resigning global
OOL)isGlobalYES if light is global

itchLight:(BOOL)flagIf flag is YES, turns receiver on

Color:(NXColor)theColorSets the receiver's color
XColor)colorReturns the receiver's color

d:(NXTypedStream *)theStreamReads the receiver from the stream
te:(NXTypedStream *)theStreamWrites the receiver to the stream
akePerforms additional initialization after unarchiving

tFrame:(const NXRect *)fRectInitializes the receiver

derRenders current frame if printing, renders current page

StartFrame:(int)start Sets counters for movie
endFrame:(int)end
incrementFramesBy:(int)skip
t)startFrameReturns the first frame number
t)endFrameReturns the last frame number
t)frameIncrementReturns the frame increment

d:(NXTypedStream *)theStreamReads the receiver from the stream
te:(NXTypedStream *)theStreamWrites the receiver to the stream
akePerforms additional initialization after unarchiving

essoryViewReturns the accessory view
AccessoryView:aViewSets the accessory view

t)runModalPresents the render panel in a model loop

t)resolutionReturns the resolution set by the user in the panel

t)numSelectedHostsReturns the number of hosts selected by the user
ar **)hostNamesReturns an array of strings for selected host names

t)browser:senderFills the panel's browser with host names
fillMatrix:theMatrix
inColumn:(int)col

tFromFile:(const char *)ribfileInitializes the receiver from a file

OOL)drawAt:(const NXPoint *)pointReturns YES if the image is successfully drawn at point
OOL)drawIn:(const NXRect *)rectReturns YES if the image is successfully drawn in rect
OOL)drawReturns YES if the image is successfully drawn

BoundingBox:(NXRect *)rectangleReturns the rectangle that bounds the image
Size:(NXSize *)theSizeReturns the size of the image

XColor)backgroundColorReturns the background color
BackgroundColor:(NXColor)theColorSets the background color

3DHider)hiderReturns the hider type for rendering images
Hider:(N3DHider)theHiderSets the hider type for rendering images

SurfaceType:(N3DSurfaceType)typeSets the surface type used for rendering images
3DSurfaceType)surfaceTypeReturns the surface type used for rendering images

d:(NXTypedStream *)theStreamReads the receiver from the stream
te:(NXTypedStream *)theStreamWrites the receiver to the stream

Initializes the receiver
tWithCamera:myCameraInitializes the receiver and sets its camera

Camera:myCameraSets the receiver's camera
Center:(const NXPoint *)center Sets the receiver's center point and radius
andRadius:(float)radius

RotationAxis:(N3DAxis)axisSets the axes about which the receiver rotates
3DAxis)rotationAxisReturns the axis about which the receiver rotates

d:(NXTypedStream *)theStreamReads the receiver from the stream
te:(NXTypedStream *)theStreamWrites the receiver to the stream

Initializes the receiver with no shader file
WithShader:(const char *)aShaderInitializes the receiver with a shader file
eFrees the receiver

Shader:(const char *)aShaderSets the receiver's shader
nst char *)shaderReturns the receiver's shader

Color:(NXColor)aColorSets the receiver's color
XColor)colorReturns the receiver's color
UseColor:(BOOL)flagSets the receiver to apply its color
OOL)doesUseColorYES if receiver applies color

Transparency:(float)alphaValueSets the receiver's transparency
oat)transparencyReturns the receiver's transparency

t)shaderArgCountThe number of arguments for the shader function
nst char *)shaderArgNameAt:(int)argIndexThe name of the indicated argument
LO_TYPE)shaderArgType:(const char *)argName
The type of the named argument
OOL)isShaderArg:(const char *)argNameYES if argName is an argument to the shader function
ShaderArg:(const char *)floatName Sets the specified argument to the specified value
floatValue:(float)floatValue
ShaderArg:(const char *)stringName Sets the specified argument to the specified value
stringValue:(const char *)stringValue
ShaderArg:(const char *)pointName Sets the specified argument to the specified value
pointValue:(RtPoint)pointValue
ShaderArg:(const char *)colorName Sets the specified argument to the specified value
colorValue:(NXColor)colorValue
ShaderArg:(const char *)floatName Gets the specified value for the specified argument
floatValue:(float *)floatValue
ShaderArg:(const char *)stringName Gets the specified value for the specified argument
stringValue:(const char **)stringValue
ShaderArg:(const char *)pointName Gets the specified value for the specified argument

Applies the shader function during rendering

d:(NXTypedStream *)theStreamReads the receiver from the stream
te:(NXTypedStream *)theStreamWrites the receiver to the stream

tInitializes and returns the receiver
eFrees the receiver and its descendants
eAllFrees the receiver, its next peer and its descendants

der:(N3DCamera *)theCameraRenders the shape and its descendants
derSelf:(N3DCamera *)theCameraOverride to implement actual rendering
derSelfAsBox:(N3DCamera *)theCameraRenders only the shape's bounding box

xtPeerReturns the shape ªto the rightº
viousPeerReturns the shape ªto the leftº
stPeerReturns the shape ªto the far leftº of receiver's peer group
tPeerReturns the shape ªto the far rightº of receiver's peer group
scendantReturns the shape ªbelowº the receiver
tDescendantReturns the farthest descendant below the receiver
cestorReturns the shape ªabove the receiver
stAncestorReturns the shape at the top of the receiver's hierarchy
OOL)isWorldYES if the receiver is at the top of its hierarchy

kPeer:aPeerInserts aPeer between receiver and its next peer
kDescendant:aDescendantInserts aDescendant between receiver and its descendant
kAncestor:anAncestorSets receiver's ancestor, returns previous ancestor
inkUnlinks the receiver, reconnects peers and descendants
up:anAncestorMakes the receiver a descendant of anAncestor
groupRemoves receiver from hierarchy, promotes descendant

Shader:aShaderSets an N3DShader for the shape
derType:(SLO_TYPE)typeReturns the N3DShader of type

OOL)doesDrawAsBoxYES if the receiver draws only its bounding box

Bounds:(NXRect *)boundingRect Returns the rectangle representing the receiver's
inCamera:theCamerabounding box in the coordinates of the camera

nvertObjectPoints:(RtPoint *)points Converts points from the receiver's coordinate system
count:(int)n to that of camera
toCamera:camera

nvertPoints:(RtPoint *)points Converts points from an ancestor's coordinate system
count:(int)n to that of the receiver
fromAncestor:(N3DShape *)theShape

nvertPoints:(RtPoint *)points Converts points from the receiver's coordinate system
count:(int)n to that of an ancestor
toAncestor:(N3DShape *)theShape

Selectable:(BOOL)flagSets whether the receiver can be selected

OOL)isSelectableYES if the receiver can be selected

Visible:(BOOL)flagSets the receiver to render when renderSelf: is invoked

OOL)isVisibleYES if receiver renders when renderSelf: is invoked

ShapeName:(const char *)aNameSets the name of the shape

nst char *)shapeNameReturns the name of the shape

RenderDelegate:aShapeSets the rendering delegate for the receiver

noveRenderDelegateRemoves and returns the render delegate

derDelegateReturns the render delegate

TransformMatrix:(RtMatrix)newTransform
Sets the receiver's transformation matrix

TransformMatrix:(RtMatrix)transformReturns the receiver's transformation matrix

ncatTransformMatrix:(RtMatrix)theMatrixMultiplies the transform matrix with theMatrix,
premultiply:(BOOL)flagpremultiplying by theMatrix if flag is YES

CompositeTransformMatrix:(RtMatrix)theMatrix
relativeToAncestor:(N3DShape *)theAncestor
Returns the matrix representing the transformation between theAncestor's space and the receiver's

InverseCompositeTransformMatrix:(RtMatrix)theMatrix
relativeToAncestor:(N3DShape *)theAncestor
Returns the matrix representing the transformation between the receiver's space and theAncestor's

:(float)yScaleFactor
:(float)zScaleFactor
leUniformly:(float)scaleFactorScales the receiver uniformly in all axes
scaleUniformly:(float)scaleFactorScales the receiver uniformly in all axes, premuliplying the transformation
nslate:(float)xTranslation Translates the receiver
:(float)yTranslation
:(float)zTranslation
translate:(float)xTranslation Translates the receiver, premultiplying the
:(float)yTranslation transformation
:(float)zTranslation

d:(NXTypedStream *)theStreamReads the camera from the stream
te:(NXTypedStream *)theStreamWrites the camera to the stream
akePerforms additional initialization after unarchiving