

TENPIX

TenPix Cloth

MAXScript Guide

GPU cloth simulation for Autodesk® 3ds Max®

Version	1.0544
For	3ds Max 2024, 2025, 2026 and 2027 (64-bit)
Graphics	NVIDIA RTX graphics card
Companion	TenPix Cloth Artist Manual

Contents

1	Introduction	4
1.1	About This Guide	4
1.2	Addressing the Modifiers	4
1.3	Solver Groups and Instances	5
1.4	Switching Modifiers On and Off	6
1.5	Return Values and the Listener	6
1.6	Units and Frames	6
2	Examples	7
2.1	Cloth from Nothing	7
2.2	Wind	7
2.3	Quality in Script	7
2.4	Switching a Modifier Off	8
2.5	A Small Tool of Your Own	8
2.6	Outfit Builder	9
2.7	Storm Stage	10
2.8	A Fringe of Threads	11
3	Modifier Properties	13
3.1	Tnpix_Solver	13
3.2	Tnpix_Garment	14
3.3	Tnpix_Gravity	15
3.4	Tnpix_Forces	15
3.5	Tnpix_Collision	16
3.6	Tnpix_Self_Collision	17
3.7	Tnpix_Material_Standard	18
3.8	Tnpix_Material_Advanced	18
3.9	Tnpix_Channel	18
3.10	Tnpix_Pin	19
3.11	Tnpix_Preserve	20
3.12	Tnpix_Stitch	20
3.13	Tnpix_Tear	21
3.14	Tnpix_Volume	23
3.15	Tnpix_Rigid	23
3.16	Tnpix_Soft_Body	24
3.17	Tnpix_Cache	25
3.18	Tnpix_Post_Process	26
3.19	Tnpix_Wrap	27

3.20	Tnpix_Threads	28
3.21	Tnpix_Threads_Object	30
4	The TnpixTool Interface	31
4.1	Simulation	31
4.2	Collision	31
4.3	Self Collision	32
4.4	Channels and the Tear Brush	32
4.5	Pins	33
4.6	Material Standard	34
4.7	Material Advanced	34
4.8	Cache, Post Process, Rigid and Wrap	35
4.9	Threads	36
4.10	Licence	37
5	The TnpixPreset Interface	38
5.1	Functions	38
5.2	Files	38
6	The TnpixGraph Interface	39
6.1	Functions	39
6.2	The Graph and Scripts	39
A	Field Names and Codes	40
A.1	Collider Row Fields	40
A.2	Material Standard Fields	40
A.3	Stock Fabrics	41
A.4	Material Advanced Fields	41
A.5	Multipliers, Presets and Bend Types	43
A.6	Other Codes	43

Introduction

1.1 About This Guide

Everything on the TenPix Cloth panels can also be driven from MAXScript: for wiring values together, batch set-ups, render-farm scripts, or tools of your own built on top of the plug-in.

This guide describes the MAXScript interface of TenPix Cloth version 1.0544 for 3ds Max 2024, 2025, 2026 and 2027. The four builds are compiled from one source and expose the same names. It has six parts:

- this chapter: how to address the modifiers and the rules that apply to every script;
- Chapter 2: worked examples, each with what you should see;
- Chapter 3: every modifier property, generated from the plug-in's own parameter blocks;
- Chapter 4: the `TnpixTool` interface, which reaches what a property cannot: painting, material values, collider rows, pins, caches and more;
- Chapter 5: the `TnpixPreset` interface for material presets;
- Chapter 6: the `TnpixGraph` interface, which opens and closes the TenPix Graph, and Appendix A: the field names and codes the functions take.

For what each setting does to the cloth, see the *TenPix Cloth Artist Manual*; this guide describes how to reach it.

1.2 Addressing the Modifiers

Each TenPix modifier is an ordinary 3ds Max modifier. Its MAXScript class name is the panel name with the spaces replaced by underscores; each class also answers to a second, compact name.

Script class	Also	Panel name	What it is
<code>Tnpix_Solver</code>	<code>TnpixSolver</code>	Tnpix Solver	The simulation
<code>Tnpix_Garment</code>	<code>TnpixGarment</code>	Tnpix Garment	Mesh rebuild (remesh or OpenVDB)
<code>Tnpix_Gravity</code>	<code>TnpixGravity</code>	Tnpix Gravity	Gravity for the Solver group
<code>Tnpix_Forces</code>	<code>TnpixForces</code>	Tnpix Forces	3ds Max space warps
<code>Tnpix_Collision</code>	<code>TnpixCollision</code>	Tnpix Collision	The collider list
<code>Tnpix_Self_Collision</code>	<code>TnpixSelfCollision</code>	Tnpix Self Collision	Self and garment-to-garment contact, layers
<code>Tnpix_Material_Standard</code>	<code>TnpixGoldMaterial</code>	Tnpix Material Standard	Stock fabrics and everyday dials
<code>Tnpix_Material_Advanced</code>	<code>TnpixMaterial</code>	Tnpix Material Advanced	The full fabric model
<code>Tnpix_Channel</code>	<code>TnpixChannel</code>	Tnpix Channel	Painted weight maps; the tear brush

Script class	Also	Panel name	What it is
Tnpix_Pin	TnpixPin	Tnpix Pin	Holds a painted region in place or on an object
Tnpix_Preserve	TnpixPreserve	Tnpix Preserve	A painted region follows the animation below the Solver
Tnpix_Stitch	TnpixStitch	Tnpix Stitch	Holds cloth to cloth
Tnpix_Tear	TnpixTear	Tnpix Tear	Tearing
Tnpix_Volume	TnpixVolume	Tnpix Volume	Volume, pressure, buoyancy
Tnpix_Rigid	TnpixRigid	Tnpix Rigid	Solid pieces inside the cloth
Tnpix_Soft_Body	TnpixSoftBody	Tnpix Soft Body	Soft flesh and simple muscles (experimental)
Tnpix_Cache	TnpixCache	Tnpix Cache	Saves and plays the simulation (.tnc, .abc)
Tnpix_Post_Process	TnpixPostProcess	Tnpix Post Process	Delta Mush and detangle after the solve
Tnpix_Wrap	TnpixWrap	Tnpix Wrap	A detailed mesh following simulated proxies
Tnpix_Threads	TnpixThreads	Tnpix Threads	Warp and weft threads at tears and painted hems
Tnpix_Threads_Object	TnpixThreadsObject	Tnpix Threads Object	A geometry object that shows a cloth's threads (not a modifier)

```
addModifier $ (Tnpix_Solver())           -- add a modifier to the selection
$.modifiers[#Tnpix_Solver].iterations = 3 -- address it by name on the stack
$.modifiers[#Tnpix_Garment].edgeLength = 0.5
```

`addModifier` puts the new modifier on top of the stack, so add them bottom-up: Tnpix Garment first, then the Solver, and the modifiers that work on its result (Tnpix Threads, Cache, Post Process) after it.

1.3 Solver Groups and Instances

The Solver simulates every object that carries the same Solver instance, together, as one Solver group. Two garments with two separate Solvers are two separate simulations and never touch. To make garments interact (a shirt under a jacket), create the modifier once and add that one object to each garment:

```
s = Tnpix_Solver()
addModifier $Shirt s
addModifier $Jacket s      -- same instance: one Solver group
```

Share the group-wide modifiers the same way: one Gravity, one Collision and one Self Collision instance across the Solver group. Tnpix Forces acts per garment: each garment is pushed by its own, and one instance shared across the Solver group pushes them alike. Group-wide settings have one value for the Solver group; where two members carry different values, the member created first (lowest node handle) wins, whichever object evaluates first. The Tear is the exception: its group-wide settings come from all the Solver group's Tears (a Map Tear before a Pressure Tear before a Stretch Tear, then the lowest Threshold). Per-garment settings (painted channels and everything bound to them, an Advanced material without a channel, pins, a Preserve, stitches, the Tear's region, Forces, Volume, Rigid, Soft Body, Garment, Threads) are read from each object. Channels are merged by name

across the Solver group: give each garment's channels their own names. A change on any member re-solves the whole Solver group.

The garments take their places in the Solver group in the order 3ds Max first evaluates them, and that order gives each garment without a layer of its own its self-collision layer. `TnpixTool.selfColLayers` reads it (Chapter 4).

1.4 Switching Modifiers On and Off

IMPORTANT

Use `tnEnabled`, not `enabled`. Every 3ds Max modifier already has a built-in `enabled` property (the light bulb), and it shadows a parameter of the same name. The TenPix modifiers expose their own on/off as `tnEnabled`:

```
$.modifiers[#Tnpix_Garment].tnEnabled = false
```

Modifier	Its own switch from MAXScript
Garment, Gravity, Forces, Self Collision, Material Advanced, Pin, Stitch, Tear, Volume, Wrap, Threads	<code>tnEnabled</code>
Collision	<code>active</code>
Solver, Channel, Cache, Post Process	None: use the light bulb (<code>enabled</code>). Post Process has <code>deltaMush</code> and <code>detangle</code> for its two passes.
Material Standard, Rigid	Their switch is named <code>enabled</code> and is shadowed by the light bulb; the light bulb turns the whole modifier off.

1.5 Return Values and the Listener

`TnpixTool` and `TnpixPreset` are global interfaces: call a function as `TnpixTool.name arg1 arg2`. Functions that change something return `true` or `false` (or a count); functions that report return a string. A refusal is a return value, never a silent success: an unknown field name, a value out of range or a missing modifier returns `false` (or `-1`, or a string that begins `*** REFUSED`) and writes nothing.

The MAXScript Listener carries TenPix's own short lines - a note, a warning or an error, each starting with `TenPix` - and nothing else during normal work. Nothing in this guide depends on a Listener line: the contract of every call is its return value and the panels' status lines.

1.6 Units and Frames

- Lengths are in the scene's system unit, as on the panels. A modifier created by script expresses its length defaults in the scene's unit, as one created from the panel does.
- Frame numbers (`activeFrom`, `restFrame`, `bindFrame`, the cache range) are counted as the timeline counts frames.
- The simulation starts at the first frame of the animation range. `TnpixTool.sim node frame` evaluates the cloth up to a frame, as scrubbing does.

Examples

Each example runs as it stands: paste it into the MAXScript Listener or a new script and evaluate it. Each says what you should see.

2.1 Cloth from Nothing

```
p = Plane length:100 width:100 lengthsegs:1 widthsegs:1 pos:[0,0,50]
addModifier p (Tnpix_Garment())
p.modifiers[#Tnpix_Garment].edgeLength = 2.0
addModifier p (Tnpix_Gravity())
addModifier p (Tnpix_Collision())
addModifier p (Tnpix_Solver())
t = Teapot radius:25 pos:[0,0,0]
TnpixTool.colAdd p t          -- the same path as the Pick Object button
```

What you should see: the plane's wireframe rebuilds into small, even triangles (Tnpix Garment at work). Play from the first frame: the sheet falls and drapes over the teapot. Leave out the last line and it falls through: an object only collides once it is in a Tnpix Collision list. Leave out the Gravity line and it floats: a Solver group with no Gravity modifier has no gravity.

2.2 Wind

```
w = Wind pos:[0,-200,80] strength:1.0 decay:0.0 turbulence:0.3 frequency:2.0
w.rotation = eulerAngles 90 0 0
addModifier $ (Tnpix_Forces())
append $.modifiers[#Tnpix_Forces].forceNodes w
append $.modifiers[#Tnpix_Forces].forceMuted 0      -- 0 = acting, 1 = muted
```

What you should see: nothing while stopped: forces act during the simulation. Play and the cloth leans and ripples away from the wind. Rotate the Wind and play again: the cloth follows the new direction. Set `$.modifiers[#Tnpix_Forces].forceMuted[1] = 1` and the wind stops acting on the cloth; the warp stays in the scene.

IMPORTANT

Pass `decay:0.0` when you create a Wind by script. A Wind made in the interface starts with Decay, Turbulence and Frequency at 0; made from MAXScript they start at 1.0, and with Decay at 1.0 the push dies off within a few units of the warp.

2.3 Quality in Script

`substeps` is the number the Solver panel shows, the same number everywhere (4 is the default).

```
s = $.modifiers[#Tnpix_Solver]
s.substeps      = 8          -- the High step, as on the panel
s.iterations    = 4
s.contactPasses = 4          -- contact passes rise with iterations
s.stretchQuality = 4
format "substeps: %\n" s.substeps
```

What you should see: the panel shows Substeps 8, Iterations 4, Contact passes 4, Stretch quality 4. The next play simulates again with the new quality: steadier cloth in fast motion and a proportionally slower solve. To simulate again without changing anything, call `TnpixTool.reset()`.

2.4 Switching a Modifier Off

```
$.modifiers[#Tnpix_Garment].tnEnabled = false
```

What you should see: the original mesh comes back in the viewport; the conversion is suspended, not lost. Set it back to `true` and the rebuilt mesh returns exactly as before. Remember: `tnEnabled`, not `enabled`.

2.5 A Small Tool of Your Own

```
rollout TenPixQuick "TenPix Cloth - Quick Tools" (
  fn setQuality panelSubsteps iter ctc stretch = (
    for o in objects where o.modifiers[#Tnpix_Solver] != undefined do (
      local s = o.modifiers[#Tnpix_Solver]
      s.substeps      = panelSubsteps      -- the panel's number
      s.iterations    = iter
      s.contactPasses = ctc
      s.stretchQuality = stretch
    )
  )
  button btnMake "Make Selected Cloth" width:150
  button btnDraft "Draft Quality (all)" width:150
  button btnHigh "High Quality (all)" width:150
  on btnMake pressed do (
    for o in selection do (
      addModifier o (Tnpix_Garment())
      o.modifiers[#Tnpix_Garment].edgeLength = 2.0
      addModifier o (Tnpix_Gravity())
      addModifier o (Tnpix_Solver())
    )
  )
  on btnDraft pressed do setQuality 1 1 1 1
  on btnHigh pressed do setQuality 8 4 4 4
)
createDialog TenPixQuick 170 110
```

What you should see: a small window with three buttons. Select meshes and click *Make Selected Cloth*: they become simulation-ready cloth with your defaults. Before a quick preview, *Draft Quality* drops every Solver in the scene to the Draft step; before the final, *High Quality* raises them all. The Quality tool on the TenPix toolbar does the same with all four steps.

2.6 Outfit Builder

Select the garment meshes in dressing order, innermost first (shirt, then jacket, then coat), and build them into one Solver group: every garment is rebuilt into even cloth, and all of them share one Solver, one Self Collision with layer ordering, one Gravity and one collider list. A second button prints a report of every cloth in the scene.

```
rollout OutfitBuilder "TenPix Cloth - Outfit Builder" (
    local body = undefined
    spinner spnEdge "Cloth density" range:[0.2,10,2.0] fieldwidth:50
    spinner spnThick "Self thickness" range:[0.05,5,0.15] fieldwidth:50
    dropdownlist ddQ "Quality" items:#("Draft (1/1)", "Normal (4/2)", "High (8/4)") selection:2
    checkbox chkWind "Add wind rig" checked:false
    pickbutton pbBody "Pick Body (collider)" width:190
    button btnBuild "Build Outfit From Selection" width:190
    button btnReport "Scene Cloth Report" width:190
    on pbBody picked obj do ( body = obj; pbBody.text = "Body: " + obj.name )
    on btnBuild pressed do (
        local order = selection as array
        if order.count == 0 then ( messageBox "Select the garments first (innermost first)."; return() )
        -- substeps, iterations, contact passes, stretch quality
        local q = #( #(1,1,1,1), #(4,2,1,2), #(8,4,4,4) )[ddQ.selection]
        for o in order do (
            -- the rebuild is per garment
            addModifier o (Tnpix_Garment())
            o.modifiers[#Tnpix_Garment].edgeLength = spnEdge.value
        )
        -- ONE Solver group: the same Self Collision, Gravity, Collision, Forces and Solver on every
        garment
        local sc = Tnpix_Self_Collision()
        sc.tnEnabled = true
        sc.layerOrdering = true
        sc.layerHoldStrength = 4.0
        sc.thickness = spnThick.value
        local grav = Tnpix_Gravity()
        local col = Tnpix_Collision()
        local fr = undefined
        if chkWind.checked do (
            local w = Wind pos:[0,-250,120] strength:0.5 decay:0.0 turbulence:0.4 frequency:2.0
            w.rotation = eulerAngles 90 0 0 -- blow across the outfit, not straight up
            fr = Tnpix_Forces()
            append fr.forceNodes w
            append fr.forceMuted 0
        )
        local s = Tnpix_Solver()
        s.substeps = q[1]
        s.iterations = q[2]
        s.contactPasses = q[3]
        s.stretchQuality = q[4]
        for i = 1 to order.count do (
            local o = order[i]
            addModifier o sc
            addModifier o grav
            addModifier o col
            if fr != undefined do addModifier o fr
            addModifier o s -- last on = on top
            -- the garment list behind the Self Collision panel: garment i sits on layer i - 1
            append sc.memberHandles (o.inode.handle)
            append sc.memberLayers (i - 1)
            append sc.memberFrictions -1.0 -- -1 = inherit
        )
    )
)
```

```

        append sc.memberThicknesses -1.0
        format "layer %: % ready\n" (i - 1) o.name
    )
    if body != undefined then (
        TnpixTool.colAdd order[1] body
        format "OUTFIT BUILT - % layers in one Solver group, body added as collider. Play.\n"
order.count
    ) else (
        format "OUTFIT BUILT - % layers in one Solver group. Pick the body on the Tnpix Collision
panel, then play.\n" order.count
    )
)
on btnReport pressed do (
    format "\n===== SCENE CLOTH REPORT =====\n"
    for o in objects where o.modifiers[#Tnpix_Solver] != undefined do (
        local s = o.modifiers[#Tnpix_Solver]
        format "% | substeps % iter % contact % | heatmap %\n" o.name s.substeps s.iterations
s.contactPasses s.stressHeatmap
        local sc = o.modifiers[#Tnpix_Self_Collision]
        if sc != undefined do format "        self: on % thickness % friction % layers % hold %\n"
sc.tnEnabled sc.thickness sc.friction sc.layerOrdering sc.layerHoldStrength
        local f = o.modifiers[#Tnpix_Forces]
        if f != undefined do for n in f.forceNodes do format "        force: %\n" n.name
        if o.modifiers[#Tnpix_Collision] != undefined do format "        colliders: %\n"
(TnpixTool.colCount o)
    )
    format "===== \n"
)
)
createDialog OutfitBuilder 210 220

```

What you should see: select your garments in order, optionally pick the body, and click *Build Outfit From Selection*. The script names each layer as it goes. Open the Self Collision panel on any garment: the garment list shows every layer, innermost first. Play: the layers settle on the body and on each other instead of passing through. A new Self Collision starts switched on; the explicit `sc.tnEnabled = true` keeps the script's intent plain and does not depend on the default.

2.7 Storm Stage

This example builds a whole animated shot, TenPix and native 3ds Max together: a ground plane, an animated pole, a flag whose edge is painted and pinned to the pole entirely from script, a Wind whose strength is keyed calm, storm, calm, a Vortex waiting muted in the list, and TurboSmooth on top of the simulation.

```

(
    animationRange = interval 0 120
    local ground = Plane length:600 width:600 pos:[0,0,0] wirecolor:gray
    local pole = Cylinder radius:2 height:150 pos:[-60,0,0] wirecolor:orange
    animate on (
        at time 0    pole.pos = [-60,0,0]
        at time 60   pole.pos = [-60,25,0]
        at time 120  pole.pos = [-60,0,0]
    )
    -- the flag is built flat at the origin so it can be painted in easy coordinates,
    -- then stood up and moved onto the pole at the end
    local flag = Plane length:60 width:110 lengthsegs:1 widthsegs:1 wirecolor:red
    addModifier flag (Tnpix_Garment())
    flag.modifiers[#Tnpix_Garment].edgeLength = 2.2
)

```

```

-- the held edge: a channel painted along the flag's left edge (x = -55 in its own space)
addModifier flag (Tnpix_Channel())
select flag
max modify mode
modPanel.setCurrentObject flag.modifiers[#Tnpix_Channel] -- painting needs the Channel panel open
completeRedraw()
TnpixTool.chanAdd flag "pole_edge" 0.0
for y = -30.0 to 30.0 by 3.0 do TnpixTool.chanPaint flag [-55, y, 0] 3.0 0 1.0
format "%\n" (TnpixTool.chans flag)
addModifier flag (Tnpix_Pin())
TnpixTool.pinSet flag "pole_edge" 1.0 pole -- weight 1.0 = follows the pole completely
addModifier flag (Tnpix_Self_Collision())
flag.modifiers[#Tnpix_Self_Collision].tnEnabled = true
addModifier flag (Tnpix_Gravity())
local w = Wind pos:[-250,-250,100] strength:0.2 decay:0.0 turbulence:0.3 frequency:2.0
w.rotation = eulerAngles 90 0 45
animate on (
    at time 0 w.strength = 0.2
    at time 45 w.strength = 3.0
    at time 80 w.strength = 3.0
    at time 120 w.strength = 0.3
)
local v = Vortex pos:[80,80,0]
addModifier flag (Tnpix_Forces())
append flag.modifiers[#Tnpix_Forces].forceNodes w; append flag.modifiers[#Tnpix_Forces].forceMuted
0
append flag.modifiers[#Tnpix_Forces].forceNodes v; append flag.modifiers[#Tnpix_Forces].forceMuted
1
local s = Tnpix_Solver()
s.substeps = 4
s.iterations = 3
addModifier flag s
addModifier flag (TurboSmooth iterations:1)
-- into place: standing up, its left edge on the top of the pole
flag.rotation = eulerAngles 90 0 0
flag.pos = [-2, 0, 115]
format "STORM STAGE READY - press Play.\n"
)

```

What you should see: a small stage appears, the script prints the channel table with *pole_edge* in it, and the Pin panel shows the channel as bound. On Play the flag hangs calm from the pole, whips hard as the keyed storm rises, then settles as the wind dies, while the pole sways and the flag's edge follows it. The Vortex waits muted:

`flag.modifiers[#Tnpix_Forces].forceMuted[2] = 0` adds swirl without touching the scene. TurboSmooth makes the rendered flag smoother than the simulated one: native modifiers above the cloth stack.

2.8 A Fringe of Threads

Tnpix Threads needs no tear to show: along painted edges it fringes a hem from the first frame. With the channel left empty, every open border of the cloth is fringed.

```

p = Plane length:60 width:60 lengthsegs:1 widthsegs:1 pos:[0,0,50]
addModifier p (Tnpix_Garment())
p.modifiers[#Tnpix_Garment].edgeLength = 2.0
addModifier p (Tnpix_Gravity())
addModifier p (Tnpix_Solver())
addModifier p (Tnpix_Threads())          -- on top: directly above the Solver
th = p.modifiers[#Tnpix_Threads]
th.whereItTears = false                  -- this sheet does not tear
th.alongPaintedEdges = true              -- channel "" = every open border
th.grainFrom = 2                         -- Combed only ...
th.combAlong = 1                        -- ... along world X, for a flat sheet
th.warpEvery = 1.0
th.weftEvery = 1.0
th.threadLength = 3.0
TnpixTool.sim p 10                      -- simulate up to frame 10
print (TnpixTool.threadsInfo p)
print (TnpixTool.threadsMakeObject p)   -- the tubes into their own object

```

What you should see: the sheet falls and its four borders carry a fringe of threads that swing as it falls. The first print is the Threads status for frame 10, among it *Along painted edges: every open border* and the tube faces of every thread. The second is the name of the new threads object, for example *Plane001_Threads*: the tubes now live in that object, and the sheet leaves its stack as the Solver made it. Set `th.viewportPercent` to show more or fewer threads in the viewport; a render makes all of them.

Modifier Properties

Every modifier property that MAXScript can read and write, generated from the plug-in's own parameter blocks, and the properties of the Tnpix Threads Object, the one TenPix object that is not a modifier. Access a property through the modifier, for example `$Shirt.modifiers[#Tnpix_Solver].iterations = 3`, and an object's directly, for example `$Sheet_Threads.threadsFrom`.

Type is the MAXScript type; **A** marks an animatable property (it takes keys, and `animate on` keys it). **Default** is the value of a newly added modifier. Lengths are in the scene's system unit; [†] marks a length default that a new modifier expresses in the scene's unit (the inch value is shown).

3.1 Tnpix_Solver

Panel name **Tnpix Solver** · also reachable as `TnpixSolver`

The simulation. Its own on/off is 3ds Max's light bulb (there is no `tnEnabled`).

Property	Type	A	Default	Range	Description
<code>substeps</code>	integer	A	4	1 – 99,999,985	The Substeps dial: the same number the Solver panel and the TenPix Graph show (4 = the default). Inside, the solver takes 15 more steps than this number each frame (4 runs 19).
<code>iterations</code>	integer	A	2	1 – 100,000	Constraint iterations per substep.
<code>contactPasses</code>	integer	A	1	1 – 16	Contact resolution passes per step. Raise it together with <code>iterations</code> and <code>stretchQuality</code> .
<code>stretchQuality</code>	integer	A	2	1 – 32	Extra passes for the stretch constraints (the edge net and the membrane) per substep. 2 is the normal solve.
<code>smoothingIterations</code>	integer	A	10	0 – 16	Surface smoothing passes per substep. 0 = off; 2 removes most of the zigzag of a dense mesh.
<code>unlimitedGround</code>	boolean		false		<i>Unlimited ground (Z = 0)</i> : an endless floor at Z = 0 for every garment of the Solver group and its threads, with no ground object and no collider.
<code>strainLimit</code>	boolean		true		The Solver's own stretch limit, followed by fabrics that do not hold their own. No panel control; the fabric's limit is set on the material.
<code>strainSweeps</code>	integer	A	8	0 – 64	Sweeps of the Solver's own stretch limit. No panel control.
<code>triangleStrain</code>	boolean		true		The Solver's triangle strain term. No panel control.
<code>strainRateLimit</code>	float	A	25.0	0.0 – 10,000	The Solver's own strain-rate limit in rest lengths per second, followed by fabrics that do not set their own. No panel control.

Property	Type	A	Default	Range	Description
breathingAllowance	float (distance)	A	0.0	0.0 – 100.0	Contact give on the panel: a slide, in scene units, before contact friction grips. Not a stretch setting.
meshContactSoftness	float	A	0.0	0.0 – 1.0	Mesh contact softness: 0 = hard contact.
meshContactFast	boolean		false		Not on the panel in this version. Leave at the default.
meshContactSmoothing	float	A	0.0	0.0 – 1.0	Mesh contact smoothing: blends contact across neighbouring points.
ccd	boolean		false		Not on the panel in this version. Leave at the default.
redetectFastBody	boolean		false		Not on the panel in this version. Leave at the default.
ccdMarch	integer	A	8	1 – 64	Not on the panel in this version. Leave at the default.
stressHeatmap	boolean		false		Stress / contact heatmap display. Diagnostic only.
particleBonds	boolean		false		Particle bonds debug display, drawn on every garment of the Solver group. Diagnostic only.
bondDistance	boolean		false		Bond display: distance bonds.
bondBend	boolean		false		Bond display: bend (hinge edge) bonds.
bondTriangleStrain	boolean		false		Bond display: triangle strain.
bondWeld	boolean		false		Bond display: welds (held only).
exactTriContact	boolean		false		Not on the panel in this version. Leave at the default.
frictionInLoop	boolean		true		Applies friction inside the solve rather than after it.
bodyStabilization	boolean		true		Damps the jolt when a body starts moving under the cloth.
gammaDecoupleBody	boolean		true		Keeps a body's push out of the fabric's damping.
bodyStrainComp	float	A	0.25	0.0 – 10.0	Lets the fabric give where it presses on a body. Mesh contact only.
collisionSystem	integer		4	1 – 4	1 = Volume Only, 2 = Volume + Proxy, 3 = Surface + Volume, 4 = Surface (True Geometry). A value outside 1 to 4 is refused by the solver.
deepRescue	boolean		false		Not on the panel in this version. Leave at the default.

3.2 Tnpix_Garment

Panel name **Tnpix Garment** · also reachable as **TnpixGarment**

Rebuilds the mesh: even triangles (mode 0) or re-surfaced from a volume (mode 1). Goes at the bottom of the stack.

Property	Type	A	Default	Range	Description
tnEnabled	boolean		true		On/off. Off shows the original mesh; the conversion is suspended, not lost.
edgeLength	float		5.0	0.0001 – 1,000,000	Target triangle edge length of the remesh, in the mesh's own units.
keepUVs	boolean		true		Carries the texture coordinates through the remesh.
mode	integer		0	0 – 1	0 = Garment (remesh), 1 = OpenVDB (volume).
voxelSize	float		5.0	0.0001 – 1,000,000	OpenVDB: voxel size.
halfWidth	float		3.0	1.0 – 100.0	OpenVDB: narrow-band half width, in voxels.
adaptivity	float		0.0	0.0 – 1.0	OpenVDB: mesher adaptivity; 0 = uniform triangles.
capOpenBorders	boolean		false		OpenVDB: close open borders before voxelizing.
shellOpenVdb	boolean		true		OpenVDB: treat an open mesh as a thin shell.
smooth	boolean		false		OpenVDB: smooth the volume before re-surfacing.
flipNormals	boolean		false		OpenVDB: reverse the result's normals.
smoothWidth	float		1.0	0.0 – 100.0	OpenVDB: smoothing width.
dilateAmount	float		0.0	0.0 – 1000.0	OpenVDB: grow the volume.
erodeAmount	float		0.0	0.0 – 1000.0	OpenVDB: shrink the volume.

3.3 Tnpix_Gravity

Panel name **Tnpix Gravity** · also reachable as **TnpixGravity**

Gravity for the Solver group. A Solver group with no Gravity modifier has no gravity. Share one instance across the Solver group.

Property	Type	A	Default	Range	Description
tnEnabled	boolean		true		On/off.
magnitude	float	A	–386.22 inch [†]	–1,000,000 – 1,000,000	Gravity in scene units per second squared, applied along direction . A new modifier starts at Earth gravity in the scene's unit (–386.22 inch, –981 cm, –9.81 m).
direction	point3	A	[0.0, 0.0, 1.0]		The axis gravity acts along.

3.4 Tnpix_Forces

Panel name **Tnpix Forces** · also reachable as **TnpixForces**

3ds Max space warps acting on the cloth. **forceNodes** and **forceMuted** are parallel arrays: keep them the same length.

Property	Type	A	Default	Range	Description
<code>tnEnabled</code>	boolean		true		On/off for the whole list.
<code>strength</code>	float		1.0	-1000.0 – 1000.0	Multiplier on every warp in the list.
<code>forceNodes</code>	node array		empty		The space warps in the list.
<code>forceMuted</code>	integer array		empty		Mute flag per warp, parallel to <code>forceNodes</code> : 1 = muted (contributes nothing), 0 = acting.
<code>zoneOn</code>	boolean		false		Active Zone: every warp acts only where the zone reaches, times the zone's value there.
<code>zoneHelpers</code>	node array		empty		The Influence Helpers that make the zone (other objects are ignored). Empty with <code>zoneOn</code> = the zone reaches nothing.
<code>zoneTime</code>	boolean		false		true = the zone works only from <code>zoneFrom</code> to <code>zoneTo</code> .
<code>zoneFrom</code>	integer		0	-100,000 – 100,000	First frame the zone works.
<code>zoneTo</code>	integer		100	-100,000 – 100,000	Last frame the zone works; before <code>zoneFrom</code> it never works.
<code>forceChannel</code>	string		""		The painted channel the forces act through: each point is pushed by its weight in it. "" (the default) = the whole cloth. A cloth without a channel of this name is not pushed.
<code>zoneInvert</code>	boolean		false		Invert: the warps act where the zone does NOT reach (one minus the zone's value), inside the zone's time. Empty <code>zoneHelpers</code> = every point is pushed.

3.5 Tnpix_Collision

Panel name **Tnpix Collision** · also reachable as `TnpixCollision`

The collider list. Every row has its own settings, which are read with `TnpixTool.rows` and changed with `TnpixTool.colSet` (see Chapter 4). The list is group-wide.

Property	Type	A	Default	Range	Description
<code>active</code>	boolean		true		On/off for the whole list.
<code>colliders</code>	node array		empty		The collider objects. Appending an object adds a row with the default settings; <code>TnpixTool.colAdd</code> , the Pick button's own path, is preferred: it refuses duplicates and objects that are not meshes.
<code>objEnabled</code>	boolean array		empty		Row storage. Read rows with <code>TnpixTool.rows</code> ; change them with <code>TnpixTool.colSet</code> .
<code>objTwoSided</code>	boolean array		empty		Row storage (see <code>objEnabled</code>).
<code>objVoxel</code>	float array		empty		Row storage (see <code>objEnabled</code>).

Property	Type	A	Default	Range	Description
<code>objShowVoxel</code>	boolean array		empty		Row storage (see <code>objEnabled</code>).
<code>objThickness</code>	float array		empty		Row storage (see <code>objEnabled</code>).
<code>objDynSlide</code>	float array		empty		Row storage: friction (see <code>objEnabled</code>).
<code>objProxyTarget</code>	integer array		empty		Row storage (see <code>objEnabled</code>).
<code>objGrip</code>	float array		empty		Row storage (see <code>objEnabled</code>).
<code>objOpenTwoSided</code>	boolean array		empty		Row storage: Open mesh (see <code>objEnabled</code>).
<code>objMeshSmooth</code>	float array		empty		Row storage (see <code>objEnabled</code>).

3.6 Tnpix_Self_Collision

Panel name **Tnpix Self Collision** · also reachable as `TnpixSelfCollision`

Self collision and layers. A new modifier starts switched on. The four member arrays are the storage behind the panel's garment list; keep them the same length and append to them in step.

Property	Type	A	Default	Range	Description
<code>tnEnabled</code>	boolean		true		On/off. A new modifier starts on; turning it on also turns <code>edgeEdge</code> on.
<code>edgeEdge</code>	boolean		true		Edge-edge contacts. Always on under modes 1 and 2.
<code>thickness</code>	float	A	0.15 inch [†]	0.001 – 10.0	Self-contact radius in scene units; two layers settle about twice this apart. A new modifier takes 0.15 inch in the scene's unit.
<code>friction</code>	float	A	0.0	0.0 – 1.0	Self-collision friction.
<code>passes</code>	integer		10	1 – 50	Relaxation passes per step.
<code>queryDiameter</code>	float		2.0	1.0 – 4.0	Contact search diameter as a multiple of the thickness; never below 1.
<code>samePieceExclusion</code>	float		2.5	0.5 – 4.0	Under <code>exclusion</code> 0 only: the same-piece radius as a multiple of the point spacing.
<code>mode</code>	integer		2	0 – 2	0 = Basic, 1 = Advanced (Optimized plus <code>collisionSubsteps</code>), 2 = Optimized (Basic plus recovery).
<code>layerOrdering</code>	boolean		false		Layer ordering: the lower layer holds, the upper gives way. Works under every mode.
<code>overrideMaterialFriction</code>	boolean		false		true = this modifier's friction wins over the materials' friction.
<code>layerHoldStrength</code>	float		4.0	1.0 – 32.0	How firmly the lower layer holds; 1 = no hold.
<code>memberHandles</code>	integer array		empty		The garment list: node handles (<code>obj.inode.handle</code>).

Property	Type	A	Default	Range	Description
<code>memberLayers</code>	integer array		empty		Layer per garment, parallel to <code>memberHandles</code> ; lower = nearer the body.
<code>memberFrictions</code>	float array		empty		Friction per garment, parallel; -1 = inherit.
<code>memberThicknesses</code>	float array		empty		Thickness per garment, parallel; -1 = inherit.
<code>exclusion</code>	integer		1	0 – 1	0 = Rest distance, 1 = Mesh ring.
<code>inelasticContact</code>	boolean		true		true = cloth pressing on cloth stops instead of bouncing (the push goes to positions, not velocity).
<code>collisionSubsteps</code>	integer		4	1 – 16	Under mode 1 only: self collision runs this many times per substep, each time on the next part of the substep's motion; the material is not solved again. 1 = exactly Optimized. The other modes ignore it.

3.7 Tnpix_Material_Standard

Panel name **Tnpix Material Standard** · also reachable as `TnpixGoldMaterial`

The Standard material keeps its dials in its own record, not in properties: read and write them by name with `TnpixTool.goldAt` / `goldSetAt` (Chapter 4 and Appendix A). Its only property is shadowed by 3ds Max's own `enabled`, so the material's check box is reached from the panel only.

Property	Type	A	Default	Range	Description
<code>enabled</code>	boolean		–		Shadowed by 3ds Max's own <code>enabled</code> (the light bulb); the material's check box is panel-only.

3.8 Tnpix_Material_Advanced

Panel name **Tnpix Material Advanced** · also reachable as `TnpixMaterial`

The Advanced material also keeps its dials in its own record: read and write them with `TnpixTool.matAt` / `matSetAt` and the other `mat...` functions (Chapter 4 and Appendix A).

Property	Type	A	Default	Range	Description
<code>tnEnabled</code>	boolean		true		On/off: the Enabled check box of the Channel (Tnpix) rollout.

3.9 Tnpix_Channel

Panel name **Tnpix Channel** · also reachable as `TnpixChannel`

Painted channels. Channels are created, painted and cut with `TnpixTool.chan...` and `tearCut` (Chapter 4). Its on/off is 3ds Max's light bulb.

Property	Type	A	Default	Range	Description
<code>showWeights</code>	boolean		false		Draws the selected channel's weights in the viewport.
<code>channelName</code>	string		""		The Name field of the panel.
<code>showWeightColors</code>	boolean		false		Shows the weights as vertex colours. Script only in this version.

3.10 Tnpix_Pin

Panel name **Tnpix Pin** · also reachable as `TnpixPin`

Holds a painted region in place or on an object. `TnpixTool.pinSet` sets channel, weight and object in one undoable call.

Property	Type	A	Default	Range	Description
<code>tnEnabled</code>	boolean		true		On/off.
<code>channel</code>	string		""		The painted channel this pin holds; empty = holds nothing. A <code>Tnpix_Soft_Body</code> or <code>Tnpix_Volume</code> with a channel above the pin on the stack lets go of the points it paints.
<code>weight</code>	float	A	1.0	0.0 – 1.0	The follow fraction: 1 = follows 100 percent, 0.5 = 50 percent. Below 1 is a soft pin.
<code>followObject</code>	node		undefined		The object the pinned points follow; <code>undefined</code> = held in place.
<code>activeTime</code>	boolean		false		true = the pin holds only from <code>activeFrom</code> to <code>activeTo</code> .
<code>activeFrom</code>	integer		0	-100,000 – 100,000	First frame the pin holds.
<code>activeTo</code>	integer		100	-100,000 – 100,000	Last frame the pin holds; before <code>activeFrom</code> the pin never activates.
<code>reach</code>	float		0.0	0.0 – 1,000,000	Scene units: painted points farther than this from the object when the pin takes hold stay free. 0 = any distance.
<code>minPiece</code>	float		1.0	0.0 – 100.0	Percent of the cloth's faces: a torn piece smaller than this lets go of the pin. 0 = never.
<code>dynamic</code>	boolean		false		true = the pinned points keep their mass and the pin pulls them; false = the object moves them directly.
<code>zoneOn</code>	boolean		false		Active Zone: the pin holds a painted point only while the zone reaches it, with its weight times the zone's value there. Works inside the Active time.
<code>zoneHelpers</code>	node array		empty		The Influence Helpers that make the zone. Empty with <code>zoneOn</code> = the pin holds nothing.

Property	Type	A	Default	Range	Description
zoneStay	boolean		false		Stays held once touched: a point the zone has reached stays held until the Active time ends. false (the default) = only the points inside the zone now.
zoneInvert	boolean		false		Invert: the pin holds where the zone does NOT reach (its weight times one minus the zone's value). With zoneStay , a point the helper has touched stays let go. Empty zoneHelpers = every painted point is held.

3.11 Tnpix_Preserve

Panel name **Tnpix Preserve** · also reachable as **TnpixPreserve**

Makes a painted region follow the animation of the mesh below the Tnpix Solver instead of simulating it: a point painted 1 sits on the animation and is ignored by every collision; between 0 and 1 it is pulled toward the animation by the Pin's Weight rule. Its place in the stack does not matter; over a Pin on the same point it wins.

Property	Type	A	Default	Range	Description
tnEnabled	boolean		true		On/off. Off, the painted points are simulated like the rest of the cloth.
channel	string		""		The painted channel that follows the animation below the Solver; "" = it holds nothing. Painted 1 sits on the animation, 0.5 follows it half way, 0 is simulated. A Tnpix_Soft_Body or Tnpix_Volume with a channel above it lets go of the points it paints.

3.12 Tnpix_Stitch

Panel name **Tnpix Stitch** · also reachable as **TnpixStitch**

Holds cloth to cloth on the same Solver, or to its own folded parts. Stitches form when the simulation starts.

Property	Type	A	Default	Range	Description
tnEnabled	boolean		true		On/off.
stitchTo	node array		empty		The garments to stitch to; each must be on this cloth's Solver. Append to it.
stitchToSelf	boolean		false		Also stitch to this cloth's own folded-close parts.
searchDistance	float (distance)		0.5 inch [†]	0.0 – 100,000	How far a painted point looks for a surface, in scene units. A new modifier takes 0.5 inch in the scene's unit.
tightness	float		1.0	0.0 – 1.0	1 = sewn onto the surface, 0 = keeps the starting gap (a rigid tack).
strength	float		1.0	0.0 – 1.0	How rigid the hold is: about 0.5 holds, 0.8 and above is sewn shut.

Property	Type	A	Default	Range	Description
channel	string		""		The painted channel that says where stitching is allowed.
threshold	float		0.5	0.001 – 1.0	Paint at or above this value stitches.
showStitches	boolean		true		Draw the stitches while the Stitch panel is open.
seamPasses	float		1.0	1.0 – 10.0	Solver passes for the seam each step; whole numbers (fractions are rounded).
stitchToHandles	integer array		empty		Storage: the targets by node handle, as the panel keeps them. Write stitchTo , not this.
offset	float (distance)		0.0	0.0 – 100,000	A gap, in scene units, the seam keeps between the two cloths (this cloth keeps it).
activeTime	boolean		false		true = the stitch holds only from activeFrom to activeTo .
activeFrom	integer		0	-100,000 – 100,000	First frame the stitch holds; the pairs are found where the cloths are on this frame.
activeTo	integer		100	-100,000 – 100,000	Last frame the stitch holds.
zoneOn	boolean		false		Active Zone: a stitch pulls only while the zone reaches its seam, with its strength times the zone's value there. Works inside the Active time.
zoneHelpers	node array		empty		The Influence Helpers that make the zone. Empty with zoneOn = no stitch pulls.
zoneStay	boolean		true		Stays sewn once touched (the default): a stitch the zone has reached keeps pulling - a zipper. false = only the stitches inside the zone now.
zoneInvert	boolean		false		Invert: a stitch pulls where the zone does NOT reach its seam (its strength times one minus the zone's value). With zoneStay , a stitch the helper has touched lets go for good. Empty zoneHelpers = every stitch pulls.
slide	boolean		false		true = the stitched points stay on the surface they are sewn to, at the stitch's gap, but can slide along it.
friction	float		0.3	0.0 – 1.0	How hard the sliding is: 0 slides freely, 1 does not slide. Used while slide is true.

3.13 Tnpix_Tear

Panel name **Tnpix Tear** · also reachable as **TnpixTear**

Tearing by painted region, by stretch or by pressure. On a shared Solver the Solver group's **tearBy**, **threshold**, **ends** and weakness range come from the first member, in creation (handle) order, that has an enabled Tear.

Property	Type	A	Default	Range	Description
tnEnabled	boolean		true		On/off. Off, the cloth simulates whole.

Property	Type	A	Default	Range	Description
tearBy	integer		0	0 – 2	0 = Map, 1 = Stretch, 2 = Pressure (needs a Tnpix Volume on the same stack).
channelName	string		""		The painted channel: where the cloth may tear (Map, Pressure) and where it is weaker (all three). Empty under Map or Pressure = nothing tears.
markedAbove	float		0.5	0.0 – 1.0	Paint at or above this value belongs to the region.
ends	integer		0	0 – 1	0 = both ends of an edge must be marked, 1 = either end is enough.
threshold	float		10.0	0.0 – 1,000,000	Map: the pull an edge takes before it opens. Pressure: the bursting pressure. Not used by Stretch. Not converted by the scene's unit.
weakMin	float		1.0	0.01 – 10.0	Weakness at the lightest marked paint (multiplies the pull or the stretch).
weakMax	float		1.0	0.01 – 10.0	Weakness at full paint.
calibSubsteps	integer		0	0 – 100,000	The Solver's Substeps (the panel's number) the threshold was typed at, written by the panel; 0 = not recorded. A label: nothing computes from it.
calibIterations	integer		0	0 – 100,000	The Solver iterations the threshold was typed at (see calibSubsteps).
growRings	integer		0	0 – 8	Points within this many edges of a painted point count as painted.
stretchRatio	float		0.1	0.001 – 100.0	Stretch mode: how much longer an edge may get before it tears, as a part of its length (0.1 = 10 percent).
zoneOn	boolean		false		Active Zone: a point may tear only where the zone reaches it (anywhere inside a helper's Far radius).
zoneHelpers	node array		empty		The Influence Helpers that make the zone. Empty with zoneOn = nothing tears.
zoneStay	boolean		true		Stays active once touched: a point the zone has reached stays able to tear after the zone moves on.
zoneTime	boolean		false		true = the zone works only from zoneFrom to zoneTo .
zoneFrom	integer		0	-100,000 – 100,000	First frame the zone works.
zoneTo	integer		100	-100,000 – 100,000	Last frame the zone works.
zoneInvert	boolean		false		Invert: a point may tear where the zone does NOT reach (beyond a helper's Near radius), inside the zone's time. With zoneStay , a point the helper has covered never tears. Empty zoneHelpers = every marked point may tear.

3.14 Tnpix_Volume

Panel name **Tnpix Volume** · also reachable as `TnpixVolume`

Gives a closed shell an inside. Per object. Keyed values drive one continuous simulation.

Property	Type	A	Default	Range	Description
<code>tnEnabled</code>	boolean		true		On/off.
<code>mode</code>	integer		0	0 – 1	0 = Keep volume, 1 = Pressure.
<code>volumeRatio</code>	float	A	1.0	0.05 – 10.0	Target as a fraction of the rest volume.
<code>softness</code>	float		0.0	0.0 – 1.0	0 holds the volume exactly; higher lets contact squash the shape.
<code>pressure</code>	float	A	0.0	-10.0 – 10.0	Pressure mode: a constant push outward, in scene units per solver step; negative pulls in.
<code>restFrame</code>	integer		0	-100,000 – 100,000	The frame whose shape is the rest volume; 0 is the first simulated frame, wherever the timeline starts.
<code>buoyancy</code>	boolean	A	false		Lifts the closed shape as if it were lighter than air. A whole object only: not used while <code>volumeChannel</code> names a channel.
<code>lift</code>	float	A	1.0	0.0 – 3.0	Buoyancy strength: 1 hovers, above 1 rises.
<code>volumeChannel</code>	string		""		The painted channel this Volume holds. "" = the whole closed object; a channel name = only the painted area, each point as firmly as its paint. Above a <code>Tnpix_Pin</code> or <code>Tnpix_Preserve</code> on the stack, its painted points are let go from them.
<code>volumeCentre</code>	node		undefined		The centre a region's volume is measured against. <code>undefined</code> = the middle of its painted points; a node = that object (a bone or helper inside the region). The cloth itself is refused.
<code>restFrameOn</code>	boolean		true		true = the rest volume is the shape at <code>restFrame</code> ; false = no rest frame: the volume follows the shape below the Solver on every frame, read through a <code>Tnpix Wrap</code> below it, times <code>volumeRatio</code> .

3.15 Tnpix_Rigid

Panel name **Tnpix Rigid** · also reachable as `TnpixRigid`

Solid pieces inside the cloth. Its switch is named `enabled`, so from MAXScript 3ds Max's own light bulb shadows it; use the light bulb or the panel.

Property	Type	A	Default	Range	Description
<code>enabled</code>	boolean		true		On/off. Shadowed by 3ds Max's own <code>enabled</code> from MAXScript.
<code>piecesFrom</code>	integer		0		0 = painted Tnpix channel, 1 = mesh elements.

Property	Type	A	Default	Range	Description
channel	string		""		The painted channel that marks the rigid regions.
threshold	float		0.5	0.0 – 1.0	Paint at or above this value is rigid.
eachRegion	boolean		true		true = every connected painted region is its own piece.
rigidity	float	A	1.0	0.0 – 1.0	How firmly each piece holds its rest shape; 0 = off.
massMul	float		1.0	0.01 – 1000.0	The pieces' weight as a multiple of the fabric's.
bounce	float		0.5	0.0 – 1.0	Speed kept on hitting a collider: 0 lands dead, 1 bounces to full height.
friction	float		0.3	0.0 – 10.0	The pieces' own friction in object and self collision.
airResistance	float		0.0	0.0 – 100.0	Slows the pieces' motion; 0 = off, 5 stops within a second.
spinResistance	float		0.0	0.0 – 100.0	Slows the pieces' spin; 0 = off.

3.16 Tnpix_Soft_Body

Panel name **Tnpix Soft Body** · also reachable as **TnpixSoftBody**

Experimental. Turns a closed object into soft flesh, solved in the Solver's own steps; with fibres, a simple muscle. Only **activation** and **tensionPoint** take keys. Setting **material** to 0 - 3 writes that material's numbers, as the panel does.

Property	Type	A	Default	Range	Description
tnEnabled	boolean		true		On/off. Off, the object simulates as cloth.
softChannel	string		""		The painted channel this Soft Body's material covers; "" = the whole object, which lets no pin's point go. Above a Tnpix_Pin or Tnpix_Preserve on the stack, its painted points are let go from them. The panel's Region list writes it.
keepVolume	boolean		true		true = it keeps its volume as it wobbles; false = it squashes like a sponge.
resolution	integer		40	8 – 200	Tetrahedra across the object's longest side. Several Soft Bodies on one object: the finest wins.
material	integer		1		0 Fat, 1 Jelly, 2 Muscle, 3 Rubber, 4 Custom. 0 - 3 write that material's numbers; a number changed afterwards makes it 4.
stiffness	float		5.0	0.01 – 100,000	How firm it is, in kPa: fat 1.5, jelly 5, relaxed muscle 10, rubber 500.
density	float		1.02	0.01 – 20.0	How heavy it is for its size, g/cm ³ (water is 1).
damping	float		0.08	0.0 – 1.0	How fast its wobble dies: 0 never, 1 settles without a wobble.

Property	Type	A	Default	Range	Description
fibres	boolean		false		true = fibres along the fibre line that pull when it is activated.
fibreStart	node		undefined		The object at the fibre line's start. With fibreEnd both undefined , the line runs along the object's own length; set both or neither. The soft body itself is refused.
fibreEnd	node		undefined		The object at the fibre line's end (see fibreStart).
strength	float		300.0	0.0 – 100,000	How hard the fibres pull at full activation, kPa.
maxContraction	float		45.0	0.0 – 90.0	How short the fibres can get at full activation, percent of their length.
tendons	float		10.0	0.0 – 45.0	How much of each end is tendon, percent of the fibre line.
tendonStiffness	float		30.0	1.0 – 1000.0	How many times stiffer the tendons are than the body.
activation	float	A	0.0	0.0 – 1.0	How hard the fibres work, 0 to 1. Key it to flex the muscle.
autoTension	boolean		true		true = the fibres tense by themselves as the fibre line's ends come closer.
tone	float		5.0	0.0 – 50.0	Auto tension: the tension it keeps, percent.
tensionPoint	float	A	0.5	0.0 – 1.0	Where along the fibre line it tenses most, and bulges: 0 at its start, 1 at its end.
tensionSpread	float		0.2	0.0 – 1.0	How wide the tense area is, as a share of the fibre line; 0 = the whole body evenly.
junction	float		0.1	0.001 – 0.45	The share of the fibre line over which a tendon turns into the body. No panel control.
relaxedFibre	float		0.05	0.0 – 1.0	A relaxed body's own fibres against stretch past their rest, as a share of the fibres' stiffness. No panel control.

3.17 Tnpix_Cache

Panel name **Tnpix Cache** · also reachable as **TnpixCache**

Saves the simulation to .tnc or .abc and plays it back. Save, load and clear with **TnpixTool.cacheSave**, **cacheLoad** and **cacheClear**.

Property	Type	A	Default	Range	Description
file	filename		–		The cache file. The extension is the format: .tnc (TenPix) or .abc (Alembic).
playFromCache	boolean		false		Serve the cloth from the file; the Solver below is not evaluated.
animationRange	boolean		true		false also selects Start..End. Use specificRange .
start	integer		0	-100,000 – 100,000	First frame to save when specificRange is true.

Property	Type	A	Default	Range	Description
<code>end</code>	integer		100	-100,000 – 100,000	Last frame to save when <code>specificRange</code> is true.
<code>frameOffset</code>	integer		0	-100,000 – 100,000	Shifts the cached frames on the timeline.
<code>retime</code>	float	A	1.0	0.0 – 1000.0	Playback speed; key it for a ramp or a freeze (0).
<code>posterizeStep</code>	integer	A	0	0 – 100,000	Every this many frames the pose is stamped and held; 0 = off.
<code>posterizeHold</code>	integer	A	0	0 – 100,000	Frames each stamped pose is held.
<code>specificRange</code>	boolean		false		false = save the scene's animation range; true = save <code>start</code> .. <code>end</code> .
<code>writeVelocities</code>	boolean		true		Alembic only: write per-point velocities for motion blur.

3.18 Tnpix_Post_Process

Panel name Tnpix Post Process · also reachable as `TnpixPostProcess`

Delta Mush and detangle after the solve; it changes nothing about the simulation. Its on/off is 3ds Max's light bulb.

Property	Type	A	Default	Range	Description
<code>deltaMush</code>	boolean		true		Delta Mush on/off.
<code>iterations</code>	integer	A	10	0 – 500	Smoothing rounds per frame.
<code>strength</code>	float	A	1.0	0.0 – 1.0	0 = the simulation as it is, 1 = the full mush.
<code>pinBoundary</code>	boolean		true		Open edges keep the simulated position.
<code>weightFrom</code>	integer		0		Where the mush may work: 0 = None (whole cloth), 1 = Vertex color (red), 2 = Tnpix channel.
<code>weightChannel</code>	string		""		The Tnpix channel that weighs the mush, by name.
<code>weightInvert</code>	boolean		false		Flips the weight.
<code>method</code>	integer		2		0 = Uniform (Max), 1 = Cotangent, 2 = Shrink-free (Taubin).
<code>step</code>	float	A	0.5	0.01 – 1.0	How far each round moves a point towards its neighbours.
<code>tangentialOnly</code>	boolean		false		Relax along the surface only.
<code>keepCreases</code>	float	A	60.0	0.0 – 180.0	Edges folded sharper than this angle (degrees) are not smoothed across; 0 = off.
<code>inwardLimit</code>	float	A	0.0	0.0 – 1.0	Allowed fraction of the move into the cloth along the normal.
<code>outwardLimit</code>	float	A	1.0	0.0 – 1.0	Allowed fraction of the move out of the cloth along the normal.
<code>detail</code>	float	A	1.0	0.0 – 5.0	Multiplier on the rest deltas: 1 = the modelled detail exactly.

Property	Type	A	Default	Range	Description
temporal	boolean		false		Smooth over time (needs frames on both sides: a Tnpix Cache or the Solver's memory below).
temporalWindow	integer		2	1 – 30	Frames on each side.
temporalFilter	integer		0		0 = Gaussian, 1 = Median.
restFrom	integer		0		0 = Range start, 1 = This frame (restFrame).
restFrame	integer		0	-100,000 – 100,000	The rest frame when restFrom is 1.
detangle	boolean		true		Detangle on/off.
passes	integer	A	3	1 – 50	Detangle rounds per frame.
thicknessFrom	integer		0		0 = From Self Collision, 1 = This value (thickness).
thickness	float	A	0.15	0.0 – 1000.0	The detangle separation in scene units when thicknessFrom is 1.
selfCollisions	boolean		true		Detangle cloth against itself.
bodyColliders	boolean		false		Detangle also moves the cloth out of the colliders listed in the Tnpix Collision below the Post Process, each at its own row's thickness.
detangleWeightFrom	integer		0		Which points the detangle may move: 0 = None, 1 = Vertex color, 2 = Tnpix channel.
detangleWeightChannel	string		""		The Tnpix channel that weighs the detangle, by name.

3.19 Tnpix_Wrap

Panel name **Tnpix Wrap** · also reachable as **TnpixWrap**

Makes a high-resolution mesh follow low-resolution proxies. **TnpixTool.wrapFill** and **wrapInfo** are its script functions.

Property	Type	A	Default	Range	Description
tnEnabled	boolean		true		On/off.
proxies	node array		empty		The low-resolution meshes that drive this one.
bindFrame	integer		0	-100,000 – 100,000	The frame where the mesh and the proxies line up.
smoothing	float		1.0	0.0 – 4.0	Blend reach over neighbouring proxy faces, in proxy edge lengths; 0 = one face.
falloff	float		1.0	0.1 – 10.0	How fast a neighbouring face's influence fades; 1 = even.
offset	float	A	1.0	0.0 – 1.0	Fraction of its height above the proxy a point keeps.
strength	float	A	1.0	0.0 – 1.0	How far points move to where the wrap puts them; 1 = fully.

Property	Type	A	Default	Range	Description
<code>weightFrom</code>	integer		0		Where per-point strength comes from: 0 = Everywhere, 1 = Vertex color, 2 = Tnpix channel.
<code>weightChannel</code>	string		""		The Tnpix channel that scales the strength, by name.
<code>limitDistance</code>	boolean		false		true = points farther than <code>maxDistance</code> from every proxy are left alone.
<code>maxDistance</code>	float		1.0	0.0 – 1,000,000	Scene units, measured at the bind frame.
<code>cutTears</code>	boolean		true		Cut this mesh along proxy tears.
<code>rememberProxies</code>	boolean		true		Keep each proxy's shape per frame in memory. No limit: every frame seen stays until the scene closes, about 12 bytes per proxy point per frame.
<code>closeCuts</code>	boolean		true		Where this mesh is a shell (two layers), close each cut along a proxy tear with a strip of faces. Works with <code>cutTears</code> on; a one-layer mesh is left as it is.
<code>rimMaterialID</code>	integer		0	0 – 65,535	Material ID of the faces that close the cuts; 0 = the ID of the outer face beside each one.

3.20 Tnpix_Threads

Panel name **Tnpix Threads** · also reachable as `TnpixThreads`

Warp and weft threads where the cloth tears and along painted hems. It goes directly above the Tnpix Solver and reads the Solver's frames. `TnpixTool.threadsInfo` reads its status lines and `threadsMakeObject` makes its threads object (Chapter 4).

Property	Type	A	Default	Range	Description
<code>tnEnabled</code>	boolean		true		On/off. Off, the cloth leaves the modifier exactly as it came in.
<code>whereItTears</code>	boolean		true		Threads are born on the frame a seam tears. Needs a Tnpix Tear on the cloth.
<code>alongPaintedEdges</code>	boolean		false		A fringe on the open borders inside <code>channel</code> .
<code>channel</code>	string		""		The channel that marks the borders to fringe, by name; "" = every open border. An edge is fringed when both its points are painted at 0.5 or more.
<code>grainFrom</code>	integer		0	0 – 2	Which way the threads run: 0 = Auto (UVs, else combed), 1 = UVs only, 2 = Combed only.
<code>combAlong</code>	integer		0	0 – 2	The axis of the combed grain: 0 = Up (world Z), 1 = World X, 2 = World Y.
<code>grainAngle</code>	float		0.0	-180.0 – 180.0	Turns the grain around each face, in degrees; 45 = bias cut.
<code>warpEvery</code>	float (distance)		0.25 inch [†]	0.0001 – 1,000,000	Spacing between the threads that run along the grain.

Property	Type	A	Default	Range	Description
weftEvery	float (distance)		0.25 inch ⁺	0.0001 – 1,000,000	Spacing between the threads that run across the grain.
threadLength	float (distance)		0.75 inch ⁺	0.0 – 1,000,000	How far a loose thread reaches out of the cloth.
randomLength	float		40.0	0.0 – 100.0	Percent: each thread's length varies by up to this share, longer or shorter.
strays	float		3.0	0.0 – 100.0	Percent of the threads that are much longer than the rest.
strayLength	float		6.0	1.0 – 100.0	How many times longer a stray is (the panel's <i>x length</i>).
seed	integer		1	0 – 999,999	Another arrangement of the same settings: which threads are long, clumped or holding.
clump	float		0.3	0.0 – 1.0	How strongly neighbouring threads bunch into strands; 0 = apart, 1 = a strand's tips meet.
clumpSize	integer		6	2 – 100	Threads per strand.
frizz	float		0.2	0.0 – 2.0	How far a thread wanders off straight, as a share of its length.
frizzFrequency	float		3.0	0.0 – 50.0	Kinks along a thread.
holdAcrossCrack	boolean		true		Some threads span the crack from lip to lip (the panel's <i>Threads that still hold across the crack</i>).
warpHolds	float		12.0	0.0 – 100.0	Percent of the crack's warp threads that still span it.
weftHolds	float		12.0	0.0 – 100.0	Percent of the crack's weft threads that still span it.
snapsAt	float		3.0	1.0 – 100.0	A holding thread pulls length out of the weave up to this many times its length, then stretches (the panel's <i>A holding thread snaps at</i>).
looseEnds	boolean		true		A torn seam without a holding thread leaves a loose thread on each side.
leanTowardCut	float		0.0	0.0 – 1.0	0 = every thread along its grain, 1 = straight out of the cut.
motion	integer		0	0 – 1	0 = Dynamic, 1 = Follow the cloth.
damping	float		0.08	0.0 – 1.0	Share of a thread's swing lost each frame, at 30 frames per second. Dynamic only.
pointsPerThread	integer		13	3 – 64	Simulated points per thread.
collideWithCloth	boolean		true		The threads stay off the cloth they grow from.
collideWithColliders	boolean		true		The threads collide with the cloth's Tnpix Collision objects.
threadThickness	float (distance)		0.02 inch ⁺	0.0 – 1,000,000	Each thread's thickness; collision keeps half of it off every surface.
tubeSides	integer		3	3 – 16	Sides of each tube.
warpMaterialID	integer		0	0 – 65,535	Material ID of the warp tubes; 0 = the ID of the face the thread grows from.

Property	Type	A	Default	Range	Description
<code>weftMaterialID</code>	integer		0	0 – 65,535	Material ID of the weft tubes; 0 = the ID of the face the thread grows from.
<code>threadCap</code>	integer		500,000	1 – 2,000,000	The most threads this modifier makes; past it, every crack is thinned alike.
<code>material</code>	integer		0	0 – 5	0 = Cotton, 1 = Wool, 2 = Silk, 3 = Linen, 4 = Synthetic, 5 = Custom. 0 to 4 write <code>bendingLength</code> , <code>damping</code> , <code>friction</code> and <code>breaksAt</code> ; changing any of those makes it 5.
<code>bendingLength</code>	float (distance)		0.35 inch ⁺	0.001 – 1,000,000	How stiff the thread is, as a length: shorter threads stand, longer ones droop.
<code>friction</code>	float		0.4	0.0 – 2.0	How much a thread grips what it lies on; 0 = slides off, 1 = stays. Dynamic only.
<code>forcesFromCloth</code>	boolean		true		The cloth's Tnpix Forces act on its threads too. Dynamic only.
<code>collideWithRoom</code>	boolean		true		The threads collide with the other objects simulated in the same Tnpix Solver.
<code>tubesGoTo</code>	integer		0	0 – 1	0 = the cloth mesh, 1 = their own object (a Tnpix Threads Object).
<code>breaksAt</code>	float		7.0	0.0 – 1000.0	Percent a thread stretches before it snaps.
<code>viewportPercent</code>	float		10.0	0.1 – 100.0	Percent of the threads the viewport shows; a render makes all of them.
<code>textureFrom</code>	integer		0	0 – 1	0 = UVs along the thread, 1 = the cloth's texture read at the thread's root.

3.21 Tnpix_Threads_Object

Panel name **Tnpix Threads Object** · also reachable as `TnpixThreadsObject`

A geometry object, not a modifier: it shows the tubes of a cloth's Tnpix Threads whose `tubesGoTo` is 1 (their own object). Make one with `TnpixTool.threadsMakeObject`, or create it with `Tnpix_Threads_Object()` and set `threadsFrom`.

Property	Type	A	Default	Range	Description
<code>threadsFrom</code>	node		undefined		The cloth whose Tnpix Threads this object shows; only a node that carries a Tnpix Threads is accepted.
<code>threadsModifier</code>	modifier		undefined		The Tnpix Threads this object was made from, set by Make the threads object; <code>undefined</code> shows the cloth's top-most Tnpix Threads.

The TnpixTool Interface

`TnpixTool` is a global function interface. It reaches what a property cannot: collider rows, the garment list's layers, painting and the tear brush, pins, material values, caches, Post Process, Rigid and Wrap, threads and the licence. Call a function as `TnpixTool.name arg1 arg2`.

- **node** is the object that carries the modifier, for example `$Shirt`. The function finds the modifier on its stack.
- **index** counts from 0. For materials it is the position among the materials of that kind on the stack, from the top (0 = the first); for collider rows it is the row number printed by `rows`. The Threads functions are the exception: they count from 1.
- Functions that change something are undoable where the panel's own gesture is, and they refresh an open panel.

4.1 Simulation

Function	Returns	What it does
<code>reset()</code>	–	Clears the stored simulation; the next evaluation simulates again from the first frame. The Solver panel's Reset Simulation.
<code>sim node frame</code>	integer	Evaluates the cloth up to <i>frame</i> , as scrubbing does, and returns how many frames are stored.
<code>touch node</code>	boolean	Tells the Collision modifier on the node that it changed, so its list is read again.

4.2 Collision

Function	Returns	What it does
<code>colAdd node collider</code>	boolean	Adds a collider row, the Pick Object button's own path. Refuses an object already in the list, an object that is not a mesh, and an object that carries this Collision modifier (a cloth is never its own collider).
<code>colCount node</code>	integer	How many rows the list holds.
<code>rows node</code>	string	Every row and all of its settings, as text.
<code>colSet node index field value</code>	boolean	Sets one setting of one row. <i>index</i> is the row as printed by <code>rows</code> ; <i>field</i> is a number from Appendix A. Values outside a setting's range are clamped to it.

```
TnpixTool.colAdd $Shirt $Body
TnpixTool.colSet $Shirt 0 5 0.2    -- row 0, field 5 (Thickness) = 0.2
TnpixTool.colSet $Shirt 0 6 0.5    -- row 0, field 6 (Friction) = 0.5
TnpixTool.rows $Shirt
```

4.3 Self Collision

Function	Returns	What it does
<code>selfCollLayers node</code>	string	The garment list's layers as the panel shows them: the number of garments, then for each garment its node handle, its name, the layer it is simulated at, and <i>set</i> when the layer was typed in the list or <i>auto</i> when it comes from the Solver group's order. (<i>no Tnpix Self Collision on this node</i>) when the node has none.

```
TnpixTool.selfCollLayers $Shirt
-- for example: "3 garment(s) | h368 Shirt 0 auto | h412 Pants 1 auto | h883 Jacket 2 set"
```

The layers stored in the list are the Self Collision's member arrays (Chapter 3). A garment with no stored layer takes its place in the Solver group's order, which `selfCollLayers` reads from the garment's Solver.

4.4 Channels and the Tear Brush

Painting from script needs the Tnpix Channel modifier to be the current object in the Modify panel, on exactly one object (an instanced Channel modifier is refused). A dab lands on the channel selected on the panel, which after `chanAdd` is the new one. The brush centre is in the object's own space, the coordinates the mesh vertices have before the node is moved.

Function	Returns	What it does
<code>chanAdd node name fill</code>	boolean	Creates a channel with every vertex at <i>fill</i> (clamped to 0–1) and selects it. Names are unique and case-sensitive. If a selection arrives from below the Channel at a sub-object level (vertex, edge or face), those vertices are born at the panel's Value; a selection left inside an object at its top level is not used.
<code>chanPaint node center radius mode strength</code>	boolean	One brush dab. Modes: 0 Add, 1 Subtract, 2 Replace, 3 Smooth.
<code>chanStroke node on</code>	boolean	<code>true</code> opens a stroke and <code>false</code> closes it: the dabs between count as one stroke, as a mouse drag does.
<code>chanApplySel node</code>	boolean	Writes the panel's Value into the selected vertices (Apply to Selection).
<code>chanRemSel node</code>	boolean	Sets the selected vertices back to 0 (Remove Selection).
<code>chanClear node</code>	boolean	Sets the whole selected channel back to 0 (Clear Channel).
<code>chans node</code>	string	The channel table, as text.
<code>chanCount node name atOrAbove</code>	integer	How many vertices carry paint at or above the value; –1 = no such channel.
<code>chanRemove node name</code>	boolean	Deletes a channel by name, with the tear-brush lines drawn into it (the Delete button).

Function	Returns	What it does
<code>chanMirror node side tolerance</code>	string	Mirror, the panel button's own function: copies the selected channel's weights from one side of the object to the other. <i>side</i> : 0 +X to -X, 1 -X to +X, 2 +Y to -Y, 3 -Y to +Y, 4 +Z to -Z, 5 -Z to +Z. <i>tolerance</i> , in the object's units, is how far a mirrored vertex may land from a vertex and still count as its twin. One undo step. Returns <i><channel>: mirrored +X to -X: N vertices took their twin's weight, M found no twin</i> , or a line beginning <i>refused</i> : with the reason (no Tnpix Channel on the node, no channel selected, a side outside 0 – 5, the modifier not the current object or on more than one object).
<code>chanSelCount node edges</code>	integer	How many components the Channel Vertices selection holds (<i>edges</i> <code>false</code>), or the Channel Edges selection (<code>true</code>), on the object the modifier is edited on; –1 when the Channel is not the current object.
<code>tearCut node channel points projDir spacing</code>	string	The Tear brush: cuts a line into the mesh inside the Channel modifier and marks it. See below.
<code>tearRemove node mode</code>	string	0 removes the last cut, 1 removes every cut; the weights are put back. Undoable. Returns <i>removed N cut(s), M remain</i> .

tearCut. Every point of the array (object space, like `chanPaint`) is projected onto the mesh that reaches the Channel modifier along *projDir* (the panel uses the view direction). Points closer than *spacing* to the previous one are dropped. The line is cut into the mesh, every channel grows to the new vertex count, and the vertices on the line are marked in the named channel at 1.0 (an empty name means the channel selected on the panel). A first or last point that misses the cloth runs the cut out to the border on that side. It needs no Modify panel. It returns *cut N of M points, +A verts, marked K in 'name'*, or a line beginning **** REFUSED*: with the reason (no channel, no mesh yet, no point on the cloth, fewer than two points after spacing).

```
select $Flag
max modify mode
modPanel.setCurrentObject $Flag.modifiers[#Tnpix_Channel]
TnpixTool.chanAdd $Flag "hem" 0.0
TnpixTool.chanPaint $Flag [-55, 0, 0] 3.0 0 1.0
TnpixTool.chanCount $Flag "hem" 0.5
TnpixTool.chanMirror $Flag 1 0.1      -- the hem painted on the -X side, given to the +X side too
TnpixTool.tearCut $Flag "hem" #([-40,0,0], [-20,4,0], [0,0,0]) [0,0,-1] 2.0
TnpixTool.tearRemove $Flag 0
```

`chanGrab` and `chanAddSel` are registered but not part of this version: they refuse with a message. Use `chanApplySel`.

4.5 Pins

Function	Returns	What it does
<code>pinSet node channel weight followObject</code>	boolean	Sets the Pin's channel, weight and object in one undoable call. <code>undefined</code> as the object holds the region in place.
<code>pin node</code>	string	The Pin's report: what it is bound to and what it holds.

The channel must be on the same stack as the Pin.

4.6 Material Standard

The Standard material keeps its values in a record, so they are read and written by name; Appendix A lists the names. Writing a value also marks it as set, which is what makes it act.

Function	Returns	What it does
<code>goldCount node</code>	integer	How many Standard materials the stack carries.
<code>goldAt node index</code>	string	The material's values, as text.
<code>goldSetAt node index field value</code>	boolean	Sets one value by name. An unknown name is refused and nothing is written; so is a <i>bendQuadK</i> outside 0 to 100.
<code>goldClearAt node index field</code>	boolean	Clears the tick of <i>matFriction</i> , <i>matGrip</i> or <i>strainRateLimit</i> and keeps the number, as unticking does. Other names are refused.
<code>goldChanAt node index channel</code>	boolean	Binds the material to a channel; <code>""</code> = the whole garment.
<code>goldPreset node index</code>	boolean	Applies stock fabric <i>index</i> (0–11, Appendix A) to the first Standard material. It writes only the values the fabric carries.
<code>goldSet node field value / goldChan node channel</code>	boolean	The same as <code>goldSetAt</code> / <code>goldChanAt</code> on the first Standard material.

```
TnpixTool.goldPreset $Shirt 1          -- Silk onto the first Standard material
TnpixTool.goldSetAt  $Shirt 0 "densityAll" 0.3
TnpixTool.goldSetAt  $Shirt 1 "bendGoldK" 12.0    -- the second Standard material
TnpixTool.goldChanAt $Shirt 1 "sleeves"
TnpixTool.goldClearAt $Shirt 1 "matFriction"
TnpixTool.goldAt     $Shirt 1
```

4.7 Material Advanced

Function	Returns	What it does
<code>matCount node</code>	integer	How many Advanced materials the stack carries.
<code>matAt node index</code>	string	The material's values and ticks, as text.
<code>matSetAt node index field value</code>	boolean	Sets one value by name (Appendix A) and ticks it. Out-of-range values are refused, not clamped.
<code>matSetPairAt node index family coeff magIdx</code>	boolean	Sets a stiffness as coefficient and multiplier row in one call: <i>family</i> is <code>"stretch"</code> or <code>"bend"</code> , <i>magIdx</i> a row of the × Multiplier list (Appendix A).
<code>matSetBendModelAt node index model</code>	boolean	The bend type: 0 Basic, 1 Shape Memory, 3 Shape Memory (Even). 2 is refused.

Function	Returns	What it does
<code>matPresetAt node index presetRow</code>	boolean	Applies a row of the Preset list: 1 Ultra Light, 2 Light, 3 Medium, 4 Heavy, 5 Leather (Debug); 0 (Custom) writes nothing. An override: every dial is reset first.
<code>matAuthoredAt node index on</code>	boolean	The Authored switch: 0 runs pure defaults, 1 your values again.
<code>matShearKClearAt node index</code>	boolean	Withdraws the shear stiffness (value kept).
<code>matChanAt node index channel</code>	boolean	Binds the material to a channel.

```
TnpixTool.matPresetAt $Shirt 0 4      -- Heavy Fabric onto material 0
TnpixTool.matSetPairAt $Shirt 0 "bend" 1.0 7  -- bend stiffness 1.0 x 1 000
TnpixTool.matSetAt $Shirt 0 "maxStretch" 8.0  -- percent, as on the panel
TnpixTool.matSetBendModelAt $Shirt 0 1      -- Shape Memory
TnpixTool.matChanAt $Shirt 0 "body_all"
TnpixTool.matAt $Shirt 0
```

The single-material forms `mat`, `matBind`, `matMul`, `matBendK`, `matBendKClear`, `matMembraneK` and `matMembraneKClear` still work and always address the first Advanced material on the stack.

An Advanced material whose channel is empty (`matChanAt node index ""`) is the default material of its own garment: it covers every vertex of that garment that no material bound to a channel claims. Give a garment only one such material; with two, only the one created last acts.

4.8 Cache, Post Process, Rigid and Wrap

Function	Returns	What it does
<code>cacheSave node</code>	integer	Save cache: frames written, or -1 if refused (the reason is in <code>cacheInfo</code> , for example the demo on a farm, or an Alembic file in the demo). It records from the first frame of the Cache's range wherever the time slider is, and puts the slider back. Frames already simulated are saved as they are.
<code>cacheLoad node</code>	boolean	Load: opens the file and plays from it.
<code>cacheClear node</code>	boolean	Clear: stops playing and forgets the file; nothing is deleted.
<code>cacheInfo node</code>	string	The file, whether it was found, the frames, whether it plays, the garments and the last status; <i>stopped=1</i> when the file is from a recording that stopped early and holds the frames it kept.
<code>postInfo node</code>	string	What the Post Process is set to and what rest pose it has.
<code>postCaptureRest node</code>	boolean	Capture rest: takes the rest pose now, at the rest frame.
<code>rigidInfo node</code>	string	The rigid pieces and points of the garment.
<code>wrapFill node</code>	integer	Fill now: evaluates the animation range into the proxy memory; frames filled, or -1 if refused (a proxy that is a live simulation).
<code>wrapInfo node</code>	string	The proxy memory's frames and size, the tick and the last message.

```
addModifier $Shirt (Tnpix_Cache())
$Shirt.modifiers[#Tnpix_Cache].file = "D:/caches/shirt.tnc"
TnpixTool.cacheSave $Shirt
TnpixTool.cacheInfo $Shirt
```

Progress while a cache records

While a Save records, TenPix Cloth calls a global MAXScript function named `tnpixCacheProgressHook`, if one exists, with two integers: the frames recorded so far and the frames this Save records. It is called once with 0 before the first frame and then after every frame, for the panel's Save button and for `cacheSave` alike. Its return value is ignored, and so is an error inside it: the recording goes on. The Deadline job that comes with the plug-in uses it to show each recorded frame as the task's progress.

```
global tnpixCacheProgressHook
fn tnpixCacheProgressHook done total = (
    format "recorded % of % frames\n" done total
)
TnpixTool.cacheSave $Shirt
tnpixCacheProgressHook = undefined    -- stop listening
```

Declare the function `global`. A farm manager that runs a job file as one block, as Deadline does, makes a function defined at the top of the file local to that block, and the plug-in looks for a global.

4.9 Threads

index counts the node's Tnpix Threads modifiers from the top of the stack, starting at 1 (not 0 as elsewhere in this chapter). The forms without *index* address the top-most Tnpix Threads.

Function	Returns	What it does
<code>threadsInfo node</code>	string	The status lines of the Tnpix Threads on this frame, as the panel shows them. (<i>no Tnpix Threads on this node</i>) when there is none.
<code>threadsInfoAt node index</code>	string	The same for the <i>index</i> -th Tnpix Threads.
<code>threadsMakeObject node</code>	string	Make the threads object: creates the Tnpix Threads Object that shows these threads, sets Tubes go to their own object, and returns the new object's name. A line beginning <i>***</i> says why it refused, for example when an object already shows these threads.
<code>threadsMakeObjectAt node index</code>	string	The same for the <i>index</i> -th Tnpix Threads.

```
addModifier $Sheet (Tnpix_Threads())    -- on top, above the Solver
$Sheet.modifiers[#Tnpix_Threads].warpEvery = 0.1
TnpixTool.sim $Sheet 30                  -- simulate up to frame 30
TnpixTool.threadsInfo $Sheet
TnpixTool.threadsMakeObject $Sheet      -- returns "Sheet_Threads"
```

4.10 Licence

Function	Returns	What it does
<code>licInfo()</code>	string	The licence state of this computer: free, demo or activated, the build (<i>build=rc, rc-beta or rc-demo</i>), the demo's limits (<i>cap=</i> vertices, <i>frames=</i> at the scene's frame rate) and <i>farm=</i> , why this session counts as a farm session or <i>no</i> .
<code>licCap()</code>	integer	The demo's vertex limit for the whole scene (25000).
<code>licActivate key</code>	string	Activate this machine with a key; returns the outcome.
<code>licCheck force</code>	string	Check now with the store (<code>true</code>) or only when the check is due (<code>false</code>).
<code>licDeactivate()</code>	string	Deactivate this machine.

In the demo build (*rc-demo*) `licInfo()` always reports *state=demo* and *key=none*, and `licActivate`, `licCheck` and `licDeactivate` refuse with a sentence that names the demo build: it takes no key.

Do not put a licence key in a script you share: it is your purchase.

`cache`, `lanes`, `instState`, `serState`, `serForceOff`, `custEditProbe`, `layers`, `debug`, `getDebug` and `debugFile` are test instruments for the product's own checks and are not part of the scripting interface; in the released build they do nothing.

The TnpixPreset Interface

`TnpixPreset` saves and loads material presets, the same files the Presets rollout of Tnpix Material Standard and Tnpix Material Advanced writes. *material* is the modifier itself, for example `$Shirt.modifiers[#Tnpix_Material_Standard]`.

5.1 Functions

Function	Returns	What it does
<code>folder()</code>	string	The presets folder: the one chosen on the Presets rollout, remembered on this computer, or <code>Documents\TenPix Cloth\Material Presets</code> .
<code>setFolder path</code>	boolean	Chooses the presets folder and remembers it.
<code>names material</code>	string	The presets in the folder that belong to this kind of material, one name per line.
<code>save material path</code>	boolean	Saves every dial of the material, with its tick, to the file.
<code>load material path</code>	integer	Loads a preset: every dial returns to its default and then takes the preset's value. Returns how many values were applied, or -1 if refused. Undoable.
<code>text material</code>	string	The material's current values in preset form.
<code>recycle path</code>	boolean	Moves a preset file to the Windows Recycle Bin.
<code>why()</code>	string	What the last <code>save</code> , <code>load</code> , <code>setFolder</code> or <code>recycle</code> did, or why it was refused. After a load it reads <i>applied N missing M unknown K bad B</i> .

5.2 Files

- Standard presets are `.tpstd` files; Advanced presets are `.tpadv` files. A preset is a text file of *name = value* lines.
- A preset stores the fabric, not its place on the garment: the channel binding is not saved and not changed by a load.
- A material of one kind refuses a preset of the other kind.

```
m = $Shirt.modifiers[#Tnpix_Material_Standard]
f = TnpixPreset.folder() + "\\Heavy denim.tpstd"
TnpixPreset.save m f
TnpixPreset.load $Jacket.modifiers[#Tnpix_Material_Standard] f
TnpixPreset.why()
TnpixPreset.names m
```

The TnpixGraph Interface

`TnpixGraph` opens and closes the TenPix Graph from a script, as the **Graph** button on the Tnpix Solver panel does. The TenPix Graph shows every Solver group of the scene and edits every setting of every TenPix modifier; it is a window for an interactive 3ds Max session (see *The TenPix Graph* in the Artist Manual).

6.1 Functions

Function	Returns	What it does
<code>open()</code>	nothing	Opens the TenPix Graph, or shows it again if it is open: a window of its own, which on 3ds Max 2025, 2026 and 2027 the user can dock by dragging it onto a side of the main window; shown again, it is where the user left it, floating or docked. Opened from a script, it opens on the scene's first Solver group.
<code>close()</code>	nothing	Closes the TenPix Graph.

6.2 The Graph and Scripts

- The graph writes the same parameters and values a script reaches through the modifier properties and `TnpixTool`; what a script changes shows in the open graph, and what the graph changes reads back in a script.
- Every change made in the graph is one entry in 3ds Max's undo, like a change made on a panel.
- The graph keeps nothing in the scene: Solver groups are formed by the modifiers, as described in *Solver groups and Instances* in Chapter 1.

Field Names and Codes

A.1 Collider Row Fields

For `TnpixTool.colSet node index field value`. Values outside a setting's range are clamped. Field 4 is not used.

Field	Setting	Values
0	Enabled	0 or 1
1	Two-sided mesh	0 or 1
2	Open mesh	0 or 1; a new row starts at 1
3	Voxel	0 = automatic; up to 20
5	Thickness	Scene units
6	Friction	
7	Proxy	0 = unlimited (the true mesh)
8	Grip	A multiplier (default 3.0)
9	Mesh Smooth	0 faceted to 1 smooth
10	Override material, as a switch	0 = Off, any other value = Everything
11	Override material	0 = Off, 1 = Unpainted cloth only, 2 = Everything

A.2 Material Standard Fields

For `TnpixTool.goldSet / goldSetAt`. Names are case-sensitive; an unknown name is refused. A new Standard material starts with the exponential membrane on (400,000 / 40,000) and friction and grip ticked.

Name	Panel control	Notes
bendGoldK	K bend	Per region
bendGoldDamp	damp bend	0 = off
dampRatioStretch / dampRatioBend	ratio stretch / ratio bend	Whole garment; 0 = off
relDamp	Relative Damp (k relative)	Whole garment; 0 = off
stretchGoldK / shearGoldK	K stretch / K shear	Per region; 0 = exponential membrane off
maxStrain	max strain	Fraction of the rest length; per region
damping	velocity damping (global)	Per region
motionDamp	Motion Damping	Per region; 0 = off

Name	Panel control	Notes
densityAll	weight (density)	Per region
warpStiff / weftStiff / shearStiff	warp / weft / shear	
membraneCompress	compress stiffness	
bendAnisoWarp / bendAnisoWeft	warp bend / weft bend	
bendQuadK	fold body k_b	0 to 100; 0 = off; outside the range refused
gammaK	collision damping (gk)	
matFriction / matGrip	friction / grip	Setting ticks them; <code>goldClearAt</code> unticks
strainLimitOn	limit stretch (hold this fabric's ceiling)	0 or 1
stretchBodyLimit	limit where it touches the body	0 or 1
strainRateLimit	strain-rate limit (/s)	Setting ticks it; <code>goldClearAt</code> unticks; 0 = off
triStrainLimit	per-triangle limiter	Integer, 0 or 1
strainOptIters / strainOptCg	Accurate strain limit (iterations) / Solve steps per iteration	Integers
mrsLevels	multi-res levels (MRS)	Integer; 0 = off
enableBend	bend enable	Integer, 0 or 1
preset	preset (the list's row)	Integer; use <code>goldPreset</code> , which also writes the values

A.3 Stock Fabrics

For `TnpixTool.goldPreset node index`.

Index	Fabric	Index	Fabric	Index	Fabric
0	Cotton (approx)	4	Wool	8	Spandex
1	Silk (approx)	5	Linen	9	Polyester
2	Denim (approx)	6	Chiffon (approx)	10	Canvas
3	Leather (approx)	7	Satin	11	Velvet

A.4 Material Advanced Fields

For `TnpixTool.matSetAt node index field value`. Setting a value ticks it. Out-of-range values are refused and the call returns `false`. A new Advanced material starts with relDamp (0.01), friction (0.35) and grip (3.0) ticked.

Name	Panel control
stretchGoldK / bendGoldK	Stretch Stiffness / Bend Stiffness as one absolute number (also written to the coefficient and multiplier pair)
shearGoldK	Shear Stiffness
edgeK / warpK / weftK	Edge Stiffness / Stretch XX (warp) / Stretch YY (weft)
relDamp	Relative Damp (k relative)
damping	Global Damping
stretchDamp / dampRatioBend	Stretch Damping Ratio / Bend Damping Ratio
membraneCompress	Compression Stiffness
foldBody	Fold Body k_b, 0 to 100
stretchDropoff / bendDropoff	Stiffness Dropoff (Stretch / Bend), 0 to 1
restLengthScale / restAngleScale	Rest Length Scale / Rest Angle Scale
normalDrag / tangentDrag	Normal Drag / Tangent Drag; any value on <i>tangentDragClear</i> withdraws Tangent Drag
massKgM2 / massMode	Mass (kg/m2) / Mass Mode: 0 = Unchanged (value kept), other = Override
scaleMultiplier / scaleMultiplierMode	Scale Multiplier; <i>scaleMultiplierMode</i> 0 withdraws it
stretchType	Stretch Type: 0 = Standard (the default), 2 = Fabric Curve. 1 is not a Stretch Type and is refused. The numbers are the stored values, not the order of the panel's list.
nlWarpSoft / nlWeftSoft	Fabric Curve: Warp / Weft soft until, in percent (0.1 to 100)
nlBiasLock	Fabric Curve: Bias lock, in degrees (1 to 80)
nlLockMul / nlSmooth	Fabric Curve: Lock stiffness (1 to 100,000; 1 = no curve) / Lock smoothness (0 to 2)
nlClear	Withdraws one Fabric Curve value, keeping the number: 0 Warp, 1 Weft, 2 Bias lock, 3 Lock stiffness, 4 Lock smoothness
strainLimitOn / maxStretch	Strain limit (0 or 1) / Max stretch, in percent
stretchHardness / stretchSweeps	Hardness / Sweeps
stretchBodyLimit / strainRateLimit	Limit where it touches the body (0 or 1) / Strain-rate limit (/s)
stretchLimitClear	Withdraws one stretch-limit value: 0 Strain limit, 1 Max stretch, 2 Hardness, 3 Sweeps, 4 Limit where it touches the body, 5 Strain-rate limit
stretchPlasticOn / Threshold / Rate / Max	Stretch Enable Plasticity (0 or 1) and its three amounts
bendPlasticOn / Threshold / Rate / Max	Bend Enable Plasticity (0 or 1) and its three amounts (radians)
bendAnisoWarp / bendAnisoWeft / bendGrainAngle	Grain: Warp Ratio / Weft Ratio / Grain Angle (degrees)
friction / grip	Friction (Tnpix) / Grip (Tnpix)

Name	Panel control
stretchMul / edgeMul / bendMul	The three multipliers (1.0 = no change)

A.5 Multipliers, Presets and Bend Types

× **Multiplier rows** for `matSetPairAt` (*mag/dx*):

Row	Multiplier	Row	Multiplier	Row	Multiplier
0–4	1e+10, 1e+9, 1e+8, 1e+7, 1e+6	5–9	100 000, 10 000, 1 000, 100, 10	10	1
11–15	0.1, 0.01, 0.001, 1e-4, 1e-5	16–20	1e-6, 1e-7, 1e-8, 1e-9, 1e-10	21	No Scaling

Advanced presets for `matPresetAt`: 0 Custom (writes nothing), 1 Ultra Light Fabric, 2 Light Fabric, 3 Medium Fabric, 4 Heavy Fabric, 5 Leather (Debug).

Bend types for `matSetBendModelAt`: 0 Basic, 1 Shape Memory, 3 Shape Memory (Even). 2 is refused.

A.6 Other Codes

Where	Values
<code>chanPaint</code> mode	0 Add, 1 Subtract, 2 Replace, 3 Smooth
<code>tearRemove</code> mode	0 the last cut, 1 every cut
<code>chanMirror</code> side	0 +X to -X, 1 -X to +X, 2 +Y to -Y, 3 -Y to +Y, 4 +Z to -Z, 5 -Z to +Z
<code>Tnpix_Solver.collisionSystem</code>	1 Volume Only, 2 Volume + Proxy, 3 Surface + Volume, 4 Surface (True Geometry)
<code>Tnpix_Self_Collision.mode</code> / <code>.exclusion</code>	0 Basic, 1 Advanced, 2 Optimized / 0 Rest distance, 1 Mesh ring
<code>Tnpix_Tear.tearBy</code> / <code>.ends</code>	0 Map, 1 Stretch, 2 Pressure / 0 Both marked, 1 Either
<code>Tnpix_Volume.mode</code>	0 Keep volume, 1 Pressure
<code>Tnpix_Rigid.piecesFrom</code>	0 Tnpix channel, 1 Mesh elements
<code>Tnpix_Soft_Body.material</code>	0 Fat, 1 Jelly, 2 Muscle, 3 Rubber, 4 Custom
<code>Tnpix_Garment.mode</code>	0 Garment (remesh), 1 OpenVDB (volume)
<code>weightFrom</code> (Post Process, Wrap)	0 None / Everywhere, 1 Vertex color, 2 Tnpix channel
<code>Tnpix_Threads.grainFrom</code> / <code>.combAlong</code>	0 Auto: UVs, else combed, 1 UVs only, 2 Combed only / 0 Up (world Z), 1 World X, 2 World Y
<code>Tnpix_Threads.motion</code>	0 Dynamic, 1 Follow the cloth
<code>Tnpix_Threads.material</code>	0 Cotton, 1 Wool, 2 Silk, 3 Linen, 4 Synthetic, 5 Custom
<code>Tnpix_Threads.tubesGoTo</code> / <code>.textureFrom</code>	0 this cloth, 1 their own object / 0 along the thread, 1 from its root

Where	Values
<code>threadsInfoAt</code> , <code>threadsMakeObjectAt</code> index	1 the top-most Tnpix Threads on the stack, 2 the next, and so on