Subject: W3d import error
Posted by Gen_Blacky on Sun, 21 Oct 2007 03:43:33 GMT
View Forum Message <> Reply to Message

i got this error importing nod stealth tank in renx

v_nod_stlth.w3d

Using coolfires w3d importer

```
   in coordsys gmPivots[p] setVert gmMeshes[i] v meshes[i].verts[v]
    )
    -- to show influenced info at v1.08
    format "%: \n" meshes[i].header.meshName
    for p = 1 to hier.pivots.count do
    (
     ivs = #()
     for v = 1 to meshes[i].vertInfs.count do
     (
      if meshes[i].vertInfs[v] == (p - 1) do (append ivs v)
     )
     if ivs.count > 1 do (format "% %: %\n" p hier.pivots[p].pivotName ivs)
    )
   )
  )
  if (anim != undefined) and (hier != undefined) then
  (
   if anim.header.frameCount > 1 then
    animationRange = interval 1 anim.header.frameCount -- edited at v1.07
   else
    animationRange = interval 0 1
   frameRate = anim.header.frameRate
   for i = 1 to anim.channels.count do
   (
    curChn = anim.channels[i]
    --curObj = execute ("$'" + hier.pivots[curChn.pivotID + 1].pivotName + "'") -- at v1.02. it should be curChn.pivotID + 1 at v1.04
    curObj = gmPivots[curChn.pivotID] -- test at v1.04
    --print curObj
    --print curChn
    if curObj != undefined then
    (
     datumPos = hier.pivots[curChn.pivotID + 1].pos
     datumRot = hier.pivots[curChn.pivotID + 1].rotation
     case curChn.flags of
```

```
(
 0x0000: --ANIM_CHANNEL_X = 0
 (
  curObj.pos.controller = linear_position()
  curKey = addNewKey curObj.pos.controller 0
  curKey.value = datumPos
  for f = curChn.firstFrame to curChn.lastFrame do
  (
   k = getKeyIndex curObj.pos.controller (f + 1)
   if k == 0 then
   (
    curKey = addNewKey curObj.pos.controller (f + 1)
    curKey.value = datumPos
   )
   else
   (
    curKey = curObj.pos.controller.keys[k]
   )
   curKey.value += [curChn.values[(f - curChn.firstFrame + 1)], 0, 0] * (inverse datumRot)
  )
  --if curChn.firstFrame == 0 then
  --(
  -- curKey = addNewKey curObj.pos.controller (curChn.lastFrame + 2)
  --)
  --else
  --(
  -- curKey = addNewKey curObj.pos.controller (curChn.firstFrame)
  --)
  --curKey.value = datumPos
 )
 0x0001: --ANIM_CHANNEL_Y = 1
 (
  curObj.pos.controller = linear_position()
  curKey = addNewKey curObj.pos.controller 0
  curKey.value = datumPos
  for f = curChn.firstFrame to curChn.lastFrame do
  (
   k = getKeyIndex curObj.pos.controller (f + 1)
   if k == 0 then
   (
    curKey = addNewKey curObj.pos.controller (f + 1)
    curKey.value = datumPos
   )
   else
   (
    curKey = curObj.pos.controller.keys[k]
   )
   curKey.value += [0, curChn.values[(f - curChn.firstFrame + 1)], 0] * (inverse datumRot)
  )
  --if curChn.firstFrame == 0 then
  --(
  -- k = getKeyIndex curObj.pos.controller (curChn.lastFrame + 1)
  -- if k == 0 then
  -- (
  --  curKey = addNewKey curObj.pos.controller (curChn.lastFrame + 1)
  --  curKey.value = datumPos
  -- )
  --)
  --else
  --(
  -- k = getKeyIndex curObj.pos.controller (curChn.firstFrame - 1)
  -- if k == 0 then
  -- (
  --  curKey = addNewKey curObj.pos.controller (curChn.firstFrame - 1)
  --  curKey.value = datumPos
  -- )
  --)
 )
 0x0002: --ANIM_CHANNEL_Z = 2
 (
  curObj.pos.controller = linear_position()
  curKey = addNewKey curObj.pos.controller 0
  curKey.value = datumPos
  for f = curChn.firstFrame to curChn.lastFrame do
  (
   k = getKeyIndex curObj.pos.controller (f + 1)
   if k == 0 then
   (
    curKey = addNewKey curObj.pos.controller (f + 1)
    curKey.value = datumPos
   )
   else
   (
    curKey = curObj.pos.controller.keys[k]
   )
   curKey.value += [0, 0, curChn.values[(f - curChn.firstFrame + 1)]] * (inverse datumRot)
  )
  --if curChn.firstFrame == 0 then
  --(
  -- k = getKeyIndex curObj.pos.controller (curChn.lastFrame + 1)
  -- if k == 0 then
  -- (
  --  curKey = addNewKey curObj.pos.controller (curChn.lastFrame + 1)
  --  curKey.value = datumPos
  -- )
  --)
```

```
  --else
  --(
  -- k = getKeyIndex curObj.pos.controller (curChn.firstFrame - 1)
  -- if k == 0 then
  -- (
  --  curKey = addNewKey curObj.pos.controller (curChn.firstFrame - 1)
  --  curKey.value = datumPos
  -- )
  --)
 )
-- --0x0003: --ANIM_CHANNEL_XR = 3
-- --(
-- -- curKey = addNewKey curObj.pos.controller f
-- -- curKey.value.x = curChn.values[(f - curChn.firstFrame + 1)]
-- --)
-- --0x0004: --ANIM_CHANNEL_YR = 4
-- --(
-- -- curKey = addNewKey curObj.pos.controller f
-- -- curKey.value.y = curChn.values[(f - curChn.firstFrame + 1)]
-- --)
-- --0x0005: --ANIM_CHANNEL_ZR = 5
-- --(
-- -- curKey = addNewKey curObj.pos.controller f
-- -- curKey.value.z = curChn.values[(f - curChn.firstFrame + 1)]
-- --)
 0x0006: --ANIM_CHANNEL_Q = 6
 (
  curObj.rotation.controller = linear_rotation()
  curKey = addNewKey curObj.rotation.controller 0
  curKey.value = datumRot
  for f = curChn.firstFrame to curChn.lastFrame do
  (
   curKey = addNewKey curObj.rotation.controller (f + 1)
   curKey.value = curChn.values[(f - curChn.firstFrame + 1)] - (inverse datumRot)
  )
  --if curChn.firstFrame == 0 then
  --(
  -- k = getKeyIndex curObj.rotation.controller (curChn.lastFrame + 1)
  -- if k == 0 then
  -- (
  --  curKey = addNewKey curObj.rotation.controller (curChn.lastFrame + 1)
  --  curKey.value = datumRot
  -- )
  --)
  --else
  --(
  -- k = getKeyIndex curObj.rotation.controller (curChn.firstFrame - 1)
  -- if k == 0 then
```

```
       -- (
       --  curKey = addNewKey curObj.rotation.controller (curChn.firstFrame - 1)
       --  curKey.value = datumRot
       -- )
       --)
      )
     )
    )
   )
   for i = 1 to anim.bitchannels.count do
   (
    curChn = anim.bitchannels[i]
    -- curObj = execute ("$'" + hier.pivots[curChn.pivotID].pivotName + "'") -- at v1.02. it should be
curChn.pivotID + 1 at v1.04
    curObj = gmPivots[curChn.pivotID] -- test at v1.04
    if curObj != undefined then
    (
     case curChn.flags of
     (
      0x0000: --BIT_CHANNEL_VIS = 0 // turn meshes on and off depending on anim frame.
      (
       defVal = (curChn.defaultVal > 0)
       curObj.visibility = defVal
       curObj.visibility.controller = On_Off()
       curKey = addNewKey curObj.visibility.controller 0
       curKey.selected = defVal
       prevVal = defVal
       byteIdx = 1
       bitIdx = 1
       for f = curChn.firstFrame to curChn.lastFrame do
       (
        curVal = bit.get (curChn.values[byteIdx]) bitIdx
        if curVal != prevVal then
        (
         curKey = addNewKey curObj.visibility.controller (f + 1)
         curKey.selected = curVal
         prevVal = curVal
        )
        bitIdx += 1
        if bitIdx > 8 then
        (
         byteIdx += 1
         bitIdx = 1
        )
       )
       --curKey = addNewKey curObj.visibility.controller (curChn.lastFrame + 1)
       --curKey.selected = defVal
      )
      --0x0001: --BIT_CHANNEL_TIMECODED_VIS
     )
    )
   ) --for end
  ) --if end
  if (cmpAnim != undefined) and (hier != undefined) then
  (
   if cmpAnim.header.frameCount > 1 then
    animationRange = interval 1 cmpAnim.header.frameCount -- edited at v1.07
   else
    animationRange = interval 0 1
   frameRate = cmpAnim.header.frameRate
   case cmpAnim.header.flavor of
   (
    0x0: --ANIM_FLAVOR_TIMECODED
    (
     for i = 1 to cmpAnim.channels.count do
     (
      curChn = cmpAnim.channels[i]
      curObj = gmPivots[curChn.pivotID] -- test at v1.04
      if curObj != undefined then
      (
       datumPos = hier.pivots[curChn.pivotID + 1].pos
       datumRot = hier.pivots[curChn.pivotID + 1].rotation
       case curChn.flags of
       (
        0x0000: --ANIM_CHANNEL_TIMECODED_X = 0
        (
         curObj.pos.controller = linear_position()
         curKey = addNewKey curObj.pos.controller 0
         curKey.value = datumPos
         for t = 1 to curChn.timeCodesCount do
         (
          fNext = curChn.values[t].keyTime
          vNext = curChn.values[t].keyValue
          if t > 1 then
          (
           fPrev = curChn.values[(t - 1)].keyTime + 1
           vPrev = curChn.values[(t - 1)].keyValue
          )
          else
          (
           fPrev = fNext
           vPrev = vNext
          )
          for f = fPrev to fNext do
          (
           k = getKeyIndex curObj.pos.controller (f + 1)
           if k == 0 then
           (
            curKey = addNewKey curObj.pos.controller (f + 1)
            curKey.value = datumPos
           )
           else
           (
            curKey = curObj.pos.controller.keys[k]
           )
           step = vPrev + ((vNext - vPrev) * (f - fPrev + 1) / (fNext - fPrev + 1))
           curKey.value += [step, 0, 0] * (inverse datumRot)
          )
         )
        )
        0x0001: --ANIM_CHANNEL_TIMECODED_Y = 1
        (
         curObj.pos.controller = linear_position()
         curKey = addNewKey curObj.pos.controller 0
         curKey.value = datumPos
         for t = 1 to curChn.timeCodesCount do
         (
          fNext = curChn.values[t].keyTime
          vNext = curChn.values[t].keyValue
          if t > 1 then
          (
           fPrev = curChn.values[(t - 1)].keyTime + 1
           vPrev = curChn.values[(t - 1)].keyValue
          )
          else
          (
           fPrev = fNext
           vPrev = vNext
          )
          for f = fPrev to fNext do
          (
           k = getKeyIndex curObj.pos.controller (f + 1)
           if k == 0 then
           (
            curKey = addNewKey curObj.pos.controller (f + 1)
            curKey.value = datumPos
           )
           else
           (
            curKey = curObj.pos.controller.keys[k]
           )
           step = vPrev + ((vNext - vPrev) * (f - fPrev + 1) / (fNext - fPrev + 1))
           curKey.value += [0, step, 0] * (inverse datumRot)
          )
         )
        )
        0x0002: --ANIM_CHANNEL_TIMECODED_Z = 2
        (
         curObj.pos.controller = linear_position()
         curKey = addNewKey curObj.pos.controller 0
         curKey.value = datumPos
         for t = 1 to curChn.timeCodesCount do
         (
          fNext = curChn.values[t].keyTime
          vNext = curChn.values[t].keyValue
          if t > 1 then
          (
           fPrev = curChn.values[(t - 1)].keyTime + 1
           vPrev = curChn.values[(t - 1)].keyValue
          )
          else
          (
           fPrev = fNext
           vPrev = vNext
          )
          for f = fPrev to fNext do
          (
           k = getKeyIndex curObj.pos.controller (f + 1)
           if k == 0 then
           (
            curKey = addNewKey curObj.pos.controller (f + 1)
            curKey.value = datumPos
           )
           else
           (
            curKey = curObj.pos.controller.keys[k]
           )
           step = vPrev + ((vNext - vPrev) * (f - fPrev + 1) / (fNext - fPrev + 1))
           curKey.value += [0, 0, step] * (inverse datumRot)
          )
         )
        )
        0x0006: --ANIM_CHANNEL_TIMECODED_Q = 6
        (
         curObj.rotation.controller = linear_rotation()
         curKey = addNewKey curObj.rotation.controller 0
         curKey.value = datumRot
         for t = 1 to curChn.timeCodesCount do
         (
          f = curChn.values[t].keyTime
          curKey = addNewKey curObj.rotation.controller (f + 1)
          curKey.value = curChn.values[t].keyValue - (inverse datumRot)
```

```
     )
    )
   )
  )
 )
 --for i = 1 to cmpAnim.bitchannels.count do
 --(
 -- curChn = cmpAnim.bitchannels[i]
 -- curObj = gmPivots[curChn.pivotID] -- test at v1.04
 -- if curObj != undefined then
 -- (
 --  case curChn.flags of
 --  (
 --   --0x0000: --BIT_CHANNEL_VIS = 0 // turn meshes on and off depending on anim frame.
 --   0x0001: --BIT_CHANNEL_TIMECODED_VIS = 1
 --   (
 --    defVal = (curChn.defaultVal > 0)
 --    curObj.visibility = defVal
 --    curObj.visibility.controller = On_Off()
 --    curKey = addNewKey curObj.visibility.controller 0
 --    curKey.selected = defVal
 --    prevVal = defVal
 --    byteIdx = 1
 --    bitIdx = 1
 --    for t = 1 to curChn.timeCodesCount do
 --    (
 --     f = curChn.values[t].time
 --     curVal = bit.get (curChn.values[byteIdx]) bitIdx
 --     if curVal != prevVal then
 --     (
 --      curKey = addNewKey curObj.visibility.controller (f + 1)
 --      curKey.selected = curVal
 --      prevVal = curVal
 --     )
 --     bitIdx += 1
 --     if bitIdx > 8 then
 --     (
 --      byteIdx += 1
 --      bitIdx = 1
 --     )
 --    )
 --   )
 --  )
 -- )
 --) --for end
)
--0x1: --ANIM_FLAVOR_ADAPTIVE_DELTA
--0x2: --ANIM_FLAVOR_VALID
```

```
   )
  ) --if end
  if pickbox != undefined then
  (
   sName = pickbox.boxName
   dotPos = findString sName "."
   if (dotPos != undefined) and (dotPos < sName.count) do (sName = subString sName (dotPos + 1) -1)
   gmBox = Box name:sName pos:pickbox.center
   gmBox.width = pickbox.extent.x
   gmBox.length = pickbox.extent.y
   gmBox.height = pickbox.extent.z
   gmBox.wirecolor = pickbox.boxColor
  )
 )
)

macroscript ImportW3D
 category: "W3D Importer"
 buttontext: "Import W3D"
 tooltip: "Coolfile W3D Importer"
 icon:#("gMax",2)
(
 cfW3DImporter()
```