```
Implementation file: tcl3dCgQuery.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dCg
       FileName:       tcl3dCgQuery.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with query procedures related to
                       the Cg module.


       Name:           tcl3dCgGetVersion - Get Cg version string.

       Synopsis:       tcl3dCgGetVersion {}

       Description:    Return the version string of the wrapped Cg library.
                       The version is returned as "Major.Minor.Patch".

       See also:       tcl3dOglGetVersions
                       tcl3dGetLibraryInfo
```

```
Implementation file: tcl3dCgUtil.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dCg
       FileName:       tcl3dCgUtil.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with miscellaneous utility
                       procedures related to the Cg module.


       Name:           tcl3dCgGetError - Check for a Cg error.

       Synopsis:       tcl3dCgGetError { contextId { msg "" } }

       Description:    contextId       : Cg context identifier
                       msg             : Additional message string

                       Check, if a Cg related error has occured.
                       The Cg context - as returned by cgCreateContext - has to
                       be supplied with parameter "contextId".

                       The procedure returns an empty string, if no error has
                       occurred. Otherwise it returns the additional message
                       string, the error number and Cg error message as supplied
                       by the Cg library as a formatted string.

       See also:       tcl3dOglGetError


       Name:           tcl3dCgGetProfileList - Get a list of Cg profile names.

       Synopsis:       tcl3dCgGetProfileList { }

       Description:    Return a Tcl list of Cg profile names.

                       The list consists of (key,value) pairs, where key is the
                       profile name, like CG_PROFILE_FP30 and value is either 1,
                       if the corresponding profile is supported, or 0, if it
                       is not available.

       See also:       tcl3dCgFindProfile
                       tcl3dCgFindProfileByNum


       Name:           tcl3dCgFindProfile - Find a supported Cg profile.

       Synopsis:       tcl3dCgFindProfile { args }

       Description:    args    : Profile names

                       Find the first profile supported by the Cg implementation
                       from the profile names supplied in "args".
                       If successful, it returns the profile name, otherwise an
                       empty string.

       See also:       tcl3dCgGetProfileList
                       tcl3dCgFindProfileByNum


       Name:           tcl3dCgFindProfileByNum - Find a supported Cg profile.

       Synopsis:       tcl3dCgFindProfileByNum { profileNum }

       Description:    profileNum : int (CGprofile)
```

```
                Find a profile name by it's numerical value supplied
                in "profileNum".
                If successful, it returns the profile name, otherwise
                an empty string.

                Note: The procedure does not check, if the profile is
                      supported. Use tcl3dCgFindProfile to check for
                      support by the underlying Cg implementation.

See also:       tcl3dCgGetProfileList
                tcl3dCgFindProfile


Name:           tcl3dCgPrintProgramInfo - Print Cg program info.

Synopsis:       tcl3dCgPrintProgramInfo { progId { progFile "Unknown" } }

Description:    progId          : Cg Program identifier
                progFile        : string

                Print the profile name and the name of the entry function
                of the Cg program identified by "progId".
                The Cg program identifier is the identifier as returned
                by calls to the cgCreateProgram familiy.
                An optional parameter "progFile" can be supplied to
                specify the name of the file containing the Cg program
                source code.

See also:
```

```
Implementation file: tcl3dHeightMap.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dDemoUtil
       FileName:       tcl3dHeightMap.tcl

       Author:         Paul Obermeier

       Description:    Tcl module to create a heightmap from a photo image.
                       This functionality is needed for NeHe tutorial 45.


       Name:           tcl3dDemoUtilHeightmapFromPhoto - Create a heightmap
                       from an image.

       Synopsis:       tcl3dDemoUtilHeightmapFromPhoto {
                               phImg flHeightScale flResolution }

       Description:    phImg           : string (Photo image identifier)
                       flHeightScale   : float
                       flResolution    : float

                       Create two Tcl3D vectors containing the vertices and
                       texture coordinates of a heightmap created from the image
                       data of photo image "phImg".
                       The height values can be scaled with "flHeightScale".
                       "flResolution" indicates how many pixels form a vertex.

                       The two vectors and the number of vertices generated are
                       returned as a Tcl list:
                       Index 0: Vertex vector
                       Index 1: Texture coordinates vector
                       Index 2: Number of vertices

       See also:
```

```
Implementation file: tcl3dFTGLQuery.tcl

       Copyright:      2007-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dFTGL
       FileName:       tcl3dFTGLQuery.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with with query procedures related to
                       the FTGL module.


       Name:           tcl3dFTGLGetVersion - Get FTGL version string.

       Synopsis:       tcl3dFTGLGetVersion {}

       Description:    Return the version string of the wrapped FTGL library.
                       The version is returned as "Major.Minor.Patch".

                       Note: FTGL does not support version numbers in the code,
                             so the version number is hand-coded here.

       See also:       tcl3dOglGetVersions
                       tcl3dGetLibraryInfo
```

```
Implementation file: tcl3dFTGLUtil.tcl

       Copyright:      2006-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dFTGL
       FileName:       tcl3dFTGLUtil.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with miscellaneous utility
                       procedures related to the FTGL module.


       Name:           tcl3dFTGLGetBBox - Get bounding box of a string.

       Synopsis:       tcl3dFTGLGetBBox { font str }

       Description:    font    : string (Font identifier)
                       str     : string

                       Return the bounding box of string "str" displayed in
                       font "font".
                       The bounding box is returned as a list of 6 values:
                       { xmin ymin zmin xmax ymax zmax }
                       "font" must be an identifier as returned by one of the
                       FTGL*Font functions (ex. FTGLBitmapFont).

       See also:
```

```
Implementation file: tcl3dGl2psQuery.tcl

       Copyright:      2007-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dGl2ps
       FileName:       tcl3dGl2psQuery.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with query procedures related to
                       the GL2PS module.


       Name:           tcl3dGl2psGetVersion - Get GL2PS version string.

       Synopsis:       tcl3dGl2psGetVersion {}

       Description:    Return the version string of the wrapped GL2PS library.
                       The version is returned as "Major.Minor.Patch".

       See also:       tcl3dOglGetVersions
                       tcl3dGetLibraryInfo
```

```
Implementation file: tcl3dGl2psUtil.tcl

       Copyright:      2006-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dGl2ps
       FileName:       tcl3dGl2psUtil.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with miscellaneous utility
                       procedures related to the GL2PS module.


       Name:           tcl3dGl2psCreatePdf - Create PDF from OpenGL content.

       Synopsis:       tcl3dGl2psCreatePdf { toglwin filename
                               { title "Tcl3D Screenshot" }
                               { drawBackground 0 }
                               { producer "Tcl3D" } }

       Description:    toglwin        : string (Togl identifier)
                       filename       : string
                       title          : string
                       drawBackground : boolean
                       producer       : string

                       Create a PDF file from current Togl window content.
                       The PDF is created from the Togl window identified by
                       "toglwin" and written to file "filename".
                       The following optional parameters set PDF specific
                       values:
                       "title" is the name of the document title as
                       listed in the document properties of the PDF file.
                       If "drawBackground" is set to true, the background
                       color of the Togl window is also used as the background
                       color of the PDf document. Otherwise the PDF background
                       color is set to white.
                       "procuder" is the name of the producer property as
                       listed in the document properties of the PDF file.

       See also:
```

```
Implementation file: tcl3dOdeQuery.tcl

       Copyright:      2007-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dOde
       FileName:       tcl3dOdeQuery.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with query procedures related to
                       the ODE module.


       Name:           tcl3dOdeGetVersion - Get ODE version string.

       Synopsis:       tcl3dOdeGetVersion {}

       Description:    Return the version string of the wrapped ODE library.
                       The version is returned as "Major.Minor.Patch".

                       Note: ODE does not support version numbers in the code,
                             so the version number is hand-coded here.

       See also:       tcl3dOglGetVersions
                       tcl3dGetLibraryInfo
```

```
Implementation file: tcl3dOglQuery.tcl

       Copyright:      2007-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dOgl
       FileName:       tcl3dOglQuery.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with query procedures related to
                       the OpenGL module.


       Name:           tcl3dOglGetVersion - Get OpenGL version string.

       Synopsis:       tcl3dOglGetVersion {}

       Description:    Return the version string of the wrapped OpenGL library.
                       The version string does not have a specific format.
                       It depends on the vendor of the OpenGL implementation.
                       Some examples:
                       1.2 APPLE-1.4.56
                       TODO

                       If no OpenGL context has been established (i.e. a Togl
                       window has not been created), the function returns an
                       empty string.

       See also:       tcl3dOglGetVersions
                       tcl3dGetLibraryInfo
                       tcl3dOglHaveVersion


       Name:           tcl3dOglHaveExtension - Check availability of a specific
                       OpenGL extension.

       Synopsis:       tcl3dOglHaveExtension { extensionName }

       Description:    extensionName : string

                       Return 1, if the OpenGL extension "extensionName" is
                       provided by the underlying OpenGL implementation.
                       Otherwise return 0.

                       Example: tcl3dOglHaveExtension GL_ARB_multitexture
                                checks the availability of the multitexturing
                                extension.

                       Note: A Togl window (and therefore a graphics context)
                             must have been created before issuing a call to
                             this function.

       See also:       tcl3dOglGetExtensions


       Name:           tcl3dOglHaveVersion - Check availability of a specific
                       OpenGL version.

       Synopsis:       tcl3dOglHaveVersion { majorWanted { minorWanted 0 }
                                             { patchWanted 0 } }

       Description:    majorWanted : int
                       minorWanted : int
                       patchWanted : int

       Description:    Return 1, if the OpenGL version fits the supplied
                       major, minor and patch level numbers.
```

```
                Otherwise return 0.

                Note: The version number of the OpenGL implementation
                      is extracted from the string returned by calling
                      "glGetString GL_VERSION". As some vendors format
                      the version in an unusual way, this function may
                      not work correctly on all platforms.

                Note: A Togl window (and therefore a graphics context)
                      must have been created before issuing a call to
                      this function.

See also:       tcl3dOglGetVersions


Name:           tcl3dOglGetVersions - Get OpenGL version information.

Synopsis:       tcl3dOglGetVersions {}

Description:    Return OpenGL version information as a list of
                (key,value) pairs.
                Keys are the following OpenGL version types:
                GL_VENDOR, GL_RENDERER, GL_VERSION, GLU_VERSION.
                Values are the corresponding version strings as returned
                by the underlying OpenGL implementation.

                Example:
                {GL_VENDOR {Intel Inc.}}
                {GL_RENDERER {Intel GMA 950 OpenGL Engine}}
                {GL_VERSION {1.2 APPLE-1.4.56}}
                {GLU_VERSION {1.3 MacOSX}}

                Note: A Togl window (and therefore a graphics context)
                      must have been created before issuing a call to
                      this function.

See also:       tcl3dOglHaveVersion
                tcl3dOglGetExtensions


Name:           tcl3dOglGetExtensions - Get all supported OpenGL extensions.

Synopsis:       tcl3dOglGetExtensions {}

Description:    Return a two-element list containing OpenGL extension
                information. The first sub-list constains all OpenGL
                extensions, the second sub-list contains all GLU
                extensions supported by the OpenGL implementaion.

                Note: A Togl window (and therefore a graphics context)
                      must have been created before issuing a call to
                      this function.

See also:       tcl3dOglHaveExtension
                tcl3dOglGetVersions


Name:           tcl3dOglGetIntState - Get OpenGL state variable.

Synopsis:       tcl3dOglGetIntState { state { numVals 1 } }

Description:    state    : GLenum
                numVals  : int

                Utility function to query an integer OpenGL state
                variable with glGetIntegerv.
                The state variable to be queried is specified as an
                GLenum in parameter "state".

                The value of the state variable is returned as an
                integer scalar value, if "numVals" is 1. If "numVals" is
                greater than 1, a Tcl list is returned.
                Note: See Appendix B of the OpenGL Red Book for a list
                      of state variables.

See also:       tcl3dOglGetFloatState
                tcl3dOglGetDoubleState


Name:           tcl3dOglGetFloatState - Get OpenGL state variable.

Synopsis:       tcl3dOglGetFloatState { state { numVals 1 } }

Description:    state    : GLenum
                numVals  : int

                Utility function to query a 32-bit floating point
                OpenGL state variable with glGetFloatv.
                The state variable to be queried is specified as an
                GLenum in parameter "state".

                The value of the state variable is returned as a
                float scalar value, if "numVals" is 1. If "numVals" is
                greater than 1, a Tcl list is returned.

                Note: See Appendix B of the OpenGL Red Book for a list
                      of state variables.

See also:       tcl3dOglGetIntState
                tcl3dOglGetDoubleState


Name:           tcl3dOglGetDoubleState - Get OpenGL state variable.

Synopsis:       tcl3dOglGetDoubleState { state { numVals 1 } }

Description:    state    : GLenum
                numVals  : int

                Utility function to query a 64-bit floating point
                OpenGL state variable with glGetDoublev.
                The state variable to be queried is specified as an
                GLenum in parameter "state".

                The value of the state variable is returned as a
                double scalar value, if "numVals" is 1. If "numVals" is
                greater than 1, a Tcl list is returned.

                Note: See Appendix B of the OpenGL Red Book for a list
                      of state variables.

See also:       tcl3dOglGetIntState
                tcl3dOglGetFloatState


Name:           tcl3dOglGetMaxTextureSize - Get maximum texture size.

Synopsis:       tcl3dOglGetMaxTextureSize {}

Description:    Utility function to get maximum size of a texture.
                The maximum texture size is returned as integer value.
                This function corresponds to querying state variable
                GL_MAX_TEXTURE_SIZE.

See also:       tcl3dOglGetIntState
                tcl3dOglGetMaxTextureUnits


Name:           tcl3dOglGetMaxTextureUnits - Get maximum texture units.

Synopsis:       tcl3dOglGetMaxTextureUnits {}

Description:    Utility function to get maximum number of texture units.
```

```
                The maximum number of texture units is returned as an
                integer value.
                This function corresponds to querying state variable
                GL_MAX_TEXTURE_UNITS.

See also:       tcl3dOglGetIntState
                tcl3dOglGetMaxTextureSize


Name:           tcl3dOglGetViewport - Get current viewport.

Synopsis:       tcl3dOglGetViewport {}

Description:    Utility function to get the current viewport.
                The viewport is returned as a 4-element Tcl list:
                { LowerLeftX LowerLeftY Width Height }
                This function corresponds to querying state variable
                GL_VIEWPORT.

See also:       tcl3dOglGetIntState


Name:           tcl3dOglGetShaderInfoLog - Get shader object log.

Synopsis:       tcl3dOglGetShaderInfoLog { shader }

Description:    shader  : Shader object

                Utility function for easier use of OpenGL function
                glGetShaderInfoLog.
                Given the shader object (as returned by function
                glCreateShader), the function returns the
                information log message as a Tcl string.

See also:       tcl3dOglGetProgramInfoLog
                tcl3dOglGetShaderSource
                tcl3dOglGetInfoLogARB


Name:           tcl3dOglGetProgramInfoLog - Get shader program log.

Synopsis:       tcl3dOglGetProgramInfoLog { shader }

Description:    shader  : Shader program

                Utility function for easier use of OpenGL function
                glGetProgramInfoLog.
                Given the shader program (as returned by function
                glCreateProgram), the function returns the
                information log message as a Tcl string.

See also:       tcl3dOglGetShaderInfoLog
                tcl3dOglGetShaderSource
                tcl3dOglGetInfoLogARB


Name:           tcl3dOglGetShaderSource - Get shader object source.

Synopsis:       tcl3dOglGetShaderSource { shader }

Description:    shader  : Shader object

                Utility function for easier use of OpenGL function
                glGetShaderSource.
                Given the shader object (as returned by function
                glCreateShader), the function returns the
                shader source code as a Tcl string.

See also:       tcl3dOglGetShaderInfoLog
                tcl3dOglGetProgramInfoLog
                tcl3dOglGetInfoLogARB
```

```
Name:           tcl3dOglGetInfoLogARB - Get shader object log.

Synopsis:       tcl3dOglGetInfoLogARB { object }

Description:    object  : Shader object

                Utility function for easier use of OpenGL function
                glGetInfoLogARB.
                Given the shader object (as returned by function
                glCreateProgramObjectARB), the function returns the
                information log message as a Tcl string.

See also:       tcl3dOglGetShaderInfoLog
                tcl3dOglGetProgramInfoLog
                tcl3dOglGetShaderSource


Name:           tcl3dOglGetStates - Get OpenGL state variables.

Synopsis:       tcl3dOglGetStates { {sortFlag "none"} }

Description:    sortFlag : string (increasing|decreasing|none)

                Query all state variables of the OpenGL library and
                return the results as a list of sub-lists. Each sublist
                contains a flag indicating the sucess of the query,
                the query command used, the key and the value(s).

                Note: This function is still incomplete. Chances are
                      high, it will never be finished.

See also:       tcl3dOglGetExtensions
                tcl3dOglGetVersions
```

```
Implementation file: tcl3dOglUtil.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dOgl
       FileName:       tcl3dOglUtil.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with miscellaneous utility
                       procedures related to the OpenGL module.


       Name:           tcl3dOglExtInit - Initialize the extension library.

       Synopsis:       tcl3dOglExtInit {}

       Description:    Initialize the OpenGL extension library.
                       If no OpenGL context has been established (i.e. a Togl
                       window has not been created), the function throws an
                       error.
                       It is recommended to add a call to tcl3dOglExtInit in
                       the create callback.

       See also:


       Name:           glMultiDrawElements - Draw multiple array elements.

       Synopsis:       glMultiDrawElements {
                               mode count type indices primcount }

       Description:    mode      : GLenum
                       count     : tcl3dVector of type GLuint
                       type      : GLenum
                       indices   : List of tcl3dVectors
                       primcount : integer

                       Tcl wrapper for the OpenGL function glMultiDrawElements.
                       As the Tcl3D Swig wrapper currently does not support
                       "void **" pointers, the multiple array elements must be
                       specified as a list of tcl3dVectors.

       See also:


       Name:           tcl3dOglGetError - Check for an OpenGL error.

       Synopsis:       tcl3dOglGetError {}

       Description:    Check, if an OpenGL related error has been occurred.

                       If no error occurred, an emtpy string is returned.
                       Otherwise a formatted string showing the error number
                       and the error message is returned.

       See also:


       Name:           tcl3dOglShaderSource - Wrapper for glShaderSource.

       Synopsis:       tcl3dOglShaderSource { shaderId shaderString }

       Description:    shaderId     : Shader handle
                       shaderString : string

                       Wrapper for easier use of OpenGL function glShaderSource.
```

```
                    In contrast to glShaderSource only the shader program
                    identifier (created with a call to glCreateShaderObject)
                    and the shader source have to be specified.

      See also:
```

```
Implementation file: tcl3dSDLQuery.tcl

       Copyright:      2007-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dSDL
       FileName:       tcl3dSDLQuery.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with query procedures related to
                       the SDL module.


       Name:           tcl3dSDLGetVersion - Get SDL version string.

       Synopsis:       tcl3dSDLGetVersion {}

       Description:    Return the version string of the wrapped SDL library.
                       The version is returned as "Major.Minor.Patch".

       See also:       tcl3dOglGetVersions
                       tcl3dGetLibraryInfo
```

```
Implementation file: tcl3dSDLUtil.tcl

       Copyright:      2006-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dSDL
       FileName:       tcl3dSDLUtil.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with miscellaneous utility
                       procedures related to the SDL module.


       Name:           tcl3dSDLGetFocusName - Convert focus state bitfield.

       Synopsis:       tcl3dSDLGetFocusName { state }

       Description:    state : int

                       Return a SDL focus state bitfield as the corresponding
                       string representation.
                       See file SDL_active.h for the definition of possible
                       bitfield values.

       See also:       tcl3dSDLGetButtonName
                       tcl3dSDLGetHatName
                       tcl3dSDLGetEventName


       Name:           tcl3dSDLGetButtonName - Convert button state bitfield.

       Synopsis:       tcl3dSDLGetButtonName { state }

       Description:    state : int

                       Return a SDL button state bitfield as the corresponding
                       string representation.
                       See file SDL_mouse.h for the definition of possible
                       bitfield values.

       See also:       tcl3dSDLGetFocusName
                       tcl3dSDLGetHatName
                       tcl3dSDLGetEventName


       Name:           tcl3dSDLGetHatName - Convert hat state bitfield.

       Synopsis:       tcl3dSDLGetHatName { state }

       Description:    state : int

                       Return a SDL hat state bitfield as the corresponding
                       string representation.
                       See file SDL_joystick.h for the definition of possible
                       bitfield values.

       See also:       tcl3dSDLGetFocusName
                       tcl3dSDLGetButtonName
                       tcl3dSDLGetEventName


       Name:           tcl3dSDLGetEventName - Convert event enumeration.

       Synopsis:       tcl3dSDLGetEventName { state }

       Description:    state : int (SDL event enumeration)
```

```
                Return SDL event related enumeration as the
                corresponding string representation.
                See file SDL_events.h for the definition of possible
                enumeration values.

See also:       tcl3dSDLGetFocusName
                tcl3dSDLGetButtonName
                tcl3dSDLGetHatName


Name:           tcl3dSDLFrames2MSF - Convert CD frames.

Synopsis:       tcl3dSDLFrames2MSF { frames }

Description:    frames : int

                Return CD frame as minutes/seconds/frames as a list
                of 3 integers.

See also:


Name:           tcl3dSDLGetTrackTypeName  - Convert track type bitfield.

Synopsis:       tcl3dSDLGetTrackTypeName { type }

Description:    type : int

                Return SDL CD track type bitfield as the corresponding
                string representation.
                See file SDL_cdrom.h for the definition of possible
                bitfield values.

See also:       tcl3dSDLGetCdStatusName


Name:           tcl3dSDLGetCdStatusName  - Convert CD status enumeration.

Synopsis:       tcl3dSDLGetCdStatusName { status }

Description:    status : int (CD status enumeration)

                Return SDL CD status enumeration as the
                corresponding string representation.
                See file SDL_cdrom.h for the definition of possible
                enumeration values (CDstatus).

See also:       tcl3dSDLGetTrackTypeName
```

```
Implementation file: tcl3dAutoscroll.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)
                       2003 Kevin B Kenny <kennykb@users.sourceforge.net>

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dAutoscroll.tcl

       Author:         Paul Obermeier

       Description:    Tcl module to create scroll bars that automatically
                       appear when a window is too small to display its
                       content.


       Name:           ::tcl3dAutoscroll::autoscroll - Create auto scrollbar

       Synopsis:       ::tcl3dAutoscroll::autoscroll { w }

       Description:    w : string (Widget name)

                       Create a scroll bar that disappears when it is not
                       needed, and reappears when it is.
                       The scroll bar "w" should already exist.

                       Side effects:
                       The widget command is renamed, so that the 'set' command
                       can be intercepted and determine whether the widget
                       should appear.
                       In addition, the 'Autoscroll' bind tag is added to the
                       widget, so that the <Destroy> event can be intercepted.

       See also:       ::tcl3dAutoscroll::unautoscroll


       Name:           ::tcl3dAutoscroll::unautoscroll - Remove scrollbar from
                       package control.

       Synopsis:       ::tcl3dAutoscroll::unautoscroll { w }

       Description:    w : string (Widget name)

                       Return a scrollbar to its normal static behavior by
                       removing it from the control of this package.

                       "w" is the path name of the scroll bar, which must have
                       previously had ::tcl3dAutoscroll::autoscroll called on it.

                       Side effects:
                       The widget command is renamed to its original name.
                       The widget is mapped if it was not currently displayed.
                       The widgets bindtags are returned to their original state.
                       Internal memory is cleaned up.

       See also:       ::tcl3dAutoscroll::autoscroll


       Name:           ::tcl3dAutoscroll::widgetCommand - Apply a widget command.

       Synopsis:       ::tcl3dAutoscroll::widgetCommand { w command args }

       Description:    w       : string (Widget name)
                       command : string
                       args    : argument list

                       Apply widget command "command" on 'autoscroll' scrollbar
                       "w". Arguments to the commands can be supplied in "args".
```

```
                Return whatever the widget command returns.

                Side effects:
                Has whatever side effects the widget command has. In
                addition, the 'set' widget command is handled specially,
                by gridding/packing the scroll bar according to whether
                it is required.

See also:       SEE_ALSO


Name:           ::tcl3dAutoscroll::destroyed - Destroy callback.

Synopsis:       ::tcl3dAutoscroll::destroyed { w }

Description:    w : string (Widget name)

                Callback executed when automatic scroll bar "w" is
                destroyed.

                Side effects:
                Cleans up internal memory.

See also:       ::tcl3dAutoscroll::map


Name:           ::tcl3dAutoscroll::map - Mapping callback.

Synopsis:       ::tcl3dAutoscroll::map { w }

Description:    w : string (Widget name)

                Callback executed when automatic scroll bar "w" is mapped.

                Side effects:
                Geometry of scroll bar's top-level window is constrained.

                This procedure keeps the top-level window associated with
                an automatic scroll bar from being resized automatically
                after the scroll bar is mapped. This effect avoids a
                potential endless loop in the case where the resize of the
                top-level window resizes the widget being scrolled,
                causing the scroll bar no longer to be needed.

See also:       ::tcl3dAutoscroll::destroyed


Name:           ::tcl3dAutoscroll::wrap - Autoscroll all new scrollbars.

Synopsis:       ::tcl3dAutoscroll::wrap {}

Description:    Arrange for all new scrollbars to be automatically
                autoscrolled.

                Side effects:
                ::scrollbar is overloaded to automatically autoscroll
                any new scrollbars.

See also:       ::tcl3dAutoscroll::unwrap


Name:           ::tcl3dAutoscroll::unwrap - Turn off automatic
                autoscrolling of new scrollbars.

Synopsis:       ::tcl3dAutoscroll::unwrap {}

Description:    Turn off automatic autoscrolling of new scrollbars.
                Does not effect existing scrollbars.

                Side effects:
                ::scrollbar is returned to its original state
```

```
See also:       ::tcl3dAutoscroll::wrap
```

```
Implementation file: tcl3dCapture.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dCapture.tcl

       Author:         Paul Obermeier

       Description:    Tcl module implementing functions for capturing window
                       contents into an image or file.
                       Note: All of the functionality requires the help of the
                       Img extension. Some of the functionality requires the
                       help of the Twapi extension and is therefore only
                       available on Windows.


       Name:           tcl3dWidget2Img - Copy widget content into photo image.

       Synopsis:       tcl3dWidget2Img { win { ign "" } }

       Description:    win : string (Widget name)
                       ign : string

                       Copy contents of widget "win" and all of its
                       sub-widgets into a photo image.
                       If "ign" is specified and not the empty string, it is
                       interpreted as a pattern for widget names, that should
                       be ignored while traversing the widget hierarchy.
                       The pattern is passed to the "string match" command.

                       Return the photo image identifier.

       See also:       tcl3dWidget2File
                       tcl3dCanvas2Img
                       tcl3dClipboard2Img
                       tcl3dWindow2Img


       Name:           tcl3dWidget2File  - Copy widget content into image file.

       Synopsis:       tcl3dWidget2File { win fileName { ign "" }
                                          { fmt "JPEG" } { opts "" } }

       Description:    win      : string (Widget name)
                       fileName : string
                       ign      : string
                       fmt      : string
                       opts     : string

                       Copy contents of widget "win" and all of its
                       sub-widgets into a photo image and save this image to
                       file "fileName". The file format handler is determined
                       with "fmt". Some formats need optional parameters.
                       These can be supplied in "opts".
                       See the Img documentation (man img) for a list of format
                       handlers and options.
                       If "ign" is specified and not the empty string, it is
                       interpreted as a pattern for widget names, that should
                       be ignored while traversing the widget hierarchy.
                       The pattern is passed to the "string match" command.

       See also:       tcl3dWidget2Img
                       tcl3dCanvas2File
                       tcl3dClipboard2File
                       tcl3dWindow2File
```

```
Name:           tcl3dCanvas2Img - Copy canvas content into photo image.

Synopsis:       tcl3dCanvas2Img { canv }

Description:    canv : string (Widget name)

                Copy the contents of canvas "canv" into a photo image.

                Return the photo image identifier.

See also:       tcl3dCanvas2File
                tcl3dWidget2Img
                tcl3dClipboard2Img
                tcl3dWindow2Img


Name:           tcl3dCanvas2File - Copy canvas content into image file.

Synopsis:       tcl3dCanvas2File { canv fileName { fmt "JPEG" }
                                   { opts "" } }

Description:    canv     : string (Widget name)
                fileName : string
                fmt      : string
                opts     : string

                Copy the contents of canvas "canv" into a photo image
                and save the image to file "fileName". The file format
                handler is determined with "fmt". Some formats need
                optional parameters. These can be supplied in "opts".
                See the Img documentation (man img) for a list of format
                handlers and options.

See also:       tcl3dCanvas2Img
                tcl3dWidget2File
                tcl3dClipboard2File
                tcl3dWindow2File


Name:           tcl3dClipboard2Img - Copy clipboard content into photo
                image.

Synopsis:       tcl3dClipboard2Img {}

Description:    Copy the contents of the Windows clipboard into a photo
                image.

                Return the photo image identifier, if successful.
                Otherwise a Tcl error is thrown.

                Note: This function is currently available only under
                      Windows and needs the Twapi extension.

See also:       tcl3dClipboard2File
                tcl3dWidget2Img
                tcl3dCanvas2Img
                tcl3dWindow2Img


Name:           tcl3dClipboard2File  - Copy clipboard content into file.

Synopsis:       tcl3dClipboard2File { fileName { fmt "JPEG" }
                { opts "" } }

Description:    fileName : string
                fmt      : string
                opts     : string

                Copy the contents of the Windows clipboard into a photo
                image and save the image to file "fileName". The file
                format handler is determined with "fmt". Some formats
                need optional parameters.
                These can be supplied in "opts".
                See the Img documentation (man img) for a list of format
                handlers and options.

                Note: This function is currently available only under
                      Windows and needs the Twapi extension.

See also:       tcl3dClipboard2Img
                tcl3dWidget2File
                tcl3dCanvas2File
                tcl3dWindow2File


Name:           tcl3dImg2Clipboard - Copy photo image into clipboard.

Synopsis:       tcl3dImg2Clipboard { phImg }

Description:    phImg : string (Photo image identifier)

                Copy photo image "phImg" into the Windows clipboard.

                Note: This function is currently available only under
                      Windows and needs the Twapi extension.

See also:       tcl3dClipboard2Img


Name:           tcl3dWindow2Clipboard - Copy window contents into
                clipboard.

Synopsis:       tcl3dWindow2Clipboard {}

Description:    Copy the contents of the top level window (Alt-PrtSc)
                into the Windows clipboard.

                Note: This function is currently available only under
                      Windows and needs the Twapi extension.

See also:       tcl3dClipboard2Img


Name:           tcl3dDesktop2Clipboard - Copy desktop contents into
                clipboard.

Synopsis:       tcl3dDesktop2Clipboard {}

Description:    Copy the contents of the whole desktop (PrtSc) into the
                Windows clipboard.

                Note: This function is currently available only under
                      Windows and needs the Twapi extension.

See also:       tcl3dWindow2Clipboard


Name:           tcl3dWindow2Img - Copy window contents into photo image.

Synopsis:       tcl3dWindow2Img {}

Description:    Copy the contents of the top level window into a photo
                image.
                Return the photo image identifier, if successful.
                Otherwise a Tcl error is thrown.

See also:       tcl3dWindow2File


Name:           tcl3dWindow2File -  Copy window contents into file.

Synopsis:       tcl3dWindow2File { fileName { fmt "JPEG" } { opts "" } }
```

```
Description:    fileName : string
                fmt      : string
                opts     : string

                Copy the contents of the top level window into a photo
                image and save the image to file "fileName". The file
                format handler is determined with "fmt".
                Some formats need optional parameters. These can be
                supplied in "opts".
                See the Img documentation (man img) for a list of format
                handlers and options.

See also:       tcl3dWindow2Img
```

```
Implementation file: tcl3dColors.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dColors.tcl

       Author:         Paul Obermeier, Victor G. Bonilla

       Description:    Tcl module to convert Tcl color names into Tcl3D
                       color lists. Color names may be specified as numeric
                       values or strings.
                       Currently accepted Tcl color names:
                               #RRGGBB
                               All names as listed in the Tcl manual pages,
                               section colors.

                       This module has been inspired by Victor G. Bonilla, who
                       wrote the first version of this file.


       Name:           tcl3dGetColorNames - Get all supported Tcl color names.

       Synopsis:       tcl3dGetColorNames {}

       Description:    Return a list of all supported Tcl color name strings.

       See also:       tcl3dFindColorName


       Name:           tcl3dFindColorName - Validate Tcl color name.

       Synopsis:       tcl3dFindColorName { colorName }

       Description:    colorName : string

                       Check, if supplied color name is a valid string color name.
                       If true, return the supplied color name, otherwise
                       return an empty string.

       See also:       tcl3dGetColorNames


       Name:           tcl3dName2Hex - Convert color name to Tcl hexadecimal.

       Synopsis:       tcl3dName2Hex { colorName }

       Description:    colorName : string

                       Convert Tcl color name "colorName" into the corresponding
                       Tcl hexadecimal representation.
                       Tcl colors are returned as string in the following format:
                       #RRGGBB

       See also:       tcl3dName2rgb
                       tcl3dName2rgbf


       Name:           tcl3dName2Hexa - Convert color name to Tcl hexadecimal.

       Synopsis:       tcl3dName2Hexa { colorName }

       Description:    colorName : string

                       Convert Tcl color name "colorName" into the corresponding
                       Tcl hexadecimal representation.
                       Tcl colors are returned as string in the following format:
```

```
                #RRGGBBAA

See also:       tcl3dName2rgba
                tcl3dName2rgbaf


Name:           tcl3dName2rgb - Convert color name to OpenGL RGB.

Synopsis:       tcl3dName2rgb { colorName }

Description:    colorName : string

                Convert Tcl color name "colorName" into the corresponding
                OpenGL RGB representation.
                OpenGL colors are returned as a list of 3 unsigned
                bytes: { r g b }

See also:       tcl3dName2rgba
                tcl3dName2rgbf


Name:           tcl3dName2rgbf - Convert color name to OpenGL float RGB.

Synopsis:       tcl3dName2rgbf { colorName }

Description:    colorName : string

                Convert Tcl color name "colorName" into the corresponding
                OpenGL float RGB representation.
                OpenGL colors are returned as a list of 3 floats in the
                range [0..1]: { r g b }

See also:       tcl3dName2rgbaf
                tcl3dName2rgb


Name:           tcl3dName2rgba - Convert color name to OpenGL RGBA.

Synopsis:       tcl3dName2rgba { colorName }

Description:    colorName : string

                Convert Tcl color name "colorName" into the corresponding
                OpenGL RGBA representation.
                OpenGL colors are returned as a list of 4 unsigned bytes:
                { r g b a }

See also:       tcl3dName2rgb
                tcl3dName2rgbaf


Name:           tcl3dName2rgbaf - Convert color name to OpenGL float RGBA.

Synopsis:       tcl3dName2rgbaf { colorName }

Description:    colorName : string

                Convert Tcl color name "colorName" into the corresponding
                OpenGL float RGBA representation.
                OpenGL colors are returned as a list of 4 floats in the
                range [0..1]: { r g b a }

See also:       tcl3dName2rgba
                tcl3dName2rgbf


Name:           tcl3dRgb2Name - Convert OpenGL RGB to color name.

Synopsis:       tcl3dRgb2Name { r g b }

Description:    r, g, b : int
```

```
                Convert an OpenGL RGB color representation into a
                hexadecimal Tcl color name string.
                OpenGL colors are specified as unsigned bytes in the
                range [0..255].

See also:       tcl3dName2rgb
                tcl3dRgba2Name


Name:           tcl3dRgba2Name - Convert OpenGL RGBA to color name.

Synopsis:       tcl3dRgba2Name { r g b a }

Description:    r, g, b, a : int

                Convert an OpenGL RGBA color representation into a
                hexadecimal Tcl color name string.
                OpenGL colors are specified as unsigned bytes in the
                range [0..255].

See also:       tcl3dName2rgba
                tcl3dRgb2Name


Name:           tcl3dRgbf2Name - Convert OpenGL RGB to color name.

Synopsis:       tcl3dRgbf2Name { r g b }

Description:    r, g, b : float

                Convert an OpenGL RGB color representation into a
                hexadecimal Tcl color name string.
                OpenGL colors are specified as floats in the range
                [0..1].

See also:       tcl3dName2rgbf
                tcl3dRgbf2Name


Name:           tcl3dRgbaf2Name - Convert OpenGL RGBA to color name.

Synopsis:       tcl3dRgbaf2Name { r g b a }

Description:    r, g, b, a : float

                Convert an OpenGL RGBA color representation into a
                hexadecimal Tcl color name string.
                OpenGL colors are specified as floats in the range
                [0..1].

See also:       tcl3dName2rgbaf
                tcl3dRgbaf2Name
```

```
Implementation file: tcl3dCompat.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dCompat.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with Tcl3D compatibility procedures.


       Name:           tcl3dCompat-0.1 - Tcl3D version 0.1 compatibility.

       Synopsis:       tcl3dCompat-0.1 {}

       Description:    Tcl3d version 0.1 has Vector and Photo commands
                       without the tcl3d prefix. So, if running a script
                       written for newer versions with Tcl3D Version 0.1
                       (Vector command exist, but no tcl3dVector command),
                       all appropriate tcl3d* commands are renamed.

       See also:
```

```
Implementation file: tcl3dConsole.tcl

       Copyright:      2006-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dConsole.tcl

       Author:         Paul Obermeier

       Description:    Tcl module implementing a simple Tk console. It is
                       used in the tcl3dsh, the Tcl3D Starpack.
                       The implementation of this console window was taken
                       from D. Richard Hipp's mktclapp.


       Name:           tcl3dConsoleCreate - Create a console window.

       Synopsis:       tcl3dConsoleCreate { w prompt title }

       Description:    w      : string (Widget name)
                       prompt : string
                       title  : string

                       Create a new console window "w". The window's title will
                       be set to "title". The prompt inside the console's text
                       widget will be set to "prompt".

                       Example:
                       tcl3dConsoleCreate .myConsole "tcl3d> " "Tcl3D Console"

       See also:
```

```
Implementation file: tcl3dFile.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dFile.tcl

       Author:         Paul Obermeier

       Description:    Tcl module with file handling Tcl3D utility procedures.


       Name:           tcl3dGetTmpDir - Get the name of a temporary directory.

       Synopsis:       tcl3dGetTmpDir {}

       Description:    Return the location of a temporary directory name.
                       The function checks the following environment variables
                       first: TMP, TEMP, TMPDIR.
                       If none of these variables are defined, standard places
                       are checked (C:/Windows/Temp, /tmp).

       See also:       tcl3dCreateTmpDir


       Name:           tcl3dCreateTmpDir - Create a unique temporary directory.

       Synopsis:       tcl3dCreateTmpDir {}

       Description:    Create a unique temporary directory.
                       Return the full path name of the created directory.
                       The new directory is created in the standard temporary
                       directory as returned by tcl3dGetTmpDir.

       See also:       tcl3dGetTmpDir


       Name:           tcl3dGenExtName - Create a name on the file system.

       Synopsis:       tcl3dGenExtName { fileName }

       Description:    fileName : string

                       Return a valid path name on the file system generated
                       from the file name specified in "fileName".
                       Use this function, if writing to a file from a script,
                       which may be running from within a Starpack.
                       If the script is not executed from a Starpack, the
                       function returns the supplied parameter unchanged.

       See also:       tcl3dGetExtFile


       Name:           tcl3dGetExtFile - Get a name on the file system.

       Synopsis:       tcl3dGetExtFile { fileName }

       Description:    fileName : string

                       Return a valid path name on the file system generated
                       from the file name specified in "fileName".
                       Use this function, if a file is needed for reading
                       from an external Tcl3D library, like font files used by
                       FTGL, and the script may be executed from within a
                       virtual file system (ex. starkit vfs or zvfs).
                       The file included in the virtual file system is
                       transparently copied onto the file system and that path
```

```
                 name is returned.
                 The path name is built using a system-wide temporary
                 directory as returned by tcl3dGetTmpDir.
                 If the script is not executed from within a virtual
                 file system, the function returns the supplied parameter
                 unchanged.

See also:        tcl3dGenExtName
                 tcl3dGetTmpDir
```

```
Implementation file: tcl3dInfo.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dInfo.tcl

       Author:         Paul Obermeier

       Description:    Tcl module to get Tcl3D related information:
                       Version, extensions, state variables.


       Name:           tcl3dHavePackage - Check availability of a specific
                       Tcl3D package.

       Synopsis:       tcl3dHavePackage { pkgName version }

       Description:    pkgName : string
                       version : string

                       Return 1, if Tcl package "pkgName" is available in at
                       least version "version". Otherwise return 0.

                       Example: tcl3dHavePackage tcl3dcg 0.3.2
                                checks availability of the tcl3dCg package in
                                at least version 0.3.2.

       See also:       tcl3d{SubPackage}GetVersion


       Name:           tcl3dGetLibraryInfo - Get library version of a Tcl3D
                       module.

       Synopsis:       tcl3dGetLibraryInfo { pkgName }

       Description:    pkgName : string

                       Return the library version corresponding to supplied
                       Tcl3D package name "pkgName" as a string. If no version
                       information is available, an empty string is returned.

       See also:       tcl3dGetPackageInfo


       Name:           tcl3dGetPackageInfo - Get Tcl3D package information.

       Synopsis:       tcl3dGetPackageInfo {}

       Description:    Return a list of sub-lists containing Tcl3D package
                       information. Each sub-list contains the name of the
                       sub-package, the availability flag (0 or 1), the
                       sub-package version as well as the version of the
                       wrapped library.

                       Example:
                       {tcl3dcg 1 0.3.3 1.5.0015}     {tcl3ddemoutil 1 0.3.3 {}}
                       {tcl3dftgl 1 0.3.3 2.1.2}      {tcl3dgauges 1 0.3.3 {}}
                       {tcl3dgl2ps 1 0.3.3 1.3.2}     {tcl3dode 1 0.3.3 0.7.0}
                       {tcl3dogl 1 0.3.3 {APPLE-1.4}} {tcl3dsdl 1 0.3.3 1.2.9}
                       {tcl3dtogl 1 0.3.3 {}}         {tcl3dutil 1 0.3.3 {}}

                       Note: A Togl window (and therefore a graphics context)
                             must have been created before issuing a call to
                             this function.

       See also:       tcl3dShowPackageInfo
```

```
Name:           tcl3dShowPackageInfo - Display package information.

Synopsis:       tcl3dShowPackageInfo {}

Description:    Display the version info returned by
                tcl3dGetPackageInfo in a toplevel window.

                Note: A Togl window (and therefore a graphics context)
                      must have been created before issuing a call to
                      this function.

See also:       tcl3dGetPackageInfo


Name:           tcl3dHaveCg - Check availability of tcl3dCg module.

Synopsis:       tcl3dHaveCg {}

Description:    Return 1, if the Cg library has been loaded successfully
                via the tcl3dCg module. Otherwise return 0.

                Note: This function is only available when loading Tcl3D
                      via a "package require tcl3d".

See also:       tcl3dGetPackageInfo
                tcl3dCgGetVersion


Name:           tcl3dHaveSDL - Check availability of tcl3dSDL module.

Synopsis:       tcl3dHaveSDL {}

Description:    Return 1, if the SDL library has been loaded successfully
                via the tcl3dSDL module. Otherwise return 0.

                Note: This function is only available when loading Tcl3D
                      via a "package require tcl3d".

See also:       tcl3dGetPackageInfo
                tcl3dSDLGetVersion


Name:           tcl3dHaveFTGL - Check availability of tcl3dFTGL module.

Synopsis:       tcl3dHaveFTGL {}

Description:    Return 1, if the FTGL library has been loaded successfully
                via the tcl3dFTGL module. Otherwise return 0.

                Note: This function is only available when loading Tcl3D
                      via a "package require tcl3d".

See also:       tcl3dGetPackageInfo
                tcl3dFTGLGetVersion


Name:           tcl3dHaveOde - Check availability of tcl3dOde module.

Synopsis:       tcl3dHaveOde {}

Description:    Return 1, if the ODE library has been loaded successfully
                via the tcl3dOde module. Otherwise return 0.

                Note: This function is only available when loading Tcl3D
                      via a "package require tcl3d".

See also:       tcl3dGetPackageInfo
                tcl3dOdeGetVersion
```

```
Name:           tcl3dHaveOsg - Check availability of tcl3dOsg module.

Synopsis:       tcl3dHaveOsg {}

Description:    Return 1, if the OSG library has been loaded successfully
                via the tcl3dOsg module. Otherwise return 1.

                Note: This function is only available when loading Tcl3D
                      via a "package require tcl3d".

See also:       tcl3dGetPackageInfo
                tcl3dOsgGetVersion


Name:           tcl3dHaveGl2ps - Check availability of tcl3dGl2ps module.

Synopsis:       tcl3dHaveGl2ps {}

Description:    Return 1, if the GL2PS library has been loaded successfully
                via the tcl3dGl2ps module. Otherwise return 0.

                Note: This function is only available when loading Tcl3D
                      via a "package require tcl3d".

See also:       tcl3dGetPackageInfo
                tcl3dGl2psGetVersion
```

```
Implementation file: tcl3dLogo.tcl

       Copyright:       2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                        See the file "Tcl3D_License.txt" for information on
                        usage and redistribution of this file, and for a
                        DISCLAIMER OF ALL WARRANTIES.

       Module:          Tcl3D -> tcl3dUtil
       FileName:        tcl3dLogo.tcl

       Author:          Paul Obermeier

       Description:     Tcl module showing the Tcl/Tk and poSoft logo.


       Name:            tcl3dLogoDestroyPoSoft - Destroy poSoft logo window.

       Synopsis:        tcl3dLogoDestroyPoSoft {}

       Description:     Destroy a previously opened poSoft logo window.

       See also:        tcl3dLogoShowPoSoft
                        tcl3dLogoShowTcl


       Name:            tcl3dLogoShowPoSoft - Display poSoft logo.

       Synopsis:        tcl3dLogoShowPoSoft { version copyright withdrawWin }

       Description:     version     : string
                        copyright   : string
                        withdrawWin : string (Widget name)

                        Display the poSoft logo in two possible ways:
                        If "withdrawWin" is set to the empty string, the logo
                        is shown in a window with decoration. This may be used
                        for displaying the logo as an action for an "About"
                        menu entry.
                        If "withdrawWin" is set to an existing window name
                        (typically the name of the main application window), the
                        logo window is shown without decoration as a splash
                        window, which automatically disappears after a second.
                        The logo window has two label widgets to display
                        additional text messages, which are specified in
                        "version" and "copyright".

       See also:        tcl3dLogoDestroyPoSoft
                        tcl3dLogoShowTcl


       Name:            tcl3dLogoDestroyTcl - Destroy Tcl logo window.

       Synopsis:        tcl3dLogoDestroyTcl { w img }

       Description:     Destroy a previously opened Tcl logo window.

       See also:        tcl3dLogoShowTcl
                        tcl3dLogoShowPoSoft


       Name:            tcl3dLogoShowTcl - Display Tcl logo.

       Synopsis:        tcl3dLogoShowTcl { args }

       Description:     args : variable parameter list

                        Display the Tcl logo with additional text messages in
                        a window with decoration. This may be used
                        for displaying the logo as an action for an "About"
                        menu entry.
```

```
                  "args" may contain any combination of the following
                  package names:
                  Tk Img Tktable combobox mysqltcl tcom

See also:         tcl3dLogoShowPoSoft
```

```
Implementation file: tcl3dToolhelp.tcl

       Copyright:       2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                        See the file "Tcl3D_License.txt" for information on
                        usage and redistribution of this file, and for a
                        DISCLAIMER OF ALL WARRANTIES.

       Module:          Tcl3D -> tcl3dUtil
       FileName:        tcl3dToolhelp.tcl

       Author:          Paul Obermeier

       Description:     Tcl module implementing a simple tool help widget.


       Name:            tcl3dToolhelpInit - Initialize toolhelp module.

       Synopsis:        tcl3dToolhelpInit { w { bgColor yellow }
                        { fgColor black } }

       Description:     w         : string (Widget name)
                        bgColor   : string
                        fgColor   : string

                        Initialize the toolhelp module.
                        The initialization function only needs to be called
                        when non-standard background and foreground colors
                        are needed.

       See also:        tcl3dToolhelpAddBinding


       Name:            tcl3dToolhelpShow - Display toolhelp message.

       Synopsis:        tcl3dToolhelpShow { x y str }

       Description:     x   : int
                        y   : int
                        str : string

                        Display the toolhelp window at widget relative
                        coordinates (x, y) with message string "str".

                        A typical usage is like follows:
                        bind $w <Enter> "tcl3dToolhelpShow %X %Y [list $str]"

       See also:        tcl3dToolhelpHide
                        tcl3dToolhelpAddBinding


       Name:            tcl3dToolhelpHide - Hide toolhelp message.

       Synopsis:        tcl3dToolhelpHide {}

       Description:     Hide the toolhelp message window.

       See also:        tcl3dToolhelpShow
                        tcl3dToolhelpAddBinding


       Name:            tcl3dToolhelpAddBinding - Add binding for a toolhelp
                        message.

       Synopsis:        tcl3dToolhelpAddBinding { w str }

       Description:     w   : string (Widget name)
                        str : string

                        Add bindings to widget "w" to display message string
                        "str" in a toolhelp window near the widget. The
```

```
                  toolhelp window is shown, when the mouse enters the
                  widget and unmapped, when the mouse leaves the widget.

See also:         tcl3dToolhelpInit
```

```
Implementation file: tcl3dTrackball.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dTrackball.tcl

       Author:         Paul Obermeier

       Description:    Simple trackball-like motion adapted (ripped off)
                       from projtex.c (written by David Yu and David Blythe).
                       See the SIGGRAPH '96 Advanced OpenGL course notes.

                       Usage overview:

                       Call tcl3dTbInit before any other trackball call.
                       Call tcl3dTbReshape from the reshape callback.
                       Call tcl3dTbMatrix to get the trackball matrix rotation.
                       Call tcl3dTbStartMotion to begin trackball movement.
                       Call tcl3dTbStopMotion to stop trackball movement.
                       Call tcl3dTbMotion from the motion callback.
                       Call tcl3dTbAnimate(1) if you want the trackball to
                            continue spinning after the mouse button has been
                            released.
                       Call tcl3dTbAnimate(0) if you want the trackball to stop
                            spinning after the mouse button has been released.

                       See ftglDemo.tcl for a real world example.

                       Modified for Tcl3D by Paul Obermeier 2006/02/02
                       See www.tcl3d.org for the Tcl3D extension.


       Name:           tcl3dTbStartMotion - Begin trackball movement

       Synopsis:       tcl3dTbStartMotion { toglwin x y }

       Description:    toglwin : string
                       x       : int
                       y       : int

                       Begin movement of the trackball attached to Togl window
                       "toglwin".
                       "x" and "y" give the actual mouse position inside the
                       Togl window.
                       This procedure is typically bound to a button press
                       event.
                       Example: bind .toglwin <ButtonPress-1>
                                "tcl3dTbStartMotion .toglwin %x %y"

       See also:       tcl3dTbStopMotion
                       tcl3dTbMotion


       Name:           tcl3dTbStopMotion - Stop trackball movement

       Synopsis:       tcl3dTbStopMotion { toglwin }

       Description:    toglwin : string

                       Stop movement of the trackball attached to Togl window
                       "toglwin".
                       This procedure is typically bound to a button release
                       event.
                       Example: bind .toglwin <ButtonRelease-1>
                                "tcl3dTbStopMotion .toglwin"
```

```
See also:       tcl3dTbStartMotion
                tcl3dTbMotion


Name:           tcl3dTbAnimate - Set the trackball animation mode.

Synopsis:       tcl3dTbAnimate { toglwin animate }

Description:    toglwin : string
                animate : bool

                Set the animation mode of the trackball attached to
                Togl window "toglwin".
                If the trackball shall continue spinning after the
                mouse button has been released, set "animate" to true.
                Set "animate" to false, if the trackball should stop
                spinning after the mouse button has been released.

See also:       tcl3dTbStartMotion


Name:           tcl3dTbInit - Initialize the trackball module.

Synopsis:       tcl3dTbInit { toglwin }

Description:    toglwin : string

                Initialize the trackball attached to Togl window
                "toglwin".
                This procedure must be called before any other trackball
                procedures, for example in the Togl create callback.

See also:


Name:           tcl3dTbMatrix - Use the trackball matrix rotation

Synopsis:       tcl3dTbMatrix { toglwin }

Description:    toglwin : string

                Use the rotation matrix of the trackball attached to
                Togl window "toglwin".
                The rotation matrix is applied to the top most OpenGL
                matrix with glMultMatrixf.
                This procedure is typically called in the Togl display
                callback.

See also:


Name:           tcl3dTbReshape - Notify trackball about a reshape.

Synopsis:       tcl3dTbReshape { toglwin w h }

Description:    toglwin : string
                w       : int
                h       : int

                Notify the trackball attached to Togl window "toglwin"
                that the size of the window has been changed to
                width "w" and height "h".
                This procedure is typically called in the Togl reshape
                callback.

See also:       tcl3dTbInit


Name:           tcl3dTbMotion - Move the trackball.

Synopsis:       tcl3dTbMotion { toglwin x y }
```

```
Description:    toglwin : string
                x       : int
                y       : int

                Move the trackball attached to Togl window "toglwin".
                "x" and "y" give the actual mouse position inside the
                Togl window.
                This procedure is typically bound to a mouse motion
                event.
                Example: bind .toglwin <B1-Motion>
                         "tcl3dTbMotion .toglwin %x %y"

See also:       tcl3dTbStartMotion
                tcl3dStopMotion
```

```
Implementation file: tcl3dVecMath.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dVecMath.tcl

       Author:         Paul Obermeier

       Description:    Tcl module to handle vectors and transformation
                       matrices.


       Name:           tcl3dVec3fPrint - Print contents of a 3D vector.

       Synopsis:       tcl3dVec3fPrint { vec }

       Description:    vec : string (Tcl3D Vector Identifier)

                       Print the contents of 3D Vector "vec" onto
                       standard output. "vec" is a Tcl3D Vector of size 3
                       and type float or double.

       See also:       tcl3dMatfPrint


       Name:           tcl3dMatfPrint - Print contents of a transformation
                       matrix.

       Synopsis:       tcl3dMatfPrint { mat }

       Description:    mat : string (Tcl3D Vector Identifier)

                       Print the contents of transformation matrix "mat" onto
                       standard output. "mat" is a Tcl3D Vector of size 16
                       and type float or double.

       See also:       tcl3dVec3fPrint


       Name:           tcl3dRadToDeg  - Convert angle from radians to degrees.

       Synopsis:       tcl3dRadToDeg { ang }

       Description:    ang : double

                       Return angle "ang" specified in radians in degrees.

       See also:       tcl3dDegToRad


       Name:           tcl3dDegToRad  - Convert angle from degrees to radians.

       Synopsis:       tcl3dDegToRad { ang }

       Description:    ang : double

                       Return angle "ang" specified in degress in radians.

       See also:       tcl3dRadToDeg
```

```
Implementation file: tcl3dVector.tcl

       Copyright:      2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                       See the file "Tcl3D_License.txt" for information on
                       usage and redistribution of this file, and for a
                       DISCLAIMER OF ALL WARRANTIES.

       Module:         Tcl3D -> tcl3dUtil
       FileName:       tcl3dVector.tcl

       Author:         Paul Obermeier

       Description:    Tcl module to handle tcl3dVectors, i.e. contiguous
                       pieces of memory.


       Name:           tcl3dVector - Create a new Tcl3D Vector

       Synopsis:       tcl3dVector { type size }

       Description:    type : string
                       size : int

                       Create a new Tcl3D Vector of size "size" by calling
                       the memory allocation routine new_"type" and create
                       a new Tcl procedure.
                       The contents of the new Tcl3D Vector are uninitialized.
                       Return the identifier (i.e. the name of the created Tcl
                       procedure) of the new Tcl3D Vector.

                       The following base types are currently supported:
                       GLbitfield      GLboolean       GLbyte    GLclampd
                       GLclampf        GLdouble        GLenum    GLfloat
                       GLint           GLshort         GLsizei   GLubyte
                       GLuint          GLushort        double    float
                       int             short           uint      ushort

                       Note: To get an up-to-date list of wrapped types, issue
                             the command "info commands new_*" after loading
                             Tcl3D or use the script "vectorTypes.tcl" in
                             directory "tcl3dUtil/test".

                             A detailled description of Tcl3D Vectors can be
                             found in the Tcl3D manual.

       See also:       tcl3dVectorFromArgs
                       tcl3dVectorFromByteArray
                       tcl3dVectorFromList
                       tcl3dVectorFromPhoto
                       tcl3dVectorFromString


       Name:           tcl3dVectorInd - Get index of a Tcl3D Vector.

       Synopsis:       tcl3dVectorInd { vec type ind }

       Description:    vec  : string (Tcl3D Vector Identifier)
                       type : string
                       ind  : int

                       Return the "pointer" to the "ind" element of a Tcl3D
                       Vector. The base Tcl3D Vector is specified with "vec",
                       the type of the Vector is given with "type".

                       Note: See the description of tcl3dVector for a list
                             of usable types.
                             This function may be used in conjunction with
                             OpenGL interleaved vertex arrays. See RedBook
                             demo "aapolyStride.tcl" for an example usage.
```

```
See also:       tcl3dVector


Name:           tcl3dVectorPrint - Print contents of a Tcl3D Vector.

Synopsis:       tcl3dVectorPrint { vec num }

Description:    vec : string (Tcl3D Vector Identifier)
                num : num

                Print the first "num" elements of Tcl3D Vector "vec"
                onto standard output.

                Note: Tcl3D Vectors behave like C vectors, i.e. they
                      do not have information about its length.

See also:       tcl3dVector


Name:           tcl3dVectorFromArgs - Create new Tcl3D Vector from
                an argument list.

Synopsis:       tcl3dVectorFromArgs { type args }

Description:    type  : string
                args  : list

                Create a new Tcl3D Vector of type "type" from given
                variable argument list.
                Return the identifier (i.e. the name of the created Tcl
                procedure) of the new Tcl3D Vector.

                Note: See the description of tcl3dVector for a list
                      of usable types.

See also:       tcl3dVector
                tcl3dVectorFromByteArray
                tcl3dVectorFromList
                tcl3dVectorFromPhoto
                tcl3dVectorFromString


Name:           tcl3dVectorFromList - Create new Tcl3D Vector from a
                list.

Synopsis:       tcl3dVectorFromList { type l { maxElems -1 } }

Description:    type     : string
                l        : list
                maxElems : int

                Create a new Tcl3D Vector of type "type" from given
                Tcl list "l". If "maxElems" is given and greater than
                zero, only the first "maxElems" are used.
                Return the identifier (i.e. the name of the created Tcl
                procedure) of the new Tcl3D Vector.

                Note: See the description of tcl3dVector for a list
                      of usable types.

See also:       tcl3dVector
                tcl3dVectorFromArgs
                tcl3dVectorFromByteArray
                tcl3dVectorFromPhoto
                tcl3dVectorFromString


Name:           tcl3dCharToNum - Convert character to integer.

Synopsis:       tcl3dCharToNum { char }

Description:    char : character
```

```
                Convert an ASCII character into the corresponding
                numeric value.

See also:       tcl3dNumToChar


Name:           tcl3dNumToChar - Convert integer to character.

Synopsis:       tcl3dNumToChar { num }

Description:    num : int

                Convert a numeric value into the corresponding ASCII
                character.

See also:       tcl3dCharToNum


Name:           tcl3dVectorFromString - Create new Tcl3D Vector from a
                string.

Synopsis:       tcl3dVectorFromString { type str }

Description:    type : string
                str  : string

                Create a new Tcl3D Vector of type "type" from given
                string "str".
                Return the identifier (i.e. the name of the created Tcl
                procedure) of the new Tcl3D Vector.

                Note: This version is very slow and is intended only
                      for converting the characters of short text
                      strings into it's numerical values to be used by
                      display lists rendering raster fonts.
                      See the description of tcl3dVector for a list
                      of usable types.

See also:       tcl3dVector
                tcl3dVectorFromArgs
                tcl3dVectorFromByteArray
                tcl3dVectorFromList
                tcl3dVectorFromPhoto


Name:           tcl3dVectorFromByteArray - Create new Tcl3D Vector from
                binary string.

Synopsis:       tcl3dVectorFromByteArray { type str }

Description:    type : string
                str  : string

                Create a new Tcl3D Vector of type "type" from given
                binary string "str".
                Return the identifier (i.e. the name of the created Tcl
                procedure) of the new Tcl3D Vector.

                Note: See the description of tcl3dVector for a list
                      of usable types.

See also:       tcl3dVector
                tcl3dVectorFromArgs
                tcl3dVectorFromList
                tcl3dVectorFromPhoto
                tcl3dVectorFromString


Name:           tcl3dVectorFromPhoto - Create new Tcl3D Vector from
                a Tk photo.
```

```
Synopsis:       tcl3dVectorFromPhoto { phImg { numChans -1 }
                { scl 1.0 } { off 0.0 }  }

Description:    phImg    : string (Photo image identifier)
                numChans : int
                scl      : double
                off      : double

                Create a new Tcl3D Vector containing the image data
                of Tk photo "phImg". The created Tcl3D Vector is of type
                GL_UNSIGNED_BYTE. If "numChans" is specified and between
                1 and 4, only the first "numChans" are copied into
                the Tcl3D Vector. Otherwise all channels available in
                the photo image are used.
                "scl" and "off" can be used to scale and offset the
                pixel values while converting.
                Return the identifier (i.e. the name of the created Tcl
                procedure) of the new Tcl3D Vector.

See also:       tcl3dVector
                tcl3dVectorFromArgs
                tcl3dVectorFromByteArray
                tcl3dVectorFromList
                tcl3dVectorFromString


Name:           tcl3dVectorToList - Copy Tcl3D Vector into a list.

Synopsis:       tcl3dVectorToList { vec num }

Description:    vec : string (Tcl3D Vector Identifier)
                num : int

                Copy "num" elements of Tcl3D Vector "vec" into a Tcl
                list and return that list.

See also:       tcl3dVectorFromList
                tcl3dVectorToByteArray
                tcl3dVectorToString


Name:           tcl3dVectorToString - Copy Tcl3D Vector into a string.

Synopsis:       tcl3dVectorToString { vec }

Description:    vec : string (Tcl3D Vector Identifier)

                Interpret the elements of Tcl3D Vector "vec"
                (which must be of type GLubyte) as a
                null-terminated string and return that string.

                Note: This version is slow and is intended only for
                      short text strings. Use this function for example
                      to convert the information returned by a GLSL
                      shader.

See also:       tcl3dVectorFromString
                tcl3dVectorToByteArray
                tcl3dVectorToList


Name:           tcl3dVectorToByteArray - Copy Tcl3D Vector into a binary
                string.

Synopsis:       tcl3dVectorToByteArray { vec numBytes {srcOff 0}
                {destOff 0} } {

Description:    vec      : string (Tcl3D Vector Identifier)
                numBytes : int
                srcOff   : int
                destOff  : int
```

```
                    Copy "numBytes" elements of Tcl3D Vector "vec" into
                    a Tcl binary string and return that string. The
                    Tcl3D Vector has be of type GLubyte.
                    "srcOff" and "destOff" may be used optionally to specify
                    an offset into the source and the destination.

    See also:       tcl3dVectorFromByteArray
                    tcl3dVectorToList
                    tcl3dVectorToString
```

```
Implementation file: tcl3dWidgets.tcl

       Copyright:       2005-2008 Paul Obermeier (obermeier@tcl3d.org)

                        See the file "Tcl3D_License.txt" for information on
                        usage and redistribution of this file, and for a
                        DISCLAIMER OF ALL WARRANTIES.

       Module:          Tcl3D -> tcl3dUtil
       FileName:        tcl3dWidgets.tcl

       Author:          Paul Obermeier

       Description:     Tcl module implementing some simple Tk widgets like
                        scrolled listboxes and text widgets, as well as
                        some widget and window handling utilities.


       Name:            tcl3dHaveAqua - Check, if windowing system is Aqua.

       Synopsis:        tcl3dHaveAqua {}

       Description:     Return true, if the windowing system is Apple's Aqua.
                        Otherwise return false.

       See also:        tcl3dShowIndicator


       Name:            tcl3dShowIndicator - Check, if button indicators
                        should be shown.

       Synopsis:        tcl3dShowIndicator {}

       Description:     Return true, if we want to show the indicators for
                        radio- and checkbuttons. Currently we do this on a Mac
                        running Aqua, because it looks very buggy otherwise.

       See also:        tcl3dHaveAqua


       Name:            tcl3dAddEvents - Add virtual events.

       Synopsis:        tcl3dAddEvents {}

       Description:     Add the following virtual events for cross-platform
                        mouse event handling:
                        <<LeftMousePress>>
                        <<MiddleMousePress>>
                        <<RightMousePress>>

       See also:


       Name:            tcl3dWinIsTop - Check, if widget is a top level window.

       Synopsis:        tcl3dWinIsTop { wid }

       Description:     wid : string

                        Return true, if widget "wid" is a top level window.

       See also:        tcl3dWinRaise


       Name:            tcl3dWinRaise - Raise a widget.

       Synopsis:        tcl3dWinRaise { wid }

       Description:     wid : string

                        Raise widget "wid" to the top of the widget layout
```

```
                 hierarchy.

See also:        tcl3dWinIsTop


Name:            tcl3dSetFullScreenMode - Put a widget into fullscreen
                 mode.

Synopsis:        tcl3dSetFullScreenMode { wid }

Description:     wid : string

                 Put widget "wid" into fullscreen mode:
                 It's size is adjusted to fit the entire screen, window
                 decoration is removed and the widget can not be resized.

See also:        tcl3SetWindowMode


Name:            tcl3dSetWindowMode - Put a widget into windowing
                 mode.

Synopsis:        tcl3dSetWindowMode { wid w h }

Description:     wid : string
                 w   : int
                 h   : int

                 Put widget "wid" into windowing mode:
                 It's size is adjusted to width "w" and height "h",
                 window decoration is enabled and the widget can be
                 resized.

See also:        tcl3dSetFullScreenMode


Name:            tcl3dSetScrolledTitle - Set the title of a
                 scrolled widget.

Synopsis:        tcl3dSetScrolledTitle { wid titleStr
                                       { fgColor "black" } }

Description:     wid      : string
                 titleStr : string
                 fgColor  : string

                 Set the title of scrolled widget "wid" to string
                 "titleStr". The text color can be optionally specified
                 with "fgColor". "fgColor" must be a valid Tk color name.
                 "wid" must be a widget name returned from
                 tcl3dCreateScrolledWidget or descendants.

See also:        tcl3dCreateScrolledWidget


Name:            tcl3dCreateScrolledWidget - Create a scrolled widget.

Synopsis:        tcl3dCreateScrolledWidget { wType wid titleStr args }

Description:     wType     : string
                 wid       : string
                 titleStr  : string
                 args      : list

                 Create a compound widget with horizontal and vertical
                 scrollbars. The type of the widget is given with "wType"
                 and must be a valid Tk widget like canvas or text.
                 "wid" is the parent frame of the compound widget and
                 must already exist.
                 The compound widget may have a title string, which is
                 given with "titleStr". If "titleStr" is an empty string,
                 no title label will be generated.
```

```
                With optional parameter "args" additional widget
                specific parameters may be supplied.
                Return the identifier to the created master widget.

                There exist several utility procedures for often used
                Tk widget types. See list below.

See also:       tcl3dSetScrolledWidgetTitle
                tcl3dCreateScrolledFrame
                tcl3dCreateScrolledListbox
                tcl3dCreateScrolledText
                tcl3dCreateScrolledCanvas
                tcl3dCreateScrolledTable
                tcl3dCreateScrolledTableList


Name:           tcl3dCreateScrolledFrame - Create a scrolled
                frame widget.

Synopsis:       tcl3dCreateScrolledFrame { wid titleStr args }

Description:    wid         : string
                titleStr    : string
                args        : list

                Create a scrolled frame widget. "wid" specifies the
                parent frame of the created scrolled widget. "titleStr"
                specifies the string displayed as widget title.
                With optional parameter "args" additional widget
                specific parameters may be supplied.
                Return the identifier to the created frame widget.

See also:       tcl3dCreateScrolledWidget
                tcl3dSetScrolledWidgetTitle


Name:           tcl3dCreateScrolledListbox - Create a scrolled
                listbox widget.

Synopsis:       tcl3dCreateScrolledListbox { wid titleStr args }

Description:    wid         : string
                titleStr    : string
                args        : list

                Create a scrolled listbox widget. "wid" specifies the
                parent frame of the created scrolled widget. "titleStr"
                specifies the string displayed as widget title.
                With optional parameter "args" additional widget
                specific parameters may be supplied.
                Return the identifier to the created listbox widget.

See also:       tcl3dCreateScrolledWidget
                tcl3dSetScrolledWidgetTitle


Name:           tcl3dCreateScrolledText - Create a scrolled
                text widget.

Synopsis:       tcl3dCreateScrolledText { wid titleStr args }

Description:    wid         : string
                titleStr    : string
                args        : list

                Create a scrolled text widget. "wid" specifies the
                parent frame of the created scrolled widget. "titleStr"
                specifies the string displayed as widget title.
                With optional parameter "args" additional widget
                specific parameters may be supplied.
                Return the identifier to the created text widget.
```

```
See also:       tcl3dCreateScrolledWidget
                tcl3dSetScrolledWidgetTitle


Name:           tcl3dCreateScrolledCanvas - Create a scrolled
                canvas widget.

Synopsis:       tcl3dCreateScrolledCanvas { wid titleStr args }

Description:    wid        : string
                titleStr   : string
                args       : list

                Create a scrolled canvas widget. "wid" specifies the
                parent frame of the created scrolled widget. "titleStr"
                specifies the string displayed as widget title.
                With optional parameter "args" additional widget
                specific parameters may be supplied.
                Return the identifier to the created canvas widget.

See also:       tcl3dCreateScrolledWidget
                tcl3dSetScrolledWidgetTitle


Name:           tcl3dCreateScrolledTable - Create a scrolled
                tktable widget.

Synopsis:       tcl3dCreateScrolledTable { wid titleStr args }

Description:    wid        : string
                titleStr   : string
                args       : list

                Create a scrolled TkTable widget. "wid" specifies the
                parent frame of the created scrolled widget. "titleStr"
                specifies the string displayed as widget title.
                With optional parameter "args" additional widget
                specific parameters may be supplied.
                Return the identifier to the created TkTable widget.

See also:       tcl3dCreateScrolledWidget
                tcl3dSetScrolledWidgetTitle


Name:           tcl3dCreateScrolledTablelist - Create a scrolled
                tablelist widget.

Synopsis:       tcl3dCreateScrolledTablelist { wid titleStr args }
Description:    wid        : string
                titleStr   : string
                args       : list

                Create a scrolled tablelist widget. "wid" specifies the
                parent frame of the created scrolled widget. "titleStr"
                specifies the string displayed as widget title.
                With optional parameter "args" additional widget
                specific parameters may be supplied.
                Return the identifier to the created tablelist widget.

                Note: A "package require tablelist" must be issued before
                      using this function.

See also:       tcl3dCreateScrolledWidget
                tcl3dSetScrolledWidgetTitle
```