

VWD's Cloth & Hair

Program Manual

Version 2.x



(c) 2019 Virtual World Dynamics

<https://www.virtualworldynamics.net>

Welcome!

Welcome to Virtual World Dynamics' Cloth & Hair Program documentation.

We're glad you've chosen VWD's Cloth & Hair (VWD for short) product and hope it will surpass your expectations. As with any complex and powerful simulation tool, there are a lot of gears that 'make it go', and VWD is no exception. Because the program packs a lot of power into a compact toolkit, the learning curve can be steep at times, but the capabilities are extensive and the results can be well-worth the effort.

In this major version release (V2.x), Cloth & Hair now provides a native interface to both Poser 9+ / Pro 2012+ and DAZ Studio 4.10+. We've added NVidia GPU support, clothing and hair import 'assistants', simple body dynamics (e.g. butt and breast jiggle), inflatables, improved wind simulation, performance optimizations, enhanced material control features, various bug fixes, and a bunch of interface tweaks and features, all resulting in an even more powerful toolkit at your fingertips. VWD also (still) interfaces with DAZ Carrara 8.5 using the great 'VWD to Carrara Bridge' 3rd party add-on product*.

This document should serve three key functions:

- Providing an overview of the program's usage, functionality, and features.
- Describing the specific program controls and functions.
- Demonstrating some mechanical usage scenarios. (A VWD **Usage Guide** with practical examples is being produced and distributed along with this guide).

We're working to make the operation of VWD self-explanatory, but it simply isn't there yet. Simulation and intra-application interactions are complex and still require that each user develop a feeling for the flow and limitations of this simulation system before they can really leverage the available power. The addition of the new VWD import assistants will help users get a good start on their simulation setups, while continuing to allow the additional fine tuning controls that will lead to optimal results in each distinct project.

By using the included examples, one can get moving pretty quickly on simple simulations, but to really grasp and control the amazing power that's available to you in this tool, we **highly recommend** that you at least peruse this document before you dig in. The power is immense, but as is often said “*with great power comes great responsibility*”, and to truly master this tool, you will need to study it, use it, consider your results, and study it some more...

Before we start, we'd really like to thank our community of supportive users for their helpful and always-friendly feedback, especially on this recent V2.x beta release that we put out there before it was up to our own standards. It seemed like the right decision, as it helped to keep our DS 4.11 users (the DS 4.11 version update 'broke' the DS bridge) working until the real version was ready, but we're also sure that it took some extra energy to manage the changes and bugs that were in there. Thanks again, and we hope it was worth the wait.

So, let's do this!

Gérald and Dan

**Philemot's VWD to Carrara bridge product is available at www.renderosity.com.*

Table of Contents

Welcome!.....	2
1 Introduction.....	5
1.1 What is VWD?.....	5
1.2 Why do we really need VWD's Cloth & Hair?.....	6
1.3 Common uses for VWD.....	7
2 System Requirements.....	8
3 Installing VWD's Cloth & Hair Program.....	10
3.1 Poser Installation.....	11
3.2 DAZ Studio Installation.....	13
3.2.1 Trouble-shooting (DAZ Studio Plugin):.....	16
3.3 The V1.x to V2.x Upgrade Utility.....	18
3.4 Testing Your Installation(s).....	20
4 Before We Really Start.....	21
4.1 VWD's Cloth & Hair... It's a way of thinking.....	21
4.2 VWD Paradigms and Workflow Ideas.....	21
4.3 A Typical Workflow Overview.....	24
4.4 Terms, Conventions, and Hints.....	25
4.4.1 VWD's Commonly Used Terms.....	25
4.4.2 Hints.....	28
5 The VWD Cloth & Hair Program Interface.....	34
5.1 The 'Scene import' Tab.....	35
5.1.1 The 'Cloth assistant' Tab.....	40
5.1.2 The 'Hair assistant' Tab.....	45
5.1.3 The 'Hair attachment' Tab.....	50
5.2 The 'Scene viewer' Tab.....	53
5.3 The 'Simulate' Tab.....	58
5.3.1 The 'Vertices interactions' sub-tab.....	68
5.3.2 The 'Simulation history' sub-tab.....	71
5.4 The 'Wind and Force Field' Tab.....	73
5.5 The 'Collision parameters' Tab.....	78
5.6 The 'Dynamic parameters' Tab.....	81
5.7 The 'Vertices selection' Tab.....	86
5.8 The 'Forces and Springs' Tab.....	94
5.9 The 'Utilities' Tab.....	104
5.10 The Mouse and Shortcuts Tab.....	112
5.10.1 Mouse and Mouse + Key Combinations:.....	113
5.10.2 Keyboard Shortcuts:.....	114
5.11 The 'Dynamic morphs' Tab (DAZ Studio only).....	115
5.12 The 'About' Tab.....	120
6 Using VWD: Concepts and Logistics.....	121
6.1 Concepts:.....	121

6.2 Logistics:	124
6.2.1 Backing-up and Restoring VWD Project Files:	124
6.2.2 Vertices and Selection tools:	129
7 VWD Utilities:	131
7.1 Utility: Animation conversion.exe:	131
8 Some Very Simple VWD Examples:	133
8.1 The Amazing Deflating Sphere! (a very basic dynamic-actor simulation):	134
8.2 A Plane and a Sphere! (a basic dynamic-actor and collision-item simulation):	144
9 Additional Resources:	148

1 Introduction

1.1 What is VWD?

At its core, VWD's Cloth & Hair (or usually *VWD* for short) is a physics-based motion simulator that allows users to treat static 3D-meshes like virtual fabric and hair, letting them drape, collide, and flow naturally over other figures and props. While simulating, these meshes can also be manually sculpted and 'helped' along as they drape. The results can then be saved for use in both still renders and animated 3D image sequences.

VWD's Cloth & Hair is also a mini 3D Swiss army knife that includes mesh welding, UV-map re-projection, simulation driven morph tool (esp. for body-jiggle), and mesh-partitioning tools that will let you take your kit-bashing to a whole new level.

VWD operates like a plugin, or 'helper program', being started from, and passing data between itself and Renderosity/Bondware's Poser application (formally sold by Smith-Micro), DAZ Inc's DAZ Studio application. It also works well with DAZ Inc's Carrara application via a great 3rd party bridge product*.

Fixed 3D-mesh hair and clothes, like dresses and pants, that were once 'frozen' polygon models can now be virtually draped onto other 3D-mesh figures using this simulation tool. They can also be tugged, wind-blown, gravity pulled, and otherwise manipulated in various ways, producing realistic and/or interesting results. These meshes can even be sliced, separated, and welded together from right within your VWD workflow.

This tool-set works on virtually any 3D polygon mesh, letting you drape the usual clothes and hair onto figures, but blankets can also be draped over beds, flags flown in the wind, seaweed undulates 'under water', and any other ideas that might occur to you in the realm of 3D-mesh objects.

The typical workflow goes something like this:

- First, you construct a normal 3D scene in one of the supported host-applications like DAZ Studio, Poser, or Carrara*.
- Then you process the 'hard' and 'soft' scene elements through the VWD Cloth & Hair simulation engine to generate realistic fabric and hair interactions against the various scene elements adding the influences of forces like gravity, wind, and friction,.
- After VWD Cloth & Hair returns those simulation results to the host-application, you render out the results using your scene's lighting and cameras.

The good news for users, is the already familiar scene preparation steps (layout, lights, camera angles, animations, etc.) don't really change at all. Nor does the output rendering process change in your host-application. The only change to your current workflow occurs when you use VWD to apply the natural draping and motion effects to your scene elements - after you've set your scene up, and just before you are ready to render the final results.

**Philemot's VWD to Carrara bridge product is available at www.renderosity.com.*

1.2 Why do we really need VWD's Cloth & Hair?



An 'autofit' dress 'following' a simple pose



The same dress draped dynamically

To generate better, or otherwise impossible fabric and hair effects in your 3D art, games, and animations...

Pre-defined shape adjustments (morphs) that try to make static 3D-mesh clothes and hair look like they're in motion may look compelling in their frozen state, but that's usually the only place they look good, and that may not be exactly what you want in *your* scene.

And, because most outfits and hair are usually designed to only fit specific figures (e.g. Poser or DAZ characters), and they are usually frozen in one of the few positions (poses) that the model-creator designed for them. This usually turns out to be 'OK' for most 3D scene-makers, but usually produces very stiff and unnatural results, especially in complex and interesting scenes or animations. Additionally, the automatic figure-following tools that try to automatically follow the figure's bone motions, poses, and animations simply can't produce anything like a physically modeled clothing drape sequence - not in static scenes, and especially not in animations.

VWD's Cloth & Hair lets you create scenes where the dynamic-actors (clothes, hair, flags, blankets, etc.) are freed from their designer's frozen constraints, and actually become soft and flexible, responding to both the simulated forces (gravity, wind, inflation, etc.) and the hard collision-items (bodies, sofas, beds, window-frames, etc.) in your 3D-worlds. The results are physically correct and look incredible with natural flowing motion, and realism.

Anyone who has tried to use modeling tools or 'deformer' utilities to produce natural looking fabric folds, flowing cloth, or hair effects on a 3D-mesh will immediately appreciate how much time and work this tool will save them, and love how easy it is to generate very natural and accurate results. To anyone who has taken the next step and tried to realistically animate these kinds of motions manually (flags in the wind or hair on a running figure, etc.), this tool is priceless.

Of course there are all sorts of things you can do with this sort of simulation engine ***that have nothing to do with clothes or figures*** – like hanging curtains, special-effects, water, terrain creation, explosions (I'll do a video someday), and simply fun and clever uses of mesh-dynamics in your 3D explorations.

1.3 Common uses for VWD

There are three primary approaches to using 3D assets in media today: Still image renders (snapshots), animation sequences (films), and 3D-Mesh manipulations for modelers, game-designers, and content producers. VWD's Cloth & Hair supports all of these production workflows:

Static: Creating 'still' images, like from a typical camera - where you use this tool to get the cloth and hair (meshes) just the way you want, and you render this resulting 'snapshot' in time. The 'static' mode is only static in the sense that your goal at the end of a simulation session is a perfect frame, frozen in 3D-space, that can be saved and rendered in your favorite host-application as a single frame scene image.

Dynamic: Creating animated image sequences, like a video camera - in this workflow, an animator produces a scene where figures and/or props step through an animated motion sequence, processed through the VWD simulations, using gravity, inertia, wind, and collisions to produce the flowing cloth and hair output frames as they interact with each other and nearby scene elements. These frames are then rendered as image sequences for use in video production.

Content and Modeling: For content producers, game-designers, and modelers, VWD is simply a mesh manipulation tool that uses simulations algorithms rather than modeling tools. Anyone who knows how to create morphs from a base OBJ file and one or more manipulated versions of that OBJ file can use VWD to create traditional drape and fold morphs in their cloth-based products, and export them for use within more involved workflows.

So when you're building scenes using arbitrary mesh items that may need to be naturally fitted to, or interact with, various items and figures in an underlying 3D-world, this might be a good place to start. Animators can also use single-frame outputs as static resources for these same purposes.

Whether you wish to create a simple image, put together a story-book or graphic novel, or produce a full-length animated masterpiece, if natural looking clothes and hair are important to your vision or modeling needs, we think you'll find plenty of value in VWD's Cloth & Hair. But enough talk, let's get something done here.

2 System Requirements

Operating System: VWD's Cloth And Hair is a Microsoft Windows-only application. It is successfully running on Windows 7, 8, and 10, although it may run on other variants of those operating system families.

Host Applications: VWD's Cloth And Hair V2.x run with built-in support for Poser 9/Pro 2012 or newer, and DAZ Studio v 4.10 or newer. Using the well-regarded VWD to Carrara Bridge product from renderosity.com and a Windows-based Python 2.7 installation, VWD 2.x also works with DAZ Carrara 8.5.

Hardware and Performance: VWD's Cloth & Hair capabilities are naturally constrained by the computing resources available to the simulation engine. All resource demands are driven by both the polygon counts within a simulation, and the interactions between active simulation elements. Mesh subdivision and hi-poly figures will have a very real impact on your simulations, but sparse meshes will not simulate well, so it is always a balance. In VWD's Cloth & Hair, textures, lights, cameras, and other non-mesh scene elements do not affect the simulation.

More memory and CPU/GPU (NVIDIA) processing power will always improve the simulation setup and execution experience. Once the simulations are complete, the hardware resources that are needed to actually render a given scene are completely dependent upon the complexity of that scene. These two workflow processes never run at the same time – After building the 3D scene, run the simulation(s) to help create the scene dynamics, then render the resulting scene.

For smaller scenes with a lower polygon 3D-mesh figures and clothing items and/or hair, we recommend *at least*:

- Intel I5 or better, running at least 3Ghz (or AMD equivalent)
- 8 Gigabytes of memory
- An extra 500 Gigabytes of available disk for the various ***Cloth & Hair*** working files, in addition to having space for your own final renders!

For a 'good' mid-range system, something like:

- Intel I7 quad-core system, with hyperthreading to 8+), running at least 3Ghz (or AMD equivalent)
- 16 Gigabytes of memory (32 is better)
- NVIDIA GeForce 700 series or better, if you choose to use VWD's Cloth & Hair GPU functions.
- A Solid-State-Drive for the ***Cloth & Hair*** installation folder, which is where many of the internal simulation files are written and referenced.
- (disk space for your rendering resources and outputs depend on the kinds of projects you produce)

For simulations involving larger 3D-meshes with complex and high-speed interactions the sky is the limit as far as hardware is concerned. As a frame-of-reference point, our program developer uses:

- Intel XEON 32-core processor
- NVIDIA 1080ti series GPU card
- 32 Gigabytes of System RAM

- SSD and traditional SATA drives on High-Speed controllers
- Terabytes of backed-up drive space...

And of course all of these specifications are rough estimates as the true performance of any system depends on the specific tasks being thrown at it!

3 Installing VWD's Cloth & Hair Program

The installation of VWD's Cloth & Hair product is done manually by copying files from the installation zipfile to your preferred locations. This process is slightly different for each host-application (Poser/DAZ Studio), but the ideas are the same.

With the newly added support for DAZ Studio (v4.10+), and the continued support for Renderosity/Bondware's Poser (V9+/Pro-2012+), here are some notes of interest:

- When running VWD V2.x version with DAZ Studio, a plugin module (DLL) must be present and loaded during the DAZ Studio start-up. Poser uses its built-in Python interpreter to interact with VWD. The mentioned Carrara bridge also depends upon its own plugin and Python interpreter.
- Multiple versions of VWD's Cloth and Hair can co-exist in both the DAZ Studio and Poser environments. (E.g. **C:\VWD1**, **C:\VWD2**, ...) so long as your start-up scripts are configured to use your chosen version.
- Both Poser and DAZ Studio can leverage the same VWD Cloth & Hair program installation, but two copies *can* be installed - one for each - with little expense in disk space and no known side-effects.
- If you are using VWD's version 1.x with of DAZ Studio version 4.10 (with the VWD to DAZ Studio Bridge product), the VWD V2.x and Bridge product plugins should co-exist without problem.

Note: The DAZ Studio 4.11 upgrade included internal changes that broke the VWD to DAZ Studio Bridge, so there's no need to have that product's plugins installed on DAZ Studio versions 4.11 or newer.

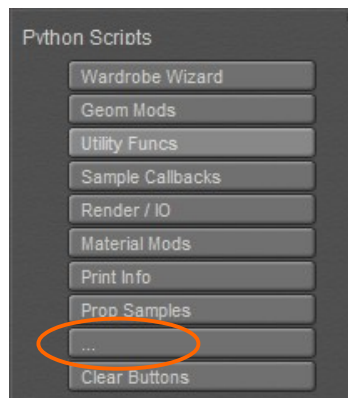
- The (sometimes large) simulation/animation files are generated and stored in the VWD Cloth & Hair core installation directory. This means that both space and access speed may be impacted by your choice of installation locations. In my case, I prefer to install my VWD Cloth & Hair core programs on my second drive, a faster solid state disk (SSD), e.g. **D:\VWD** for an added performance boost.
- Users that purchase the VWD 2.x Upgrade product from the original Renderosity brokered VWD V1.x product will receive an upgrade extraction tool that will create the normal installation package after finding the original Version 1.x **VWDClothAndHair.exe** file. After that the normal installation procedures will apply.

We'll cover the full installation process for both products next.

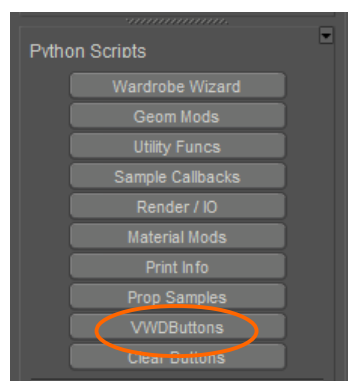
3.1 Poser Installation

Install the core VWD Cloth & Hair program and documentation:

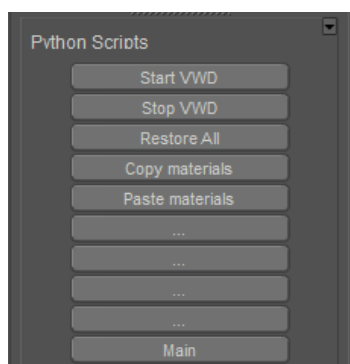
- Extract the VWD Cloth & Hair program files from your installation zipfile.
- From the resulting zipfile **...\installation** folder, copy or move the **VWD** folder to a location of your liking.
- While you may prefer to install these program elements in a traditional Poser-like location (e.g. a 'runtime' folder location like **Runtime\Python\poserScripts**), it can go most anywhere, and we recommend installing that folder in a low-level drive directory like **C:\VWD**, allowing quick access, and preventing potential 'path' issues later.
- Launch Poser, and from the main Poser interface, and look for the 'Python scripts' menu.



- Find the '...' button near the bottom (or click on an expendable button while pressing the **Alt** key, and it will clear that function, clearing a new '...' button), and click on it (or any available '...' button...)
- In the pop-up dialog, browse to your VWD installation directory (e.g. **C:\VWD**), and select the **VWDButtons.py** file in the installation directory.
- The '...' button should now change to **VWDButtons** :



Click on the new **VWDButtons** button, and you will see the new **VWD** control buttons in a standalone menu:



The VWD menu commands:

- **Start VWD** : launches the program interface on the right of the screen.
- **Stop VWD** : ends an active relationship between VWD and Poser. This button will, in theory, never be used, because the loop will stop automatically when VWD stops. However, sometimes (very rarely), it happens that the communication becomes locked. This button is there to stop the Python script.
- **Restore All** : clears the VWD-created scene actors - (**xxx_VWD**) from the Poser interface and restores the original scene items to their visible state.
- **Copy materials** : saves the materials of a selected scene item. This is used to restore materials to the VWD scene items (e.g. MyHair_VWD), because when a scene is reloaded, the _VWD actors have no materials.
- **Paste materials** : applies the currently 'Copied' materials to the selected scene actors. This is used to restore materials to the VWD actors (e.g. MyHair_VWD) when a scene is reloaded and the _VWD actors have no materials.
- **Main** : restores the default python script buttons to the Python Scripts interface in the Poser interface.

Your VWD installation for Poser is now complete. We'll detail the program startup, features and usage in later sections.

Note: The program documentation and usage guides are located in the main VWD program's **Documentation** folder. (e.g. **C:\VWD\Documentation**). Version-specific notes and issues are also located in the main VWD program's **_V1.2.333.4444** directory (where the version number is real, e.g. **C:\VWD_V2.1.892.6290**).

3.2 DAZ Studio Installation

First, install the core VWD Cloth & Hair program:

- From your same installation zipfile...
- From the zipfile's **...\installation** folder, copy the **VWD** folder to a location of your liking. We highly recommend installing that folder in a hard-drive root folder like

C:\VWD (allows for quick access, and may prevent 'path' issues.)

- (Remember the location you've chosen... We'll need to browse there a bit later.)

Second, install the appropriate 32/64 bit VWD plugin(s) into your DAZ Studio program **plugins** folder(s) (not your content folders!):

- Extract the VWD Cloth & Hair program files from your installation zipfile.
- From the zipfile's **...\installation\VWD\DAZ_Studio_Files\Program_4.10+\plugins** folder, find the correct version of your **VWDEExchange_x64.dll** or **VWDEExchange_x32.dll** plugin file (64 or 32 bit):

...\installation\Program_4.10+\plugins\VWDEExchange_x64.dll on 64bit systems

...\installation\Program_4.10+\plugins\VWDEExchange_x32.dll on 32bit systems

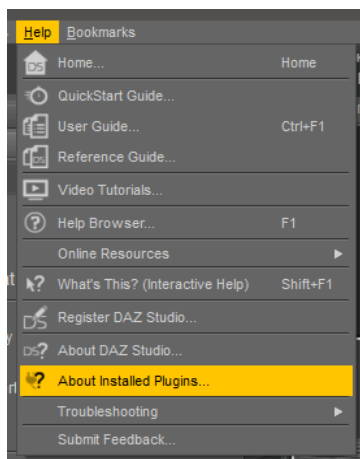
- Copy each to the proper DAZ Studio program's **plugins** folder.

typically **C:\Program Files\DAZ 3D\Studio4\plugins** on 64bit systems

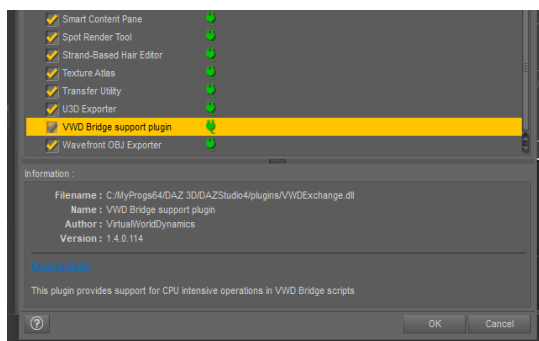
typically **C:\Program Files (x86)\DAZ 3D\Studio4\plugins** on 32bit systems

Important: If you have installed and/or used any of the VWD 2.x beta versions that required the **VWDEExchange.dll** files (note the subtle name difference!) in the **plugins** directory, please remove these files from your 32bit and/or 64bit DAZ Studio program's plugins directories as *they will conflict*.

If you restart DAZ Studio and this plugin is installed properly, in the main menu, the **Help->About Installed Programs...** menu...



find the VWD plugin in the list (near the bottom), which should look something like:



(See the troubleshooting section if you don't see the expected results - you may have a conflict with another similar plugin).

Lastly, install the User-Facing DAZ Studio Scripts and Documentation:

- From the zipfile's ...**installation\My DAZ 3D Library** folder, find the **Scripts** and **ReadMe's** folders.
- Open (any one of) your DAZ Studio content libraries, and copy these two folders (**Scripts** and **ReadMe's**) into your preferred DS content library folders of the same name.

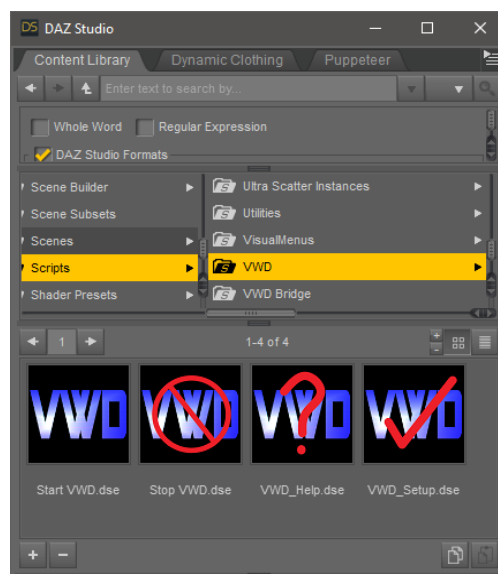
typically **C:\Users\Public\Documents\My DAZ 3D Library**

Even if these target directories (**Scripts / ReadMe's**) already exists, this copy action will merge the folder contents, which is correct.

- **Note:** The program documentation and usage guides are located in the main VWD program's **Documentation** folder. (e.g. **C:\VWD\Documentation**) and version-specific notes and issues are also located in the main VWD program's **_V1.2.333.4444** directory (where the version number is real, e.g. **C:\VWD_V2.1.892.6290**).

- If you restart DAZ Studio, or refresh the proper content library, you will see the new VWD scripts in (one of) the DAZ Content libraries, like:

My DAZ 3D Library : Scripts : VWD : Cloth and Hair (V2.x) content library folder:



The scripts:

- ***Start VWD*** : This script launches the VWD Cloth & Hair main program. The main VWD Cloth & Hair program interface will appear on the right edge of the screen after a few moments. It will also prompt you to 'find' the main VWD program if its location is not known during the start-up attempt. This should only occur once.
- ***VWD_Setup*** : This script lets you (re)configure the main ***Start VWD*** script to 'find' the VWD Cloth & Hair core programs on your system, and 'tells' the ***Start VWD*** script where they are. You should only need to run this one time after your installation, but you can run it whenever needed.
- ***Stop VWD*** : This script signals the ***Start VWD*** script program to stop interacting with the VWD program and exit. Because the ***Start VWD*** script normally exits when the user finishes running their VWD program simulations, this script only needs to be run when the communication between the VWD program and DAZ Studio gets out of sync – especially when the VWD simulation program exits abnormally.
- ***VWD_Help***: This script opens up the VWD **Documentation** folder in a Windows Browser window, allowing you to view the available VWD documentation. This **Documentation** folder is found in the main VWD program installation folder, so this script will prompt for the user to find that directory if it hasn't already been located in any of the other VWD activities.

The first time you run ***Start VWD***, ***VWD_Setup***, or ***VWD_Help***, you will be prompted to locate your VWD program directory with a Windows file-system browser, where you must find and select your **VWDClothAndHair.exe** executable program (in your newly installed VWD program folder - e.g. **C:\VWD**)

3.2.1 Trouble-shooting (DAZ Studio Plugin):

If the DAZ Studio *help->about installed plugins...* dialog doesn't show the “*VWD's Cloth & Hair Bridge*” plugin as being present and active (green plug) down near the bottom of the list, none of the DAZ Studio VWD activities will work. You'll have to track down that plugin issue before you can do much else.

DAZ Studio plugin conflicts:

- **Users of DAZ Studio 4.11 and above:** be sure to check-for *and remove* both the **VWDExchange.dll** and **VWDSupport.dll** plugins from the DAZ Studio program's **../plugins/** directory!

(e.g. **C:\Program Files\DAZ 3D\DAZStudio4\plugins\VWDExchange.dll** and/or **C:\Program Files\DAZ 3D\DAZStudio4\plugins\VWDSupport.dll** – just remove them.).

Neither version is useful since DS 4.11 was released and will only cause conflicts if either of those plugins is present. Depending on the architecture folder, you should only have the **VWDExchange_x64.dll** and **VWDExchange_x32.dll** plugins present in those directories.

- **Users of DAZ Studio 4.10:** You should be able have the current V2.x release co-exist with *philemot's* great “*VWD to DS Bridge*” product for VWD V1.x. That product's plugin is called **VWDSupport.dll** and should be able to operate along with VWD's 2.x release (NOT the betas!) plugins should you wish to have both versions available.

Note: this DS bridge product will not work beyond DAZ Studio Version 4.10 due to updates made to their 4.11 and newer systems.

- Users of any of VWD's Cloth and Hair V2.x beta releases should simply remove any DLLs called **VWDExchange.dll** from their DAZ Studio plugins folders (both 32 and 64 bit), as these plugins are now obsolete and likely to cause conflicts.
- **Users of DAZ Studio 4.9 and below:** You must use the DS Bridge product AddOn to access VWD 1.x (only). VWD 2.x does not work with DAZ Studio versions below 4.10.

DAZ Studio users, Resetting the current path to the VWD main program directory:

In some cases, you may wish to reset your main VWD program directory and have the DS-based VWD scripts prompt you for the current location. From within DAZ Studio's Content library, in the VWD scripts folder, you can remove or manually edit the small text-file containing the path to the main VWD installation directory folder:

- Browse to your DS Content Library folder that contains the VWD Scripts.
- In DS, right-click on the window background to use the *Browse to folder* option.
- In that file browser window, remove (or edit) the **VWDDir** file (or **VWDDir.txt** if you show full filenames in Windows). This file contains the path to your VWD program directory and will be recreated the next time you use the **Start VWD** script.

- Optional: You can choose this same **VWDdir.txt** file and manually edit the path if you wish (experts only!). Note that the path separators are backwards from the usual default Windows path separators. It should look something like: **C:/VWD** unless you've installed VWD in an alternate location.

3.3 The V1.x to V2.x Upgrade Utility

Users that purchase the VWD 2.x upgrade product to upgrade their original renderosity.com V1.x product receive an upgrade extraction utility program that will generate a complete installation zipfile package after finding and verifying the original Version 1.x (specifically V1.1.522.3630) **VWDClothAndHair.exe** program file. After that zipfile package is extracted, the normal VWD installation instructions will apply (full installation instructions are included).

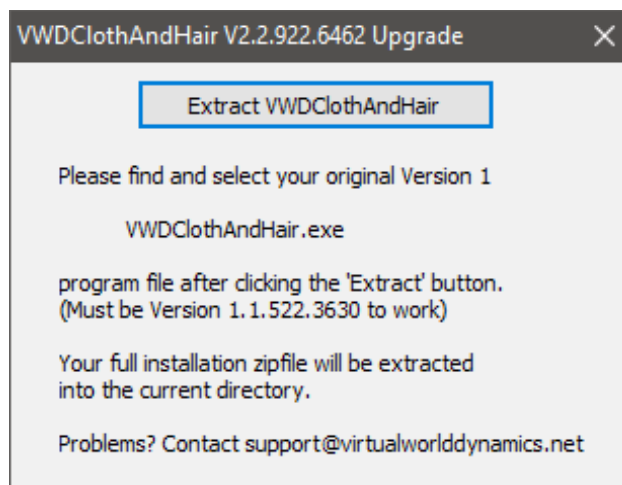
This upgrade utility simply verifies that you have a legitimate copy of the original VWD Cloth and Hair executable program (**VWDClothAndHair.exe**) on your machine. This V1.x executable program file does *not* have to be installed or located in any particular folder, it simply has to be available to the upgrade utility program for verification of ownership. Upon starting the extraction program, you will use a file-manager browser to locate and select this particular file for verification.

To use this extraction tool:

- Download the **VWDClothAndHair_V2.2.924.6464_upgrade.zip** zipfile from your download source (your renderosity.com or VWD product library). Extract the content to a local working folder (e.g. your Windows Desktop should be fine). You should now have a new folder (**VWDClothAndHair_V2.2.924.6464_upgrade**) containing the upgrade utility and some basic instructions and release information.
- In the newly extracted **VWDClothAndHair_V2.2.924.6464_upgrade** folder, locate the **VWDClothAndHair_V2.2.924.6464_upgrade.exe** program.

Note: Optionally, you can move or copy that extraction program to a workspace folder of your choice – somewhere you can extract and locate the resulting VWD installation zipfile (e.g. your Windows Desktop or any working folder). The extracted zipfile should only be around 25 Megabytes in size for a complete VWD V2.x installation, smaller if you have a partial update version.

- In your working space, run the **VWDClothAndHair_V2.2.924.6464_upgrade.exe** utility program to open a window that looks *something* like:



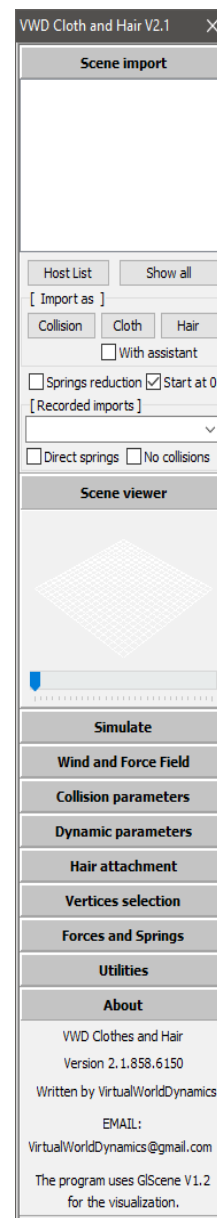
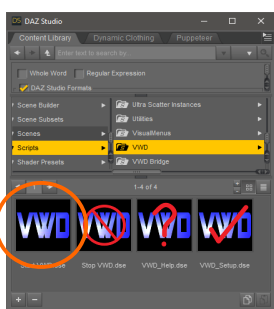
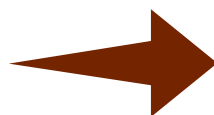
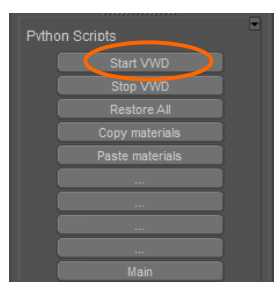
- Click on the **Extract VWDClothAndHair** button, and use the browser to find your **Version V1.1.522.3630** copy of **VWDClothAndHair.exe** and select it from that browser.

- If the selected VWD program file is the proper version, a new zipfile for the current version (**VWDClothAndHair V2.2.924.6464 Upgrade.zip**) will be created in the local folder where you started the upgrade utility program, and you should be ready to start the 'official' installation process that precedes this section.

Note: It's probably a good idea to save a copy of the newly extracted installation zipfile in your favorite content backup location for future use and reference.

3.4 Testing Your Installation(s)

To start the VWD Cloth & Hair main program, find your newly installed host-application startup script (***Start VWD***), and run it.



If the program elements are installed properly, you should get a tall thin VWD program interface on the right side of your screen (see above). The ***About*** tab at the bottom of the interface should show the version you are running (the contents will differ from the above screenshot version).

4 Before We Really Start...

4.1 VWD's Cloth & Hair... It's a way of thinking.

First of all, from here on out, even though the formal name of this product is *VWD's Cloth & Hair*, let's make life easier and call it '**VWD**' for short. It's simply what we call it everywhere else (internal messages, forums, etc.), so let's just go with it. Yeah, VWD...

As far as using it, well, the interface is a little bit unorthodox, but not only does VWD grow on you, but there are some really clever interface features that you start to wish were in your other programs. We're still working on interface enhancements that will help simplify some of the complexities of simulations, but getting the internal simulation engines right is still our highest priority.

The simulation workflow requires what some users might consider a new way of thinking - to help separate and manage the necessary preparation, execution, and adjustment stages available (and required) in a typical simulation process. But, it is not so much a radical difference from good organizational thinking, as it is an identification, appreciation, and division of the distinct stages that make up any *simulation* workflow. Add the all-important idea of 'springs', attachments, and forces to the mix, and it *will* make you think a bit. But we think the results are worth it...

Really getting the most out of VWD also requires that you let go of the usual "set-pose then render, repeat..." modal way of thinking. VWD certainly lets you work that way, but it also lets you sculpt your dynamic-actor meshes in real-time, while it's simulating! Consider turning on wind for a short while to get most of the hair where you want it, then dragging wisps around for 'the right look', then sending the result to the host application, saving that version, then back in VWD, switching the wind direction around, and watching until the hair swirls back until another completely different look is achieved, saving that, then ...

It's also good to know that there are a few tools within VWD that allow you to manipulate 3D-meshes that have nothing to do with simulation. Some of these tools simply let you load a mesh, run some actions on that mesh that will result in new files on your system, or elements in your host-application scene. These utilities have their own mini-workflows and have little or nothing to do with the usual workflow of VWD's simulation functions. Odd, but sometimes very handy.

4.2 VWD Paradigms and Workflow Ideas

Your stuff - The hard and the soft...

In all cases, using VWD requires that you take all of your scene's relevant 'hard things' - we call them *collision-items* - like figures, furniture, props, etc., and import them into VWD from your host-application. You then import *one* of your 'soft things' - we call it the *dynamic-actor* - like a shirt, a sheet, or a hair item. (Occasionally we'll call it the *dynamic-cloth-actor*, or *dynamic-hair-actor* for clarity...)

Important idea!: As odd as it sounds, when you import a soft sexy looking animated 3D female figure to drape your 3D clothes onto, she's actually the 'hard' collision-item in the simulation!!! Even in her sultry or warrior-like animations, she might as well be made of concrete as far as her simulating hair or clothes are concerned! This is sometimes hard to get your mind around, but it makes sense if you take the perspective of the simulation engine, and consider that the hair and clothes are soft and flowing *against* the 'harder' stuff (her).

Once imported and configured with some starting fabric and hair characteristics or properties, you can fine tune your VWD session's items by selecting various mesh areas or sections of your imported items and applying the appropriate material properties for precise stiffness, stretch, collision, and self-interaction effects that will help you produce exactly the right look and feel with your 3D-mesh items.

The forces...

In the Simulate tab there are adjustable forces like gravity, inertia, and air-resistance that you can adjust for realism or effect. There's also a tab dedicated to wind and force-fields that will let you insert both constant and varying forces into your simulations, both static and animated. Many of these forces can be adjusted during the simulation too!

The processing engine...

Next, you configure your simulation engine settings, both mechanical: GPU/CPU and processors, and virtual: intra-frame interpolation settings, etc.

The simulation...

You then run the simulation. In this phase, your imported dynamic-actor (cloth or hair) will become a flowing mesh, interacting with any-and-all collision-item(s) in the VWD scene. In a *static* scene (not animated), the dynamic hair or cloth should respond to the scene forces(like gravity and wind), and generally settle into the natural drape and final position after a while. In a *dynamic* scene (animated), the dynamic hair or cloth will also respond to the scene's forces and collision-item(s), and it will also interact with the changes occurring in each animated frame as it is requested from the host-application, one-at-a-time, responding to those changing collision-item influences as they occur.

Tweaking things...

Just when you believe the simulation configuration is 'done' and you're ready to sit back and watch..., even more of VWD's power becomes evident! At any point while the simulation is running, you can either pause it to make adjustments, or you can just let it run, saving the results as they complete. In the *static* mode, you can use your mouse to manually tug and pull at parts of your dynamic-actor that may 'need some help' in any way you see fit, **while the simulation engine is still running**. You can then pause, adjust, and resume the simulation repeatedly, or stop and save the result when it's exactly 'right'...

Sometimes this process gets interesting when your 3D-mesh items have been designed without welds, and parts fall off during the simulation. Or, when a belt is supposed to stay on a pair of pants. Or, when hair strands are not welded to the skull-cap and the hair falls to the ground... VWD is designed to help manage these scenarios, but... you have to help it do so - because each mesh is unique in its design and density variability (and there are a *lot* of odd meshes out there!).

What's amazing is rather than having to exit VWD pull your 3D-mesh into a modelling program and make welds and similar fixes, most of these adjustments can be made from within VWD, just before the simulation phase begins. Attachments, collision, scaling, softness - can all be set on the entire mesh or small parts and accessories like buttons or ribbons.

Rendering...

This is done back in your host-application, but there are a few residual things that using VWD sometimes adds to this last step. In some cases, VWD may need to be used to re-apply textures to its own simulation copy, and it may be necessary to 'reload' all of the simulations in a scene, to re-coordinate a multi-item animation for the rendering sequence. These steps aren't that involved or complicated, but they add an extra step to that last pre-rendering step, and bear mentioning here.

Generally...

The biggest challenge for users of any simulation tool is getting a feel for the relationship between the various 3D-mesh item characteristics (e.g. polygon density, layout, and construction), the simulation settings (stiffness, stretch, attachments, forces, collisions, and springs, springs, and more springs...) and how they relate to each other, such that the desired effect(s) can be realized.

Because each 3D-mesh item is unique in its construction (polygon density, seams, welds, etc.), so are the respective 'correct' settings for each! Consistent meshes will promote consistent configurations and results. Varied meshes will take more effort, but VWD has the capacity to handle those as well, and that capability may be just what's needed to make your project(s) pop. Clothing items from the Poser cloth-room and DAZ's d-Force meshes are fairly dense and properly welded for simulations, so there are plenty of well-made clothing meshes to work with. Even the strand-based hair products that are hitting the market can be finessed to work well with VWD!

So, yes, if you're reading this correctly, you're probably starting to realize that this configuration process becomes an art in itself, an adventure, like a pallet of paints and a blank canvas... and may sometimes require a good number of iterations and experiments before getting your results 'just right'.

But while 'just right' is sometimes amazing, and always fascinating to watch as the simulations run, there's more... you can pause the thing, and get in there and tug and pull on the fabrics and hair with your mouse!, all while the simulation is running! How cool is that?! Really! Once you get things set up and running, then start to tug at 'living' cloth and hair, you'll be hooked!, and your results will almost always impress!

4.3 A Typical Workflow Overview

The typical workflow used to manage the scene setup in your host-application, the VWD simulations, and rendering processes back in your host-application, goes something like:

1. In your usual way, build a scene in your host-application - DAZ Studio, Poser, (or Carrara - with the Add-On Bridge product). If desired, set up any animations in the scene.
2. Start VWD from your host-application.
3. Within the VWD session you will:
 - Define each session item's role in the simulation *as you import it* (one or more collision-items, and a single cloth or hair dynamic-actor). While anything in the scene can be imported, you only need to import items that will be actively involved in the simulation: figures, clothes, hair, props, furniture, etc.
 - Apply additional material properties and characteristics (stiffness, friction, stretch, etc.) to any of the imported elements.
 - Enable and adjust any forces that will impact the scene/simulation (e.g. wind/gravity/inertia, etc.).
 - Configure the simulation settings (CPU/GPU, processors, etc.)
 - Run the simulation(s), pausing to adjust and tug at things if you like
 - Reset/adjust/simulate until you achieve the desired results.
 - Send the results back to the host program.
4. Repeat the above cycle step for each dynamic element in your scene (e.g. hair, then pants, then shirt, then sheets, etc.) until all of the dynamic elements have been simulated to your liking.
5. Back in your host-application, reload and (re)activate any/all of the VWD simulation animations (if needed) and re-apply any textures (if needed), then preview and finally render your image or animation sequence.

Reminder: Each active VWD simulation session can only have *one* active dynamic-actor (hair or cloth) item being simulated 'against' one or more collision-item(s) at a time.

Hint: To produce a scene with multiple simulated dynamic-actors (e.g. a scene with hair, a dress, blankets, etc.), each dynamic-actor will need to be simulated in its own VWD session, many of which will become the underlying basis for the next simulation session, now acting as collision-items - in a sort of cumulative 'layering' model. This is exactly how layered clothes are handled (e.g. a coat over a shirt, over a figure, etc.). This process will be covered later in this document.

There are some basic examples of the setup and use of the VWD tools near the end of this document. The above information is intended to generate a 'feel' for the VWD workflow before you start using it.

4.4 Terms, Conventions, and Hints

These are a few of the things we think everyone should know before using VWD.

4.4.1 VWD's Commonly Used Terms

Simulation has a language of its own, and we have terms we've also come to rely upon to exchange ideas on the use and control of VWD's Cloth & Hair. We can't emphasize enough how much more you'll enjoy the learning and use of this tool if you peruse and become familiar with our own way of describing and managing some of these ideas that permeate this tool and simulation environment. There's just no shortcut to taking a few minutes to sneak these terms into your new 3D-simulation domain.

- **Virtual World Dynamics (VWD)** - our company. Also the nickname for our *Cloth & Hair* product...
- **VWD's Cloth & Hair (VWD)** - Our flagship product. Allows users to convert virtual frozen clothes and hair into virtual flowing gowns and lockes. Everyone just calls it *VWD* for short...
- **dynamic-actor** - Our term for the one item in a simulation session that 'flows', interacting with the things in the scene that aren't dynamic. Note that there can only be one dynamic-actor in a VWD session. For each dynamic element in a complex 3D scene, a new VWD simulation session must be run. In such cases, many of the early dynamic-actors will take on non-dynamic but animated roles in later sessions.
- **collision-item** - Our term for the fixed or animated 'hard' thing(s) in a VWD session. There can be many, and they can either be still or pre-animated. A window frame may be a collision-item for a set of dynamic-actor curtains. More commonly, a walking female figure collision-item may be wearing a dynamic-actor dress that reacts to her body's animated motion and gravity.
- **xxxx_VWD** - This is how we describe an arbitrary VWD working item that is added to your scene when an item is imported into VWD as a dynamic-actor (cloth or hair), where **xxxx** is the name of the original scene item. VWD hides the original item in your scene and does its magic on this stand-in unrigged mesh-duplicate of that item. It is generally OK to remove these **xxxx_VWD** scene items when you are resetting your scene and any simulations, but if you are actively working on a scene/simulation, this item is the simulation target and where the results are applied, so removing it won't damage anything, but will remove your work/results... e.g. If you have a scene item called **My_Dress**, and you import it as a dynamic cloth actor into VWD, a new item will appear in your scene called **My_Dress_VWD**.
- **Selection** - Much like a word-processor, where you select words before you cut and paste them... After some base settings and properties are assigned to the actors and items being imported, almost all of the fine-tuned properties and characteristics of the VWD sessions fabric, hair, and collision-items are assigned to 'selected' mesh vertices and areas.

This process will be very natural to those that do a lot of 3D modeling but it can be pretty simple for those that don't model as well. For anyone new to VWD, it will take some playing with the available selection tools to appreciate how to get *just the vertices you want* for a particular effect. Worth mentioning here, is that the mesh navigation and selection memory buffer tools are simple but amazingly powerful. Once you've gotten used to them, you will wish you had them available on all of your other mesh manipulation tools.

- **Soft-Selection** - This selection method extends the current selection in an outward direction, having a reduced influence as the distance increases from that current selection. It is used to soften transitions between various cloth or hair properties (e.g. very firm to very soft). It is not always available in all selection scenarios.
- **Springs** - These very important virtual connections between various vertices or mesh-points drive almost every behavior you see in VWD's simulations. They are stretchy, stiff, long, short, ... and *you* set them the way you want them - like it or not... Like paint on a canvas, you paint your stiffness and attachment properties onto selected areas of your 3D-meshes, and when you get it right, you'll watch in fascination as the simulation magic unfolds. Yes, if you master VWD, you'll be thinking in 'springs'...
- **by-neighborhood** - This is a method of scanning for nearby 3D-mesh vertices from a given location or current selection-set. When using the by-neighborhood method of selection scanning, only the distance between the relevant points matter, and where on the mesh(es) they reside is irrelevant. If the points are within the sphere of the specified distance radius, they are 'included', usually in a selection-set. In cases like pleats on a dress with zig-zag folds, or layers of hair meshes, points can be selected across 'air gaps', or across open space between meshes. While this is good for some selection scenarios, it can also be used together with the other vertex selection methods (e.g. by-extension described below).
- **by-extension** - This is a method of scanning for nearby 3D-mesh vertices from a given location or current selection-set. When using the by-extension method of selection scanning, all connections between vertices or mesh-points have to occur across contiguous edge-connections on the nearby mesh structure. This means springs will only be added to the selection when they are located across the same mesh surface by hopping from connected vertex to vertex within the given distance. While this is good for some selection scenarios, it can also be used together with the other vertex selection methods (e.g. by-neighborhood described below).
- **Inter-penetration** (poke-through) – This is when meshes intersect or overlap *through* each other. Also called poke-through. This occurs most often when collision is either disabled on or between meshes, or when meshes are moving quickly into each-other, or when mesh polygons are too big to calculate collision effectively.
- **Rigidity** - We use this word a lot in this program and documentation. It's our way of describing flexibility and softness. More springs and less softness make things more rigid, stiff, inflexible, 'untwisty', etc.... (it's actually pretty tough to articulate some of these properties, given the many uses we've attached to some of these words - e.g. soft music, a soft throw, soft fur, a soft impact, and soft jello, etc....)
- **Nail to** - Our current way of describing attaching or gluing things together. These things are nailed together using our virtual *springs* (see above) of various softness, or stretchiness. e.g. *Nail to collision*, is when you attach all or part of a dynamic-actor's mesh to a selected part of a collision-item - like dress straps to a figure's shoulders, etc.
- **DAZ Studio (DS), DAZ, DAZ Inc., DAZ Carrara, dForce** - These are the names and terms of a 3D image and animation rendering tool (DAZ Studio), and the company that produces it (DAZ Inc.). DAZ Inc. also produces 3D content for use in 3D workflows and sells them as a brokerage. Virtual World Dynamics acknowledges and respects all copyright and trademarks related to DAZ Inc. and their products. <https://www.daz3d.com>

- **Poser, Renderosity, Bondware** - These are the names related to a 3D image and animation rendering tool (Poser), and the company that produces it (Renderosity/Bondware Inc.). Renderosity is also a 3D model, software, and general 3D resource brokerage. Renderosity sells VWD's Cloth & Hair products, Poser 3D software and a variety of 3D products. Virtual World Dynamics acknowledges and respects all copyright and trademarks related to Renderosity/Bondware Inc. and their products.
<https://www.renderosity.com>
- **Windows** – This is the trademarked Microsoft Inc. operating system series (Windows XP/7/8.1/10, etc.) that VWD's Cloth & Hair runs on. Virtual World Dynamics acknowledges and respects all copyright and trademarks related to Microsoft Inc.

4.4.2 Hints

As you use any software tool and develop your own favorite workflows, you'll learn things as you go. Here are a couple of tips, techniques, and mechanical quirks that we have learned while using VWD in no particular order...

- **Release notes and Known Issues:** Please review the release notes that are included with your version in the version-specific release-notes that are included in each release package! There may be known issues (a button that is present but simply isn't working...) in the release that should be described in these notes and may save you some grief and time. These release notes should always be available in the core VWD directory that looks something like:

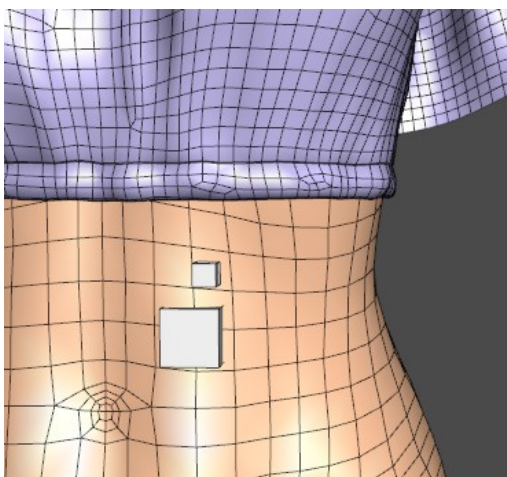
`C:\VWD\2.2.333.4444\VWD_V2.2.333.444_release_notes.txt`

(or wherever you installed the core VWD directory). These kinds of notes are specific to the latest releases, so it does not make good sense to include these transient issues in this core documentation.

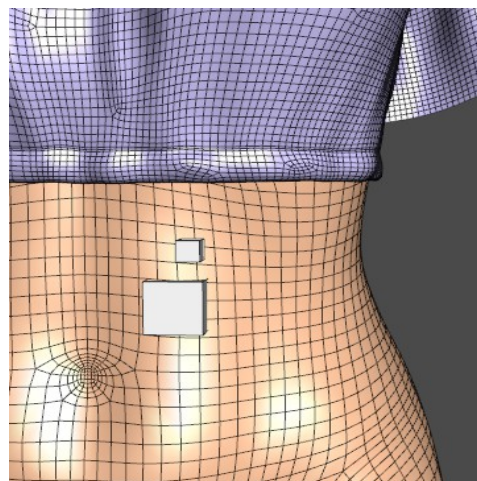
- **Only one dynamic-actor (simulated item) is allowed in each VWD session** - This is mentioned all over this document, but it is really important, so it's also here as a *hint*. There can be many items in a VWD simulation session, but the dynamic-actor is the only element in the scene that will be simulated during that session. Everything else will be available for the dynamic-actor to interact with (or against).

Note: To create an image with multiple simulated items (e.g. a simulated hair, simulated dress, and simulated blankets on a bed), you will need to run VWD at least three (3) times - once for each simulated item. This is true in both static and dynamic/animated scenes. Simulated items are layered in sequential sessions.

- **DS 'Smoothing modifier':** When using DAZ Studio, imported collision-items must have their smoothing function removed completely to work properly with the VWD simulation engine. If smoothing is enabled in DS on a clothing item that is animated by the host, it will probably 'explode' in VWD during the simulation. This smoothing can be re-added to the item, re-activated, and configured once the VWD simulations are completed and returned to DAZ Studio session.
- **Mesh density** – The polygon density of the various 3D-meshes has a *huge* effect on how that mesh will behave during the simulation processing! 'Sparse' meshes, or meshes that have large polygons, will be look blocky and often simply confuse or break the simulation engine. Dense meshes, or meshes with a large number of polygons will simulate beautifully, but more slowly, and react slowly to dynamic deformation adjustments. So, what are the 'best' polygon densities in your scene items to shoot for when using VWD? Here are some rough guidelines:



Genesis 8 Female and dforce top, both at their 'Base' resolutions. Probably OK for simulating in VWD.



Genesis 8 Female and dForce top at 'High' resolution (level 1 smoothing in DAZ Studio). Better for simulating in VWD, but slower.

While each host-application uses its own units of measure internally, the typical (above) DAZ Inc. figure 'open' surfaces (not faces, hands, etc.) seem to *average* around $\frac{3}{4}$ inch per polygon in base resolution in DAZ Studio units, and $\frac{1}{4}$ inch polygons in level 1 smoothed sub-division 'High Resolution' mode. In the images above, the larger cube in G8F's belly is one DAZ Studio inch per edge. The smaller cube is one DAZ Studio centimeter per edge. This will give you an idea how those figures and clothes meshes are designed and what a decent mesh range is for VWD. Using items with polygons that are much larger or smaller will diminish your VWD simulations in quality or time, respectively. Technically, this means good polygon density from a DAZ Studio frame or reference is around 9 to 16 polygons per square inch, or a bit less than 1 polygon per centimeter, but you have success with variations from this.

Hint: Like I do in the above images, I recommend taking a look at clothing items made for DAZ Studio's dynamic tools like dForce or dynamic-clothing-control, and/or Poser's Cloth-Room products. These are usually pretty close to the right density for VWD simulations. Bring one of these items into a new scene along with your own scene item(s), then set the viewport to one of the mesh viewing modes (in DS I prefer *hidden-line* or *wire-shaded*), and zoom around the items and compare the polygon sizes to the dynamic items and see if they are similar in mesh-density.

Note: In a given scene, collision-item meshes can be less dense and still be useful, but sparse meshes used in simulations and renders will show more apparent mesh structure (facets and corners) and may suffer from more poke-through issues (especially in animations).

- **Toggle Buttons** - I made up this term, which describes a bunch of clever VWD program buttons that have multiple functions, which can only be seen/activated (toggled...) by floating over them with your mouse cursor while having the Windows **Alt** or **Ctrl** key pressed.

Important: The text on any/all of these dual-mode "toggle buttons" is italicized in the VWD interface!!!

Most of these dual-use 'toggle buttons' are logical, in that the function of the default button is inverted when 'toggled' by the **Ctrl**-float-over action, but a few are just convenient and completely unrelated in their functions... All of these buttons are documented in the sections below, but it's *good to know that they exist* early in your VWD explorations!

One easy example to try is in the **Forces and springs** tab, the **Apply inflate** button: float over it with your mouse cursor (no buttons pressed), then press and hold the **Ctrl** key as you move the mouse slightly, and watch the button's caption change to **Remove inflate**! Obviously, if you click the button in this state, it will do as the caption indicates.

Note: Some of the buttons will also automatically change their purpose depending on the state of a VWD process. Most notably, in the **Simulate** tab, the **Start dynamic simulation** button will change to **Stop dynamic simulation** while running, then **Send animation to host** when the simulation is complete. Similarly, the **Start static simulation** will change to **Stop static simulation** (which is really more like a pause button).

- **There are multiple display modes** for the items in the VWD **Scene viewer** tab - Try pressing the 'a', 'h', and/or 'o' keys while the mouse is within the **Scene viewer** viewport. The active-item's display style will change! Press each button again (they are toggles) and the display will return to its previous mode. You can press any of these buttons at any time to help in your navigation and vertex selection efforts. Details are added below, but this is good to know early...
- **Reset/Fresh start in VWD** - To really get a fresh start with a VWD session and scene, you need to clean out any VWD items from your host-application scene, and clean out the VWD working files so you don't get any residual working files competing with your new session files. First, remove all of the files in your scene that have the **_VWD** suffix (just select and remove them), and then open VWD and first thing: run the **Delete all "Exchange" files** tool from the **Utilities** tab.

Important: Remember that **Delete all "Exchange" files** clears *all* of your available VWD animations from its animation file-cache, which removes all (past) results from all projects. Backup the Exchange folder if this may be a problem (see **Backing-up and Restoring VWD Project Files** for details). You may also wish to run the **Restore program parameters** from that same **Utilities** tab to ensure that some residual settings are being applied to your simulations without your knowledge or intention.

- **Consider animating your figures 'into' their final positions/poses** – Non-animators can use the new **Pose to animation** function (see section 5.9, 'The Utilities tab') to generate a short transition animation. Even for those of you who do not animate, you might consider running a very simple pose-to-pose animation from your base T/A-pose to your intended final pose and position. VWD can take a cloth or hair mesh from any configured pose/position and drape it to its final resting drape just fine, but you may find that you achieve better results by starting a scene with both the cloth/hair and figure in their as-shipped A-poses - fitted and stable. Starting from this stable point, set up a 3-5 second settling time (90-150 frames @30fps), then have the figure move into their final pose over a 5 second time-range, letting that pose settle for another 5 seconds at the end.

Not only will the clothes and hair fall more naturally as the animation progresses, but you can pause and help the dynamic-actor find its way to where you want it to go and settle. When you send the sequence of frames back to the host-application, you can render any of the resulting frames, rather than just the static frame that was produced in the static render mode.

- **Question:** What are these **xxxx_VWD** files that are created in my host-application scenes when I run a VWD session? (e.g. After I simulate my **CoolDress** scene item in VWD, there's a new **CoolDress_VWD** item in my scene tree and my **CoolDress** is hidden...).

These files are mesh duplicates of the original items that are then imported into VWD for the simulation session. These are the mesh-files that VWD actually works with, leaving your original scene items alone (other than to hide them after VWD successfully returns the new simulated version of the original item to your original scene).

If you decide that there's no remaining value to having these **xxxx_VWD** files in your scene, or you want to start a new VWD session from absolute scratch, you can remove these items from your scene and your scene will be just like you started, and VWD can be run again from scratch with no ill effects.

Note: Do not place these **xxxx_VWD** files in your DAZ Studio groups or hierarchies while you are in the process of generating simulation results. Only *after* you have completed all VWD simulations should you consider temporarily grouping these items so you can manipulate the groups as a unit. If you wish to re-apply the textures or animations to the **xxxx_VWD** items, they must be visible to the VWD program at the top of the DAZ Studio scene tree hierarchy. Once re-applied, you can re-group these items for further adjustments and rendering.

Also of related interest: These scene objects are un-rigged standalone duplicated meshes. In DAZ Studio, these meshes are stored right within the scene or scene-subset *.duf files, meaning that these saved scene-files may grow rather large if hi-poly items are being simulated in VWD. They can be exported as items using the scene-subset or export options of your host-application.

- **Switching between VWD and your host-application** - Changes can be made to your host-application's scene while VWD is still open, and by pressing the **Host List** button in the **Scene import** tab, you can access and import any new scene elements into VWD without having to restart your session.

Hint: While I tend to try to finalize my DS/Poser scenes before working in VWD, there might be times that I need a small collision or nail-to sphere in my VWD scene to help with a simulation effect. By quickly adding this to my DS/Poser scene and pulling it into my active VWD session, I can keep my workflow moving forward.

- **Avoid abrupt motions in animation sequences** - Be sure your animations are fairly smooth, especially if you are melding motions together in your host-application (e.g. using BVH files and aniblocks). Abrupt movements and jerkiness between frames will become very apparent in some of your VWD results, as the simulation engine senses, and accurately 'renders', these rough transitions. Upping the frame-rate may help in fast animations, and lowering the frame-rate may result in very rough simulation results.
- **Backing up VWD project files** – If you wish to return to a VWD-based project (esp. animated/dynamic scenes), you must save some specific VWD working files and restore them when you return to the scene. If you use the VWD Utilities tab to clean out your Exchange files, you will have to redo your VWD work! Buried in the main VWD directory (e.g. C:\VWD\...) are a few folders that contain VWD presets as-well-as the workfiles and result-files created by VWD during the simulation sessions.

Because some of these files may get large and inspire periodic clean-outs after working on a few projects (especially animations!) it may be wise to backup some of these files with your project's scene-files after the project is complete or put on hold. There is a section on how to do this elsewhere in this document: **Backing-up and Restoring VWD Project Files**.

- **Reset controls to their default values** - You can reset specific tab, button, or field's setting by pressing the interface item while pressing the **right-mouse button** and **Alt** key together, returning the item's setting to its default value(s). If you are resetting a tab, all of that tab's controls and values should return to their default settings too. To reset all of the VWD tab interface settings at one (all of the program's controls), open the *Utilities* tab, and press the *Restore default parameters* button.
- **Resetting a static or dynamic simulation** - It's documented below, but worth a mention as a 'hint'. To reset a simulation to its starting state after it's been run and stopped or finished:

In the VWD Simulation tab, press and hold the **Alt** key as you **left-click** the appropriate *Start simulation* button that you used to run the previous simulation (static or dynamic).

When trying out many settings, it's really handy to set things up, try a simulation, then reset things, then make some more adjustments and try another simulation run - using this reset function.

Hint: You can switch between running dynamic and static simulations once the simulation state has been reset, but always use the **Alt** / *Start simulation* sequence on the same button you used to start the current simulation...

Important: Changing between the CPU and GPU (either way) after a simulation reset will probably not work the way you want. Not recommended, even if it allows you to do it. But it might work...

- **Saving Presets** - Across the VWD program tabs, many of the preset 'save' functions available to you are not very obvious or conventional. Generally, if you find that you can type characters into a preset pull-down field, that means you can probably create your own preset (from the current settings) by simply inserting your own filename in that field and pressing Enter.

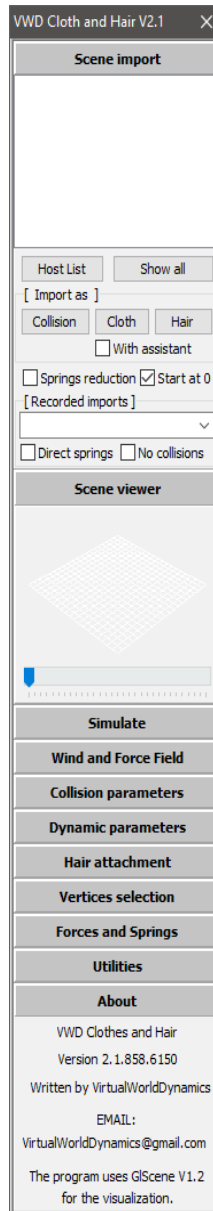
Important: When saving, there is currently no visible confirmation (just a Windows alerts sound), no overwrite prevention/protection, and the new item doesn't appear in the pulldown listbox until the next time you run a new VWD session, so it's hard to know if the preset was actually created. Sometimes I use Windows to browse to the programs preset folders to be sure.

- **Adjust session properties while simulating** - A very powerful feature in VWD is the ability to pause/stop an active simulation (static and dynamic/animated) at almost any point, make major changes to almost any property on any element, and continue the simulation, which will then continue, responding to the new state of the properties. One example would be to have a strong wind blowing a flag on a pole and then having the flag 'let go' of the pole half-way through the animation, and blow away... Those who 'see' this feature's power will do amazing things with VWD.
- **Different scenes with the same item names:** If you use the same item names in a variety of your host-application's scenes, and these scenes also have the same animation length, it's possible that pre-recorded VWD animation cache-files for your collision items will have the same names and result in side-effects into your animations (not be the animation you expected). The two best ways to manage this side-effect is:
 - Save, then clear your Exchange directory between projects and scenes via the *Utilities* tab's *Delete all "Exchange" files* button (Remember to back them up first if you have other project results you need to save!).

- When importing collision-items into your VWD session, consider checking the **New animation** box in the **Collision parameters** tab. This will tell VWD to ignore existing animation cache-files. You can also select an oddly-behaving collision-item in your scene, open the **Collision parameters** tab and press the **Record** button to force a rebuild of that animation cache-file from the current scene's animation.
- **Scene recording:** When you choose to use the VWD **Video** recording function in the **Simulate** tab, you should set your viewport view once and probably leave it as-is until the simulation is complete. Moving the viewport view *while* the recording is active will also record that viewport motion, and will result in very abrupt (not smooth) frame jumps in the recording.
- **Poser vs DAZ Studio Vertex order:** Poser and DAZ Studio seem to look at the imported vertices of the same item differently. This means that many of the files that might be useful for presets and animations will not be portable between these applications. This means that dynamic morphs, rip-files, hair vertices-sets and other presets like this may need to be created and saved in their respective applications (twice...). Animation cache files and simulations will also probably not transfer across the applications.

5 The VWD Cloth & Hair Program Interface

After starting VWD program, you will see the tall thin application interface on the right edge of your monitor. The interface is composed of a vertical stack of tabs that each allow you to perform their various functions while using the VWD application.



Of note, while almost all of these header 'tabs' act as buttons that expand or reduce their own window's interface elements, some of these headers have additional mouse-click/keyboard controls that can be quite useful.

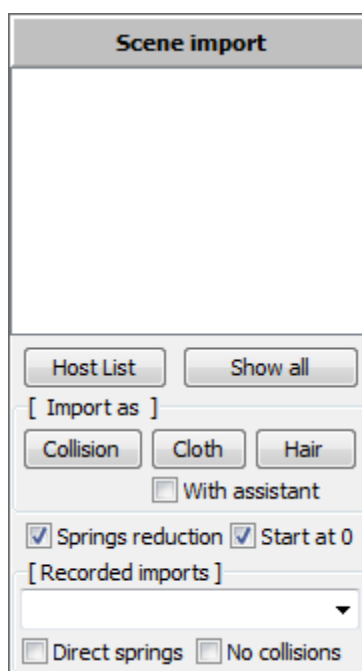
Before we describe the operation of the VWD application, let's take a look at each tab and consider what it's for, and what controls it contains. This will become a handy reference for later.

These windows will be described as they appear in the interface, in top-to-bottom order, with sub-windows being described just under the parent windows that produce them. Except for the *Scene viewer* viewport, these windows appear in their most-likely-to-be-used order.

5.1 The 'Scene import' Tab

This tab is where the items that will participate in this VWD simulation session are listed, imported, and assigned to their roles in the simulation (collision-item(s) or dynamic-actor).

Important: This tab is *also* used to select items that may be acted upon in other VWD tabs and functions. We call this the active-item in this document. This means that this tab is also a master 'item' browser, used to select items from your session's available imported items - for when you wish to adjust and configure them. There are some special commands in this listbox to manage item selection (like **Ctrl**-click on something to hide everything else, etc.) that are covered in more detail below.



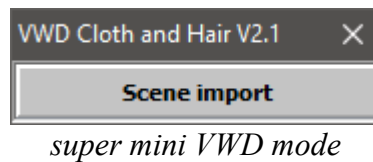
Important Ideas:

While there can be more than one collision-item (chairs, figures, props, other clothes, etc.) in a given VWD simulation session, there can only be **one** dynamic-actor (hair, clothing, blanket, flag, etc.) in any active VWD session. (We use multiple sequential VWD sessions to simulate multiple dynamic-actors in a given scene. More on this later).

Once a scene item is imported and assigned its role (collision, cloth, or hair), this role cannot be re-assigned without restarting the session. (e.g. once an item is assigned to be a dynamic **Cloth** or **Hair** actor, it cannot be changed to a **collision-item** during this same VWD session). If you mis-assign an item during your import, or change your mind about its best role, you'll need to exit the VWD session and do the import steps again.

Scene import tab header controls:

- **Click** on the **Scene import** tab header while pressing **Alt** to reset this tab's settings to their default values.
- **Reduce this tab** to just its header in the normal interface by **double-clicking** on the header. Repeat this to re-open the tab (a toggle).
- **To reduce the entire VWD program interface** to just the tab header, **right-click** on this tab header. Repeat this to restore the full interface (a toggle). This is useful when you wish to 'hide' the program without exiting:

**Scene import controls:**

- **The listbox area** just below the tab header contains the list of elements available in the host-application's scene interface. Click on any non-imported item to select it for the importing process.

Note: If the selected item is already imported (having a suffix: [collision], [hair], or [cloth]), it will become the new active-item throughout the program.

Items created during previous simulation sessions will have the '**_VWD**' suffix (e.g. **MyHair_VWD**). These elements can only be imported as collision-items.

Tip: **Alt-clicking** on a listbox item having the '**_VWD**' suffix will remove the item from both the listbox and the host-application scene. This is an easy way to remove a previously simulated item.

- **active-item:** At any time, clicking on a listbox item changes the currently selected item in the program to become the **active-item**. It should flicker red for a moment. This is often necessary when configuring the properties of a given collision or dynamic element (selection, rigidity, nail-to, etc.)
- **Hide all others** (no visible button) : At any time, if you activate this Scene import tab and **Ctrl-select** an **imported listbox item**, all of the other imported items in the scene will be hidden. Use the **Show all** button to re-display them. This is very useful for working on configuring a particular item without the distraction of other scene items. This sometimes behaves oddly when using the **Vertices selection** tab controls, but clicking on the same listbox item a second time, without the Ctrl-key pressed, usually generates the expected display result.
- **Host list:** asks the host-application to refresh the list of all available objects in the host-application scene. Note: If you add items to your host-application scene *after* you've started or are using VWD, then press this button, you should be able to see and import any of those new scene items.

- **Show all:** (re)display all of this session's actively imported components within the **Scene viewer** viewport. Items that you have hidden during your vertex selection activities can be made visible again using this button. (This has nothing to do with items in the scene on the host-application).

Import as groupbox: These controls are used to define the role of the current import item in this VWD session.

Hint: When simulating multiple layered items in a scene, a single item may first be imported as a *dynamic cloth actor* that's simulated and saved, usually interacting with a figure. Then in a follow-on session, its xxx_VWD simulated result item will then be imported as an underlying collision-item that the next dynamic-actor will interact on top of, as the next layer.

- **Collision:** Import the selected item as collision-item.

After pressing this button, the available control tabs will change, showing only the **Collision parameters** tab, which contains parameters that must be set when defining a collision-item. Once the parameters are set, press the **Collision** button again. This will commit those settings and return to the default interface.

After a successful import, the item name will be followed by '[Collision]' in the listbox, and appear in the VWD **Scene viewer** viewport.

- **Cloth:** Import the selected item as a dynamic-actor of the *cloth* type.

When chosen, this dynamic-actor will be the active simulation element in this session (remember: there can only be one dynamic-actor in each simulation session).

After pressing the **Cloth** button, the available control tabs will change, showing only the **Dynamic parameters** tab, which contains parameters that must be set when defining a dynamic-actor. Once these parameters are set, press the **Cloth** button again. This will commit those settings and return to the default interface.

After a successful import, the item name will be followed by [Cloth] in the listbox, and appear in the **Scene viewer** viewport.

- **Hair:** import the selected item as a dynamic-actor of the *hair* type.

When chosen, this *actor* will be the animated element in this simulation session (there can only be one dynamic-actor in each VWD simulation session).

After pressing the **Hair** button, the available control tabs will change, showing only the **Dynamic parameters** tab, which contains parameters that must be set when defining a dynamic-actor. Once these parameters are set, press the **Hair** button again. This will commit those settings, and then display a new tab, where you will select the vertices of hair that will be bound to the vertices of the figure's head.

After a successful import, the item name will be followed by [Hair] in the listbox, and appear in the **Scene viewer** viewport.

- **With assistant** (def = unchecked): When checked, the import process will open specific *Assistant* tabs to guide a simplified configuration of the selected import item.

Important: While the new *Cloth* and *Hair assistant* tabs offer some convenient shortcuts to the original VWD material properties configuration process, they also include some optimizations that may help solve memory issues during the simulation setup and execution phases of VWD operation. It is recommended to try using these assistants if you run into memory issues during the dynamic-actor properties setup or simulations.

- **Springs reduction** (def = unchecked): When checked, an extra processing task will be run during the material adjustment configuration steps to remove any redundant 'springs' that might slow down the simulation calculations (recommended). VWD generates 'springs' between vertices to control stiffness and stretch, and this process removes unneeded 'springs' when they occur.
- **Start at zero** (def = checked): When this is *unchecked* and the simulation is *static*, you can start the simulation at any available animation frame from the host-application. This frame determined by setting the frame-pointer location in the *Scene Viewer* tab. This is meaningless in dynamic/animated simulations.

Beta Bug! : The *Start at zero* function is known to have problems. We recommend leaving this option checked, and using your host-application to prepare the scene to the poses and positions you prefer. You can also run your simulation as a dynamic animation, let it finish, then scrub to the desired frame in the *Scene viewer* viewport, then press and hold the **Shift** key down as you use dynamic deformation controls to tweak your dynamic actor, then press the *Send pose to host* button (not *Send animation to host*) to send that one frame to your host-application.

Recorded imports groupbox:

- **The combobox:** Located near the bottom of this tab, this drop-down combobox contains the list of available **Recorded Import Presets (.RIP)**. These presets contain collections of import actions that are either automatically saved from your recent simulation import sessions, or explicitly saved by you. If you re-use a particular scene multiple time, saving and restoring the base import configuration can save a lot of time, and ensure consistency between simulation sessions.

Already imported scene elements are not affected by the **.RIP** file-based import process, as only items that are not already in the current scene will be imported from a **.RIP** file.

To save a new **.RIP** file, simply enter a new name in the combo box after you've imported some or all of your import items for a simulation session. If the specified **.RIP** exists, it will be update/overwritten (without warning).

Note: The two presets: **Last.rip** and **Restore.rip**, are automatically created and managed by VWD, so it's best not to name your own presets to these two names...

- **Direct springs** (def = unchecked): When checked, the program will look for, and re-use previously configured stiffness/stretch settings for the import item (if available from a previous use of this item).

VWD material characteristics are defined by applying so-called 'springs' of various stretch and stiffness between the various mesh vertices. Re-using these previously recorded springs can save time during the import process. Sometimes refreshing springs is desirable (fresh start), so this function has been added as a choice.

- **No collisions** (def = unchecked): When checked, any collision-items appearing in a **Recorded Import File (RIP)** will be ignored during the processing of that import file, and they will not be imported.

This feature is most useful when using the same dynamic-actor in varied scene situations, specifically when the collision-items differ between simulation sessions. This can save time during the import process.

Additional Scene import notes:

- When collision-items and a dynamic-actor are properly successfully imported, images and animations of these items will appear in the VWD **Scene viewer** viewport.
- As items of any type are imported into the VWD scene, they are displayed in a simple 'clay'-like mode by default, using arbitrarily assigned light colors (these do not relate to the original scene item materials and can be changed in the **Utilities** tab using the **Change color for current** button).
- You can reset specific button or field's setting by **Alt**-right-clicking on the interface item, returning the item's setting to its default value.
- **Recorded Springs** files (**.RSP** files) are recorded and stored in the main VWD program **Recorded Springs** folder, using the **.RSP** file extension. These files can be manually saved up and restored for future use. These files can be renamed more meaningfully using a Windows Explorer file manager/browser.
- **Recorded Import** files (**.RIP** files) are created and stored in the main VWD program **Recorded Imports** folder, using the **.RIP** file extension. These files can be manually saved and restored for future use. These files can be renamed more meaningfully using a Windows Explorer file manager/browser.

5.1.1 The 'Cloth assistant' Tab

When an item is imported in the *Scene import* tab as **Cloth** with the *Use assistant* option, this *Cloth assistant* tab opens, allowing access to the simplified cloth material and parameter options. Using this function as a dynamic-actor from DAZ Studio requires that the hair item is either parented or fit-to the figure, as it relies upon the VWD 'animated by host' function.

These presets and options are designed to generate useful cloth characteristics and to adjust the most commonly used cloth simulation and interaction settings. While these settings may work as configured, they also serve as a good starting point for non-optimal meshes that may need further refinement and tuning.

These settings control the most commonly used characteristics that are applied to dynamic-actors (the cloth actor in this case):

- Simulation type (static or dynamic/animated)
- Self-collision options (tight dresses may not need this, parts of flowing robes may)
- Cloth Characteristics (soft/thin or stiff/thick)
- Connections between meshes (e.g. 'gluing' dress straps to a figure's shoulders)
- Layer creation (This mixes cloth and collision for self-collision use)
- Areas of extra stiffness in the mesh (e.g. the belt-line around a figure's waist)
- Simulation forces (new or existing)

Cloth assistant controls:

Simulation groupbox: The type of simulation you want to generate: Static or Dynamic - there is no default, you must choose one.

- ***Static:*** Prepare for a static simulation - only one frame from the scene will be used from the host program. This 'frozen' scene frame will serve as the basis for the simulation, and only the dynamic-actor (cloth) will move in the scene. Even if the host-application has an animation sequence, only one frame will be brought into VWD.
- ***Dynamic:*** Prepare for a dynamic/animated simulation - the animated frames in the host-application scene will each be considered during the simulation. If the imported collision-items in the current session are animated in the host-application, they will perform their motions, interacting with the *dynamic cloth actor* in the simulation. (e.g. an animated figure will perform its motions within a simulating dress). **Note:** Using this function as a dynamic-actor from DAZ Studio requires that the hair item is either parented or fit-to the figure, as it relies upon the VWD 'animated by host' function.
- ***Fast*** (def = unchecked): Enable extra simulation settings that account for rapid mesh movements that can lead to poke-through and mesh 'explosions' during a simulation. It is never a problem to enable this, but it will add computation time to a simulation.

Use self-collision groupbox : Determine whether or not to enable the self-collision checking algorithms in the simulation. Enable this when you find your cloth overlapping and flowing through itself. This is especially likely to happen with the cloth is moving quickly in the simulation. This will add extra computation time to a simulation.

- ***Yes / No*** (no default, you must choose): Enable or disable the self-collision checking in the simulation.
- ***Complex*** (def = unchecked): Add extra computation steps when the cloth is likely to self-collide quickly or with extra energy. This will increase computation time, but may solve poke-through problems.
- ***Add selection/See selection*** (toggle button): Add the current selection to the self-collision memory.

If self-collision checking is enabled on this dynamic cloth actor, you must use the selection functions to identify which areas of the cloth actor should be subject to this checking process. It is easiest to select the entire cloth and press the ***Add selection*** button - and all of the cloth will be added to the checking phase. However, if a cloth item only has some parts that are likely to self collide, you can (should) only select those parts and add them to the selection memory buffer. Often the long drape of a dress will be added, but the tighter fitting upper torso parts of the same dress are not, as they are not likely (or able) to self-collide. Less selection means less computation time.

This sequence (select and add) accumulates, and can be used multiple times - use the ***Reset*** button (below) to clear this buffer (and start again, if you wish).

To see the currently selected vertices that will be checked for self-collision during the simulation, float the mouse cursor over the **Add selection** button and press on the **Ctrl** key. This will change the caption of the button to **See selection**. Pressing on this button will display (in red) the currently selected self-collision memory contents on the item.

- **Reset:** Clear/reset the self-collision memory buffer for this dynamic cloth actor. This is most often used when you wish to restart your selection process on the item.

Kind of material groupbox: Select the material characteristics for this dynamic cloth actor. These material choices determine the relative stiffness of the cloth actor. **Note:** this material stiffness is applied to the entire cloth actor, and only uses the 'by-extension' stiffening method.

- **Cloth material presets** - assuming an average mesh density in the dynamic-actor, these presets will set the cloth behave as described:

Very fluid: thin, light, silky material

Fluid: slightly more substantial, but still fairly soft

Reference: a commonly used cotton-like cloth setting

Stiff: a stiffer canvas or denim-like material (jeans or lighter jackets)

Very Stiff: A stiff, heavier material, like leather or neoprene

Again, these settings may vary greatly with the polygon density and design of the 3D-mesh being used, so use these settings as a starting point and experiment with each item.

Note: For all of these hair characteristic settings, if you are not satisfied with the initial results, you can adjust these parameters by going to the **Simulation history** tab (found in the **Simulate** tab) and adjusting the settings there. Also, you may have to stop, pause, or reset the current simulation to open the **Simulation history** tab and edit these settings.

- **Tie all vertices to collision** (def = unchecked): bind the vertices of this dynamic-actor to the nearby collision-item.

This function 'glues' the dynamic cloth actor firmly to any nearby collision-item(s). This is useful when connecting skin-tight cloth items like bikinis or leotards to a figure, or creating a clingy wet t-shirt effect with clothing. It applies to the entire dynamic clothing actor (there is no vertex selection option). It also stiffens the cloth as part of the effect.

Vertices tied to collision groupbox: Similar to the above setting, these parameters allow you to select specific collections of vertices and tie them to the nearby collision actor(s) with various softness options:

- **Hard / Soft** : Set how firmly the dynamic-actor (cloth) is attached to the nearby collision-items.

The **Hard** option will firmly glue the items together, and the **Soft** option will allow some extra 'play' between the meshes, and may help prevent poke-through and interpenetrations. (Tech note: the **Hard** option is like the *Nail to collision* setting in the **Forces and springs** tab, and is uses a softness setting of 0.1.)

Extend with soft selection groupbox: Extend the current vertices collection on the mesh outward, using a *soft-selection* method, which will generate a soft graduated transition between the fully attached vertices and the un-attached vertices. This graduated softening of the springs results in a more realistic simulation.

- **Use** (def = checked): This enables the soft-selection feature, increasing the current selection outward in a soft/graduated manner.
- **10 / 20 / 30** (def = 10): The soft/graduated extension distance added to the currently selected vertices. The larger this value, the farther out the graduated selection will influence the attachment forces in simulation.
- **Add selection/See selection** (toggle button): Add the current selection to the **Extend with soft selection** memory buffer.

If **Extend with soft selection** is enabled on this dynamic cloth actor, you must use the selection function to identify which areas of the cloth actor should be subject to this attachment process. Typically, you will select an area of a dress, like an elastic waist area and add this with a soft-selection value of 10, and press the **Add selection** button to commit this attachment.

To see the current list of vertices that are involved in the soft-selection buffer, float the mouse cursor over the **Add selection** button and press on the **Ctrl** key. This will change the caption of the button to **See selection**. Pressing on this button will display (in red) the current **Extend with soft selection** memory contents on the item. To see the actual soft-selection gradation, un-check then re-check the **Use** checkbox. This feature clearly shows how the soft-selection extension will be applied to the dynamic cloth actor.

- **Reset**: Clear/reset the current **Extend with soft selection** memory buffer for this dynamic cloth actor. This is most often used when you wish to restart your selection process on the item.

Note: These attachment and soft-selection values are set to reasonable defaults to simplify the assistant, you can always modify all these settings later using the **Simulation history** edit functions (from the **Simulate** tab).

Vertices for a new layer groupbox: Define a new layer for the current cloth, allowing you to create additional complex self-collision behaviors and relationships on the dynamic cloth actor.

- **Add selection/See selection** (toggle button): Add the current selection to the **Vertices for a new layer** memory buffer.

Pressing the **Ctrl** key while the mouse is over the **Add selection** button changes the caption of this button to **See selection**. Clicking this button will show the current **Vertices for a new layer** selection memory on the cloth item (in red).

- **Reset**: Clear/reset the current **Extend with soft selection** memory buffer for this dynamic cloth actor. This is most often used when you wish to restart your selection process on the item.

Rigid vertices groupbox: allows you to create attachments between vertices of the same cloth. This function will be useful to attach a belt or a bow to a dress for example.

- **Add selection/See selection** (toggle button): Add the current selection to the **Rigid vertices** memory buffer.

Pressing the **Ctrl** key while the mouse is over the **Add selection** button changes the caption of this button to **See selection**. Clicking this button will show the current **Rigid vertices** selection memory buffer on the item (in red).

- **Reset**: Clear/reset the current **Rigid vertices** memory buffer for this dynamic cloth actor. This is most often used when you wish to restart your selection process on the item.

Extend with soft selection (def = checked): Extend the current vertices collection on the mesh outward, using a *soft-selection* method, which will generate a soft gradated transition between the fully attached vertices and the unattached vertices. This gradated softening of the springs results in a more realistic simulation. The amount of soft-selection extension influence is fixed, and currently cannot be viewed in this interface.

Simulation groupbox: Commit these settings and start the dynamic cloth actor simulation.

- **Dynamic wind** groupbox: Basic wind force control options that can be set before starting the simulation.
 - **Use last** (def = unchecked): enable the last recorded wind force sequence during the simulation.
 - **Compute** (def = unchecked): Generate a new wind force sequence.
- **Simulate**: depending on the state of the **Start** checkbox (below), Either close the **Cloth assistant** tab and immediately start the simulation with the current settings, or open the **Simulate** tab where you can make adjustments to the simulation settings and manually start the simulation,
 - **Start** (def = checked): Immediately start the simulation with the current **Simulate** tab settings as they are. Uncheck this to open the **Simulate** tab instead (to make further adjustments, or check settings), and that tab will open when you press the **Simulate** button.

Note: Using this function with a dynamic-actor from DAZ Studio requires that the hair item is either parented or fit-to the figure, as it relies upon the VWD 'animated by host' function.

5.1.2 The '*Hair assistant*' Tab

When an item is imported in the *Scene import* tab as **Hair** with the **Use assistant** option, this *Hair assistant* tab opens, allowing access to the simplified hair material and parameter options.

These presets and options are designed to generate useful hair characteristics and to adjust the most commonly used hair simulation and interaction settings. While these settings may work as configured, they also serve as a good starting point for non-optimal meshes that may need further refinement and tuning.

These settings control the most commonly used characteristics that are applied to dynamic-actors (the hair actor in this case):

- Simulation type (static or dynamic/animated)
- The hair characteristics (length, bend, softness)
- Connections between the hair elements and the figure's head
- Areas of extra firmness in the hair (e.g. curls, hair buns, ponytails, braids, etc.)
- Simulation forces (new or existing)

Important: The hair in a dynamic/animated simulation must be parented or fitted-to its figure when using the *Hair Assistant*. If the hair is not 'attached' to the figure (and follows it while scrubbing the timeline) in the host-application timeline/animation, it will probably not work right in the VWD simulation.

***Hair attachment* tab header controls:**

- **Mouse-click** on the ***Hair attachment*** tab header while pressing **Alt** to reset this tab's settings to their default values.
- **Reduce the tab** to its header by **right-clicking** on that header. Repeat this to re-expand the tab (a toggle).

***Hair assistant* controls:**

Simulation groupbox: The type of simulation you want to generate: Static or Dynamic - there is no default, you must choose one.

- ***Static***: Prepare for a static simulation - only one frame from the scene will be used from the host program. This 'frozen' scene frame will serve as the basis for the simulation, and only the dynamic-actor (hair) will move in the scene. Even if the host-application has an animation sequence, only one frame will be brought into VWD.
- ***Dynamic***: Prepare for a dynamic/animated simulation - the animated frames in the host-application scene will each be considered during the simulation. If the imported collision-items in the current session are animated in the host-application, they will perform their motions, interacting with the dynamic hair actor in the simulation. (e.g. an animated figure will perform its motions with this hair attached to its head).
- ***Fast*** (def = unchecked): Enable extra simulation settings that account for rapid mesh movements that can lead to poke-through and mesh 'explosions' during a simulation. It is never a problem to enable this, but it will add computation time to a simulation.

Hair length groupbox: Set the hair length. No default is set, you must choose a length.

- ***Long***: The hair is well-below the shoulders.
- ***Middle***: The hair is at the shoulder level.
- ***Short***: The hair is above the shoulders, around ear length.

Kind of hair groupbox: Set the hair shape or body type. Soft bodied hair has very little inherent shape and falls straight, while full-bodied hair is stiffer and will hold waves and curls. No default is set, you must choose a hair type.

- ***Straight***: The hair strands are soft and straight without any waves.
- ***Wavy***: The hair is soft, but has some waviness but is not curly.
- ***Curly***: The hair is firm, curly, and/or frizzy.

Fluidity groupbox: Set the connectivity between the hair strands. This affects how the entire hair moves together as a group. It can be loose like individual strands, or connected, like a helmet. No default is set, you must choose a fluidity level.

- **Fluid:** Hair strands are independent and can move freely relative to the other hair strands.
- **Soft:** Hair strands are softly linked together. (usually the most realistic).
- **Sticky:** Hair strands are firmly linked with each other. This method can be used to create a wet hair effect, or to temper hair motion during fast moving animations.

Note: For all of these hair characteristic settings, if you are not satisfied with the initial results, you can adjust these parameters by going to the **Simulation history** tab (found in the **Simulate** tab) and adjusting the settings there. Also, you may have to stop, pause, or reset the current simulation to open the **Simulation history** tab and edit these settings.

- **Tie all vertices to collision** (def = unchecked): **Apply** a soft attachment (*Nail to collision*) between all of the hair and the figure. This will greatly limit the motion of the entire dynamic hair actor in the simulation, but may be a desirable effect in certain situations. This might be used to create a wet hair effect.

Vertices tied on the head groupbox: Attach the parts of the hair to the head of the figure.

Note: The amount of effort required to fully attach a hair mesh to a figure really depends on how well the hair mesh was designed and built. Some are quite simple, and others will take some thought, finesse, and patience.

- **Search:** Scan the hair mesh elements for a skull-shaped CAP to use as the hair foundation. (Not all hair products have this element). Use the distance value to the left to cover more area in the search.
 - **[distance value]** (def = 1.0, range: 0.0 to 2.0): Distance from the hair CAP to scan for hair. The larger this value, the more of the nearby hair elements will be found and selected

Note: The hair vertices that are selected in this procedure will be firmly attached to the figure, which will limit the flowing motion of the hair. In this process, we recommend that you try to select as much of the roots (near the scalp) of the hair as possible, but not too much more.

- **[+/-]:** Adjust the **Search** distance value (+/- 0.1 per button press)
- **Select vertices from collision:** Use this to create a selection on the hair, composed of the vertices on the hair that are very close to the selected vertices on the nearby collision item (usually the collision figure). These vertices can then be adjusted further and eventually applied as attachment vertices to that figure. This lets you use selected vertices on the figure to help determine which hair vertices are very close to the figure and good candidates for attachment.

Hint: To select the vertices on the figure's head, change the active-item to be the figure by opening the **Scene import** tab, Click on the figure (its vertices will turn blue), then Ctrl-click on the figure to hide everything but the figure. Then use the usual VWD vertices selection tools to select the head (or parts of the head). When you are satisfied, press the Select vertices from collision button, and your hair should appear again, with the calculated vertices selected. You might wish to clear your Vertices memory buffer (MC) and add these newly selected vertices to the buffer (M+) in case you need to re-apply these selected vertices later during the session.

- **Extend with soft selection** groupbox: Extend the current vertices collection on the mesh outward, using a *soft-selection* method, which will generate a soft graduated transition between the fully attached vertices and the un-attached vertices. This graduated softening of the springs results in a more realistic simulation.
 - **Use** (def = checked): This enables the soft-selection feature, increasing the current selection outward in a soft/graduated manner.
 - **10 / 20** (def = 10): The soft/graduated extension distance added to the currently selected vertices. The larger this value, the farther out the graduated selection will influence the attachment forces in simulation.
 - **Hard** (def = unchecked): Use firm springs to attach the selected vertices to the figure's head. This will preserve the selected hair shape but reduce the motion during the simulation. Uncheck this to soften the attachment springs, allowing more motion during the simulation.
- **Add selection/See selection** (toggle button): Add the current vertices selection to the hair attachment vertices memory list. These vertices will be attached to the figure's head/scalp.

Note: This function (select and add) can be used multiple times - use the **Reset** button to the right to clear this buffer if you wish to restart this selection process.

To see the current list of vertices that will be attached to the figure's head, float the mouse cursor over the **Add selection** button and press on the **Ctrl** key. This will change the caption of the button to **See selection**. Pressing on this button will display (in red) the current **Vertices tied to head** memory contents on the item.

- **Reset:** Clear/reset the current **Extend with soft selection** memory buffer for this dynamic cloth actor. This is most often used when you wish to restart your selection process on the item.

Rigid vertices groupbox: Attach the selected parts of the hair to the rest of the (nearby) hair. This is how you might attach bows, buns, or other parts of the dynamic hair actor together, so the simulate better, or so they don't fall apart during the simulation.

Note: This does *not* attach parts of the hair to the figure or other nearby collision-items! This particular function only connects (possibly disconnected) parts of the hair mesh/product *together*. If a hair product includes a ribbon or bow that is not welded to the mesh, it may fall off during the simulation. This function helps address that problem.

- **Add selection/See selection** (toggle button): Add the current vertices selection to the current **Rigid vertices** list. These vertices will be attached to the nearby hair elements (using the 'by-neighborhood' method).

Note: This function (select and add) can be used multiple times - use the **Reset** button to the right to clear this buffer if you wish to restart this selection process.

To see the current list of vertices that will be attached to the nearby hair elements, float the mouse cursor over the **Add selection** button and press on the **Ctrl** key. This will change the caption of the button to **See selection**. Pressing on this button will display (in red) the current **Rigid vertices** memory contents on the item.

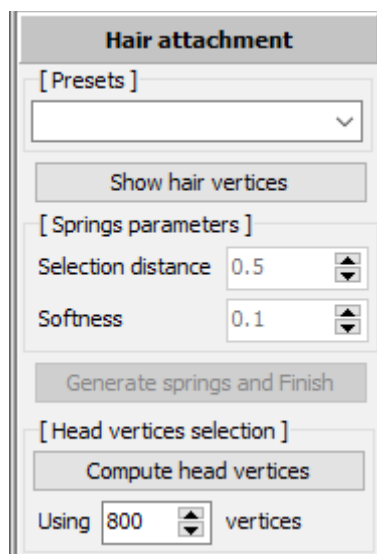
- **Reset:** Clear/reset the current **Rigid vertices** memory buffer for this dynamic cloth actor. This is most often used when you wish to restart your selection process on the item.

Simulation groupbox: Commit these settings and start the dynamic cloth actor simulation.

- **Dynamic wind** groupbox: Basic wind force control options that can be set before starting the simulation.
 - **Use last** (def = unchecked): enable the last recorded wind force sequence during the simulation.
 - **Recalculate Dyn** (def = unchecked): Generate a new wind force sequence.
- **Simulate:** depending on the state of the **Start** checkbox (below), Either close the Hair **assistant** tab and immediately start the simulation with the current settings, or open the **Simulate** tab where you can make adjustments to the simulation settings and manually start the simulation,
 - **Start** (def = checked): Immediately start the simulation with the current **Simulate** tab settings as they are. Uncheck this to simply *open* the **Simulate** tab instead (to make further adjustments, or check settings), and that tab will open when you press the **Simulate** button.

5.1.3 The '*Hair attachment*' Tab

After importing a dynamic hair actor (using the *Scene import* tab and *not using the Hair assistant*), and then setting up the general self-collision and rigidity of the hair in the *Dynamic parameters* tab, you will use the settings in this *Hair attachment* tab to attach this dynamic hair actor to the figure.



The general workflow in this tab is to make selections on both the figure's scalp, and the hair's skull-cap (or equivalent selections), then have VWD connect them with simulation 'springs'. Once attached, you can make further adjustments to the dynamic hair actor and/or collision-items, and run the simulation.

To switch the current selection item - between your collision-item figure and the dynamic hair actor - open the *Scene import* tab at the top of VWD's window, and *select the imported item* that you wish to work on. Also, by **Ctrl-clicking** *again* on this item, you can show *only* this item while you are working on it.

Note: This particular tab is only available when importing a dynamic hair actor *without the Hair assistant*, and it appears only after completion of the hair actor's *Dynamic parameters* configuration stage. Once completed, this attachment configuration stage is *not* available to the user during this VWD session. A session restart and import must occur to revisit this settings tab.

Note: Using this 'manual' configuration option in place of the *Hair Assistant* tab settings may be desirable or necessary for certain effects, but also may result in the generation of additional simulation springs, resulting in memory issues in complex scenes. The *Hair assistant* may use VWD's *Animated by host* technique to notably reduce the dynamic-hair actor's areas that are involved in the simulation.

***Hair attachment* tab header controls:**

- **Alt-click** on the ***Hair attachment*** tab header to reset this tab's settings to their default values.
- **Reduce the tab** to its header by **right-clicking** on that header. Repeat this to re-expand the tab (a toggle).

***Hair attachment* controls:**

Presets groupbox: This is where any available figure scalp vertices selection presets will be available, or can be saved after a new selection set is created.

- [*Preset combo box*]: Pre-configured vertex groups for well-known characters. Click on the right-side arrow of the combobox and choose from the list of available presets.

Hint: Hair attachment presets exist for DAZ's ***Victoria4, Genesis, Genesis2, Genesis3, Genesis8*** figures. The variations (Lo-Rez, Hi-Rez, etc.) are options that use different vertex collections. Try each one to find your preference. Each selection will replace the previous preset settings (preset selections do not accumulate).

Hint: By manually removing or adding vertices to the selection by dragging the mouse while pressing the **Alt** or **Ctrl** key and/or using the ***Vertices selection*** tab tools, you can adjust the head vertices selection that will be attached to the dynamic hair actor. Remember to **save this work** as a preset by typing your new selection set name in the above *preset combobox* and pressing **Enter**. It can then be recalled in a future session.

Show hair vertices: Commit the collision-item figure's attachment selection vertices set to memory, and display the hair, in vertices selection mode.

Warning: *Only* press this button *after* completing your figure's head vertices selection process, as you cannot return to that selection mode once this button is pressed without a full VWD session restart.

Hint: To switch the current selection item - between your collision-item figure and the dynamic hair actor - open the ***Scene import*** tab at the top of VWD's window, and *select the imported item* that you wish to work on. Also, by **Ctrl-clicking** on this item, you can show *only* this item while you are working on it.

***Springs parameters* groupbox:**

- **Selection distance** (def = 0.5, range: 0.1 to 5.0): Distance from the selected head vertices where hair vertices will be included in the attachment of the head and hair. Increasing this number will include hair vertices that are the farther from the figure's head vertices.

Hint: Try to keep this distance value as low as possible and still include as *many* of the hair strand 'roots' near the figure scalp as possible, but try not to include very *much* of each root - just the base.

Hint: By manually removing or adding vertices to the existing selection by dragging the mouse over vertices while pressing the **Alt** or **Ctrl** key and/or using the **Vertices selection** tab tools, you can adjust the head vertices selection that will be attached to the figure's head. This is useful for attaching parts of the hair that have no mesh areas that are close to the figure's scalp, like hair buns, pony tails, braids, etc. Using all of these tools to design a complete selection set is recommended on complex hair.

- **Softness** (def = 0.1, range 0.01 to 0.5): Softness of the springs attaching the selected hair vertices to the selected head vertices. Higher values are softer. This setting may affect the general firmness of the selected hair vertices during the simulation.

Generate springs and Finish: Once both sets of vertices selections (the figure's and the head's) are completed, press this button to commit the attachment of the hair to the figure with newly generated springs between the vertices selection sets.

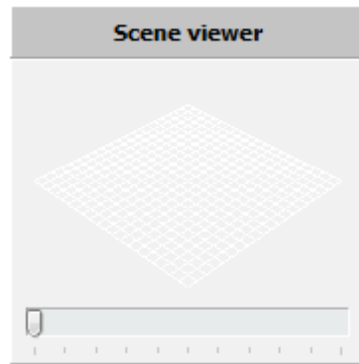
Head vertices selection groupbox: A tool to help optimize the head vertices selection process.

- **Compute head vertices:** This helper function find the longest vertex pairs in the current selection (created using manual selection, presets, or both), and saves no more than the number found in the **Using [] vertices** box below in the resulting selection. This is a simple optimization tool, and does not need to be used.
 - **Using [] vertices** (def = 810, range: 200 to 2000): No more than this number of vertices in the current figure's scalp selection set will be saved. The vertices making up the longest pairs will be saved up until this count. All other selected vertices will be removed from the selection.
 - **Note:** If the number of selected vertices are less than the **Using [] vertices** value, there will be no effect when **Compute head vertices** button is pressed.
 - **Warning:** This button/function becomes unavailable once the **Show hair vertices** button is pressed, since that button commits the head vertices selection set to memory. The only way to re-activate this button/function is to restart the VWD session and re-import the dynamic hair actor.

5.2 The 'Scene viewer' Tab

This humble little tab is central to most of VWD's functions as it displays the current session's imported scene items, allowing you to:

- Preview the scene elements as you import them (before the simulations)
- Select and assign various characteristics to the various collision, hair, and cloth items
- Monitor active simulations and review the simulation results (static and animated)
- Interact with running simulations, selecting and pulling on the meshes during the active simulations.



Most of these activities are context-specific and will be described in further detail.

While the only *visible* controller in this tab is the animation slider (detailed below), tab display-mode controls and additional powerful scene navigation and manipulation controls are activated using combinations of mouse, mouse-button, and keyboard commands (detailed below).

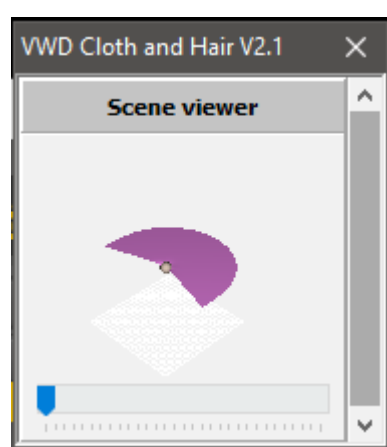
It may be worth-while to take a quick minute and run through and familiarize yourself with these controls *before* you need them. Being lucid with these controls can save you time and prevent frustration.

Scene viewer Tab header controls

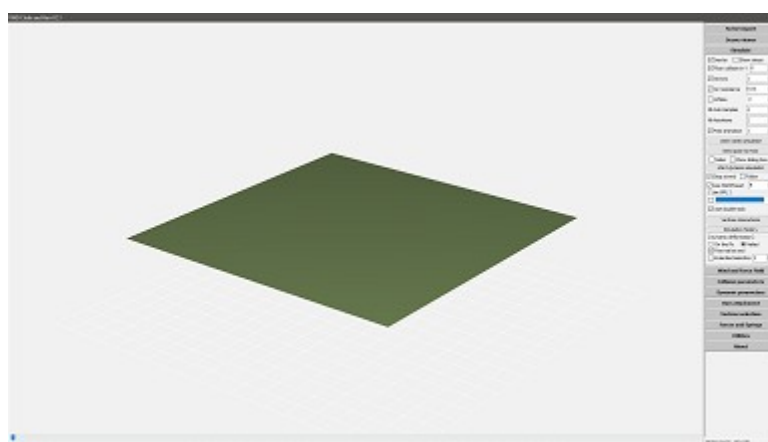
The *Scene viewer* offers three distinct window display sizes, which are controlled with mouse-**click** combinations on the tab header itself :

- **Full-screen<->mini** mode: toggle between the mini-viewer (shown above) and the full-screen mode by clicking or double-clicking on the tab, or **Ctrl**-clicking the tab. The tab will switch between large and small modes as you repeat this sequence.

The full-screen mode is used extensively throughout VWD program use, especially when selecting mesh elements and applying various characteristics to those selections. It is also the best way to view the full detail of the simulation results - before, during, and after the simulations are run.



mini-viewer



full-screen mode

- **Scene viewer 'mini' or 'solo' mode:** To reduce the *entire* VWD interface to a single small *Scene viewer* tab, **right-click** on the *Scene viewer* tab. Once in this mode, you can **double-click** on the same tab and the *Scene viewer* will toggle between full-screen/solo and mini/solo mode, hiding the rest of the tab headers

This mode is handy for keeping an eye on longer animation simulations, while you use your system for other tasks, and the control commands described below will work in this mini-viewer as well. To restore the interface to its normal state, **right-click** on the tab header.

Scene active-item viewing modes:

When items of any type are imported into the VWD scene, they are displayed in a simple 'clay' mode by default and are assigned arbitrary pastel colors (these do not relate to the original scene item materials and can be changed using the *Change color of current* button in the *Utilities* tab).

Items can be also shown in different display modes at any time. After selecting the item in the *Scene import* tab's listbox, the item will become the “active-item” in the scene. In addition, the item's *Scene viewer* display mode can be changed with the following key presses *while the mouse is in/over the display viewport*:

- **Transparent view** (toggle): press the 'o' key and the active-item will become partially transparent

- **Mesh view** (toggle): press the 'a' key and the active-item's mesh will be displayed with no 'skin' on the mesh.
- **Hidden line view** (toggle): press the 'h' key to display the active-item using the 'hidden line' view, where the mesh is visible with an underlying mesh skin.

As each of these controls is a toggle, it is easy to cycle between these display modes at any time.

Scene viewer controls:

- **Scrubber:** The only **on-screen control** in this tab is the sliding frame pointer 'scrubber' at the bottom of the screen's viewport. Clicking and holding the right mouse button, while on this scrubber will allow you to drag it back and forth ('scrubbing') across the animation sequence. This can be done before and after a dynamic/animated simulation. Additional keyboard commands to manipulate this slider are described below.

When the mouse cursor is within the *Scene viewer viewport* (small or full-screen), a collection of mouse and keyboard combinations are available to allow for easy navigation of the VWD 3D-space viewer. These view navigation controls can be operated at any time, whether working with a scene item, running and interacting with an active simulation, or viewing the resulting simulation results - static or animated.

The mouse and keyboard based controls:

The mouse must be in/over the ***Scene viewer viewport*** for these controls to operate as described.

- **Zoom:** The use of the mouse wheel allows zooming in the scene. Pressing the **Alt** key while using the mouse wheel reduces the zoom speed for more accurate positioning.
- **Rotate:** Right-click and hold the mouse in the ***Scene viewer*** tab, and move the mouse to rotate the scene items. The scene should rotate around the barycenter of the active scene item, unless that rotation point has been 'recentered' (see below)
- **Reset Rotation and Zoom:** Double-click anywhere in the ***Scene viewer*** viewport to reset the 'camera' view. This will also reset the scene's rotation-center to the barycenter of the current active/selected item in the scene.
- **Recenter Rotation:** Move the mouse to where you would like the rotation-center to be (on the active-item), and press the mouse-wheel (center button). This should become the new rotation-center of the scene.
- **Reframe item in window (Panning):** While pressing the **Ctrl** key, right-click and drag the mouse in the direction you wish to slide the scene view.

- **Follow the active-item** (toggle): Press (and release) the '**f**' key to have the scene camera center-on and follow the current active-item in the scene. This is particularly useful when simulating during animations.
- **Top view:** Press '**5**' on the numeric keypad (numlock on...)
- **Front view:** Press '**8**' on the numeric keypad (numlock on...)
- **Back view:** Press '**2**' on the numeric keypad (numlock on...)
- **Right view:** Press '**6**' on the numeric keypad (numlock on...)
- **Left view:** Press '**4**' on the numeric keypad (numlock on...)

'Scene viewer' Navigation During Simulations:

- All of the above controls can be applied to adjust your scene view, allowing you to actively review and assess the simulation results while it is simulating.
- During the simulation, **clicking with the middle button** in a free area of the viewer window will position the center of rotation of the scene to the first/nearest collision element barycenter.
- In the **Simulate** tab, the **follow** checkbox can be pressed at any time (or press the '**f**' key), and the camera view will actively center-on and follow the scene elements as the simulation progresses. This is useful when tracking a rapidly moving animated item as it migrates through a simulation.

Viewing animations:

As items are imported into a VWD session, the animated scene elements will 'perform' in a looping cycle in the *Scene viewer* tab. This also occurs once a dynamic simulation has completed. Any animation sequence can be controlled and reviewed using the on-screen slider (click and drag) and/or the following keyboard controls:

- **Stop:** Pressing **ESC** stops the animation.
- **Resume:** Pressing **F1** resumes the animation.
- **Stop/Start:** Pressing the space-bar toggles the animation's stop/running status.
- **Previous Frame:** The **left arrow key** (←) moves the animation to the previous frame.
- **Next Frame:** The **right arrow key** (→) moves the animation to the next frame.
- **First Frame:** The **up key** (↑) moves the animation to the first frame.
- **Last Frame:** The **down key** (↓) moves the animation to the last frame.

Additional *Scene viewer* Notes:

- The imported items in the VWD session are displayed in the ***Scene viewer*** tab with randomly applied colors. These have no relationship to the colors or textures of the same items in your host-application. If you wish to change these colors in your session view for any reason, press the ***Change color for current*** button in the ***Utilities*** tab, near the bottom of the VWD interface.
- **Vertex Selection:** While actively selecting VWD scene item's vertices in the ***Vertices selection*** tab, all of the above mouse and keyboard navigation modes can be used, allowing you to optimize your view angles, enabling fine-grained control in your vertex selection process. Vertex selection tools will be discussed later, in the ***Vertices selection*** tab section.
- **The 'Current Selection' or 'active-item':** Some of the ***Scene viewer*** controls operate on the currently selected session item. This selection occurs by opening the ***Scene import*** tab (top of the VWD interface), and clicking on the desired imported item in the listbox. This item must already be imported and available in the current VWD session.

Because there is no ongoing indication of the current selection in the ***Scene viewer*** viewport, you can press the '**a**' (mesh) or '**o**' (transparent), or '**h**' (hidden line) key to temporarily change the current active-item's display type, as a quick way to visually indicate which item is currently active. Press that same key again to return to your previous viewing mode.

Note: these key-presses must occur *while the mouse is in/over the display viewport*:

5.3 The 'Simulate' Tab

This tab is where the parameters and settings that drive the simulation itself are set. Here's where you'll set the more global forces: (like gravity and inertia). You'll also configure where the scene simulation is calculated (CPU or GPU), whether to record the simulation as a video. You will also adjust settings for the granularity of some of the calculation algorithms, and whether this is a 'static' or 'dynamic' (animated) simulation, and a few other controls that help tweak the simulation behavior. This is where you can reload and restart your current simulation, if you want to make some adjustments with the current scene setup.

Simulate

☒ Inertia ☐ Show stress

☒ Floor collision in Y 0

☒ Gravity 1

☒ Air resistance 0.02

☐ Inflate 10

Nb Sub Samples 4

Nb iterations 2

☒ Fast animation 1

Start static simulation

Send pose to Host

☐ Video ☐ Show dialog box

Start dynamic simulation

☒ Stop at end ☐ Follow

☒ Use Multithread 8

[Use GPU]

☐ GeForce GTX 760 Ti OEM

☐ Double sticks ☒ Sticks CPU

Vertices interactions

Simulation history

[Dynamic deformation]

☐ On the fly ☒ Nailed

☒ Free nail at end

☐ Extended selection 2

Some of these settings are configured *before* the simulation begins, while others can be actively changed *while* an active simulation is in progress or is paused.

There are two simulations modes available in the VWD program:

- **Static** : In this mode, nothing but the single cloth or hair dynamic-actor (there can only be one dynamic-actor in a simulation session) will change during the simulation. This item will react to collision-items, itself (self-collision), simulations forces (wind, gravity, etc.), and/or user controlled mesh adjustments which can be applied while the simulation is running.

- **Dynamic** (animated): In this mode, some or all of the collision-items in the scene can be moving - as pre-configured animations from the host-application -h while the single Hair or Cloth dynamic-actor is responding to these items, as well as itself (self-collision), simulation forces (wind, gravity, etc.), and/or user controlled mesh adjustments which can be applied while the simulation is running.

In both cases, wind forces can be enabled to add wind dynamics to the simulations. The simulations can also be paused and then resumed *after* making adjustments to mesh positioning, some of the simulation parameters, and/or dynamic-actor material settings.

Video recordings of the *Scene viewer* screen can be set to be recorded with the simulation if done so before the simulation is started.

When this tab is opened/active, the number of 'springs' involved in this simulation session is displayed at the bottom of the VWD interface. It is often useful to keep track of resource consumption using numbers like these.

Be sure to look for the instructions (below) describing how to **completely reset a simulation back to the initial starting conditions**. From there, you can make adjustments and restart simulations from scratch (many times). This is *very* useful when exploring the settings and fine-tuning your results!

***Simulate* tab header controls:**

- **Alt-click** on the *Simulate* tab to reset this tab's settings to their default values.
- **Reduce the tab** to its header by **right-clicking** on that header. Repeat this to re-expand the tab (a toggle).

***Simulate* controls:**

- **Inertia** (def = checked): enable whether the forces of inertia will be involved in the simulation. Disabling inertia will probably generate a result that is too 'soft' to be realistic, but may be useful for special effects (e.g. under water scenes) or when local vertices displacements without animation are needed. This force will work in concert with the '**node weight**' value assigned to the dynamic-actor in the *Dynamic parameters* tab.

Hint: This checkbox can be changed during the simulation to settle wild mesh motions (e.g. hair going wild). It's also useful when fine-grained control is needed during **dynamic deformation** (active mouse tugging and pulling on a simulating mesh).

- **Show Stress** (def = unchecked): show colorized simulation stress forces on the dynamic-actor while the simulation is running. This might be handy to help determine why mesh elements are behaving in odd or undesirable ways (e.g. inter-penetrating). It's handy but it also consumes some calculation time.
 - Simulation springs that are compressed (unchanged, or less than their original length) will be green.
 - Simulation springs that are stretched (greater than their original length) will vary from green to yellow to red. The red value corresponds to the original length * 1.2.

- **Floor collision in Y** (def = checked): Enable the local VWD ground-plane in the simulation (the location defaults to zero in Y). When enabled, dynamic-actors will collide with, and come to rest on, this plane.

Floor Collision Y value (def=0): set this Y value to a non-zero value to move the *Floor collision plane* up or down in the simulation.

- **Gravity** (def = checked): enable gravity as a force on the dynamic-actor during the simulation. This value of gravity does *not* directly correspond to gravity on earth. The default value of '1' results in a gravity value that changes with each scene, based on the size of the specific elements in the current scene. The related value of the gravity setting can be adjusted (see below). When unchecked, gravity forces will be ignored in the simulation.

Gravity value (def = 1): a multiplier to increase or decrease the current scene's gravity setting. the default value of 1 does not correlate to an earth gravity (see above). Setting a higher value will increase this gravity force, and a lower setting (e.g. 0.5) will reduce it. A zero value will disable the force (but calculate it anyway!), and a negative value can be used to invert the force. You may have to experiment with this setting in certain scenes.

- **Air resistance** (def = checked): enable air resistance forces in the simulation. Unchecking this will result in less dampened motion in the scene, which may be useful for certain effects.

Air resistance value (def = 0.02, range: 0.0 to 1.0): increase or decrease this value to adjust the motion dampening effects of the air in a simulation. Adjustments to this may help generate more realistic results on a dynamic-actor.

- **Inflate** (def = unchecked): enable inflation forces on the dynamic-actor's mesh. This force will only impact meshes with defined normals, and will 'push' or 'pull' based on the facing direction of these normals.

Inflate value (def = 10, range: -200 to 200): inflation force amount. Increase or decrease this value to control pressure to/from the affected direction.

- **Nb Sub Samples** (def = 4, range: 1 to 32): the number of 'virtual keyframes' inserted between animation frames to help generate better collision detection and management in the simulation. Intended to help reduce inter-penetrations (poke through), the default value of 4 seems to work in most general cases. If you experience inter-penetrations in faster animations and/or self-collision scenarios, may wish to increase this value - at the expense of calculation time.
- **Nb Iterations** (def = 2, range: 1 to 32): the number of times the simulation is run for each calculation cycle. The default value of 2 appears correct in almost all cases. If simulation issues do not seem to be solved by changing other parameters, increasing this value may help.
- **Fast move** (def = checked): add additional calculations to help prevent inter-penetrations (poke through) and other negative effects of fast-moving animations - especially in collision and self-collision scenarios. This box is checked by default to avoid increasing the value of the more-expensive **Nb Sub Samples** (see above). When unchecked, these extra calculations will not be enabled.

Fast move value (def = 1, range: 1 to 3): increase this value to increase the number calculations applied to preventing dynamic-actor mesh inter-penetrations. Higher numbers will prevent more inter-penetrations at the expense of calculation time. Increase this value before increasing the *Nb Sub Samples* value above.

- **Reset the simulation without a VWD session restart:** Once a simulation has been initiated, many of these settings can no longer be changed, nor can the alternate simulation type (static/dynamic) be run (those buttons are disabled). By pressing the **Alt** key while left-clicking the *Start dynamic simulation* or *Start static simulation* button (whichever has been used), the scene elements will be reset, and both the *Start static simulation* and *Start dynamic simulation* buttons will be available again, along with the relevant *Simulate* control settings.

Hint: When a simulation is reset (described above), many of the other Cloth & Hair configuration settings will also become available again for additional adjustments! When a simulation is running, these windows and settings become unavailable until the simulation is reset, or a new VWD session is (re)started.

Note: While the CPU/GPU control settings may be available after a simulation reset, the simulations may not work as expected when switching between them. This effect is under investigation. Use with caution.

Simulation modes:

VWD simulator works in two similar but distinct modes:

- **Static:** where your selected items are imported from a single frame in the host-application and effectively 'frozen' in VWD. The specified dynamic-actor (Hair/Cloth/Flags/etc.) in the simulation then interacts with:
 - The 'frozen' collision elements (figures, furniture, the ground, etc. - even if animated, they are considered to be 'frozen' in each frame!)
 - Any active simulation forces (gravity, inertia, wind, air-resistance, etc.)
 - Itself - self collisions

Static simulations do not request animation frames from the host-applications, and simply calculate the scene as presented once, returning the resulting simulated dynamic-actor to the host-application for rendering.

- **Dynamic:** where the simulation engine requests the available animation frames from the host-application, then calculates the result of each frame change, based on:
 - The current state of the dynamic-actor in the simulation (usually coming from the previous simulation frame)
 - Any active simulation forces (gravity, inertia, wind, air-resistance, etc.)
 - The new status of the animated collision-items (even if they are not simulating, some are moving and changing shape between frames, then they are all frozen in each frame!)
 - Itself - self collisions

This process continues until completed, paused, or stopped by the user, generating a series of motion records that are returned to the host-application as animation frames.

The specific control buttons for the Static and Dynamic simulations are few, but interesting in that they *change their state*, depending on how they are used! This is powerful, but not always obvious.

Static simulation controls:

- ***Start static simulation:*** Start a simulation using the **static** mode. The current frame state will be used, with all imported collision-items frozen in 3D-space, and the dynamic-actor being simulated and interacting with these items. Once a static simulation is running, consider these changes and controls of interest:
 - Once the static simulation has started, the ***Start static simulation*** button changes to ***Stop static simulation***, allowing you to stop the static simulation exactly where it is. This can also be considered a 'pause' function. You can **restart** and **pause** the simulation by pressing this button repeatedly (a toggle).
 - The use and strength values of gravity, inertia and air-resistance can be changed while actively running the simulation, or when paused.
- ***Send pose to host:*** Send the state of the VWD scene's dynamic-actor (hair/clothes/etc.) to the host-application (Poser/DS/etc.). The host-application's version of the dynamic-actor will *immediately* be updated to reflect the state of the dynamic-actor in the VWD scene at that moment.

Also of note:

- The ***Send pose to host*** button can be pressed at any time during the Static simulation - active or stopped. This is a useful feature for saving progressive variations of your simulation session, but you may wish to return to your host-application and save/export any/all of these 'snapshots' to their own scene or scene subsets for future review. You can then return to VWD and simulate further...
- Unlike the Dynamic simulation's equivalent function, this button does *not* end the simulation and return you to your host-application. However, you *can* bring the host-application to the front at any time, and work with the results, then return back to VWD for further simulations and adjustments, repeating this cycle for as long as you like.

Hint: When the dynamic-actor's simulation state is sent to and updated in your host-application, you can optionally pause the simulation in VWD, then bring up the host-application and save that result in your preferred form. You can then return to VWD, restart the simulation, make more changes, then ***Send pose to host*** again, repeating this process for any simulation state(s) that you may wish to save.

Hint: When doing a *dynamic simulation* (animated), this button can be also used to save snapshots of the animation and treated as described above. This can be done when the *dynamic simulation* is paused or completed, and with any available frame. One advantage of this is the ***Send pose to host*** results can be saved with a host application scene, and animations cannot.

- **Manual mesh manipulations** can be applied to the scene's dynamic-actor during (and after?) the simulation by pressing and holding the **Shift** key during active or paused simulations. Releasing the **Shift** key will end this manipulation mode (you can repeat this process). Note that the **Inertia** setting will be disabled after using this manipulation mode, and can be re-enabled by simple re-checking the **Inertia** checkbox at any time. More details about this mode and its options will be detailed in a later section.
- **Reset the simulation:** Once you have started a Static simulation using the **Start static simulation** button, the **Start dynamic simulation** option will be disabled for the remainder of this VWD session. By pressing the **Alt** key with the **Start static simulation** button, the scene will be reset, and both static and dynamic simulations will be available again. Press either button to begin a new simulation.

Note: While the CPU/GPU control settings may be available after a simulation reset, the simulations may not work as expected when switching between them. This effect is under investigation and may be fixed in a future update. Use with caution.

- When you've completed your simulation session, you can stop the simulation, optionally **Send pose to host**, and then exit the VWD program (close the main VWD window with the 'X' in the upper right corner), returning to your host-application to enjoy your results.

Dynamic simulation controls:

- **Video** (def = unchecked): enable **Scene viewer** tab recording during this dynamic simulation session. This can be useful to replay a simulation at the 'real' playback speed or to record simulations for tutorials and/or demonstrations. It can also be used to review an early part of an interrupted simulation, helping you decide if the entire simulation should be completed. If the simulation calculations can't be completed at 30 frames per second (e.g. too many vertices in the scene), this recording may have problems. The video width and height settings should match the size of your current monitor.
- **Show dialog box** (def = unchecked): display the video CODEC selection dialog, allowing you to choose your preferred video encoder from the available CODECs on your system (**Note:** the free XVID CODEC is a good choice as it is well-supported).
- **Start dynamic simulation:** Start a simulation using the **dynamic** mode. For each animation frame in the host-application, VWD will request and update the scene elements at that frame and continue the simulation, responding to the changes occurring in the scene as they interact with the current state of the dynamic-actor. Once a static simulation is running, consider these changes and controls of interest:
 - Once the dynamic simulation has started, the **Start dynamic simulation** button changes to **Stop dynamic simulation**, allowing you to stop/pause the static simulation at any time. This can also be considered a 'pause' function. You can **restart** and **pause** the simulation by pressing this button repeatedly (a toggle). Upon completion of the simulation, this button becomes **Send animation to host** (described below).
 - The length of this simulation is dependent upon the number of animated frames registered in the host-application.
 - The dynamic simulation will process all frames, and in fact it must do so if you wish to send the results to the host-application without error.

- The activation checkboxes and strength values of gravity, inertia and air-resistance can be changed while actively running the simulation, or when paused.
- Once you have started a dynamic simulation using the ***Start dynamic simulation*** button, the nearby ***Start static simulation*** option will be disabled for the remainder of this VWD session. By pressing the **Alt** key with the ***Start dynamic simulation*** button, the entire scene and simulation will be reset, and both static and dynamic simulations will be available again. Press either button to begin a new simulation.

Hint: When your dynamic simulation is stopped/paused or complete, the ***Send pose to host*** button (a little bit above this button) can be also used to save snapshots of the animation and treated as described above. This can be done when the dynamic simulation is paused or completed, and with any available frame. One advantage of this is the ***Send pose to host*** results can be saved *with* a host application scene, and animations cannot.

- ***Send animation to host:*** Only after the *entire* dynamic simulation is complete (all animation frames are processed) will this button be available (in place of the ***Start/Stop dynamic simulation*** button). Pressing this button will send the simulation results to the host-application, and *terminate the current VWD session*, returning your control back to the host-application with the results of your simulation session.

Warning: Using the ***Send animation to host*** button to send partially simulated animations (where not all frames are simulated) to DAZ Studio may seem to work, but may corrupt the internal DAZ Studio scripting system, and possibly lead to undesirable errors later in the same DAZ Studio session. Please let the simulation complete all animated frames and *then* use the ***Send animation to host*** button. If you happen to see this problem in DAZ Studio, simply save your work and restart DAZ Studio to refresh the scripting sub-system.

- ***Per frame customization*** in a completed animation simulation: Anytime before using the ***Send animation to host*** button, you can pick a frame in the ***Scene viewer*** viewport's animation sequence slider control, press and hold the **Shift** key, acknowledge the warning, pick any frame in the animation series, and then perform static simulation actions on that frame.

You can perform a full static simulation on the given frame, starting from the state of the objects in that frame, or you can simply tug at parts of the dynamic-actor's mesh and 'fix' small issues. Note that only this frame will be affected, and it will likely be out-of-sync with its surrounding frames in the animation, but there may be cases where this feature is useful for special effects or small frame-fixes. Hint: disable inertia, gravity (and wind?) to ensure that only your intentional changes will impact the scene in this mode.

This one-off frame editing sequence can be repeated for any-and-all frames in the current ***Scene viewer*** tab animation sequence (simply move the ***Scene viewer*** frame pointer to the desired frame and press **Shift** and adjust accordingly...). You can then ***Send animation to host*** as you would normally.

- To **reset** an entire Dynamic simulation (returning the scene elements to their original 'imported' states), press the **Alt** key while mouse-clicking on the ***Start dynamic simulation*** button.

Important Note: If this button is labeled *Send animation to host*, you can change the button's function back to *Start dynamic simulation* by pressing the **Alt** key while the mouse is simply *over* the button, and it will switch the button's function back to *Start dynamic simulation!* This is very handy, but not at all obvious.

- **Stop at end** (def = checked): Remain in the VWD application after the simulation has completed processing all of the animation's frames. When unchecked, these frames will automatically be transferred to the host-application when the simulation is complete, and the VWD program will terminate, returning you to the host-application.
- **Follow** (def = unchecked): follow the dynamic-actor with the *Scene viewer* camera as the simulation processes the animation frames. This is useful when the interesting actors migrate out of the camera view as the animation progresses. When unchecked, the camera does not move with the scene elements. Note that the camera view can always be adjusted manually.
- **Use multithread** (def = checked): Use the CPU and any multi-threading available on multi-processor machines. This capability significantly accelerates simulation calculations.

Use multithread value (def = numprocs * 2): Number of processing threads to use during simulations. More is usually better for the simulation speed, but the 'best' value will vary with each system. Probably should not be bigger than twice the number of 'real' cores in your system. This should be automatically set by VWD, but you can experiment.

Important: This checkbox is where you determine whether your GPU or CPU will be used to run the current simulation. If the *Use multithread* button is checked, the CPU will be used, otherwise the GPU will not be used.

Use GPU groupbox:

- **Use GPU** (def = depends on context): Use an available GPU subsystem if available on your system, these might be able to help increase your simulation processing speed. When checked, the GPU that's showing in the combobox will be used. The default seems to change depending on the use of configuration assistants.

Use GPU combobox (def = system dependent): Select an available GPU subsystem/processor, if any are available.

Note: While the CPU/GPU control settings may be available after a simulation reset, the simulations may not work as expected when switching between them after a simulation reset (Alt-click on the 'Start ...' button). This effect is under investigation and may be fixed in a future update. Use with caution.

Note: The simulation results using CPU may differ from those generated by a GPU. The differences will vary with the meshes and forces used, so experimentation is required...

- **Use double stick** (def = unchecked): Enable extra simulation stabilizing calculations. This feature is recommended when extra CPU or GPU resources are available to refine the simulation results. Enable this function when you want to use high parallelism on a dynamic-actor composed of relatively few vertices. When unchecked, these stabilizing optimizations will not be used in the simulation.

- ***Sticks CPU*** (def = checked): Enable extra simulation stabilizing calculations. As above, this is recommended when meshes 'explode' or otherwise behave oddly during a simulation. It is always safe to enable this optimization, but it may add time to the simulation processes.

Other dynamic simulation controls:

- ***Vertices interactions*** (opens a new tab): Establish a custom relationship between vertices on collision-items and vertices on the scene's dynamic-actor. Pressing this button will open a new tab (***Vertices interactions***) that is used to configure these relationships. The details of the ***Vertices interactions*** tab will be described in a later section.
- ***Simulation history*** (opens a new tab): (Re)load and optionally edit parameters used in a saved or recent simulation session. This can be particularly useful when trying many different settings on the same scene. The controls and use of this tab will be described in a later section.

Dynamic deformation groupbox:

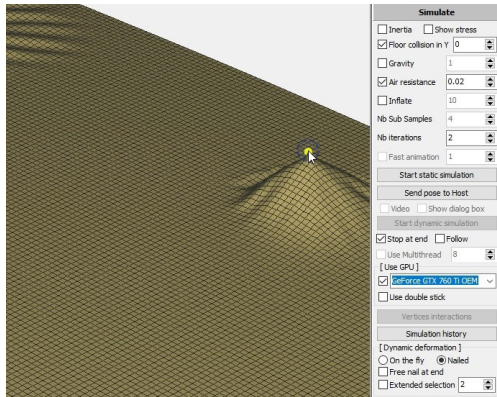
The Dynamic deformation tools provide a powerful way to make live adjustments to a static simulation scene's dynamic-actor while actively simulating, or to adjust frames of a completed dynamic/animation simulation. Initiated by pressing and holding down the **Shift** key, the VWD simulator will allow you to grab and drag single or multiple vertices on the dynamic-actor mesh, and move them around within the running simulation. disabling ***Gravity*** and/or ***Inertia*** in the ***Simulate tab*** may improve the precision of this process.

When dragging a vertex or vertices, the sticky-ness of the mouse *and* mesh can both be adjusted to facilitate various levels of control and natural flow in the deformations:

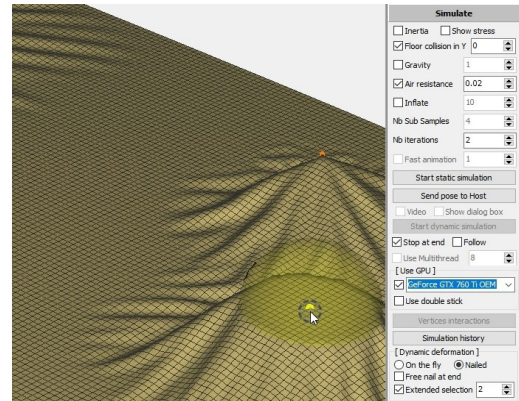
- ***On the fly / Nailed*** (def = Nailed): When performing Dynamic deformations, the mouse can be set to be very sticky (***Nailed***), or semi-sticky (***On the fly***). When ***Nailed***, the active vertex or vertices track the mouse cursor exactly. When ***On the fly***, the active vertex or vertices will lag and 'let go' of the mouse cursor after some brief motion, resulting in a lighter but repeated tugging effect on the mesh.
- ***Free nail at end*** (def = checked): After releasing the mouse when dragging vertices, the mesh can either be 'nailed' exactly where it was released (unchecked), or it can be left to float, responding freely to any active simulation forces occurring while the **Shift** key remains pressed.

Note: To release a 'nailed' vertex, re-check the ***Free nail at end*** checkbox and **float over the point, rolling the middle button of your mouse** (a bug?) until it turns red, then click on the point while it's still red. The nail point should disappear and release the mesh.

- **Extended selection** (def = unchecked): Rather than dragging only one vertex at a time (default, w/ no Extended selection enabled), this option adds an extended sphere of deformation influence around the mouse selection cursor. Vertices within this sphere will be included in the deformation adjustment. When unchecked, this feature is disabled.
 - *Extended selection value* (def = 2, range:): the radius of the sphere-of-influence, where vertices within this sphere will be included in the deformation adjustments.



*Normal single vertex
dynamic deformer*

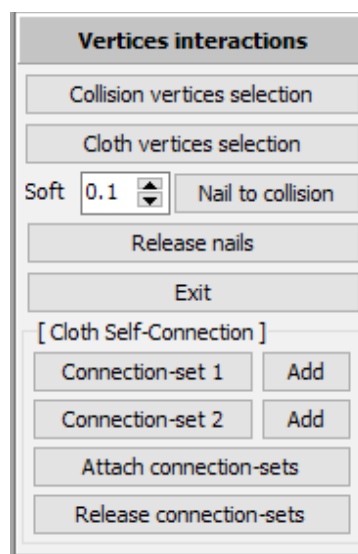


*Extended selection
dynamic deformer*

5.3.1 The '*Vertices interactions*' sub-tab

The *Vertices interactions* tab offers two powerful functions:

1. Situation-specific attachments can be created between the dynamic-actor and nearby collision-items, effectively gluing them together for some or all of the simulation. This is useful for situations like attaching a handkerchief to a figure's hand, or earrings to a figure's earlobe.
2. This tab also provides tools (in the *Cloth Reduction vertices* groupbox), to connect disconnected elements on the same dynamic cloth actor. This is a very simple way to glue disconnected cloth parts together if they cannot both be attached 'together', using a common collision-item. Some un-welded cloth meshes (that fall apart) can be fixed with this tool.



This tab is accessed from the *Vertices interactions* button in the *Simulate* tab. This button is only available before the simulation starts or when an active simulation is paused (using either the *Stop dynamic simulation* or *Stop static simulation* in the *Simulate* tab, depending on which simulation you're running).

By simply selecting vertices on each item and 'nailing' them together (*Nail to collision*), you can virtually 'glue' meshes together during the simulation. Because items can be attached and released at any time before or during a (paused) simulation, some very powerful effects can be realized in both static and dynamic/animated scenarios. One example might be for an animated figure to pick up and drop a cloth handkerchief using this feature.

This is also where you can connect disparate dynamic-actor mesh areas together as well, essentially welding parts of the dynamic-actor's mesh(es) together in the simulation. This allows you to 'glue' different parts of your dynamic-actor mesh together as if they were welded. If you change the state of these welds during the simulation, you can create scenarios like velcro hooks attaching to the loops of the same clothing during an animation.

Vertices interactions tab header controls:

- **Alt-click** on the **Simulate** tab header while pressing **Alt** to reset this tab's settings to their default values.
- **Reduce the tab** to its header by **right-clicking** on that header. Repeat this to re-expand the tab (a toggle).

Vertices interactions controls:

The following attach/detach actions can be performed anytime before a simulation is started, or during a simulation after temporarily pausing the simulation using the **Stop dynamic simulation** button in the **Simulate tab**:

- **Collision vertices selection:** Place the currently selected collision-item vertices into the collision-item vertices list/array. These vertices will be attached to the active dynamic-actor.
- **Cloth vertices selection:** Place the currently selected dynamic-actor vertices into the dynamic-actor vertices list/array. These vertices will be attached to the active collision-item vertices.
- **Nail to collision:** Attach the selected collision-item vertices and dynamic-actor vertices. The softness of the connection is set in the field on the left.
- **Soft** (def = 0.1, range 0.01 to 1.0): Softness value of the springs connecting the collision-item with the dynamic cloth actor using the **Nail to collision** button on the right. Higher values result in softer connections, meaning these connections will have more 'play' during the simulation.

Tech note: this connection necessarily uses the extension by-neighborhood method to attach the two items.

- **Release nails:** Release the attached vertices between the items.
- **Exit:** Leave this tab, with any committed settings intact.

Cloth Self-connection groupbox: Connect different parts of the same dynamic cloth actor's meshes together - as if temporarily sewing or gluing them together during the simulation. Sometimes a dynamic-mesh is composed of separate parts that aren't welded together. This can also be used for other attachment effects as well.

- **Connection set 1 / Add:** Create the first selection-set for the connection (to Connection-set 2).
- **Connection-set 2 / Add:** Create the second selection-set for the connection (to Connection-set 1).
- **Attach connection-sets:** Create a connection between the first and the second selection groups. This connection is firm (very little softness, or 'play').

Hint: You can repeat this 'connecting' procedure multiple times on the same dynamic cloth actor, meaning you can attach multiple buttons, sleeves, ribbons, and/or other parts of the same cloth item to a common/core section of the same cloth item. This could also be used in more novel situations, like attaching ropes to a hot-air ballon, etc.

- ***Release connection-sets:*** Remove any/all connecting springs between the two connection-set groups. This button can quickly undo a lot of work...

Note: It may be necessary to remove the **gravity** and **inertia** forces in the ***Simulate*** tab to achieve the desired effect.

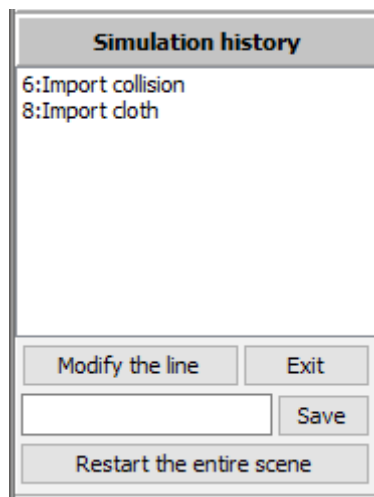
5.3.2 The '*Simulation history*' sub-tab

The *Simulation history* sub-tab lets you re-run the series of 'actions' that were taken to create the current scene, effectively restarting the current session/scene from scratch, and pausing just before the simulation process would start.

This tab is accessed from the *Simulation history* button in the *Simulate* tab.

Before restarting the session/scene, it also allows you to make edits to each recorded 'action' that makes up the scene. You can also save the current series of 'actions' - either as-is (when you opened this tab), or after making edits.

This tool is particularly useful for saving, reloading, and adjusting complex scene import and setup configurations. Automatically generated settings from the various assistant tools appear here and can also be saved, loaded, and edited as well.



When this tab is opened, a listbox is displayed, containing a list of the VWD 'actions' that were taken to create the current scene, including all of the steps taken to import and configure the currently loaded scene elements. Note that some of these steps may have been executed by the import assistants. These list elements are labeled with their action (e.g. "Import", etc.) and their related 'role' in the scene (e.g. collision, hair, cloth).

Simulation history tab header controls:

- **Alt-click** on the ***Simulate*** tab to reset this tab's settings to their default values.
- **Reduce the tab** to (only) its header by right-clicking on that tab. Repeat this to re-expand the tab (a toggle).

Simulation history controls:

- **Select and Edit the Imported scene item:** Upon mouse-clicking on a listbox item, the appropriate parameter tab (e.g. collision, hair, cloth) is displayed with the item's current settings, allowing you to make edits to those settings. (base settings only - ***Forces and springs*** settings cannot be edited in this dialog)
- **Modify the line:** If a scene item is selected from the listbox and changes are made in the relevant parameters tab, this button will 'commit' those changes to this VWD session, and the next simulation run will reflect these settings. Such changes will also be used by the ***Save*** function (below) when used after this button is pressed.
- **Exit:** Exit this history dialog and return to the ***Simulate*** tab. If any changes were made and the ***Modify the line*** button was pressed for each change, these changes will still be used in the next simulation, otherwise this button acts like a 'Cancel' action, and the current session settings remain unchanged.
- **[/] Save:** Save the current state of the simulation history actions (in the listbox above) using the filename typed in the field to the left and then pressing **Enter**. Note: There is no confirmation, and it will overwrite any existing file of the same name. If edits have been made and committed using the ***Modify the line*** button, these edits will be saved, even if the scene is not restarted.

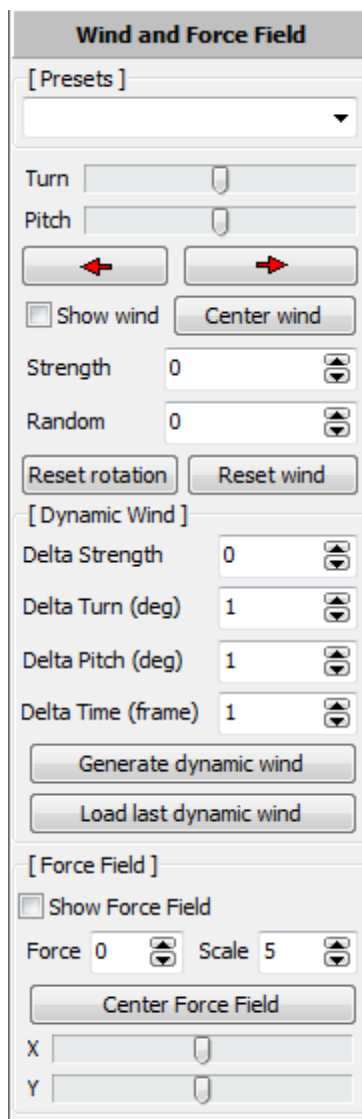
These ***saved*** actions can be loaded in a future session using the ***Scene import*** -> ***Recorded imports*** loading combobox list.

- **Restart the entire scene:** Press this button to clear the current VWD scene and reload the actions specified in the listbox. This means that if any changes were made to the scene items ***and*** the ***Modify the line*** button was pressed, those changes will be reflected in the current reload process. If no edits were made (or the ***Modify the line*** button wasn't pressed), the scene settings that were active when the ***Simulation history*** button was pressed (and this tab was opened) will be used in the scene reloading process.

This button may be useful to simply refresh a current scene exactly like it was loaded and configured - in doing so, cleaning up any memory and processing artifacts that might negatively impact a complex simulation. Kind of a scene reboot.

5.4 The 'Wind and Force Field' Tab

To enable external wind and/or a directional force that will interact with your session's dynamic-actor (cloth or hair), open the **Wind and Force Field** tab and adjust the controls as you see fit.



The **Wind** function lets you set direction and power ranges for a randomizing calculation that creates a varying wind-like force over the frames of an active animation (for dynamic/animated simulations). The main direction and amount of variation from that direction can be set, as well as the amount of wind and the strength, length, and regularity of gusts that will occur during the simulation. These settings can be generated, saved, and re-used for multiple dynamic-actor sessions (like clothes and hair items that should all experience the same wind settings).

The **Force Field** function is similar to the **Wind**, but acts with a constant force as if radiating from a *spotlight* or point-source during the simulation.

Wind and Force field tab header controls:

- **Alt-click** on the *Wind and Force Field* tab to reset this window's settings to their default values.
- **Reduce the tab** to (only) its header by **right-clicking** on that header. Repeat this to re-expand the tab (a toggle).

Wind and Force field controls:

Important: Remember that *none* of these *Wind* settings will affect the dynamic-actor during the simulation unless some or all of its vertices have been enabled with the *Apply react to wind* button in the *Forces and springs* tab!!! You can also deactivate vertices that are currently set to be affected by wind by selecting them and then pressing that *Apply react to wind* button while pressing the **Ctrl** key (it will change to *Remove react to wind*).

Note: Other objects in the scene do not impede the wind forces. The wind generator generates a parallel force relative to the dynamic-actor.

Presets groupbox: Presets, or collections of settings for pre-generated or saved wind force sequences can be saved and/or reloaded in this section.

- [*combobox*]: Use this combobox to read and save recorded wind settings.

To **save** the current wind sequence, simply type a meaningful name in the combobox and press **Enter**.

To **load** an existing wind setting sequence, click on the combobox and select the name of the desired preset.

Hint: It is often desirable to generate and save a 'good' wind preset for an animation, then re-use that same preset for each of the dynamic-actor simulation sessions in that animation project. This way, all of the dynamic-actors in the scene (hair, clothes, etc.) will seem to be coordinated and responding to the same wind force during the final animation sequence.

Note: These wind settings preset files are stored on your computer in a file folder called

.../VWD/Recorded Wind/

located in your VWD main program folder (e.g **C:\VWD\...**)

The wind settings files are named with a **.WND** suffix. You can use any Windows file-management tool to save, restore, and rename these files as you see fit. This may be useful when saving and archiving related project files or your favorite settings files for later use.

Note: The wind preset files may be limited to being used in simulation/animation sessions that match the animation frame-count of the wind preset file. This may be fixed in a future update.

- **Turn** (def = 0; range: -180 to 180): Set the left/right/horizontal directional element of the wind. This is a baseline starting point value for generated wind variations and tends to be the 'average' direction of a generated wind sequence.
- **Pitch** (def = 0; range: -90 to 90): Set the up/down/vertical directional element of the wind. This is a baseline starting point value for generated wind variations and tends to be the 'average' direction of a generated wind sequence.
- **Left / Right Arrows**: Offset controls for displaying the wind direction indicator (when visible) in the **Scene viewer** viewport. This is useful when the indicator arrow interferes with the scene elements. These controls do not change any of the wind force settings (direction or strength, etc.). Only the wind indicator location is changed for better viewing.
- **Show wind** (def = unchecked): Show the wind direction indicator in the scene viewer. This helps to orient the desired wind direction when using the **Pitch** and **Turn** sliders.
- **Center wind**: Place the wind indicator arrow (if visible) at the center of the current **Scene viewer** tab view - which is the position of the camera target, not necessarily the center of the scene. This does not change the actual wind settings - just the location of the wind direction indicator.
- **Strength** (def = 0, range: 0.0 to 50.0): Strength of the wind force. (Roughly the average force when randomized...)
- **Random** (def = 0, range: 0.0 to 3.0): Amount of randomizing that will be applied to the wind force. Higher values will increase the random influence. This value is also applied to the **Dynamic wind** groupbox settings (below).
- **Reset rotation**: Zero the **Turn** and **Pitch** directional controls.
- **Reset wind**: Reset all wind settings, except rotations (**Pitch/Turn**), to their defaults.

Dynamic Wind groupbox: Wind force settings that vary over time (Only relevant during dynamic/animated simulations). Higher values for these settings result in more volatile/dynamic changes in those characteristics over the simulation time-frame. Thoughtful use of these settings can result in very realistic simulations.

- **Delta Strength** (def = 0.1, range 0.1 to 30.0): Strength variation range - the range, +/- of the randomized **Strength** sequences during the animated simulation. e.g. A value of 10 will confine the **Strength** range between -10 and +10 for a total of 20 units of **Strength** variation during the animated simulation.
- **Delta Turn**: (def = 1, range 1 to 150): **Turn** variation range - the range, +/- of the randomized **Turn** sequences during the animated simulation. e.g. A value of 10 will confine the **Turn** range between -10 and +10 degrees for a total of 20 degrees of **Turn** directional variation during the animated simulation.
- **Delta Pitch**: (def = 1, range 1 to 45): **Pitch** variation range - the range, +/- of the randomized **Pitch** sequences during the animated simulation. e.g. A value of 10 will confine the **Pitch** range between -10 and +10 degrees for a total of 20 degrees of **Pitch** directional variation during the animated simulation.

- **Delta Time:** (def = 0.1, range 0.1 to 30.0): **Time** variation range - the range, +/- of the randomized **Time** ranges during the animated simulation. e.g. A value of 10 will confine the **Time** range between -10 and +10 time units for a total of 20 units of **Time** variation during the animated simulation.
- **Hint:** Longer times produce slower/smoothier changes in the active characteristics (**Strength/Turn/Pitch**). Shorter times produce quicker and more volatile changes in those influences, resulting in more abrupt changes in the wind forces.
- **Generate dynamic wind** (for dynamic/animated simulations): Generate a sequence of semi-random wind values for the frames of this animation. These values will act as a changing wind continuum, where the variation between frames will be minimal, but the variation across the overall animation sequence will be driven by the above settings (especially 'delta' settings).

Note: The generated wind sequence will start with the **pitch**, **turn**, and **strength** settings that are set when you click on this button. This means if you wish to create a wind sequence that starts and varies from a particular direction, you should set the initial **pitch**, **turn**, and **strength** settings to your preferred starting settings before using this function.

- **Load last dynamic wind** (for dynamic/animated simulations): Load the most recently generated wind sequence again for this dynamic/animated simulation, rather than generating a new wind sequence from the above settings.

Hint: Using this option is the only way to generate a collection of cloth and hair simulations in the same scene, that respond to the *same* wind forces. Using this option during the import process of a blouse, skirt, and hair in the same scene will result in consistent wind forces during these three simulation sessions.

Hint: To see and verify that the specific wind sequence is what you would like (e.g. it has enough variation, etc.) before you run your simulation, you can slide the *scrubber* control in the **Scene viewer** tab viewport *as you watch* the Wind forces **Pitch** and **Turn** sliders and the **Strength** and **Random** values vary through the sequence of frames. Enable the **Show wind** check box to see the directional indicator vary through the sequence.

- **Note:** The initial conditions and animated sequence of the wind are currently stored in the VWD program's **Exchange** folder in two files: The **Hair animation.rsp** file contains the sequence of wind settings that are applied to each frame, and **Saved Wind.rsp** file contains the initial settings that were used to generate that sequence file. These files are both overwritten each time a new wind sequence is generated, and must be manually saved (with your scene) and restored, if you wish to return to a scene later. The number of wind variation settings (one for each scene frame) in the **Hair animation.rsp** file depends upon the current scene's animation frame count (plus some random extra frames). This means that the same wind-frame sequences probably will not work well across different host-application scenes unless the frame counts are the same.

Force field groupbox: The **Force Field** settings. This force is similar to gravity, but more flexible in its controls (source location, strength). This feature can be useful to modify a dynamic cloth actor with too many folds for example, but it can have a many useful applications. It is easiest to treat this force like a point light, where the energy emanates from a source point and travels in all directions. Objects in the scene do not impede the forces. Distance from the force doesn't impact energy levels, but will impact apparent direction (e.g. the farther away the force generator, the more parallel the force will be to a dynamic item).

Important: Remember that *none* of these **Force Field** settings will affect the dynamic-actor during the simulation unless some or all of its vertices have been enabled with the **Apply force Field** button in the **Forces and springs** tab!!! You can also deactivate vertices that are currently set to be affected by these forces by selecting them and pressing the **Apply force Field** button while pressing the **Ctrl** key (It will change to **Remove Force Field**).

- **Show Force Field** (def = unchecked): Show the **Force Field** indicator in the scene viewer. This helps to orient the desired **Force Field** direction when using the **X** and **Y** sliders.
- **Force** (def = 0.0, range: -200 to 200): Amount of directional energy that will be applied to each dynamic-actor vertex during the simulation, as sourced from the **Force Field** generator (use the **Show Force Field** checkbox to view the indicator).
- **Scale** (def = 5, range 1 to 9): Set the *size* of the **Force Field** indicator. This does not change the actual **Force Field** settings - just the size of the **Force Field** indicator.
- **Center Force Field:** Place the **Force Field** generator at the center of the current **Scene viewer** viewport - which is based on the position of the camera, *not* necessarily the center of the scene.

Note: remember that the **Force Field** generator can be compared to a spot-light that sends out its force in all directions. The location of this indicator will be the source of the **Force Field**.

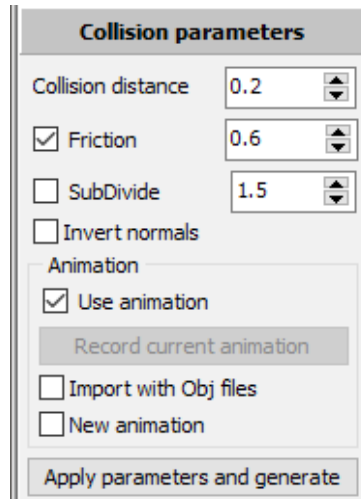
- **X / Y** (arrow sliders) : Use these sliders to move the **Force Field** generator left/right (X) or up/down (Y) on the **Scene viewer tab** viewport. (The **Force Field** generator's direction is always relative to your current view).

Note: remember that the **Force Field** generator can be compared to a spot-light that sends out its force in all directions. The location of this generator will be the source of the **Force Field**.

Note: Even if only a small number of vertices on a mesh are configured to respond to the **Force Field** during the simulation, the rest of the mesh will respond to those areas as might be expected, depending on the stretch, scale, and softness of the springs that connect the nearby mesh. This can be used for some very interesting and useful effects.

5.5 The 'Collision parameters' Tab

The *Collision parameters* tab is where the available collision-item parameters are set as each is imported. These settings determine how the dynamic-actors in a scene will respond to each collision-item during the simulations.



There can be multiple collision-items in a scene, and each may interact with the dynamic-actor as the simulation progresses. As in the real world, not all items in a scene are alike, and characteristics (like surface friction and certain 3D-mesh features) can be adjusted here to help generate the desired effects.

Collision parameters tab header controls:

- **Alt-click** on the *Collision parameters* tab to reset this tab's settings to their default values.
- **Reduce the tab** to its header by **right-clicking** on that header. Repeat this to re-expand the tab (a toggle).

Collision parameters controls:

- ***Collision distance*** (def = 0.2, range: -1.0 to 1.0): This distance setting determines where the dynamic-actor mesh in the scene will begin to 'feel' the influence of *this* collision-item's mesh and be repelled. It would be expected that small distances here are preferred for realism, but many simulations will be improved by using larger values. Fast-moving animations, sparse 3D-meshes, and simulations that suffer from inter-penetrations and/or poke-through may show improvement when this value is higher. When this value is too large, it may become obvious that meshes are repelling even though they aren't yet touching. The default value of 0.2 is a recommended starting point.
- ***Friction*** (def = checked): Enable friction forces on the surface of this collision-item. The amount of friction involved is determined by the value setting on the right. When unchecked, no friction forces will be applied between this item and the dynamic-actor (this collision-item will behave as if it's very smooth and slippery).
 - ***Friction*** value (def = 0.6, range: 0.0 to 1.2): The friction value of this collision-item. Lower values result in less friction - 0.0 is slippery (no friction), and 1.2 is sticky. A value of 0.3 seems to work well in most cases. This is only meaningful if ***Friction*** is enabled.
- ***Subdivide*** (def = unchecked): Add mesh density to the collision-item to improve interactions with the dynamic-actor. The amount of subdivision is determined by the value setting on the right. When unchecked, the mesh is not subdivided for the simulation. This is useful when the dynamic-actor interpenetrates the collision-item, especially on sparse collision-item meshes, or fast animations. This subdivision is only active during the simulation and does not affect the original collision-item.
 - ***Subdivide*** value (def = 1.5, range: 0.5 to 3.0): Any polygon edges longer than this length will be subdivided *until they are equal or smaller than this length*. Making this value too small may result in many subdivisions, impacting memory and performance during the simulation. This value is only meaningful if ***Subdivide*** is enabled.

Hint: When collision meshes are too sparse for the simulation (too fast or the mesh densities don't match well), and inter-penetrations/poke-through occurs, setting this to 0.7 or so seems to generate a collision-item mesh that's dense enough to interact well.

- ***Invert normals*** (def = unchecked): When checked, the collision-item's polygon face normals will be inverted/reversed during the import.

Animation groupbox: These settings are relevant when running dynamic simulations, where the simulation happens across animated frames in a scene.

- **Use animation** (def = checked): If this collision-item is part of a dynamic simulation and it is animated, this should be checked. However, if the collision-item is *in* an animated scene but *not* animated (e.g. like furniture items), uncheck this setting and the item will only be imported once, saving import and processing time on each additional simulation frame.
- **Record current animation:** Record the animation sequence of the current collision element and cache it locally to a VWD animation cache-file. This initial processing step will increase the speed of any dynamic/animated simulations that follow. It also has the effect of making the simulations flow more smoothly during the simulations, giving a better visual sense of the final result.

Note: The **Record current animation** option may only be available after the item is imported. To explicitly pre-record the animation cache-file, select the (already imported collision) item in the **Scene import** tab, open the **Collision parameters** tab check the **Use animation** button, check the **New animation** button, then press the **Record** button and wait for the animation to complete in the **Scene viewer** viewport. You can then continue to operate VWD normally, and the animation cache will be used when needed.

Note: If you only run a simulation once, this feature won't save you any time. If you run the simulation multiple times in one session, or in multiple sessions, the simulation will run more smoothly (less delay between frames) because of this pre-processing.

- **Import with Obj files** (def = unchecked): Use for collision-items that don't simulate Rather than process the animated collision items with raw mesh information, create and use temporary OBJ files during the simulation. This processing method is slower, but it allows some imported collision-items to work if they fail to simulate otherwise. You probably only need to use this when a dynamic simulation fails to start (probably because of an import file problem).
- **New animation** (def = checked): Force a fresh calculation and caching of the collision-item's animation frames from the host-application. Any previously recorded animation cache files will be replaced. If unchecked, VWD will scan the cache-directory for a plausible recording cache-file (vertex count and frame-count) and use it, if found.

Warning: You will need to (re)enable this option (to refresh the animation cache-file) if you have changed any of the item's timeline animation without having changed the number of frames on the timeline between VWD sessions.

- **Note:** A new animation cache-file will be generated when this button is pressed and the simulation is actually run. The cache-file will automatically be used for all sessions after that, assuming:
 - The **New animation** button is not pressed,
 - The cache-files are not cleared or removed
 - The frame-count on the timeline has not changed since the animation cache-file was generated.
- **Apply parameters and generate:** Prepare and import the current collision item using the current parameters. This is the same as clicking the larger **Collision** button in the **Scene import** tab when visible.

5.6 The 'Dynamic parameters' Tab

When importing a dynamic cloth or hair actor, a variety of settings can be set to establish the general nature of the actor. The **Dynamic parameters** tab is where you configure these parameters.

The screenshot shows the 'Dynamic parameters' tab with the following settings:

- Stretch:** 0.01
- Softness:** 0.01
- Node weight:** 1
- [Self Collision]**
 - ☐ Apply distance: 0.5
 - Detection distance:** 1
 - ☒ Detection by extension
 - Method:** By distance
 - Direction:** Two directions
- ☐ SubDivide: 1
- [Extended stiffness]**
 - ☒ Use vertices extension
 - Extension count:** 3
 - Softness:** 0.01
 - ☐ Use vertices neighborhood
 - Distance:** 0.5
 - Softness:** 0.01
 - ☐ Distance min: 0.25
- ☐ Tetra meshing: 0.01
- Apply parameters and generate** button

These **Dynamic parameters** define the overall nature of the dynamic cloth or hair actor. Many of these characteristics can be further adjusted in other tabs through the process of selecting specific areas of the cloth or hair actor and applying specific settings to those areas, but the initial base settings that define the cloth or hair actor are set in this tab.

Note: In the previous version of VWD, this tab was the only way to configure the dynamic-actor properties during the import process. With the addition of the new **Cloth** and **Hair assistant** tabs, using this 'manual' configuration option *in place of* the **Cloth** or **Hair Assistant** tab settings may be desirable or necessary for certain effects, but also may result in the generation of additional simulation springs, resulting in memory issues in complex scenes. The **Cloth** or **Hair assistant** may use VWD's *Animated by host* technique and other optimizations to notably reduce the dynamic-actor's mesh areas that are involved in the simulation. Using these features *with* the **Cloth** or **Hair assistant** tab settings is fine, and often expected.

Dynamic parameter tab header controls:

- **Alt-click** on the *Dynamic parameter* tab to reset this tab's settings to their default values.
- **Reduce the tab** to its header by **right-clicking** on that header. Repeat this to re-expand the tab (a toggle).

Dynamic parameter controls:

- **Stretch** (def = 0.01, range 0.0001 to 1.0): Level of the elastic force between mesh vertices along the polygon edges. This force determines how much the polygon edges will tend to stretch and try return to their initial state. Higher values result in more stretchy/elastic actors.
- **Softness** (def = 0.01, range: 0.0001 to 1.0): The softness, or tendency for a polygon face to bend between vertices. Higher values result in surfaces that deform or bend more easily across the surface.
- **Node weight** (def = 1, range: 0.01 to 100.0): Adds virtual density/mass to the dynamic-actor. While this setting should not need adjustment, it may be useful for specific effects.

Self collision groupbox: Detect and prevent self-inter-penetration and self-poke-through within the dynamic-actor's own mesh. There are two checks available in this method: distance of detection and distance of repelling.

- **[enable self-collision]** (def = unchecked): Enable the self-collision processing on this dynamic-actor using multiple detection methods (described below). Uncheck this to disable this self-collision prevention method.
 - **Apply distance** (def = 0.5, range: 0.1 to 1.0): The distance between dynamic-actor mesh vertices where repelling will begin to occur. This works well for meshes that approach each other face-to-face, or flat-surface to flat-surface. Long flowing robes, flags, and bed-sheets may work well with this method.
 - **Detection distance** (def = 1, range: 0.6 to 2.0): For each vertex on the dynamic-actor, this is the shortest distance between nearby vertices where we believe a self-collision is not possible. This value is powerful, but if it is set too high it can result in long processing preparation times. Use larger values for faster moving mesh interactions. Tight clothes like leotards can use low values, and pleated skirts should use higher values.

Hint: This self-collision detection method ensures that pairs of vertices of the dynamic-actor never get closer than this distance. This is particularly useful in edge-to-edge collisions like ribbons, some hair meshes, and pleated dresses - situations where face-to-face mesh collisions won't be well detected and repelled.

- **Detection by-extension** (def = unchecked): Check this to enable distance detection using the by-extension method. Uncheck this for distance detection by-neighborhood.

Detection by-extension only considers the distance between vertices that are connected by edges on the mesh (imagine a radius on the mesh around each vertex). This detection method should be used if the original dynamic-actor has many folds, like a pleated skirt.

Detection by-neighborhood considers the distance between any vertices on the mesh, from any direction (imagine a sphere around each vertex). Detection by-neighborhood is faster, but not good for all detection situations. Use this for flowing robes or bed-sheets and blankets.

- **Method** (def = *By distance*): Choose *By distance* to base your calculations on the distance between vertices. Choose *By normals* to base the distance calculations on the space between polygon face normals.
- **Direction** (def = Both directions - only enabled if the **Method** is set to *By normals*): Set which polygon 'facing direction' to use for the distance detection calculations. Choose *Interior*, *Exterior*, or *Both directions*, depending on your dynamic-actor's mesh. Choosing *Both directions* is probably the safest setting, but will take more time to simulate.

Hint: In some cases, it may be useful to work in only one direction. For example, if you want to be sure the self-collision will only repel the vertices in the direction of the dynamic-actor's normals.

- **Subdivide** (def = unchecked): Check this to apply temporary subdivision to all dynamic-actor mesh edges that are longer than the distance set on the right. This will improve simulation results on 'sparse' meshes with larger polygons. If this distance is too short, it may result in many vertices, and consume a lot of memory and calculation time.

Important: the checkbox and value only have an effect when using **Poser** as the host-application. To get the same effect in **Daz Studio**, you'll need to enable and apply *subdivision* to the dynamic-actor item within DAZ Studio *before* importing it into Cloth & Hair.

Extended stiffness groupbox: To add stiffness to the current dynamic-actor, there are two methods available that work differently and therefore generate different effects. You can use **none**, **one**, or **both** of these stiffening methods to generate the desired effect.

- **Use vertices extension** (def = unchecked): By adding simulation springs from each vertex outward to each connected vertex (in a flat radius on the mesh), the dynamic-actor's mesh will become stiffer. By 'extending' out farther, or adding more 'hops' away from each starting vertex, more springs will be added, and more springs will overlap, adding more stiffness to the entire mesh. Check this box and set the parameters below to enable this stiffening method. One use of this method is to stiffen dynamic hair strands, but leave them independent of each other.
- **Extension count** (def = 3, range 1 to 12): The number of iterations, or outward hops, from each vertice that are included in the extension process. This number is like a radius of vertices around a center vertice (picture a spider web), so a few hops can quickly add a lot of stiffening area and force.
- **Softness** (def = 0.1, range: 0.0001 to 1.0): Softness of the springs that are generated using this stiffening method. The higher the value, the softer the springs in the by-extension area. Reduce this value if you wish to stiffen your dynamic (hair or cloth) actor.

Hint: When stiffening by vertices extension. In this case, the extension of the vertices is made using the mesh structure. In an extension, the vertices of the faces containing the vertex being processed are selected. This extension is linked to the mesh structure. A stiffening using the same value may give a very different effect because it depends on the size of the faces.

- **Use vertices neighborhood** (def = unchecked): By connecting springs from each vertex outward in all directions to any nearby vertices (in a sphere), even if not adjacent on the mesh, stiffness can be added to the surface, and *between* surfaces. One use for this is to add stiffness to hair strands, and also add attraction springs between the strands.
 - **Distance** (def = 1, range: 0.1 to 2.0): Distance from each vertex where connecting/stiffening springs are generated. Any vertices with the sphere of this radius will be connected with new springs.
 - **Softness** (def = 0.1, range 0.0001 to 1.0): Softness of the springs that are generated using this stiffening method. The higher the value, the softer the springs in the by-neighborhood area. Reduce this value if you wish to stiffen your dynamic (hair or cloth) actor.
 - [*Distance min checkbox*] (def = unchecked): Check this and set the distance to the right to skip stiffening spring creation on vertex pairs that are closer together than the distance set on the right. This optimization will save simulation time with very little impact on the results, if the distance value is kept fairly low. Stiffening between close vertices does not have much effect.
 - *Distance min* (def = 0.5, range = 0.001 to 2.0): Vertices closer than this distance will not have springs generated. If the distance is small, these reduced springs will not be noticed in the simulation. We recommend starting with a number that is 1/2 the **Distance** value set above.
- **Tetra meshing** (def = unchecked): Generate a tetrahedral mesh from the current mesh. This is like the simulator's inflation function, but with more substance – like filling a closed dynamic cloth actor mesh with air or jello. The mesh must a closed volume. This generation is fully automatic and requires that the free GMSH program is installed and available. The simulation will work with the external springs and also with the internal generated springs. This function can be useful to simulate a soft pillow or balloons.
 - [*Tetra meshing softness*] (def = 0.01, range: 0.001 to 2.0): Softness of the internal tetrahedron-mesh springs (not the surface mesh springs).

Note: see the **Utilities->Find GMSH program** button to define the location of the GMSH program after you've downloaded and installed it.

Important: Until further notice, please download and install the Version 3.0.6 version (32/64 bit) from <https://gmsh.info>, via their archive folders at:

<http://gmsh.info/bin/Windows/gmsh-3.0.6-Windows64.zip>

or

<http://gmsh.info/bin/Windows/gmsh-3.0.6-Windows32.zip>

as it seems that their newer versions have introduced some incompatibilities with VWD.

- **Apply parameters and generate:** Complete the dynamic-actor preparation process and import it into the VWD session, using the above configuration settings.

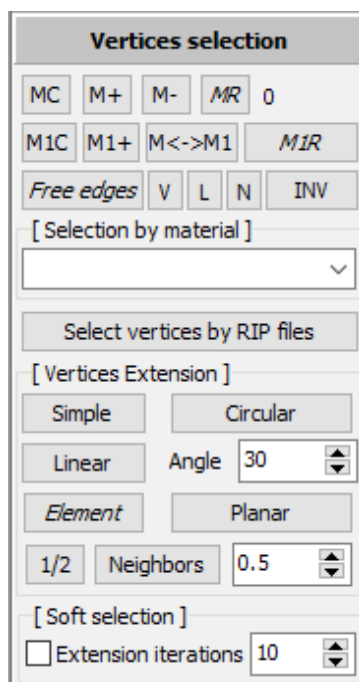
Note: This button will behave exactly like the ***Import cloth*** or ***Import hair*** button(s) when they are visible in the ***Scene import*** tab.

Other *Dynamic parameter* notes:

- Be watchful of the way these settings can easily consume memory and simulation time if they are not applied carefully. Sometimes this is necessary to achieve the desired effect, but other times you can save memory and processing time through thoughtful use of these settings.
- Clothes or hair are often fairly soft when imported into VWD as dynamic-actors. Additional stiffening is not unusual. must often be stiffened because the base rigidity is often too soft.
- These parameters apply to all of the dynamic-actor elements. There are no selection assignments in this step. Additional stiffening of specific selected areas can be applied in the ***Forces and Springs*** tab.

5.7 The 'Vertices selection' Tab

The *Vertices selection* tab is probably one of the most important tools in the VWD program. Almost every aspect of VWD's mesh preparation phases requires that the user select groups of mesh vertices, and assign them to some sort of function or behavior - critical functions like stiffness, attachments, collisions - all require that some or all of the mesh vertices be selected into groups before the settings can be applied.



Within this selection toolkit are some powerful controls, including traditional buffers and management tools to let you add, remove, and swap collections of selected vertices. There are also some 'convenience' selection buttons like the 'free edges' tool which can save a large amount of manual selection effort.

A very useful feature is the Selection by material combobox that lists the mesh's built-in material groups (if any), and allows you to select each, and using the memory buffer tools, add any number of these material groups together to form a complete collection of vertices for some action.

Specialized selection extension tools will help you add to vertex groups with handy algorithms that can save you a lot of manual vertex selection, especially on larger complex meshes.

After using these tools, especially the more specialize features, you will find many of your other modeling tools lacking, and wish you had these tools in those programs as well.

Before using these selection management tools, please review the mouse navigation controls in “4.2 The '*Scene viewer*' tab”, as they will be equally important in your selection activities.

Simple mouse-based selection:

Before starting in on the details of the *Vertices selection* tab controls, it worth a brief overview of the actual selection process, which takes place in the *Scene viewer* tab, using your mouse to click and drag over the mesh vertices of interest. Selected vertices will change color to red from their default unselected blue color.

The typical selection process goes something like this:

- You'll enter a VWD tab where the selection target item is already assigned as the active-item, or, explicitly assign your active-item by opening the *Scene import* tab and selecting an imported item from the listbox.

Hint: The active-item in the *Scene viewer* tab can be viewed in three distinct modes in the display:

transparent (press 'o' - a toggle)
hidden line (press 'h' - a toggle)
open mesh (press 'a' - a toggle)

while the mouse is within the *Scene viewer* viewport

Hint: To hide everything else but the active-item in the display, **Ctrl**-select the active-item in the *Scene import* tab. Press the Show all button below that listbox to re-display all of the currently imported items in the VWD session.

Hint: Only currently imported items can become the active-item in a VWD session. Currently imported items in the *Scene import* listbox have one of the following suffixes: [Collision], [Hair], or [Cloth]

- Using the mouse/display navigation controls detailed in “4.2 The '*Scene viewer*' tab”, especially the right-click and drag action, rotate and zoom to best show the vertices of interest in the *Scene viewer* tab.
- **Selection:** To select vertices in the default selection mode, you will surround the vertices of interest with an enclosing shape (default = rectangle). This is done by left-click and holding the mouse button on a starting location near the vertices of interest, and dragging the mouse-over those vertices until they are surrounded by the enclosing indicator. You then let go of the mouse, and the enclosed vertices will become red, indicating that they are selected.
- At this point you may have the vertices you want and can simply execute the assignment task at hand. Should you need more vertices, or wish to remove some, there are many ways to do this using the tools described below, but the simplest way is to add or remove vertices from the selection:
 - **Add** new vertices to the selection by pressing and holding the **Ctrl** key while you left-click and drag around the vertices you wish to add to your collection. When you release the mouse, these newly selected vertices will be added to the existing set of red vertices.
 - **Remove** some or all of the existing selected vertices by pressing and holding the **Alt** key while left-clicking and dragging the mouse around parts of the *existing* red selection. When you release the mouse button, these enclosed red vertices will change to blue, indicating that they are no longer part of the current selection set.

- **Precision selection mode** allows you to select groups of arbitrary individual vertex points, as if painting. This function is actuated by pressing and holding the **Shift** key while floating over the vertices of interest (no mouse key should be pressed!). The temporarily 'selected' vertices will change color to yellow as you paint over them, and then change to red when you release the **Shift** key. As with any of the selection modes, Adding to existing or removing from existing selections is the same: press and hold the **Alt** or **Ctrl** key, *then* press and hold the **Shift** key, while floating over vertices with the mouse. The affected single vertices will change color to yellow to confirm the vertices that will adjust the current selection. When the **Shift** key is released, the selection adjustment is committed.
 - **Adding:** Pressing and holding the **Ctrl** key first, then the **Shift** key, while *floating over* vertices of interest with the mouse, will *add* these yellow vertices to the current selection when you release the **Shift** key.
 - **Removing:** Pressing and holding the **Alt** key first, then the **Shift** key, while *floating over* vertices of interest with the mouse, will *remove* these yellow vertices from the current selection when you release the **Shift** key.
 - **Warning:** If you get an error message pop-up while using this selection mode, press **OK** and then press the **Shift** key again, by itself, to bring the interface back under 'control'. Once this happens, this error may repeat often, and it may take a restart of the entire VWD session to escape this issue. This is a known issue that may be fixed in a future update.

When a number of separate mesh areas need to be collected as a set, the **Vertice selection** memory buttons become quite useful. By repeating the above process of selecting areas, and adding them to one of the two memory buffers with the **M+** button, you can accumulate a diverse set of vertices on a mesh. By then pressing the **MR** button, you can display and activate the entire selection set (displayed in red). With this selection set visible, you can then use the above selection (**Alt/Ctrl - drag**) techniques to further refine the selection set.

When the selection set is to your liking, you can 'apply' it to the current VWD task, be it spring additions, spring adjustments, attachment selections, etc.

With these steps in mind, the actual **Vertices selection** tab controls usefulness and relevance should make a bit more sense.

Vertices selection controls (Selection buffer functions):

- **MC:** Clear any vertices from the main selection memory buffer (this does not clear the actual current selection, just the selection memory buffer).
- **M+:** Add the currently selected vertices to the main memory's current contents.
- **M-:** Subtracts the currently selected vertices from the main memory's current contents.
- **MR:** Recall and display the vertices currently stored in the main memory buffer, replacing any selected vertices on the mesh item.

Hint: If you press the **Ctrl** key as you float your mouse-over the **MR** button, the caption changes to **Fic**. Pressing this button reloads the most recent **MR** selection set used in the previous VWD session.

Hint: The value displayed in the top right shows the number of selected vertices currently stored in the main memory buffer.

- **MIC:** Clear any vertices from the secondary selection memory buffer
- **MI+ :** Add the currently selected vertices to the secondary memory buffer's current contents
- **M<->MI:** Swap the current selection-sets between the main memory and secondary memory buffers
- **MIR:** Recall and display the vertices currently stored in the secondary memory buffer, replacing any selected vertices on the mesh item.

Hint: If you press the **Ctrl** key as you float your mouse-over the **MIR** button, the caption changes to **Fic**. Pressing this button reloads the most recent **MIR** selection set used in the previous VWD session.

Vertices selection controls (Utils and modes):

- **Free edges** (Ctrl-toggle button over **Sel edges**): Select the outer/free edges of the active-item mesh, replacing the current selection. **Free edges** are defined as those having no neighboring faces on all their segments. This feature is useful for selecting the edges of dress straps to attach to a figure's shoulders and neck.
- **Sel edges** (Ctrl-toggle button under **Free edges**): Select only the outer edges of the current selection area(s). When you press and hold the **Ctrl** key while floating over the **Free edges** button with your mouse, the button caption changes to **Sel edges**. Press this **Sel edges** button to select only the outer edge vertices of the *current selection*. This function is very useful if you want to apply an attachment to a part of the cloth which has no easy selection method. (A simulated belt on a dress for example).
 - **Note:** Although the **Sel edges** function shares the **Free edges** button, it is completely unrelated to that function. The **Sel edges** function only scans for the outer-most vertices of the current selection, meaning the real free edges will be selected.

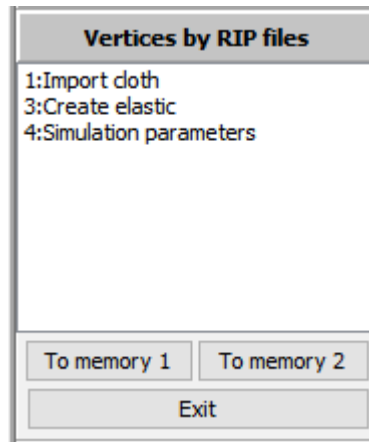
- **Note:** The *Sel edges* function does *not* add additional vertices to a selection-set. It preserves the outer-most vertices of all existing selection areas in the current selection-set. If there are multiple separate selection areas, the outer-edge vertices will remain selected for all of those areas.
- **Note:** When you press the *Sel edges* button, large parts of the mesh may turn red for a few moments, as if selected. Let it do its calculations for a while longer and the correct effect of the function will be displayed.
- **V** (def = disabled, a toggle): When enabled (light blue), only the vertices that are visible under the selection indicators (rectangle or lasso) will be considered. This is not the default, and may take extra time to isolate these visible vertices. The default behavior (button is light grey) is to select everything visible, and then everything on the mesh behind the visible selection area - as seen and projected from the viewer/camera's perspective.
- **L** (def = disabled, a toggle): Mouse-dragging selection mode, rectangle (light grey) or lasso (light blue). When enabled (light blue), the selection tool for the mouse vertices selection is a lasso. When disabled (light grey, default), the selection tool for the mouse vertices selection is a rectangle.
- **N** (def = disabled, display toggle): Display the polygon normals of the *currently selected vertices*. This function is a display enhancement, and does not alter the current selection in any way.
- **INV**: Invert the selected and unselected vertices on the active-item. Although it is not really a toggle, pressing this button again returns the selection to the previous state, etc.

Selection by material groupbox: If the active-item's imported mesh item has any defined material groups, these will be listed in the *Selection by material* pull-down listbox.

- **[material group listbox]**: Click on this listbox to see if any material groups are defined for this active-item. Select one and the vertices that make up that material group will be selected on the active-item. This selection will replace the current selection-set.

Hint: You can use the memory-buffers to gather multiple material groups into a final selection (e.g. apply a material group, press **M+**, and repeat for each material group, then press **MR** to apply/show the new consolidated selection).

Select vertices by RIP files: Use an existing RIP (Recorded Import Properties) file to create a selection-set on the active-item. Pressing this button will generate a file-browser, allowing you to find and select a RIP-file for this item that contains some selected section(s).



1. Once chosen, any available compatible selection-sets that were defined in the RIP-file are available as a selection-set to save to one of the two memory buffers. Once you press **Exit and return the Vertices selection** tab, you can apply the selection set with the **MR** or **M1R** buttons, depending on how you just saved the selection-set.
- **Important:** The selected RIP-file must be based on the exact same mesh/element as the target selection item

Vertices selection controls (selection helpers):

These selection tools use the computer to help automatically select vertices based on simple relationship ideas, like vertex lines or loops - similar to many selection tools in mesh modeling programs.

- **Simple:** Add to the existing vertices collection, by finding nearby unselected vertices that are directly connected to the same mesh polygons as the currently selected polygons. Repeated use of this button will expand the selection set much like a water drop soaks into a napkin.
- **Circular** (only on dynamic-actors): Select a line of vertices that make the most direct path between two selected vertices. To use this, you must select two vertices that are some distance apart, and press this button. If there is a 'solvable' path between these two selected vertices, those vertices in the path will be selected. If not, an error message will be displayed. There can only be two vertices selected when you press this button.
- button requires the selection of two vertices. The action of this button is to try to find a circular solution for these two vertices, ie to try from one vertex to arrive to the other using the most direct way. This action can not be solved, in this case, an error message appears.
- **Linear:** Select all vertices in a line starting from two selected vertices. Scan vertices in both directions on the mesh, selecting the vertices for as long there are vertices in that direction with an angle less than the value defined by the **angle** degrees value (set in the field to the right). The exact value is 180 - **angle** degrees.

angle (def = 30, range: 1 to 60 - uses same **angle** field as the **planar** button): when moving along a line of vertices in a given direction, consider only those vertices that vary from a straight line in that same direction, between 0 degrees and this amount. If there is more than one successful result, select and follow only the vertex path that is closest to 0 degrees.

Note: This linear selection is not necessarily composed of adjacent connected vertices, but may cross polygon faces, skipping the adjacent vertices that connect those corners.

- **Element / Selection** (Ctrl-toggle button): Specific selection options

Element (dynamic-actors and collision-items): With at least one vertex selected on the active-item (blue vertices), press the **Element** button to select the rest of the connected vertices on that active-item's section to have VWD find and select the rest of the vertices on that element.

Note: If selected vertices are present on multiple *elements* of a mesh item (like multiple hair strands, or a belt and pants mesh), all of the elements having *any* selected vertices will be completely selected. This can be useful, especially when certain elements are woven together and only parts of the meshes are visible or accessible.

Note: This may take a while on large or complex meshes. This is a very powerful for selecting specific parts of unwelded and/or partitioned meshes - such as a belt mesh element that is part of a pair of pants product. This process can be paused by pressing the **Esc** key (while the mouse is *not* in the **Scene viewer** tab!). This will show the current selection, and can be resumed by pressing the **Element** key again.

Selection (dynamic-items only): While pressing the **Ctrl** key and mousing over the **Element** button, it will change to **Selection**. After creating a vertices selection-set *on a cloth mesh* and placing it in the first memory buffer (**M**), then, by making a smaller new selection within that area and pressing the **Selection** button (optionally multiple times), the active selection-set on the mesh will expand to eventually fill the vertices-set stored in the first memory buffer (**M**).

Note: This function only operates on dynamic-actors (cloth/hair).

Note: When the growing selection reaches the edge of the selection area stored in the main memory buffer, it will spill over into the rest of the item, much like the related **Element** function.

- **Planar:** Starting from the current selection(s), scan outward in all directions, selecting vertices that are 'flatter', or less **angle**-d than the current **angle** value (set in the field above). Smaller angle values will result in selections of 'flatter' areas. Any added vertices will be added to the current selection.
 - **angle** (def = 30, range: 1 to 60 - uses same **angle** field as the **linear** button): when moving along a plane of vertices in a multiple directions, consider only those vertices that vary in flatness in all directions, between 0 degrees and this **angle** amount. The selection will 'work around' areas that are steeper than the specified angle.

Note: This scanning process may take a while on large or complex meshes. Keep an eye on the selected vertex counter at the very bottom of the VWD interface window. This process can be paused by pressing the **Esc** key (while the mouse is *not* in the **Scene viewer** tab!). This will show the current selection. Processing can be resumed by pressing the **Planar** key again.

- $\frac{1}{2}$: Reduce the current selection by evenly/randomly removing half the vertices in the current selection-set. This function tries to maintain an even distribution ratio of the original vertices across the impacted areas.

Note: This function can be useful when rigidifying and the vertex count is high. In this situation, reducing the original vertices selection probably won't change the results by much, but may save some calculation time.

- **Neighbors:** Add any nearby vertices to the current selection based on their distance from the current selection (set in the field to the right) using the by-neighborhood selection method. This type of extension is useful when linking nearby elements having no structural relationship with the currently selected vertices.

Neighbors value (def = 0.5, range: 0.01 to 2.0): Vertices that are this close or closer to the currently selected vertices will be added to the current selection. This uses the by-neighborhood selection method.

Hint: This button is similar to the **Simple** button up above, but is constrained by the distance value, and can 'jump' across 'air gaps' in the mesh - good for connecting unwelded buttons to shirts, etc.

Soft selection groupbox:

- **Extension iterations** (def = unchecked): Enable **soft-selection**, using the distance value to the right as the influence distance from this selection.

[*soft-selection extension iterations value*] (def = 10, range: 1 to 99): The number of times the expansion scan will be run from the currently selected vertices to generate the soft-selection. The distance of influence, measured from the currently selected vertices, where the selection will be extended outward from those vertices, in a graduated pattern. Across this distance, the selection strength will diminish from full strength to no strength. The color pattern of the selected vertices will indicate the distance and strength of the soft-selection setting results.

Note: This soft-selection process might take a few moments to update after being enabled or after the distance value has been changed. Please let the calculations finish, and you should see the resulting soft-selection effects on the active-item.

Hint: There are many places where soft-selection will generate smooth transitions between spring strengths, attachments, and/or force effects.

Note: not all effects will support soft-selection. In these cases, if the soft-selection is influencing vertices, those vertices will behave as if they are fully selected.

5.8 The 'Forces and Springs' Tab

This tab is where very precise settings of your dynamic-actor and/or collision-item(s) can be configured and refined (after setting up the overall base material characteristics of the item during the import process).

Forces and Springs

Soft 0.1 [up/down] Nail to collision

Collision dist 0.4 [up/down] Apply

Node weight 1 [up/down] Apply

[Scale and Softness (%)]

☒ Scale ☐ Softness

Value 90 [up/down] Apply

☐ Horizontal ☐ Vertical

No collide for current collision

No collide for current cloth

No self collide for current cloth

Self collision by extension

Apply fixed vertices

Apply react to wind

Apply inflate

Apply force Field

Animated by Host

Add layer Col Dist 0.2 [up/down]

☒ By faces Cut in 2 Cut all

[Rigidify Cloth or Hair]

[Use vertices extension]

Count 3 [up/down] Rigidify

Softness 0.1 [up/down]

[Use vertices neighborhood]

Distance 1 [up/down] Rigidify

Softness 0.1 [up/down]

☐ Distance min 0.5 [up/down]

☐ Selection ☐ Neighbors

The controls in this tab make up the heart and soul of the VWD system, as this is where the precise nature of a session's dynamic-actor and/or collision-item(s) will be configured and refined. Whether on the entire mesh, or on very specifically selected sub-sections, your dynamic-actor's stiffness, collision, wind, inflation, force-fields, scaling, layers, and softness are among the many settings that can be configured here. Some of the collision-item interactions can be adjusted using this tab's controls as well.

“Select and apply” - That's the simplest way to describe the usual operation of the **Forces and Springs** tab's controls. Use the **Vertices selection** tab navigation controls to select *exactly* where to apply each of this tab's custom settings to optimize the item's behaviors during your simulations.

The **Forces and Springs** tab controls:

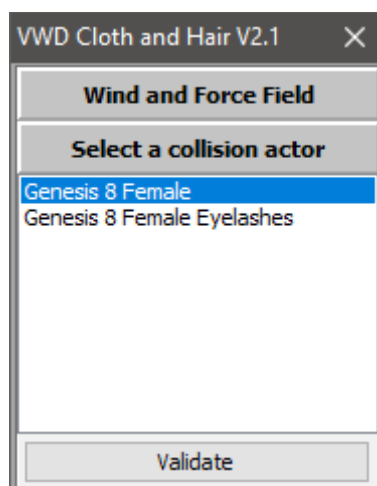
Hint: Because virtually all of the controls in this tab act upon *selected* areas of the session's current active-item, we recommend reviewing the selection tips found in “4.7 The '**Vertices selection**' Tab” and the scene navigation tips found in “4.2 The '**Scene viewer**' Tab” to familiarize yourself with the powerful vertex selection and scene navigation features of VWD.

- **Nail to collision:** Attach the currently selected vertices on the dynamic-actor (cloth or hair) to the nearest vertices on a nearby collision-item. If multiple collision-items are available in this session, a pop-up selection tab will require that you choose *one*. These attachment springs will be created using the **Soft** value to the left. Higher **Soft** values will result in more 'play', or movement, between the attached items; lower values will attach the items together more firmly.
 - **Soft:** (def = 0.1, range: 0.01 to 1.0): The softness of the attachment springs that are created when the **Nail to collision** button (to the right) is pressed. A higher **Soft** value corresponds to a softer attachment; an attachment with more 'play' between the dynamic-actor and the collision actor. A lower **Soft** value will attach the items together more firmly.

For all selected vertices on the dynamic-actor (hair or cloth), the program scans for the nearest vertices on the active collision-item and creates attachment springs of this softness value between these two items.

Note: This select/apply sequence can be repeated multiple times using different selection-sets and Soft values, to achieve precise effects.

Note: As mentioned above, if multiple collision-items are available in this session, a pop-up selection tab will require that you choose *one* for this **Nail to collision** attachment:



- **Collision dist** (def = 0.4, range: 0.01 to 1.0): Add or update the collision distance settings on the currently selected vertices.

Apply: commit the current **Collision dist** value to the currently selected dynamic-actor vertices.

Note: This select/apply sequence can be repeated multiple times using different selection-sets and **Collision dist** values, to achieve precise effects.

Hint: If you want to avoid importing a previously simulated cloth as a collision-item beneath a dynamic hair actor, this function can be used to fake this effect by creating a larger collision buffer between the bottom of a dynamic hair actor and the underlying collision-item figure.

- **Node weight** (def = 1, range = 0.01 to 100.0): Set the weight of the currently selected dynamic-actor vertices. This might be useful when trying to adjust the amount of 'body' in specific sections of dynamic hair or cloth that don't seem to respond to stiffness adjustments quite the way you want.

Hint: A single mesh can have varied weights assigned to different areas. For example, varied weights can be assigned as if half a beach towel got wet. To reset the entire mesh to the default, select it all and set the **Node weight** back to 1.0 and apply.

Note: While the node weight range is from 0.01 to 100, my experiments with a Force field of 50 on a hanging plane indicate a weight range of 0.1 to 20 on an average density mesh work pretty well. Values near the range ends tended to explode in my tests.

Note: The default value of 1.0 does not correlate to any particular earthly weight unit, and is simply a good starting value for most typical mesh dynamics. Remember also that the VWD default **gravity** setting of 1.0 is also unrelated to any earthly standard, but a useful starting point for most simulations.

The **Scale and Softness** groupbox: These controls allow you to modify the length and/or the softness of the springs corresponding the currently selected vertices in one or both mesh directions.

- **Scale / Softness** (def = Scale): When the **Apply** button (below) is pressed, any springs between the selected vertices on the dynamic-actor will be modified as follows:
 - If the **Scale** radio button is enabled, the current *length* of the mesh springs between the selected vertices will be reduced by the percentage value set in the **Value** field (below) during the simulation. This will effectively shrink the dynamic-actor by that percentage. The directional influence of this reduction will depend on the values of the **Vertical** and **Horizontal** checkboxes (below) *when the Apply button is pressed*.
 - If the **Softness** radio button is enabled, the current *stiffness* of the mesh springs between the selected vertices will be reduced by the percentage value set in the **Value** field (below) during the simulation. The directional influence of this softening will depend on the values of the **Vertical** and **Horizontal** checkboxes (below) *when the Apply button is pressed*.

Value (def = 90, range: 10 to 1000): Percentage of *change* in the Scale (length) and/or Softness of the currently selected dynamic-actor vertices. 100 = no change in scale, 90 = 10% reduction, 110% = 10% increase in scale.

Apply: Commit the current **Scale /Softness** + **Value** settings to the currently selected vertices on the dynamic-actor.

Note: This action sequence can be repeated multiple times using different selection-sets and unique **Scale / Softness / Value / Horizontal / Vertical** settings, to achieve precise effects.

Horizontal / Vertical (checkboxes: def = unchecked): Enable softness and shrinkage influences in either or both horizontal and/or vertical directions.

Technical Description: “The value will be multiplied by the cosine of the angle made between the line used by the two vertices of the *spring* and the *reference direction* (V or H). These scale and softness functions can be applied in a direction.”

Force and Springs (discrete settings): This group of settings buttons will generally enable or disable the described effect on the active-item's currently selected vertices. Like many of the settings in this tab, these buttons can be used multiple times on the target items (dynamic or collision) using distinct selection-sets for each button-press.

- **No collide for current collision:** Remove collision sensing from the selected vertices on the active collision-item. In scenarios where extreme poke-through or inter-penetrations may cause simulation problems, it may be helpful to remove collision sensing settings from certain mesh areas.

Note: This action operates on the active collision-item, *not* the dynamic-actor!

Hint: Collision sensing is an extremely powerful tool but it is very expensive to compute. Because it is sometimes easy for us to observe that some parts of the cloth pose no risk of colliding with a nearby dynamic-actor (or won't be noticed!), manually removing the vertices of these areas from the collision sensing process can save considerable time. Always use this feature if you can.

Because the import process always applies its general settings (like collision) to the entire collision-item mesh, any 'help' you can add to the optimization process is worth considering. The use of this function is a good way to help save computation time.

Hint: To re-add collision sensing to an item, use the **Collision dist** button and settings at the top of this tab.

- **No collide for current cloth:** Remove collision sensing from the selected vertices on the current dynamic-actor.

Hint: Collision sensing is an extremely powerful tool but it is very expensive to compute. Because it is sometimes easy for us to observe that some parts of the cloth pose no risk of colliding with a nearby collision-item (or won't be noticed!), manually removing the vertices of these areas from the collision sensing process can save considerable time. Always use this feature if you can.

Because the import process always applies its general settings (like collision) to the entire dynamic-actor mesh, any 'help' you can add to the optimization process is worth considering. The use of this function is a good way to help save computation time.

Hint: With some conforming clothes, sometimes two thin layers are modeled very closely together to produce a volume or thickness effect. Similarly, clothing items like belts are modeled very closely to an underlying garments like a dress. During the simulation's *collision testing*, problems can occur when the underlying figure tries to push the dress mesh *through* the nearby belt mesh. One solution is to (in this case) use this tool to prevent the vertices of the dress that are located just under the belt from interacting with the collision-item (usually the figure) by selecting these specific dress vertices then disable collision prevention on them with this button. Hint: The use of two memory buffers in the **Vertices selection** tools makes this precision selection process fairly straightforward.

Hint: To re-add collision sensing to an item, use the ***Collision dist*** button and settings at the top of this tab.

- ***No self collide for current cloth:*** Remove the self collision sensing settings from the selected vertices of the dynamic element.

Hint: Self collision sensing is an extremely powerful tool but very expensive in computing time. Because it is sometimes quite obvious to observe that some parts of the cloth pose no risk of self collision (e.g. tight clothes like leotards, socks, etc.), manually removing the vertices of these areas from the self collision processing can save considerable time. Always use this feature if you can.

Note: Because the import process always applies its general settings (like self-collision) to the entire dynamic-actor mesh, any 'help' you can add to the optimization process is worth considering. The use of this function is a good way to help save computation time.

- ***Self collision by-extension / Self collision by-neighborhood*** (a Ctrl-toggle button): Set or change the self-collision distance detection computation method of the currently selected vertices on the dynamic hair or cloth actor.

If specific areas of your dynamic or hair actor have self collision problems, you can set the self-collision detection method on the selected vertices to either the 'by-extension' or the 'by-neighborhood' algorithms. (These two sensing methods are described elsewhere in this document).

Important: This button is a Ctrl-toggle: By default, this button is configured to modify the self-collision method using the by-extension method. However, when you press and hold the **Ctrl** key with the mouse floating over the ***Self collision by-extension*** button, the caption of this button changes to ***Self collision by-neighborhood***, allowing you to apply the by-neighborhood method to the selected vertices.

- ***Apply fixed vertices / Remove fixed vertices*** (a Ctrl-toggle button): Fix (freeze) or Release the currently selected vertices on the dynamic-actor to the current location in the scene's 3D-space. When 'fixed' these selected vertices will not move or respond to any simulation forces at all. They won't move in the 3D-space during the simulation...

Note: When vertices are 'frozen', the neighbors of these 'frozen' vertices will move normally but their motion will obviously be limited by their attachment to these frozen vertices...

Hint: This is a great way to hang blowing curtains over a window without having to use VWD's attachment features. Select the top vertices of the curtain and click the ***Apply fixed vertices*** button.

Important: This button is a Ctrl-toggle: By default, this button is configured to fix or freeze the currently selected vertices in their current location throughout the simulation. However, when you press and hold the **Ctrl** key with the mouse floating over the ***Apply fixed vertices*** button, the caption of this button changes to ***Remove fixed vertices***, allowing you to release any frozen vertices in the current selection-set.

- ***Apply react to wind / Remove react to wind*** (a Ctrl-toggle button): Enable or disable wind force sensing on the dynamic-actor's currently selected vertices.

Some dynamic-actors may have a mix of mesh elements, some of which should or should not respond to wind forces (wind force sensing is off by default). An example might be the pole on a flag and pole item. You can configure which elements and areas of your dynamic-actor respond to the wind forces with this tool.

Important: This button is a Ctrl-toggle: By default, this button is configured to enable wind force-sensing on currently selected vertices. However, when you press and hold the **Ctrl** key with the mouse floating over the *Apply react to wind* button, the caption of this button changes to *Remove react to wind*, allowing you to disable the wind force-sensing on the currently selected vertices.

- *Apply inflate / Remove inflate* (a Ctrl-toggle button): Enable or disable inflation force sensing on the dynamic-actor's currently selected vertices.

Some dynamic-actors may have mesh elements that should or should not respond to inflation forces (Inflation sensing is off by default). An example might be the different parts of a hot-air balloon. You can configure which elements and areas of your dynamic-actor respond to the inflation forces with this tool.

Hint: The inflation forces in the simulation engine work in the direction of the mesh's defined normals (if any). To see how these mesh normals are defined, use the little 'N' button to toggle their visibility in the *Vertice selection* tab.

Important: This button is a Ctrl-toggle: By default, this button is configured to enable inflation force-sensing on currently selected vertices. However, when you press and hold the **Ctrl** key with the mouse floating over the *Apply inflate* button, the caption of this button changes to *Remove inflate*, allowing you to disable the wind force-sensing on the currently selected vertices.

- *Apply force field / Remove force Field* (a Ctrl-toggle button): Enable or disable wind force sensing on the dynamic-actor's currently selected vertices.

Some dynamic-actors may have a mix of mesh elements, some of which should or should not respond to Force fields (Force field sensing is off by default). An example might be the hat mesh in a hat/hair product. You can configure which elements and areas of your dynamic-actor respond to the Force fields with this tool.

Important: This button is a Ctrl-toggle: By default, this button is configured to enable Force field sensing on currently selected vertices. However, when you press and hold the **Ctrl** key with the mouse floating over the *Apply force field* button, the caption of this button changes to *Remove force field*, allowing you to disable the Force field sensing on the currently selected vertices.

- *Animated by host:* Control the currently selected vertices on the dynamic-actor using the animation motions from host-application (if available). A hybrid dynamic-actor/collision-item behavior This allows you to mix host-application based animation motions that will work well with the VWD simulation. The simulation type must be dynamic for this to be meaningful.

Hint: One example might be when a figure is waving a flag on a pole and the flag and pole are considered to be one mesh item when imported as a dynamic cloth actor. Selecting the flag pole and applying this property should result in the desired effect, where the pole acts as a non-simulated and animated collision item, and the flag is dynamic.

This feature is used in the **Hair assistant** to attach the vertices on the head of the figure. In that case, the mesh vertices are not really attached to anything in the VWD session, but instead follow their motion directions as defined in the host-application animation timeline.

- **Add layer:** Convert the currently selected vertices on your dynamic cloth or hair actor into a 'local' collision actor within that dynamic-actor. The collision distance property is set in the **Col dist** field to the right. You can create up to 5 separate collision-item sections on a single dynamic-actor in a session.

Similar to self-collision on a dynamic-actor, this differs in that the new collision-item mesh area is still part of the dynamic-actor, but does not respond to the simulation - becoming 'rigid'.

For mesh products that mix soft and hard parts of something together, like buttons on a shirt, or have and hat and hair mesh, being able to treat the different parts separately from with VWD (rather than breaking them apart in a separate modeling program and importing them separately...) is very useful.

- **Col Dist** (def = 0.2 , range: 0.01 to 1.0): Collision distance of the newly converted selection-set that is converted to a 'local' collision-item in the dynamic-actor mesh. The usual collision distance properties apply here and are described elsewhere in this document.

Note: In some cases, this self-collision management technique may be faster and more precise than the default self-collision mechanism used in the simulations.

- **Cut in 2 / Export** (a Ctrl-toggle button – DAZ Studio only - dynamic-actor): When using VWD from Daz Studio, this button allows you to quickly isolate and create new mesh items back in your DAZ Studio scene, from selections on the current dynamic-actor. The first method, **Cut in 2**, creates two new items from the selected and un-selected parts of the current dynamic-actor mesh. The second method uses the Ctrl-toggle function of this button, **Export**, and simply creates a new mesh item from the currently selected vertices.

Meshes created using the **Cut in 2** button are created and appear back in the DAZ Studio scene-tab. The first mesh will be created from the currently selected mesh vertices (as cut the **By faces** cutting method described below) and will be named like the original import item with a **_Cut1** suffix. The remaining mesh vertices (the unselected vertices) will also be added to your DAZ Studio scene as a new item, with the original item name with the **_Cut2** suffix. Both of these objects will have their UV-map information preserved.

Using the Ctrl-toggle **Export** function of this button (click on the button while pressing the Ctrl key), you can create an entirely new mesh from the currently selected vertices. The resulting mesh will be added to the DAZ Studio scene using the original item's name and the **_Selection** suffix.

This feature is a powerful ad-hoc mesh-editor and can be useful for modifying clothing items from within VWD when using DAZ Studio. For example, you can use this tool to combine parts of two dresses in a simulation sequence: part of one dress for the top and part of the other for the bottom.

Note: To use any of these newly created DAZ Studio scene items, you'll have to open the Scene import tab and import and manipulate them as you would any host-application's scene-items.

Hint: The selected vertices do not have to be contiguous using either the **Cut in 2** or **Export** buttons.

- **By faces** (def = checked): Enable/Disable the *By faces* cutting method:

Enabled: The first object (filename_**Cut1**) or **Export** object will only contain the mesh's polygons/faces that are completely surrounded by selected vertices. (filename_**Cut2** will contain the remaining vertices)

Disabled: The first object (filename_**Cut1**) or **Export** will contain the mesh's polygons/faces that are have any contact with the selected vertices. (filename_**Cut2** will contain the remaining vertices)

- **Cut all / UseMTL** (a Ctrl-toggle button – DAZ Studio only): When using VWD from Daz Studio, the **Cut all** button allows you to quickly create new mesh items back in your DAZ Studio scene from sections of your current dynamic-actor, delineated using cut-lines composed of selected vertices. Each section between the vertex cut lines will appear in your DAZ Studio scene-tab named like the original item with the **_Selection** suffix (all will named identically, so, rename them to taste!). The selected vertices have to be in the first memory before to start this function.

The Ctrl-toggle of this button, **useMTL**, creates an independent mesh scene-item in your DAZ Studio scene-tab for each unselected material group in the dynamic-actor. As above, each resulting independent mesh item in the DAZ Studio scene will be named after the original item with a **_Selection** suffix. There may be many, and they will all have the same name, so you may wish to renamed them.

Hint: The **Cut all** feature works very well with the **Sel edges** (Ctrl-toggle the **Free edge** button!) and **Linear** tools described in the **Vertices selection** tab . These tools can be used with direct selection or a material selection. To use this feature effectively, you must fully understand the **Vertices selection** memory management systems (the **M+/M-** buttons, etc.).

Hint: With this tool, you can easily do things like remove the sleeves of a dress from within the VWD session.

The **Rigidify Cloth and Hair** groupbox: Configure the stiffness properties of the dynamic cloth or hair actor's currently selected vertices. This stiffening effect can be configured using the same two methods used during the initial importing process: Stiffening by-extension, and *by-neighbor*.

- **Use vertices extension:** Add springs (and the current **Softness** setting) to the currently selected vertices, and to those vertices that are directly connected within **Count** vertex hops *on the mesh*.

This method allows the selected mesh areas to be stiffened relative to their nearby mesh vertices, but to still remain autonomous, or separated from nearby un-connected vertices and mesh elements. This is often used to get specific simulated motion effects on certain types of hair meshes, but can also be used to good effect in many other kinds of scenes and meshes.

- **Rigidify:** Press this to apply the current **Softness** setting to the dynamic-actor's currently selected vertices, and to those nearby *connected* vertices within **Count** hops of the currently selected vertices.
- **Count** (def = 3, range: 1 to 12): Add new springs to the currently selected vertices-set, using the current **Softness** setting, and include all nearby connected vertices *within this number of hops* in all directions, from the currently selected vertices-set.

- **Softness** (def = 0.01, range 0.01 to 1.0): The **Softness** value will define the softness of the currently selected vertices springs, and any new springs added, based on the **Count** value. The higher the value, the softer the springs will be. User lower values to add firmness to the selected area.

Note: This select/**Ridigidify** sequence can be repeated multiple times using different selection-sets and **Softness/Count** values, to achieve precise effects.

Hint: When using the by-extension stiffening method to ensure some autonomy and separation between the selected mesh areas, you may need to pay extra attention to the self-collision settings to achieve your intended results, as these autonomous meshes will interact more than meshes that are connected to their neighbors. I picture softer hair without hair spray, when using this stiffening method.

- **Use vertices neighborhood:** Add springs (and the current **Softness** setting) to the currently selected vertices, and to any nearby vertices that are within the specified **Distance** (in all three directions - like a sphere) from the currently selected vertices-set.

Not only does this stiffening method add springs to the nearby *connected* vertices on the same mesh surface, but it also adds any other nearby vertices that are *not directly connected* as well. This stiffening method has the effect of stiffening the *volume* surrounding the currently selected vertices, as well as the nearby mesh surfaces. This is somewhat like using hair-spray to stiffen the strands of hair individually, as well as stiffening the overall hair by grouping the strands together.

- **Rigidify:** Press this to apply the current **Softness** setting to the dynamic-actor's currently selected vertices, and to any nearby vertices within Distance, in any direction.
- **Distance** (def = 1, range: 0.01 to 1.0): The inclusion distance, as measured from the currently selected vertices-set, where any other nearby vertices will be connected, using springs set to the current **Softness** setting. This distance is a radius, and acts as a sphere, which may include vertices that are not limited to those on the same mesh surface.
- **Softness** (def = 0.1, range: 0.0001 to 1.0): This **Softness** value will be applied to the currently selected vertices springs, and any new springs added, based on the **Distance** value. The higher the value, the softer the springs will be. User lower values to add firmness to the influenced area.
- **Distance min** (def = checked): The distance to skip stiffening spring creation on vertex pairs that are closer together than the distance set on the right. This optimization will save simulation time with very little impact on the results, if the distance value is kept fairly low. Stiffening between close vertices does not have much effect.
- The **Distance min** (def = checked): Enable this to remove all of the existing or generated springs involved in this **Rigidify** action that are shorter than the **Distance min** value (to the right). This is an optimization that removes shorter springs (that won't impact the simulation much) from the current **Rigidify** action's vertex selection set, saving simulation time without much effect on the results.

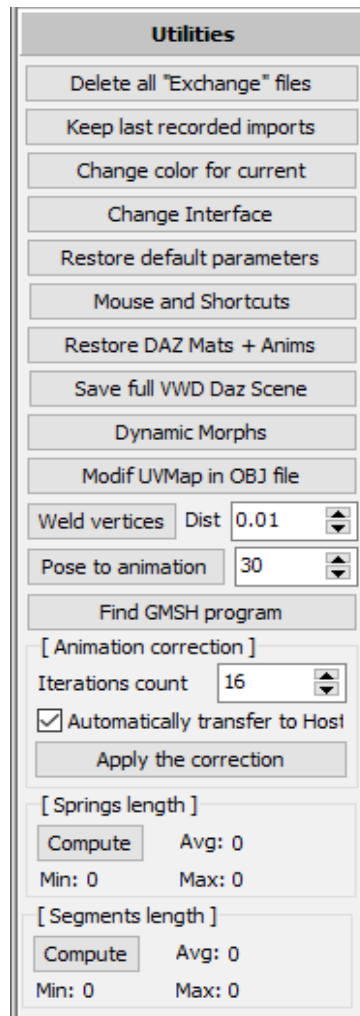
Distance min value (def = 0.5, range: 0.01 to 2.0): If the above checkbox is enabled, any springs involved in this **Rigidify** action (existing selection, or generated by-extension) will be removed. A value equal to half the **Distance** value works fine.

- **Selection** (def = unchecked): When pressing the **Rigidify** button, only apply the stiffening (w current **Softness**) on the currently selected vertices-set, *leaving the vertices that are selected by the **Distance min** extension unchanged.*
- **Neighbors** (def = checked): When pressing the **Rigidify** button, only apply the stiffening (w current **Softness**) on the extended neighbor vertices (those found within **Distance**) of the currently selected vertices-set, *leaving the properties of the currently selected vertices unchanged.*

Note: This entire select/**Ridigidify** sequence can be repeated multiple times using different selection-sets and apply-properties (e.g. **Softness/Distance**), to achieve very precise effects.

5.9 The 'Utilities' Tab

The *Utilities* tab contains various functions which are independent from the simulation settings but are very useful when operating in a VWD session.



Some of these utilities help manage the VWD environment, some present information pop-up tabs, and others start and run standalone utility programs that help with some mesh-specific need that might come up when running VWD.

The Utilities:

- **Delete all Exchange files:** Remove all the working files inside the VWD core program's *Exchange* folder. In a recommended installation path, this folder would be **C:\VWD\Exchange**. This folder contains working mesh, control, and animation files that are created when VWD is running - specifically when items are imported into VWD, and when dynamic/animated simulations are run. There is no feedback – the files are simply removed.

Note: The dynamic/animated simulation output files can become large when long animation sequences are processed.

Hint: Sometimes odd VWD problems can be resolved by restarting VWD and removing these files. Any required files will be regenerated except previous simulation results (see the warning below).

Note: You can also use any Windows file-manager/browser to review, manage, and remove these files as well. This button is just a shortcut - a convenience tool.

Important: While most of the working mesh and control files in this folder will be recreated when a new simulation is configured and run within a VWD session, *the results from any previous animation simulations will be removed* when this function is used. **This means that any simulation results will need to be regenerated by re-running the simulation if these files are removed!** Before pressing this button, you may wish to browse to this folder and back up the contents of this directory and store them with your project's scene files so that you can restore them later and re-apply these simulation/animation results to your host-application's scene items when reloading the original scene file (e.g. duf or pz3, etc.)

- **Keep last recorded imports:** Copies the most recent recorded inputs files (Last.rip/Restore.rip) to files that will not be over-written (date-based names in the same directory? Verify.)
- **Change color for current :** Change the color for the active-item actor (currently selected in the listbox of the *Scene import* tab). Press repeatedly to try different colors.

When an item is imported into VWD, a random 'clay' color is assigned to it. Sometimes, the collision actors and/or the dynamic-actor have similar colors, making it difficult to distinguish between the various items during the configuration and simulation.

- **Change interface** (a toggle): Switch between the bright and dark VWD application interfaces.
Important: This will restart the VWD session!
- **Restore default parameters:** Restore the default VWD parameters in all of the interface tabs. Some of the changed tab settings are retained across sessions, so this is a good way to reset everything in VWD back to a familiar starting point.

- **Mouse and Shortcuts:** Display a new tab showing the available keyboard shortcuts and mouse actions available in the interface. (Press the **Exit** button at the bottom of the tab to return to the **Utilities** tab). These settings are also documented elsewhere in this manual.
- **Restore DAZ Mats+Animations** (DAZ Studio only): Re-apply the original surface material settings and animations to the imported and selected collision-item. If no specific items are imported and selected, *all* of the **xxxx_VWD** items in the scene will have their materials and animations restored when you press this button (if they are available in the VWD results cache/Exchange folder).

This fixes an issue that occurs in reloaded Daz Studio scenes, where the **xxxx_VWD** items lose their surface materials and animations when the DAZ Studio scenes are reloaded.

- **Restore all items:** To use this function to re-apply the original DAZ Studio item surface materials to *all* of your grey **xxxx_VWD** versions of the items:
 1. ... After reloading a previously simulated scene into DAZ Studio (grey items)
 2. Start VWD
 3. Press this **Restore DAZ Mats+Animations** button.
 4. Exit VWD...
 5. Return to your host-application and continue.
- **Restore specific items:** To use this function to re-apply the original DAZ Studio item surface materials to one of your grey **xxxx_VWD** version of the item:
 1. ... After reloading a previously simulated scene into DAZ Studio (grey items)
 2. Start VWD
 3. Import the grey **xxx_VWD** item(s) as collision-items
 4. As you import each collision-item, press this **Restore DAZ Mats+Animations** button.
 5. When all items are restored, you can exit VWD and render normally back in DAZ Studio. You do not have to do anything else in VWD after the materials are restored.
 6. Return to your host-application and continue.
- **Save full VWD DAZ Scene:** As a backup assistance and DAZ Studio scene reloading tool for advanced users, this button tells VWD to copy key animation/project elements to a separate sub-folder in main VWD program's ...**\Exchange\Scene** folder. This new folder contains key elements of the current VWD session information, including the current import state settings (RIP files) and the dynamic simulation result frames. It also includes a DAZ Studio script that can be used to reload these scene elements, their materials, and any dynamic animations into your current DAZ Studio scene without having to start VWD.

Usage: The creation of the scene backup with this tool is easy enough: Click on the **Save full VWD DAZ Scene** button – VWD will create the directory and copy the files to it.

Reloading the scene in DAZ Studio is slightly more involved, as you need to find and run the newly created DAZ Studio loading script for that scene. Here is an example:

1. **To generate the backup folder** from an active VWD session that is working from a DAZ Studio scene named **my_scene.duf**, click on the **Save full VWD DAZ Scene** button in the **Utilities** tab. VWD will create the directory which is named after the current DAZ Studio scene-filename (e.g. **my_scene.duf** => **C:\VWD\Exchange\Scene\my_scene**), and copy the relevant files to it. (You may also wish save this directory for later scene work - zip it up or simply move it to a place where you can find it when you are working with your scene).
2. **To reload the relevant files for this scene** at a later time, open DAZ Studio and use its 'file->open' menu to browse to find the **my_scene.dsa** scene-loading script from the VWD program directory (e.g. **C:\VWD\Exchange\Scene\my_scene\my_scene.dsa**) and open it. This should load the saved DAZ Studio scene and map the materials and animations onto those **xxxx_VWD** items in that scene, generally placing things in a 'ready to render' mode.

Important: If you wish to completely re-produce a scene for later recording and you have used a dynamic wind sequence in an animated scene, you will need to also save the **.../Exchange/Animated hair.rwd** file somewhere (in that same folder, perhaps), and restore that file (and the Saved wind.rwd) file later, back in the **.../Exchange** folder, when working with the scene. This is not necessary when simply reloading and re-rendering a completed scene.

Important: Apparently some DAZ Studio systems do not honor the file paths and content locations in these backup folders! Until we figure out why, you can re-construct the backed-up scene by:

- Copying the *.dyn files from this backup folder back into your main VWD installation's **.../Exchange/Scene** folder (e.g. **C:\VWD\Exchange\Scene\NewScene_SimPlane_VWD_Plane_VWD.dyn**).
 - Opening the backed-up scene file (e.g. **my_scene.duf**) in DAZ Studio
 - Opening VWD in DAZ Studio and use the **VWD Utilities** tab's **Restore DAZ Mats + Anims** button to restore the textures and animations to that scene.
 - Close VWD and continue working in DAZ Studio with the restored scene.
- **Dynamic Morphs** (helper utility, new tab)): Create simple *body dynamics* animations (e.g. Butt/Breast jiggle) using the VWD simulation engine with an animated collision-item prop or figure. Because this function maps user-specified morph-controllers to VWD simulation results, the animated collision-item must have these morph-controllers available for this function to be practical and usable.

This button opens a new tab where you can create, save, and/or reload any available *dynamic morph* settings. The **Dynamic morphs** interface is described in section **5.10 Dynamic Morphs** in this document.

- **Modif UV-map in OBJ file** (helper utility): Written specifically to create new UV-maps that reduce or eliminate texture map distortion on items that are created with odd UV maps, this tool takes an existing OBJ file that contains UV-map information and creates a completely new copy that contains new UV-map information that reflects the current X/Z layout of the item in VWD's 3D-space. It is a powerful way to remap the UV information in a mesh item by using VWD simulation tools to 'flatten' the item onto the 3D-space 'floor', then wrapping those flattened mesh coordinates into a new OBJ file of that same item.

When to use: If you wish to apply your own arbitrary texture to a simulated VWD item and find that your texture is compressed and stretched dramatically across the cloth, it means that the designer probably created an unorthodox UV-map that doesn't reflect the natural layout of the mesh. This tool can help create a UV-map that more closely reflects the mesh's natural density variations and results in consistent texture mapping onto that mesh. (e.g. after using this, the squares of a checkerboard are still square across the entire cloth item.)

Note: The use of this tool is a part of a sequential process that takes a series of steps that move the meshes into and out-of VWD using a sequence of VWD actions. It is powerful, but it uses VWD in a very unconventional workflow. This function also uses the **Cut all** function in **5.8 Forces and Springs** and will require several steps to separate the mesh sections.

Usage: This is better explained with a video, but the steps follow (roughly):

1. Create a scene with the item to remap in your host-application.
2. Import it into VWD as a dynamic-actor (cloth).
3. Within the **Vertices selection** tab, create the optimal cut-lines to enable flattening of the mesh and progressively save these cut-lines to main memory buffer 1 (M+) until they are all saved.
4. In the **Forces and Springs** tab, click on the **Cut all** button to make the cuts
5. Close VWD, and return to DAZ Studio and look for those newly cut parts in your scene tab. (rename if desired).
6. Save all of the parts (and only those) into a new **OBJ** (hide everything else and export as OBJ, using DAZ defaults in OBJ save settings)
7. Close VWD
8. Re-import this new OBJ into DAZ Studio as a new scene item
9. Re-open VWD and import and flatten this item with VWD simulator as a dynamic cloth actor, then **Send Pose to host** in the **Simulate** tab
10. Click on the **Modif UV-map in OBJ file** button, select the newly partitioned and flattened scene item, having the **_VWD** suffix. A new **OBJ** is now created in the same directory with a **_Modif.obj** suffix.
11. Open VWD, **Weld vertices** of the new **xxxx_Modif.obj** (into a new obj), close VWD
12. Import this result **xxxx_Modif_Weld.obj** back into DAZ Studio and apply your texture(s) and/or open VWD to run simulations on it, etc.

At the end of this process, you should have a welded OBJ file with a newly replaced UV-mapping. The big idea of that process is to use VWD's simulation to flatten the mesh, then use that format to generate a new UV-map that can be applied to the OBJ file.

- **Weld vertices** (helper function): Weld together mesh vertices that are located very closely each other in a VWD-generated (**xxx_cut.obj**) OBJ file. The **Dist** value (distance) will be used to indicate how close together the vertices must be for them to be welded by this tool. Welding these disconnected vertices together in a mesh file will help keep dynamic hair and cloth actors from falling apart during the simulations.

While there are sophisticated tools within VWD that allow you to attach mesh segments together using simulation springs, this utility fixes the real problem by welding the mesh vertices in the VWD-generated (**xxxx_cut.obj**) OBJ file together, as if they were modeled that way. It reads from a VWD-created obj file and generates a new welded version. You then load this into your host-application and then into VWD as a new dynamic cloth or hair actor. The file will be named the same thing as the original, with a new **_Weld.obj** suffix. (e.g. welding **my_mesh_cut.obj** will generate a new obj file called **my_mesh_cut_Weld.obj** in the same directory folder).

- **Dist** (def = 0.01, range: 0.01 to 99.99): Maximum distance between vertices in an OBJ file to consider welding them together. Vertices closer than this distance will be welded together and saved to a new OBJ file.

Important: This utility can only be used on the VWD-generated **xxxx_cut.obj** working files that are found in the VWD program's **Exchange** folder (e.g. **C:\VWD\Exchange\my_item_cut.obj**).

Important: This function does *not* interact with anything else in VWD when it is called – it is just a small independent utility that you can access from within VWD. It simply loads a mesh (**xxxx_cut.obj**) of your choosing, and produces a distinct new (welded) mesh with a new **_Weld.obj** suffix, that is completely independently of anything else going on in VWD. To use this newly created welded mesh, it must be loaded into your host-application (normal obj import steps) and then made available to VWD using the **Scene import** tab's **Host list** refresh function where it can be imported normally (probably as a dynamic cloth or hair actor).

Usage: Without having to do *anything* else in VWD (just start VWD), open the **Utilities** tab and click on the **Weld vertices** button to bring up a browser that is used to select your VWD-generated **xxxx_cut.obj** item from your VWD program's Exchange folder:

1. You browse your system for the unwelded item (probably an OBJ file that VWD created as a working file in your VWD program's Exchange folder (e.g. **C:\VWD\Exchange\your_thing_cut.obj**)).
2. Open that file, and wait for the tool to complete. (There are no indicators of progress other than the spinning Windows cursor). It may take a few moments.
3. Once complete, look for the newly created welded OBJ file of the same name with the **_Weld** suffix (e.g. **C:\VWD\Exchange\your_thing_cut_Weld.obj**).
4. If you wish to use this new item as a cloth or hair item in your VWD session, you must first add and place the new obj item (e.g. **your_thing_cut_Weld.obj**) into your host-application's scene (import it, etc.), then use the **Scene import** tab's **Host list** function to 'see' this new item (e.g. **your_thing_cut_Weld**) and import it into VWD and work with it as you would any other imported item.

- **Pose to animation** (DAZ Studio only def = 30): Used to help with static images, this simple pose transition tool creates a short animation sequence to help smooth the flow of clothing elements from their initial fitting (zero-d 'A' or 'T' pose) to your desired pose position.

Important: This will completely replace any active/existing animation on your DAZ Studio timeline! Do **not** use this tool when working on an active animated DAZ Studio scene!

Hint: To use this, start with a non-animated scene with your undressed figure set to the desired pose on the first frame. Then set the number of frames you wish to transition between your figure at a zero-d pose and your new target pose, then press the **Pose to animation** button to generate the short/simple animated transition and place it on the DS timeline. VWD will then close, and you can then fit your clothes/hair on the figure in first frame, and use VWD's dynamic simulation mode to smoothly migrate the clothes for your final static image. Once the dynamic simulation frames are completed in VWD, scrub to the last frame in the **Scene Viewer** tab, and use the **Start static simulation** button (or 'Shift/simulate mode') to continue to perform additional draping adjustments on the VWD scene. Press the **Send pose to host** button at anytime to save the current pose back to DAZ Studio.

- **Find GMSH program:** Press this button to bring up a Windows browser that you will use to locate your version (3.0.6!) of the **GMSH.exe** meshing program (if installed). This external utility creates meshes that are used in VWD's volumetric simulations.

The **GMSH.exe** program is a stand-alone open-source utility that VWD uses to generate meshes for its volumetric simulations. It is available at the GMSH website at:

<https://gmsh.info>

and can be downloaded and installed anywhere on your system. The above function is used to browse-to and find the program in your installation location.

Important: Until further notice, please download and install the Version 3.0.6 version (32/64 bit) from <https://gmsh.info>, via their archive folders at:

<http://gmsh.info/bin/Windows/gmsh-3.0.6-Windows64.zip>

or

<http://gmsh.info/bin/Windows/gmsh-3.0.6-Windows32.zip>

as their newer versions have introduced some incompatibilities with VWD.

Note: VWD uses the **GMSH.exe** program to help create meshes for its volumetric simulations. It is **not required** for any other VWD use or its general functionality!

Animation correction groupbox: (**Currently not enabled/working!**) Apply self-collision correction (mesh adjustment) to an animated collision-item that intersects with itself. This very useful if you want to create a simulation that works properly when the collision-item intersects with itself. This can create mesh problems when the intersection forces a dynamic cloth or hair actor to also intersect or inter-penetrate with the surrounding (or pinching) meshes. This can cause problems with the simulation and produce undesirable results. Use this when a figure's thighs inter-penetrate, or when arms (near the armpits) interpenetrate with the torso (near the breasts), or any other places where the figure intersects with itself during an animation or pose.

Warning: this function is currently not enabled/working.

- **Hint:** A typical scenario where this is useful - A scene with an animated figure wearing pants, crossing his/her legs. When the animated legs pass through each other, they pinch the pants mesh through the leg meshes as well, causing odd simulation results. This function fixes the problem area by 'warping' the collision-item in in the places it would intersect itself, leaving a small amount of space for the pants (or other clothing item) to also not intersect.

- **Iterations count** (def = 16, range: 4 to 64) Number of iterations to add to the collision detection in this calculation. This value should be higher in animations that move quickly.
- **Automatic transfer to Host** (def = checked): Transfer the animation to the host-application at the end of the simulation.
- **Apply the correction:** Start the self-collision detection/correction on this animated collision-item.

Springs length groupbox: computes the **length of the springs** of the *currently selected vertices*. If no vertices are selected, the computation will be made on all the vertices.

- **Compute:** Start the computation.
Avg: The average length of the springs.
Min: Length of the shortest spring(s).
Max: Length of the longest spring(s).

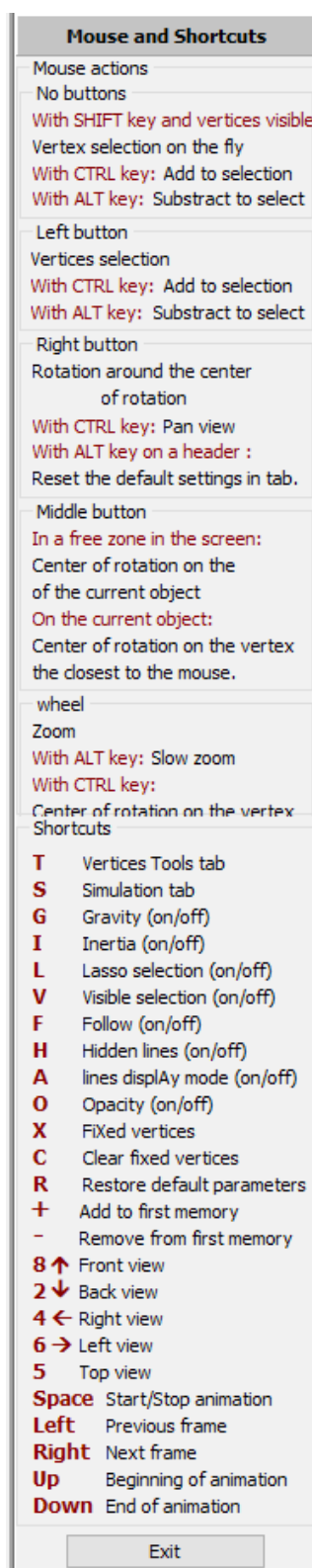
Segments length groupbox: computes the **length of the segments** of the *currently selected vertices*. If no vertices are selected, the computation will be made on all the vertices.

- **Compute:** Start the computation.
Avg: The average length of the segments.
Min: Length of the shortest segment(s).
Max: Length of the longest segment(s).

5.10 The *Mouse and Shortcuts* Tab

There are a number of very handy Mouse Controls and Keyboard Shortcuts available in certain VWD contexts, especially when working in the Scene Viewer, Simulation, and Vertices Selection tabs.

Note that the mouse has to be located within the tab window/viewport for some of these keyboard shortcuts and commands to work as expected.



Mouse and Shortcuts controls: Many of these mouse, mouse+key, and keyboard combinations are context-specific, meaning that they will only work in certain program modes or contexts – e.g. selecting vertices on a model only makes sense in the Scene viewer tab. Here are the currently available commands:

5.10.1 Mouse and Mouse + Key Combinations:

- **Scene Viewer** tab/viewport: No button pressed – selection (on the fly)
 - Drag w/ **Shift** key: Vertex selection (single vertex)
 - Drag w/ **Ctrl** key: Add to current selection
 - Drag w/ **Alt** key: Subtract from current selection
- **Scene Viewer** tab/viewport: Left button pressed – selection (rectangle or lasso area)
 - Drag w/ no key: New vertices selection
 - Drag w/ **Ctrl** key: Add to current selection
 - Drag w/ **Alt** key: Subtract from current selection
- **Scene Viewer** tab/viewport: Right mouse pressed – viewport navigation
 - Drag w/ no keys: Rotate the camera around the current center of motion
 - Drag w/ **Ctrl** key: Pan the camera
- **Scene Viewer** tab/viewport: Middle button - set center of rotation
 - Click on item mesh: Set center of rotation to nearest vertex
 - Click on empty background field: Set center of rotation to current active-item

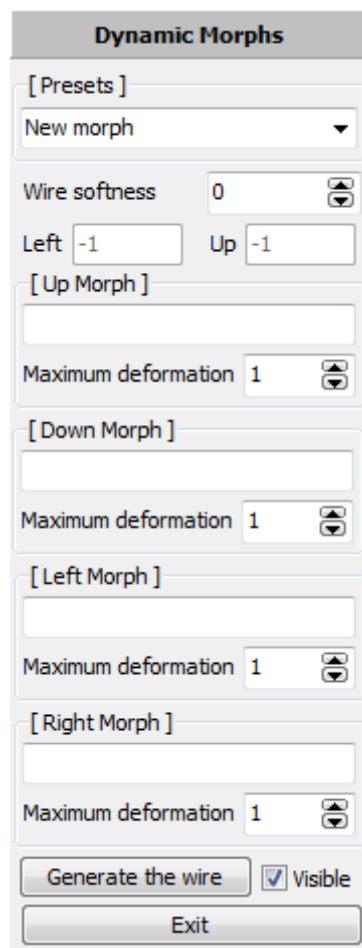
Hint: Change the current active item by clicking on the item name in the **Scene import** tab.
- **Scene Viewer** tab/viewport: Middle button, rolling – zoom/rotate
 - Roll w no keys: Faster zoom in/out
 - Roll w **Alt** key: Slower zoom in/out
 - Roll w **Ctrl** key: Center of rotation on vertex closest to mouse

5.10.2 Keyboard Shortcuts:

- **Scene viewer** tab (navigation, item display modes)
 - 2 / 8** : Back/Front view
 - 4 / 6** : Right/Left view
 - 5** : Top view
 - A** : Display active-item lines (all)
 - H** : Display active-item hidden lines
 - O** : Display active-item surfaces (opacity)
- **Scene viewer** tab (frame scrubber)
 - Space** : Start/Stop running animation
 - Right Arrow** : Next frame
 - Left Arrow** : Previous frame
 - Up Arrow** : Beginning of animation (first frame)
 - Down Arrow** : End of animation (last frame)
- **Simulation** tab
 - G** : Gravity toggle (can be changed during simulations)
 - I** : Inertia checkbox toggle (can be changed during simulations)
 - F** : Follow checkbox toggle (can be changed during simulations)
- **Forces and Springs** tab
 - X** : Set fixed vertices
 - C** : Clear fixed vertices
 - R** : Restore default parameters
- **Vertices selection** tab
 - L** : Toggle Lasso selection mode (lasso/rectangle)
 - V** : Toggle Visible selection mode (projection vs visible)
 - +** : Add selection to first memory buffer
 - : Subtract selection from first memory buffer
- Tab headers (all):
 - Right-click** with **Alt** key: reset parameters in the tab to their defaults (**Note**: not all parameters currently have defaults).
- Quick Tab Selection/Navigation
 - T** : Vertices selection tab
 - S** : Simulation tab

5.11 The 'Dynamic morphs' Tab (DAZ Studio only)

The *Dynamic morphs* tab is where you can generate simple soft body dynamics (belly/butt/breast/etc. jiggle) animations for a figure.



This function only works from DAZ Studio VWD sessions, but the results can be exported from DAZ Studio to Poser compatible pose sequences and transferred to Poser as a base animation. These dynamics are generated by mapping the motions of specifically 'placed' springs on an animated figure to specific morph-controller settings. These spring-to-controller mapping series are generated using the VWD simulation engine and sent back to the host-application, where they can be save and exported as normal keyframed animation sequences.

The general workflow of this utility is to 'attach' a flexible virtual spring 'wire' to the center of your desired 'jiggle spot' (a single vertex) on an animated collision-item (usually an animated figure, maybe on the belly-button or a breast nipple). After defining the four desired morph-controller names (up/down/left/right) from the figure's specific morph-controller set (e.g. *pCTRLrBreastUp-Down*), a simulation is run and the motion values of the flexible 'jiggle spot' spring 'wire' are mapped to the chosen morph-controllers, generating a series of animated key-framed morph-controller frames of the animation/simulation in the host-application's timeline.

Note: This utility behaves differently from the rest of the VWD simulation processes in a very important way. This particular simulation tool generates its animation-data for an animated *collision-item* rather than the usual dynamic-actor (In fact, for this type of simulation, no dynamic-actor needs to be in the scene). Also, instead of generating the usual full-mesh motion frames as with dynamic-actor meshes, the output of the **Dynamic morphs** simulation is a set of morph-controller key-frame sequences for each virtual spring 'wire' - each having it a set of four morph-controllers (up/down/left/right) positions for each sequence frame. As with the usual VWD simulations, these results are sent back to the host-applications and placed on the timeline. These are accessible as traditional key-framed morph-values from within the host-application and can be saved/edited as such.

Dynamic morphs overview:

Unlike the usual VWD simulations, which generate changes based on the entire dynamic-actor mesh, the **Dynamic morphs** process tracks the changes that occur on small, user-placed 'wire' springs that are attached to an animated collision-item - usually an animated figure. These 'wires' follow the animation movements as springs would, and the result values of these spring motions are converted to four directional morph-controller values. These morph-controller properties are assigned and weighted by the user and vary for each particular figure and body location (breasts/glutes/etc.).

To add more than one wire to the figure, after setting-up a wire with its properties, press the **Generate wire button**, then repeat the process again. Each time you press **Generate the wire**, a new spring 'wire' will be displayed on the selected figure vertex. Press the **Exit** button (after saving to a new preset?) to commit all of these spring 'wires' and be taken to the **Simulate** tab where you will start the simulation phase.

Important: Do not change the default simulation settings when simulating the Dynamic morphs animation wires. Leave the simulation settings as they are and the simulation will work as expected. Changing these settings will likely result in odd results or errors.

Press the **Start dynamic simulation** button to run the simulation. Let it finish, then press the same button, which should now be the **Send animation to host button**, and VWD should copy the data to the host-application and terminate, returning you to the host-application with the new morph key-frames written to the timeline. If you named the morph-controllers correctly, there will be a set of four new real/traditional key-frames on your timeline for each wire that was included in the simulation (e.g. left-breast: up/down/left/right, right-breast: up/down/left/right, left-glute: up/down/left-right, right-glute: up/down/left/right). If any are missing, there's probably a controller-name typo in the dynamic morph preset (manual edit is the only way to fix, or redo the entire preset in VWD).

Dynamic morphs settings:

Presets groupbox: Available *wire settings* presets.

- **[preset listbox]**: Click on this listbox field to view and select from any available *wire settings* presets.

Hint: Saving presets - After creating your own configured wires, type a new name for your preset into this list-box field and press **Enter** to save the wire settings as your own new preset.

Note: Only select wire presets that are known to be generated for the target active-item. Applying wire presets to the incorrect figure figure will certainly result in the unexpected results.

- **Wire softness** (def = 0, range: 0 to 0.1): The softness used for the simulation wires. Higher values result in softer spring wires.
- **Left / Up** (read only values): The **Left** (X, sort of) and **Up** (Y, sort of) locations for the wire in the collision-item mesh.

Note: These two values are informational and defined when you choose the reference vertex for this wire. They are not modifiable. They only seem to be meaningful when loaded from a preset, and will only show one of the 'wire' spring locations if there are more than one.

Directional morph-controller settings:

The four parameter-groups that follow are where you assign the directional morph-names and weighting values that are used to map the 'wire' spring motions that are generated by the simulation onto the figure's body part(s). Each direction requires a morph name that correlates with the figure's body-part spring wire and that body-part's appropriate directional morph-controls.

Hint: Even though the current **Wire softness** value (above) will help control the overall response of the wire-to-morph mapping in this tool, by setting this **Maximum deformation** value higher or lower, you can also fine-tune the response of this specific morph in the final simulation.

Important: While many morph-controllers may come in pairs (e.g. separate BreastUp and BreastDown controllers), many morphs generate their end-to-end range by using positive and negative morph values on the same morph-controller (e.g. BreastUpDown). This means that in some cases the down-morph control may in-fact be a negative morph value for the same up-morph control or an 'up-down morph-controller'. This is very typical. For example, the upward breast 'wire' spring motion in a DAZ Genesis 3 Female will probably be mapped to a *positive* value of that figure's breast up/down slider:

pCTRLrBreastUp-Down. Rather than using a specific down morph (which may not exist), the down motion of the 'wire' spring will probably be mapped to a *negative* value of the same **pCTRLrBreastUp-Down** morph controller. Be sure to consider the actual range of the morphs as well, as some range from -1 to 1, and others are arbitrary to the morph-controller range (e.g. one of the G3F morph-controllers ranges from -12 to 25)

Up Morph groupbox defines settings for the Up morph.

- [text] The name of the desired **Up** morph-controller in Daz Studio. (e.g. **pCTRLrBreastUp-Down** for the G3F breast **Up** morph) as found by looking within the properties of the desired Up morph for this figure).
- **Maximum deformation** (def = 1, range: 0 to (+) MorphMax **OR** (-) MorphMax): The maximum value you want to apply to your selected **Up morph** range during the simulation. This value is not necessarily the same 'max' value defined in the Daz Studio properties.

Down Morph groupbox defines settings for the Down morph.

- [*text*] The name of the desired **Down** morph-controller in Daz Studio. (e.g. *pCTRLrBreastUp-Down* for the G3F breast **Down** morph) as found by looking within the properties of the desired Down morph for this figure).
- **Maximum deformation** (def = 1, range: 0 to (+) MorphMax **OR** (-) MorphMax): The maximum value you want to apply to your selected **Down morph** range during the simulation. This value is not necessarily the same 'max' value defined in the Daz Studio properties.

Left Morph groupbox defines settings for the Left morph.

- [*text*] The name of the desired **Left** morph-controller in Daz Studio. (e.g. *pCTRLrBreastSide-Side* for the G3F breast **Left** morph) as found by looking within the properties of the desired Left morph for this figure).
- **Maximum deformation** (def = 1, range: 0 to (+) MorphMax **OR** (-) MorphMax): The maximum value you want to apply to your selected **Left morph** range during the simulation. This value is not necessarily the same 'max' value defined in the Daz Studio properties.

Right Morph groupbox defines settings for the Right morph.

- [*text*] The name of the desired **Right** morph-controller in Daz Studio. (e.g. *pCTRLrBreastSide-Side* for the G3F breast **Right** morph) as found by looking within the properties of the desired Right morph for this figure).
- **Maximum deformation** (def = 1, range: 0 to (+) MorphMax **OR** (-) MorphMax): The maximum value you want to apply to your selected **Right morph** range during the simulation. This value is not necessarily the same 'max' value defined in the Daz Studio properties.

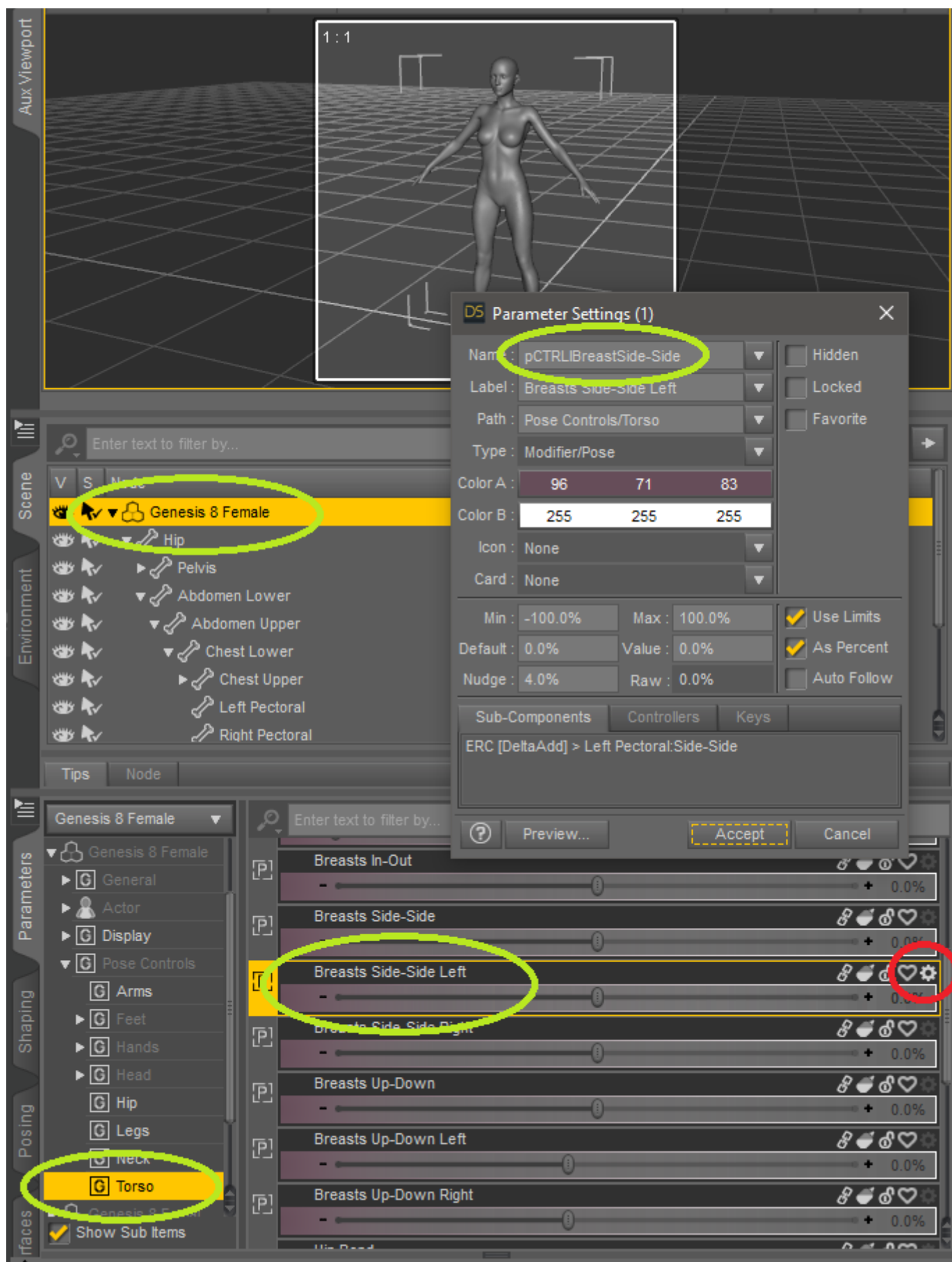
- **Generate the wire:** Press this button to generate the spring wire from the current selection. It will be visible in the simulation if the **visible** checkbox (to the right) is set.

Note: Because you may wish to generate multiple spring wires on a figure, this button commits the wire to memory and *remains in this tab*, ready for further wire creation and saving. Press **Exit** to commit the current figure and spring wires to the Simulation phase.

- **Visible (def = checked):**
- **Exit:** Press this to return to the Utilities tab. Any unsaved Dynamic morph settings may be lost.

The current version of VWD V2.x contains a small set of wire presets for some of the more common DAZ figures. You can load and analyze these to get a idea of how the wire mappings work.

In DAZ Studio, these morph-controller names (e.g. *pCTRLrBreastUp-Down*) are very specifically defined within each figure and need to be copied from the morph slider properties settings of that figure.



How to find the proper controller name (transform, pose, or morph/shape).

5.12 The 'About' Tab

The *About* tab is where you can find information about the program version, contact details, and licensing.

About
VWD Cloth and Hair
Version 2.2.922.6462
Written by VirtualWorldDynamics
Email:
<input type="text" value="support@virtualworldynamics.net"/>
Site:
<input type="text" value="https://www.virtualworldynamics"/>
Support:
<input type="text" value="https://virtualworldynamics.net/s"/>
YouTube Channel:
<input type="text" value="https://www.youtube.com/channe"/>
Patreon:
<input type="text" value="https://www.patreon.com/VWD"/>
The program uses GScene V1.2 for the visualization.

6 Using VWD: Concepts and Logistics

Before we embark on a few very simple examples to help 'articulate' the typical VWD workflow, it might be worth a few minutes to consider what's going on in the 3D dynamics simulation process. Knowing these ideas might really help guide your decision making in the best direction possible.

6.1 Concepts:

As mentioned before, there are two distinct approaches to using VWD, and knowing how **you** want to use these tools is critical to how you should approach your workflow...

You generally create a static scene in your host-application (e.g. Poser or DAZ Studio), add figures and props, lights, cameras, etc. Then apply clothing and/or hair to the figures, pose them (the clothes generally follow the poses, albeit stiffly...). Then start VWD. In VWD, import the figure (as a collision base), import a clothing item, tell VWD about the cloth and collisions you want, add/adjust any forces (gravity, etc.) then run the simulation and watch *and help* it drape. Send the results back to your host-application, save/render, etc.

Before physics-based dynamic clothing simulators were available, 3D-meshes (like dresses) were all fixed polygon models with additional pre-defined morph (shape) adjustments, and/or 'conforming' adjustments that would allow the clothing meshes to 'stick' to specific figures, stiffly following internal bone-rig motions. Talented model artists could generate beautiful clothing products, but the available motions and drapes were limited to the adjustment options that the artists/modelers built-in to their products.

To get the most out of VWD (VWD) in the shortest time-frame, there are a few important ideas that will *really* help get you going. These ideas relate to how VWD's program design impacts the workflow. Here are some of the bigger ideas :

- Before starting, the VWD user must realize that even in a static/still-image rendering context 'dynamic' simulations are ... dynamic. They happen over time. A scene simulation may run for a short time, or a very long time. Still-frame scenes may take a few seconds to 'settle', while longer animations may run for minutes or hours. This idea is new to some users, but quickly makes sense as you begin to understand and master the VWD dynamic workflow.
- Because the available 3D-Mesh products (dresses, hair, etc.) vary immensely in their mesh design and polygon density, the cloth-control settings will also vary accordingly in VWD. This will have a dramatic effect on the VWD Cloth & Hair simulation settings that will define the general behavior of the 3D-meshes in the simulations. This means that every mesh you work with will probably require different settings to get the 'same' results (e.g. getting the same 'silk' feeling with two dresses, may require different settings for each).
- In the most basic VWD scenes, the user will usually dress a figure in a 'conforming' outfit, and pose that figure, with the clothes following - often clumsily. You will then run VWD, import/configure the items and then watch the simulation drape.

This works well-enough, but for the best results, you'll probably prepare a simple figure in a t-pose to final pose animation, usually over a couple of seconds (30 to 60 frames). This pose-to-pose stage is actually simple animation, and the process is done in the host-application (DS/Poser/...). For our purposes, this preparation is quite simple, and does not require any previous animation expertise. That said, it is ***the best*** way to get the most natural results from VWD.

- Rough scenario : You'll use your host-application to place your (human) figure in a 'fitting' t-pose at frame 0, then you'll go to frame 30 (or so) and apply your desired final pose and location. Then go back to frame 0, and dress/fit your 3D clothing to that figure (still in the t-pose). You then use VWD to import these scene elements (a figure and cloth or hair), apply the material settings to the cloth or hair, then simulate this motion sequence, with the clothes following the figure's pose transition motion - from the fit pose to the desired pose. You can then let the simulation settle for a few frames, or pick any frame you like before it settles for a more active look, and send it back to the host-application, and render it !
- The only adjustments that animators will have to make when using VWD is the addition of some start-up frames, so that the clothes, hair, and other dynamic-actors will have time to establish some continuous motion before the desired scene sequences are generated. Kind of a simulation 'ease-in' idea: getting things going before you get to the 'good stuff'.
- Rather than running its simulations on the original scene mesh items, VWD automatically duplicates the dynamic-actors (hair, clothes, flags, blankets, etc.) used in a VWD simulation, and uses *those* items in the simulations. The original items are left intact and simply hidden from view in the host-application. These duplicate items are named identically to the original items, adding a `_VWD` suffix.
- In both static and animated scenes, the simulation process can be started, paused, and resumed multiple times as the simulation progress, allowing for manual mesh adjustment and some simulation setting changes as the calculations progress. This sometimes feels like you're painting on moving water, but it is very powerful, and allows for an amazing amount of control, and it's kind of fun. You can also choose to simply set things up, and let them run, trusting the final result.
- Virtually every VWD simulation will involve 'hard' things (like figure bodies or sofas) that can be animating through time - either moving from a simple fitting position (t-pose), or doing a crazy dance, etc.. We call these elements **collision-items**, as they are not affected by the simulation, but rather they affect the 'softer' things in the simulation.
- Every VWD simulation must have something 'soft' to simulate - hair, clothes, flags, blankets, etc. We call these **dynamic-actors** or sometimes '**actors**' for short. There can only be one of these dynamic-actors in a given simulation session.
- While you can have many collision-items in an active scene, you can only have **one dynamic-actor being simulated at a time in a given VWD simulation session**. This is important because it impacts the entire VWD workflow if you wish to include multiple dynamic-actors (e.g. hair and multiple clothing items) in a scene! We use a 'layered' approach to produce scenes like this. We cover this workflow in detail later.
- Rough scenario : To produce a scene with multiple dynamic-actors (e.g. walking figure, hair, pants, t-shirt) in VWD, you will typically run multiple simulations, where the dynamic-actor outputs of earlier simulations *become the* collision-items in the follow-on simulations, where the newer dynamic-actors interact and collide with the earlier simulation results. For example, you might simulate pants on a figure in a walking scene, then simulate a t-shirt on top of those pants and figure in the next simulation phase, then simulate hair on top of the t-shirt, pants, and figure in the final run. VWD allows (and requires) this kind of control, but produces very realistic simulations where the clothes and hair are actually responding to the layers underneath them.

- Because of limitations found in the host-applications, VWD materials and animations are currently **not** saved in the saved application scene files (e.g. my_scene.duf or my_scene.pz3)! When these scenes are reloaded, neither the materials or animations are immediately available to the **xxx_VWD** scene items until you manually 're-apply' these effects using the VWD application. You must do this each time you reload a scene, and the simulated animation files must still be available in the VWD core program directories. If those files are removed, the simulation animation files will have to be restored from some sort of backups, or they will have to be regenerated.
- While the term 'Cloth' in the VWD Cloth & Hair context is pretty clear (clothes, sheets, flags, or other arbitrary pieces of fabric), the term 'Hair' is broadly defined as being dynamic mesh items that are attached to one of the collision items defined in the simulation session. Things like flags on poles, horse manes, frills on dresses, and many other items of this sort can all be considered 'Hair' in the VWD context: Groups of soft things (dynamic-actors) attached to hard things (collision-items).

Of course there are many more controls and specifics on the actual usage of VWD's Cloth & Hair toolkit, but these general concepts will hopefully instill a sense of how the package works, allowing you to better navigate the program and its design paradigms.

Finally, there is a separate but very closely-related usage guide: *VWD's Cloth & Hair Usage Guide*, that's intended to provide step-by-step examples of real-world usage scenarios being presented using various VWD capabilities. Video tutorials are also being produced. Access to these and any other available resources will be provided at the end of this document.

6.2 Logistics:

6.2.1 Backing-up and Restoring VWD Project Files:

During a VWD simulation session, a collection of files is created. While many are disposable working files, others are the results of your session and may be worth saving. Here are some tips on saving the work you've done in VWD for possible future re-use. These saving tips cover saving your general VWD presets, static simulations, and dynamic simulations. Note: you should also backup your DS/Poser scene files and

General Notes:

- VWD was not designed to facilitate many of the backup techniques that are described below. All of the working files and presets are currently generated and stored in subfolders of the main VWD program directory (e.g. C:\VWD\...). Backing up these files will have to be done by manually copying or 'zipping' them into your preferred backup locations (presumably with the project's scene and content libraries) using any Windows file manager (Explorer) and/or archiving tool like WinZip, 7zip, WinRAR, etc.
- Because VWD saves and finds its files so transparently, you may not realize that many of these working and results files are being generated, used, and saved in the background! As the VWD system is used over time, these directories with presets, working files, and simulation result-files will grow larger - especially for animators! At some point it may seem prudent to clean out these old files, which is quite easy using the *Utilities tab's Delete all "Exchange" files button!* (Careful!!!) But once that is done, virtually none of your older scenes will reload completely, as some of their key VWD dependencies have been removed. Hence this section. Right after a project is completed, consider saving your host-application scene-file, and any or all of the VWD loading presets and result animation cache-files along with it - in a consistently named and safe place, maybe zipped up with a reminder 'readme.txt' file for your future reading pleasure!
- There are three key ideas to be sensitive to as you consider your backup scheme(s):
 - Are these static or dynamic simulation scene elements being saved?
 - Which files to backup?
 - Where they come from?, and where to put them back!
- **Animation lengths and contents matter!** In any scene, if you change the length or motions in the animation, virtually all of the pre-generated animation-related work and result files will become obsolete, and restoring older versions that no longer match will not work as you might expect (or at all). Your wind, recorded presets, dynamic morphs, and hair vertices files will still be relevant, but not the animated simulation. In the case where you update the length and/or content of an animation, all of the
- The VWD folders and the save-worthy files they contain are currently located in your main VWD program directory (e.g C:\VWD\...) as follows:
 - ...\\VWD\\Exchange : Most of the active working files. most of these can/will be reproduced by VWD from the original scene-file.

[Optional] - The *.vwd files that are named after a nearby *.obj files (e.g. dress.obj and dress.verts) contain the 'recorded' animation of the obj file's animation. This would have been generated as a time-saving pre-calculated cache-file by pressing the **record** button in the **Collision parameters** tab after loading or selecting a loaded collision-item. Saving and restoring these files will save you from having to repeat the recording step again and increase the simulation speed.

Important: The VWD **Utilities tab** button **Delete all “Exchange” files** will remove all of the files in this directory! If you press this button, there will be no confirmation or second chance. The files in this directory will be gone.

- ...\\VWD\\Exchange\\Scene : These are the simulation animation results files and can grow large. They are stored in the Poser cloth-room animation format and can be reloaded by Poser and/or DAZ Studio when the relevant simulation source item (xxx_VWD) is loaded as a collision-item. You can see the simulation results played on the item when you (re)import it as a collision-item (after simulating it as a dynamic-actor).

*.dyn: animation results from dynamic simulations. These can get large. These are only meaningful if the original animation length has not changes, and the nearby collision-items in the scene have not had their animations changed since the simulation was run. Restore these if you are simply reloading an unchanged scene for re-rendering.

Important: The VWD **Utilities tab** button **Delete all “Exchange” files** will remove all of the files in this directory! If you press this button, there will be no confirmation or second chance. The files in this directory will be gone.

- ...\\VWD\\Dynamic Morphs (presets) : These are the presets for any dynamic morph configurations that have been pre-configured on specific figures as described in the **Dynamic morphs** section of this manual. Any additional presets that you create will also be saved here as well.

*.dmo : These are the VWD and user's Dynamic morph preset files that contain the 'wires' and morph controller and weight assignments described in the **Dynamic morphs** section of this manual. You'll want to save any of these files that you created while using the Dynamic morphs configuration dialog. (Recommend: save any that were not shipped with VWD).

- ...\\VWD\\Hair Vertices (presets) : These are the presets that define the hair attachment vertices selection sets that are used when attaching dynamic hair actors to figures. The VWD presets are labeled for their figures. User presets are saved here as well.

*.hrv : General presets for head/scalp vertices selections that will be used to attach hair meshes to specific figures. You'll want to save any of these files that you created while using the hair attachment configuration dialog. (Recommend: save any that were not shipped with VWD).

- ...\\VWD\\Recorded Imports (presets) : These are the so-called RIP file presets that record your session import and configuration steps that occur before a simulation. These can be useful when re-working or re-using a set of figures, clothes, and collision-items in a variety of scenes. For example, a graphic-novel artist may have one or more figure/outfit and/or figure/hair RIP files that will allow for a quick load and 'fit' of the dynamic-actor to the figure for a given scene. This can be very handy.

***.rip:** General import presets. VWD stores two special RIP files: Last.rip and Restore.rip, which are reserved for reloading the last scene, whatever it is/was. Any other RIP files are probably related to recent item imports (VWD creates these too, using the name of the item), and any user-saved RIP files will be saved here too. You'll want to save any of these files that you created while importing and configuration your cloth/hair/collision-items. (Recommend: save any that were not shipped with VWD).

Note: VWD has a *No collisions* button in the **Scene import** tab, that will allow you to load these presets ignoring any collision-items that may be saved in the RIP file. This makes these presets less scene-specific and more generally useful.

Note: You can create these RIP presets at anytime (between item imports) to isolate your items if desired. You can also load multiple RIP files in a single session, although there will still only be one dynamic-actor (hair or cloth) allowed in a VWD session at a time.

- **...\VWD\Recorded Springs** (presets) : These are presets that VWD saves when you apply various spring settings (stiffness/attachment/collision) to a dynamic hair or cloth actor.

***.rsp:** Presets for any recorded springs that were defined during import and loading of you scene's dynamic-actor. These are named based on the dynamic-actor name. You'll want to save any of these files that you created while importing and configuration your cloth/hair/collision-items. (Recommend: save any that were not shipped with VWD).

- **...\VWD\Recorded Wind** (presets) : Wind presets, used during animations. This directory contains the VWD presets and any 'computed' presets you may have generated and saved. Note that these presets may be especially useful in creating consistent wind effects in multiple dynamic-actor simulation sessions (e.g. wind on hair, a dress, and a nearby plant should all be fairly consistent in a scene, so re-using a wind preset will be desirable.

***.wnd :** Presets for VWD 'standard' wind settings and any user-saved wind settings. You'll want to save any of these files that you created option. (Recommend: save any that were not shipped with VWD).

- The dynamic wind sequence files (save these to use as the same wind source in a scene with multiple simulation targets in future sims or re-dos) Note: these are overwritten for each wind sequence generation. You need to save these between projects!
 - **.../Exchange/Saved wind.rwd :** The wind settings (randomness and change amounts) – overwritten for each wind sequence generation. You need to save these between projects!
 - **.../Exchange/Animated hair.rwd :** The actual wind sequence settings use in each frame – overwritten for each wind sequence generation. You need to save these between projects!

Various Backup Schemes:

Here are some backup schemes to consider. Remember to backup your related scene-files and content folders too - these backup recipes are just for the VWD-related workfiles and presets that you might want to use later.

- **The brute-force method:** To be absolutely certain you've backed up everything you need for a project (and more), zip-up or copy the entire VWD base directory to a separate backup medium. Maybe add a **readme.txt** file of your own with date, project, and other notes for the backup. Noting that the entire VWD system is only about 30 Megabytes, having multiple copies of these extra files won't really be a big problem for most users.
- **The nearly brute-force method :** As above, but you only backup the folders described above. It'll save you a few megabytes... Maybe add a **readme.txt** file of your own with date, project, and other notes for the backup.
- **Saving your work from a static simulation scene/session :** During a static VWD simulation session, the **Send pose to host** button in the **Simulate** tab will immediately update the DAZ Studio scene with the current state of the dynamic-actor cloth or hair. At this point (and without leaving VWD), you can bring DAZ Studio up, and save the present state of the simulated item as a scene-subset (save-as->scene-subset), and only select the current VWD item (xxxxx_VWD) checkbox, and it will be saved in the DS duf file as it currently is displayed. This can be reloaded later (it may need to have textures re-applied).

In this mode, you may also wish to save the project's VWD session configurations:

- recorded import (rip) files
- recorded springs (rsp) files
- recorded wind (wnd) files

And any general configuration presets that you used while working on this scene:

- dynamic morph (dmo) files
- hair vertices (hrv) files

Note: because *static* scenes don't record animations or sequential simulation results, I don't think anything in the ...**\Exchange** or ...**\Exchange\Scene** directories needs to be saved. These files can be regenerated when the scene is reloaded into a new VWD session. Save these results in your host application, and backup your presets as recommended here.

- **Saving your work from a dynamic (animated) simulation scene/session :** During a dynamic/animated VWD simulation session, the **Send animation to host** button in the **Simulate** tab will immediately update the DAZ Studio animation timeline with the current state of the dynamic-actor cloth or hair from the simulation (VWD will then exit...). At this point you can bring DAZ Studio up and review the simulation results on the DS timeline.

Important: As noted in other sections of this document, after a VWD simulation session, the animation sequence is loaded into the host application memory, but can't be saved into a native DAZ Studio or Poser scene-file as is! Only the usual scene information can be saved as a scene preset (or subset), and the animation has to be reloaded into memory from the VWD simulation results file using VWD. This is why those simulation files must be saved and restored later when you're re-activating a scene is reloaded. VWD will also be required to reload these animations into DS memory. The reset of the scene can be reloaded normally (although the _VWD items may need to have textures re-applied).

In this mode, you may also wish to save the project's VWD session configurations:

- recorded import (*.rip) files
- recorded springs (*.rsp) files
- recorded wind (*.wnd) files

Any general configuration presets that you used while working on this scene:

- dynamic morph (*.dmo) files
- hair vertices (*.hrv) files

The animation working and simulation results. (critical!)

- animation recording cache (*.vwd) files for the various collision-items
- animated simulation results (*.dyn) for the dynamic-actors (critical!)

The dynamic wind sequence files (save to use as the same wind source in a scene with multiple simulation targets in future sims or re-dos)

- The wind settings (randomness and change amounts) – .../Exchange/**Saved wind.rwd**
- The actual wind sequence settings use in each frame – .../Exchange/**Animated hair.rwd**

All of these file-types and their locations are described in more detail above.

Important: These files must be restored to their original locations (or at least to their VWD 'home' directories if you're using different systems). VWD will only look in its default locations for each of these file-types.

Besides the obvious good-practices value of any backup regimen, The extra value of backing up these various files is:

- Being able to open and re-activate an existing scene, simply for review and re-rendering it - as it is...
- Being able to produce new scenes using assets and settings from an existing scene that involved VWD and simulation settings.
- Being able to collect all of your base figure and outfit settings and re-install them in a new VWD installation, or share them with others. Hair vertices, Dynamic Morphs, Wind settings - can all be created and shared with others who may have the same products and can take advantage of your efforts.

With these tips and VWD workfile and preset descriptions, you should be able to save as much or little of your VWD settings as you wish, perhaps saving you some time and trouble later!

6.2.2 Vertices and Selection tools

The *Vertices selection* tab is an amazing little tool that improves the mesh-selection process immeasurably, because VWD users need to select meshes a lot... The powerful memory-buffer tool gives you a couple of extra virtual hands to manage and master your vertex selection efforts. Mix this with some lucid navigation skills, and you can get a lot done very quickly and accurately.

By letting you 'hold on to' lots of little vertex collections as you navigate around a complex session item (like a hair actor, or complex dress actor), you can accumulate all of the little parts you need as you rotate, zoom, and angle your way into complex 3D-mesh geographies, grabbing and storing only the vertices you want and need for your tasks.

You can throw them into these handy memory accumulators, and add them, remove them, invert them, etc. until you have the perfect vertex collection, ready to use as the context requires.

Just so you know, this selection process is absolutely critical to really mastering the power of VWD. It is how you adjust the stiffness of things, attach things together, assign influences like attraction and repulsion, and generally tell VWD what areas (all, some, or none) you wish to affect with your settings. Firming up ponytails: you need this. Making buttons firm, then gluing them to a coat: you need this. Attaching hair to a head: you need this. Making ribbons collide differently than a t-shirt: you need this... You get the point...

Important: While this process is very much visually driven and happens entirely in the *Scene viewer* tab, your approach to this selection process will certainly determine your efficiency and enjoyment of this necessary procedure. It is used a lot, so a bit of reading here might really save you a lot of time and energy, and prevent a good amount of frustration.

Mouse Navigation:

First, some quick review: Please review the mouse and keyboard navigation techniques outlined in section “4.2 The *Scene viewer* tab” before taking on anything but the simplest of selection tasks. You will thank yourself countless times if you choose to familiarize yourself with the *Scene viewer* navigation tools, so, let me repeat those here:

The mouse must be in/over the *Scene viewer* viewport for these controls to operate as described.

- **Zoom:** The use of the mouse wheel allows zooming in the scene. Pressing the **Alt** key while using the mouse wheel reduces the zoom speed for more accurate positioning.
- **Rotate:** Right-click and hold the mouse in the *Scene viewer* viewport, and move the mouse to rotate the scene items. The scene should rotate around the barycenter of the active scene item, unless that rotation point has been 'recentered' (see below)
- **Reset Rotation and Zoom:** Double-click (left button) with the mouse anywhere in the *Scene viewer* tab to reset the camera view. This will also reset the rotation-center to the barycenter of the current active/selected item in the scene.
- **Recenter Rotation:** Move the mouse to where you would like the rotation-center to be (on the active-item), and click the mouse-wheel (center button). This should become the new rotation-center of the scene.

- **Reframe (Panning):** While pressing the **Ctrl** key, right-click and drag the mouse in the direction you wish to **shift** the scene view.
- **Follow the active-item** (toggle): Press (and release) the '**f**' key to have the scene camera center-on and follow the currently active-item in the scene. This is particularly useful when simulating during animations.
- **Top view:** Press '**5**' on the numeric keypad (numlock on...)
- **Front view:** Press '**8**' on the numeric keypad (numlock on...)
- **Back view:** Press '**2**' on the numeric keypad (numlock on...)
- **Right view:** Press '**6**' on the numeric keypad (numlock on...)
- **Left view:** Press '**4**' on the numeric keypad (numlock on...)

active-item display modes: let's not forget the available active-item display modes. The active-item, the one you're probably selecting vertices on, can be assigned as *active* in the **Scene import** tab, where you Left-click on it (use **Ctrl**-click to hide everything else, if you want some clear space in the viewport...). This item can then be displayed in a couple of different ways in the **Scene viewer** tab, each having some value in this vertex selection process. My favorite view mode is probably the hidden line display ('**h**'), but sometimes the open mesh mode ('**a**') is useful too. The full description is in the “4.2 '**Scene viewer**' tab” instructions, but again, I'll repeat them here for easy/relevant access:

The Session's active-item viewing modes:

When items of any type are imported into the VWD scene, they are displayed in a simple 'clay' mode by default and are assigned arbitrary pastel colors (these do not relate to the original scene item materials and can be changed in the **Utilities** tab).

Items can be also shown in different display modes at any time. After selecting the item in the **Scene import** tab's listbox, the item will become the active-item in the scene. In addition, the active-item's **Scene viewer** display mode can be changed by pressing the following keys *while the mouse is in/over the display viewport*:

- **Transparent view** (toggle): press the '**o**' key and the active-item will become partially transparent
- **Mesh view** (toggle): press the '**a**' key and the active-item's mesh will be displayed with no 'skin' on the mesh.
- **Hidden line view** (toggle): press the '**h**' key to display the active-item using the 'hidden line' view, where the mesh is visible with an underlying mesh skin.

As each of these controls is a toggle, it is easy to cycle between these display modes at any time.

7 VWD Utilities

The VWD Cloth & Hair installation comes with some small windows-based utilities that may help you get your data into or out of your VWD workflows. These are described in this section.

7.1 Utility: *Animation conversion.exe*

Within the base VWD installation, there is a small Windows-based stand-alone animation-file format converter called *Animation conversion.exe* (e.g. **C:\VWD\Animation conversion.exe**) that allows you to convert the animations that were generated in your VWD sessions (*.DYN) to an alternative format (*.PC2).

It is a specialized tool that will probably be very useful to a small number of users who also use the 3DS Max 3D-application environment in their workflow.

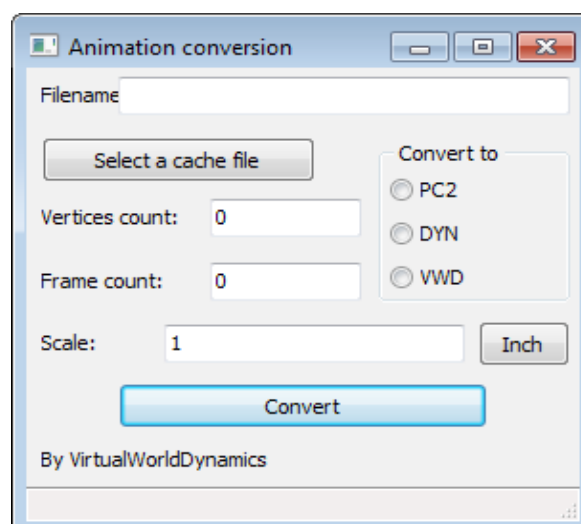
Some technical background information:

- When VWD runs a dynamic/animated simulation, the mesh alterations that occur across each frame are stored in a VWD file (*.DYN) that is based on the format used in Poser Cloth-Room simulations (good for compatibility).
- This file is the only record of the animated mesh sequence, and can be loaded into a DAZ Studio item, but it cannot be saved in DAZ Studio (e.g. as a scene-file or pose-preset) because DAZ studio does not support this kind of mesh-animation file.
- The PC2 format is a similar format that is used by the 3DS Max 3D application.

This *Animation conversion.exe* utility is a small stand-alone conversion tool that will let you convert the VWD *.DYN file(s) directly to the 3DS Max *.PC2 formats for use in 3DS Max workflows.

To use this utility:

- Use your Windows file-browser browse to your base VWD installation directory (e.g. **C:\VWD\...**)



- Double-click on *Animation conversion.exe* to open it.
- Find your source file by clicking on the **Select a cache file** button and browse to find your VWD animations file. By default it will be located in your VWD installation's **Exchange\Scene** folder (e.g. **C:\VWD\Exchange\Scene*.dyn**), and named something like your VWD item.

For my **Plane** scene item (a test of a cloth blanket), VWD created a working file in my scene called **Plane_VWD**, and the dynamic simulation animation file is located in VWD's folder and is called **NewScene_SimPlane_VWD_Plane_VWD.dyn**. **This naming is based on the Poser naming convention, so it will be useful in Poser.**

Once you select the file, it should populate the utility fields with information about that file.

- Select your file target format, (probably **PC2**)
- Start the conversion by pressing the **Conversion** button.
- When the conversion is complete, the word *Termine'* is displayed at the bottom of the interface.
- You can now close the utility if you wish by pressing the 'X' in the upper-right corner, or you can find and convert additional files.

All of the output files should be generated in the same directory as the original source files.

Warning: This utility will happily overwrite any existing output files of the same name without notification. It is a simple tool, so be careful.

- Find your file: Use your Windows file-browser browse to your VWD animation cache directory (e.g. **C:\VWD\Exchange\Scene\...**) and look for you new output file, which should be roughly the same size as the original VWD *.dyn file.

This utility should be able to convert from any of the three available formats to the other two formats. This might be really handy for users with a bi-directional workflows.

8 Some Very Simple VWD Examples

While there are a couple of comprehensive usage guides (e.g. *VWD's Cloth & Hair Usage Guide*) and various tutorials that will really help get you producing good results from VWD, here are couple of 'simple' tests to try with some basic 3D-mesh primitives that everyone can run through to get a feel for the VWD interface and workflow.

Of course, everyone who starts out with VWD wants to take their gorgeous detailed 25-part 3D Victoria4 Princess Ball Gown, throw it onto a Genesis 8 Female figure, and have her sashay across the Ballroom Floor scene and produce the beginnings of their epic period-piece animation or still-image...

Well, the examples that follow aren't exactly that, but will probably be appreciated as reference resources for the underlying ideas and usage of specific features, and maybe help you get started on the path to your epic ballroom scene...

Spheres and planes aren't really very interesting in terms of results, but through their simplicity, perhaps these simple examples can help jump-start your intriguing projects.

All of these examples assume that you have successfully installed your VWD program and that it is functioning properly. These simple examples are a good way to verify that...

This first example will be a bit tiresome in its detail (really), but the follow-on tutorials will assume that you will then know how to make a basic 3D scene with some primitives, start VWD, import both types of VWD items (collision-items and a dynamic-actor) into a VWD session, and will move more quickly along, helping to demonstrate the targeted features.

8.1 The Amazing Deflating Sphere! (a very basic dynamic-actor simulation)

In this very verbose/detailed example, we import a simple sphere primitive from our local host-application into VWD as a *dynamic-actor*. This is what we call the only item that is actually simulated in a VWD session.

Note: It's worth mention before we start that this HowTo will describe a *static* simulation session. This is slightly different from a *dynamic*, or animated simulation, which works using the same ideas, but involves animated scene elements and VWD's requesting animation frames from the host-application.

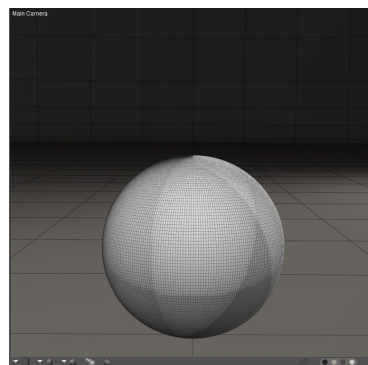
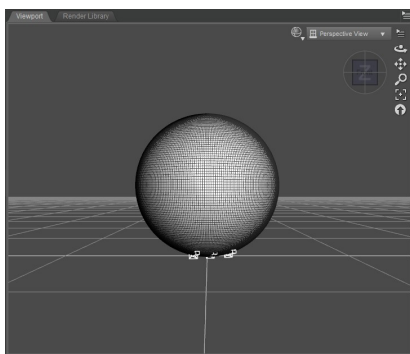
In that importing process, we will first assign the sphere to be a cloth-actor, using the default values, and we will simulate it, reset it, simulate it again, then return the results as a single static simulation snapshot result to the host-application where it can be saved and/or rendered.

Starting out: Create a brand new scene your preferred host-application:

- Create a simple sphere, and specify enough polygons to simulate well.

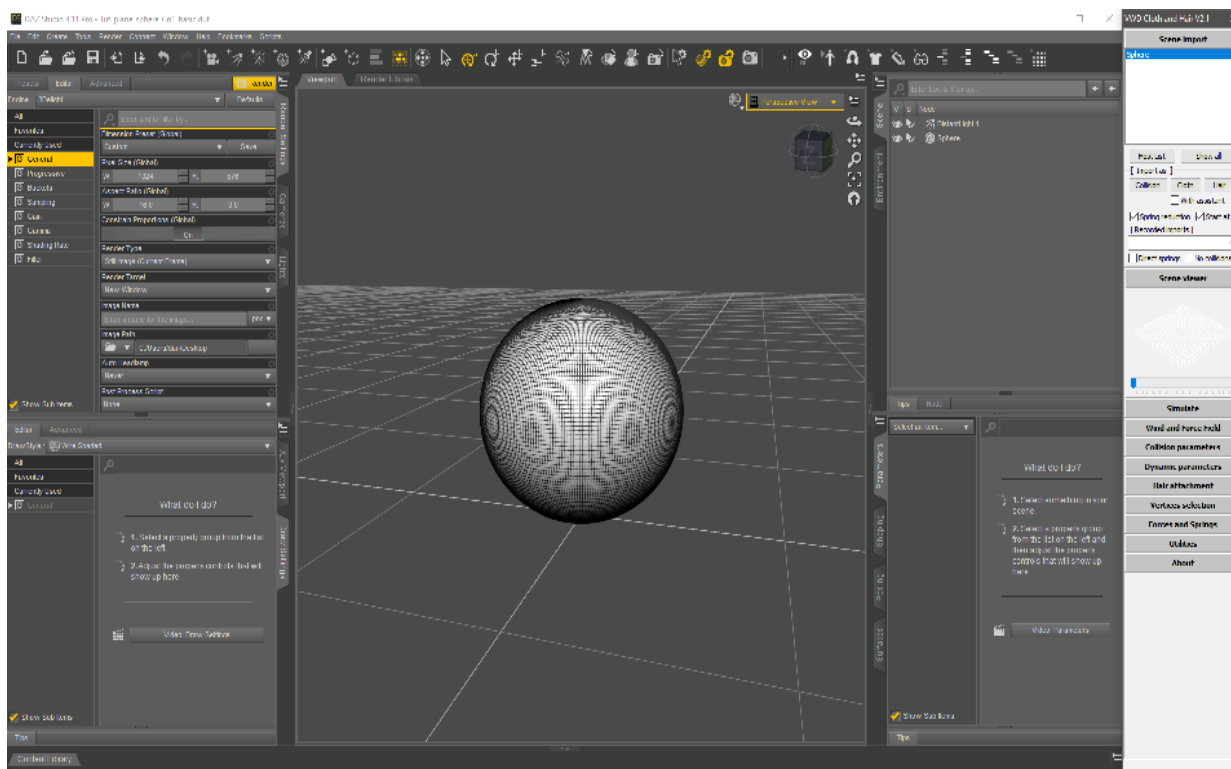
Important: As you start to use VWD it is really a good idea to take notice of the density of the polygons in the items you'll be using in your projects! The sphere that we'll be creating below is a decent average mesh-density starting point and closely matches Dforce, Poser-ClothRoom, and DAZ dynamic-clothing control (optitex) mesh densities. I would bet that Marvelous Designer meshes are also similar in density.

- So, in DAZ Studio (DS), import the OBJ version of the same sphere from the included resources folder in your installation package (C:\VWD\Working_Files\OBJ meshes\DS Sphere.obj) by simply dragging it from the folder, right into the DAZ Studio viewport, and using the DAZ Studio pull-down import settings. (You can create a sphere in DS if you cannot find these files.)
- In Poser, import the OBJ version of the same sphere from the included resources folder in your installation package (e.g C:\VWD\Working_Files\OBJ meshes\Psr Sphere.obj) by simply dragging it from the folder, right into the Poser viewport, and using the default(?) import settings.



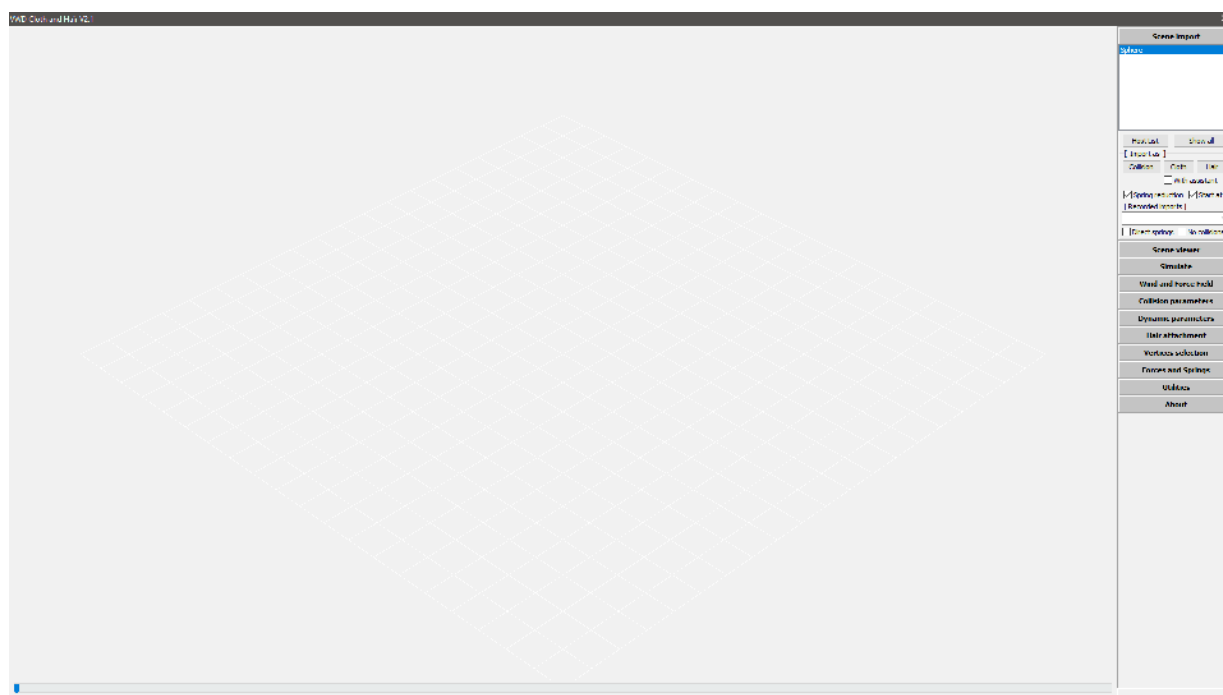
- Now, start VWD!
 - In DS, browse to your VWD scripts installation folder and double-click on your *Start VWD.dsa* script...

- In Poser, click on the python scripts '...' menu item, browse for your VWD scripts, and open the **VWD buttons.py** script. Then press on that new button, followed by pressing on the **Start VWD** button in the updated menu... Once VWD starts it should find its place along the right end of one of your monitors, but it can be dragged and operated in any of your monitors. When using DAZ Studio with our Sphere and a running VWD program window, it should something like:

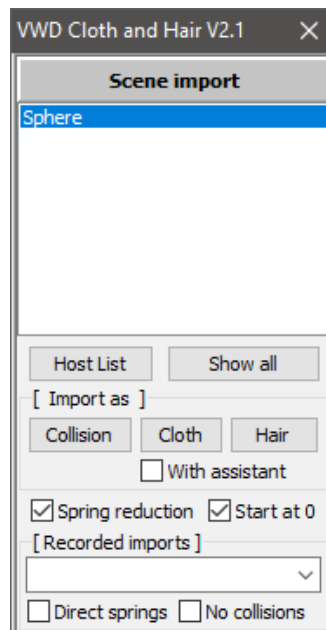


(I'll go with DAZ Studio for most of the rest of this example, but the ideas are the same.)

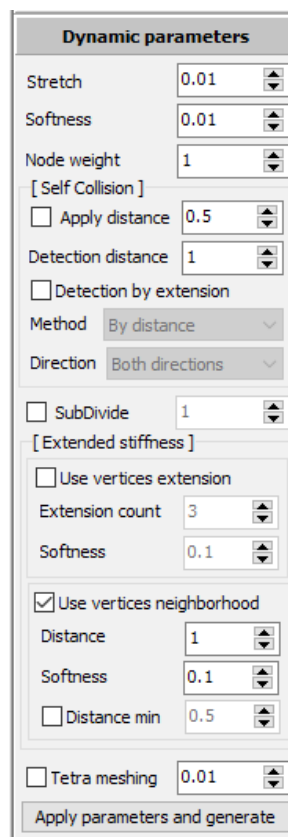
- Make VWD 'big' (full-screen) by clicking on VWD's **Scene viewer** tab header. (Click it again to reduce it back to the little display... - it's a toggle)



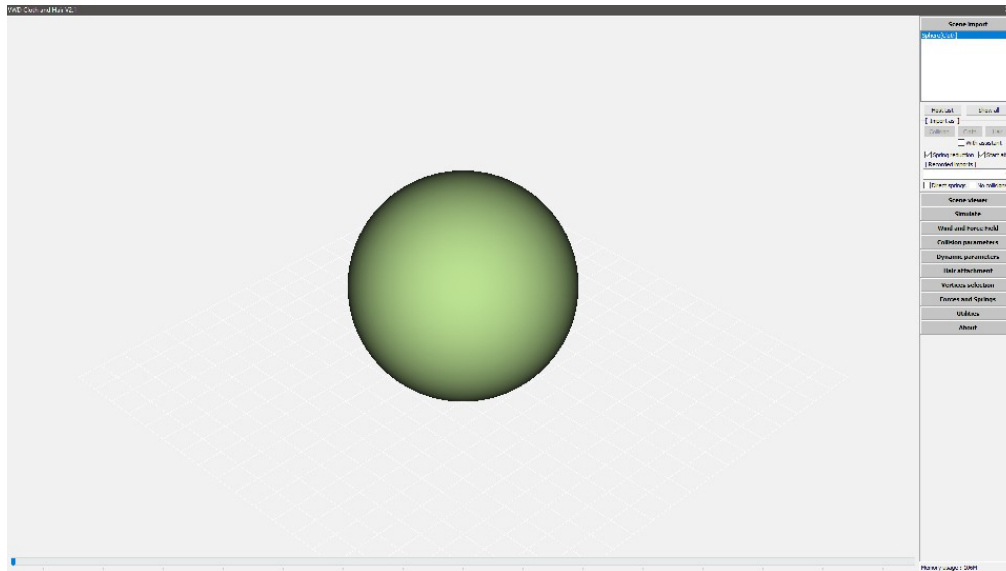
- **Importing your Sphere as a dynamic-actor (cloth):** Find the Scene import tab at the top of VWD, and click on the Sphere item in your import listbox to select it.



- Make sure the little *With assistant* checkbox is not checked and click on the *Cloth* button to actuate the *Dynamic parameters* tab.



- This tab is where you will set the basic/global settings of the Sphere. We will start out using the defaults, but this is where you assign the dynamic-cloth-actors their initial and overall stiffness and related properties. (You can adjust all or specific parts of the mesh items after this step).
- Click on the *Apply parameters and generate* button at the bottom of this tab, or the **Cloth** button back up in the *Scene import* tab at the top of the interface. Both buttons do the same thing in this context.



- You've successfully imported a Sphere into VWD and made into cloth! Note that the import listbox has added this item's role **[Cloth]** to the item name, indicating that is now imported, and because it's selected, it is also the active-item in the program, which becomes important later in the program.
- I like to see the mesh on the models, so I like to move the mouse into the *Scene viewer* viewport, and press the 'h' key on my keyboard and show the mesh. (press it again to hide the mesh again, it's a toggle). You can try the 'a' and 'o' keys to see the other display modes that are available to you. They're all toggles, so press them multiple times to see and hide their display effects.

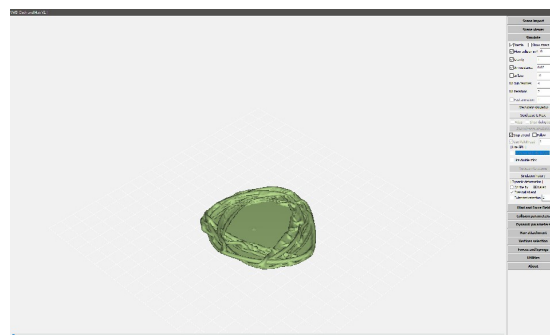


Note: I'll mention here that if you accidentally lose your sphere by navigating it out of the viewport, just double-click anywhere in the viewport and the current active-item (the sphere in this case) should jump right back to the middle of the viewport. The viewport navigation tips are described in detail earlier in this manual ("4.2 The 'Scene viewer' Tab")

- **Running the Simulation:** Now for the fun part – to simulate this virtual cloth Sphere... Click on the ***Simulate*** tab header to open the simulation settings. There are all sorts of settings here, but like before, let's leave them all at their defaults *except to enable the CPU* (vs GPU if you have one), by clicking on the ***Use Multithread*** checkbox near the middle of the ***Simulate*** tab.

Simulate	
<input checked="" type="checkbox"/> Inertia	<input type="checkbox"/> Show stress
<input checked="" type="checkbox"/> Floor collision in Y	0
<input checked="" type="checkbox"/> Gravity	1
<input checked="" type="checkbox"/> Air resistance	0.02
<input type="checkbox"/> Inflate	10
Nb Sub Samples	4
Nb iterations	2
<input checked="" type="checkbox"/> Fast animation	1
Start static simulation	
Send pose to Host	
<input type="checkbox"/> Video	<input type="checkbox"/> Show dialog box
Start dynamic simulation	
<input checked="" type="checkbox"/> Stop at end	<input type="checkbox"/> Follow
<input checked="" type="checkbox"/> Use Multithread	8
[Use GPU]	
<input type="checkbox"/>	GeForce GTX 260 TI OEM
<input type="checkbox"/> Use double stick	
Vertices interactions	
Simulation history	
[Dynamic deformation]	
<input type="radio"/> On the fly	<input checked="" type="radio"/> Nailed
<input checked="" type="checkbox"/> Free nail at end	
<input type="checkbox"/> Extended selection	2

To start a basic static simulation, simply click on the **Start static simulation** button and ... watch your wonderful sphere deflate into a pile of chaotic meshiness! (**Experiment:** Put the mouse over the viewport and press the 'h' key again to hide the mesh lines and see the un-lined sphere ... in a different way).



Our sad little deflated sphere!

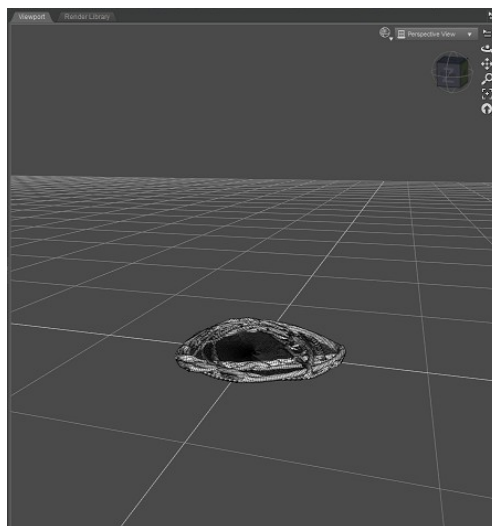
- Note that while the simulation is running, the original **Start static simulation** button changes to **Stop static simulation**... So to stop the simulation, click on that button and your simulation is done*, and we can now send it back to the host-application.

* Well, to honest, I lied a bit. When we clicked on the ***Stop static simulation***, we only *paused* the simulation! We can let it continue simulating some more if we like by clicking on the ***Start static simulation*** button (yes, the button changes back and forth between *stop* and *start...*) but it probably wouldn't be a whole lot of fun as most of the action in this simulation seems to have settled...

Hint: At this point, let me inject some good-to-know wisdom about VWD: While we won't do it here in this simple example, you can continue to do all kinds of things to this seemingly completed simulation! This is what makes VWD users so fanatic about the tool! While the simulation is running, you can turn off gravity, adjust wind, disable inertia, and otherwise tweak many of the VWD settings ***while the simulation is running!*** Then... after 'stopping' (pausing!) the simulation, you can press the **Shift** button to do 'bursts' of simulating, and during those 'bursts', you can click-and-drag the mesh around with your mouse... then continue the full simulation again to let it settle again. In principle, you could easily spend an hour straight in this single continuous session creating one or many 'perfect' deflated sphere 'sculpture(s)'

Only after you've played with this a bit will you realize the infinite potential sculpting that you can do with VWD; sculpting that would take hours in a traditional modeling program. We'll do some videos that really showcase this capability, but please take a moment to consider that VWD is ***NOT*** designed as a “set it and watch it” type tool. It *can* be used that way, but its power (and fun) is in its set, run, tweak, tug, run some more, save, run some more, save another version, workflow design.

- **Saving the results:** We can send this blob of a 'sphere' back to the host-application by clicking on the ***Send pose to Host*** button. Go ahead and try it, then pull up your host-application! There's a blob of deflated sphere in the middle of your Poser or DS scene viewport! You can save this as a scene or scene subset (that's what I do), or you can export this blob as an OBJ file, or anything else your host-application can do with a scene item.



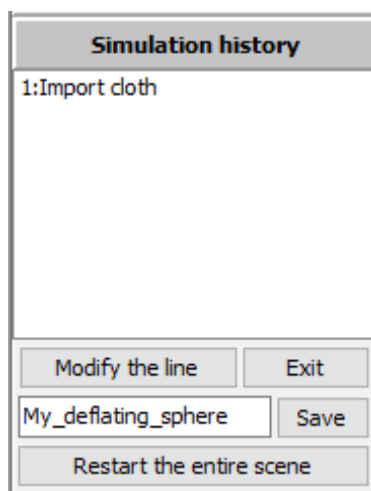
- **Resetting the simulation:** But wait!, there's a little bit more to this example. How about if we wish to reset the entire simulation and start from scratch without having to re-import everything each time? Let's do that.

To reset the static simulation at any point during your session, float the mouse over the ***Start static simulation*** button, and **Alt**-click on that button (**Alt**-clicking is when you press the **Alt** key *while* clicking on the button). Your Sphere should pop back to full inflation! You can do this *any* time, and *many* times! Remember that you can also save your item pose to your host-application, save it there, then return to VWD and play/save/reset/play/save/reset ... as many times as you like, to generate a whole collection of simulation results. Of course, this makes a lot more sense when you learn about the other available manipulations at your disposal in VWD, but for now...

- **Interactive 'burst' simulations:** OK, here's our last trick for this session: You can interactively simulate in 'bursts' using the **Shift** key!

So, with a shiny new reset Sphere in your viewport, go ahead and try it: First, make sure the mouse cursor is anywhere within the ***Scene viewer*** viewport and simply press the **Shift** key... then let go... then press it again... Cool, eh? Just wait 'til we get to dynamic-deformation, wind forces, and the many other VWD cool-isms!

- **Saving what we can:** Before we exit, let's save our VWD session's *settings*. Note that this only saves the current import item *list* and their *settings*, and does **NOT** save the actual simulation results or any meshes! Static simulation results are always saved using the ***Send Pose to Host*** button and then manually saving the results from your host-application! That said, to save the current session's item list and the item's VWD properties (stiffness, stretch, etc.), in that same ***Simulate*** tab, find and click on the ***Simulation history*** button, and simply type a filename of your liking (e.g. My_deflating_sphere) and press the ***Save*** button.



To return to the ***Simulate*** tab, click on the ***Exit*** button. (This tab has other features but we'll cover them later).

Note: After clicking the ***Save*** button, not much visible seems to happen, but in the background, these settings are now stored in the main VWD program '**Recorded import files**' folder and can be reloaded using the **Scene import** tab's ***Recorded imports*** field. Click in the field and look for your new filename.

Important: When accessed later during a future import, these recorded import files expect the various items they were created with to be available by name from the host-application scene, so don't load a new scene with different items and expect these VWD import files to do anything useful if the original items are not available in the active host-application scene.

- **Leaving VWD:** Finally, you can save the results of your session (***Send pose to Host***), or... you can simply exit VWD and return to your host-application by clicking on the 'X' in the upper right corner of the VWD window.
- **Side-effects of VWD:** Back in your host-application, you might notice that a second sphere object has been added to your scene-list (or Props list in Poser) called **Sphere_VWD** (or **xxxx_VWD** where "xxxx" is your sphere's name). You may also notice that your original item is now hidden in the scene. When a cloth or hair actor is imported into VWD, the simulation is actually run on an unriggered *duplicate* mesh, which is added to your scene and named with the **_VWD** suffix. This means that all of your renders, animations, and other VWD results will be done with these **xxxx_VWD** files.

It won't 'break' a scene-file to remove these files and unhide the original items, but for now, understand that these are how the VWD results are added to your scenes, leaving your original items alone. These new item duplicates do add the new mesh information (polygons/vertices, etc.) to your scene-file, so your scene-file may jump in size when you save it after your VWD sessions.

So that's a wrap for this HowTo session. In this session we:

- Started a new scene-file
- Added an item (Sphere)
- Started VWD
- Imported the item as a *dynamic-actor*, of the 'cloth' type (not 'hair' or 'collision') in the **Scene import** tab, using simple defaults.
- We opened the **Simulate** tab, and set it to use the CPU
- We learned how to start and stop a simulation with the dual-mode **Start/Stop dynamic simulation** button
- We watched the Sphere deflate (cool!)
- We **Sent the pose to Host** and saved the result in our host-application.
- We came back to VWD and learned how to **reset** a static simulation, and how to do interactive 'burst' simulations with the **Shift** key.
- We learned how to 'save' the current items and property settings for the VWD session using the **Simulation history** tab.
- Finally, we exited VWD and went back to our host-application and produced some of the best renders of deflated beach-balls that the world has ever seen! (or not...)

This little simple session really covers the essence of VWD operation. A few things change when we run dynamic-simulations (simulations that contain animated elements and all frames are saved...), but the ideas are the same. From here, you will add 'hard' things (we call them collision-items) to your scene so the dynamic actors can interact with things in your scenes, and then you start to fine-tune your dynamic-actor mesh areas (e.g. belts on dresses, hair and ribbons), attach things that aren't welded together on the mesh, and play with all of the available forces in the simulation engine.

8.2 A Plane and a Sphere! (a basic dynamic-actor and collision-item simulation)

In this example, we import two items, a simple sphere and a plane, from our local host-application into VWD as a collision-item and dynamic-actor, respectively. The collision-item sphere will have the dynamic-actor plane draped over it, like a beach-towel on a basket-ball. This will be a static simulation, and generate a snapshot of the simulated plane which will be sent back to the host-application for saving and/or rendering.

Our approach will be to import each item into VWD using the simple VWD defaults. One item will be a collision-item (the sphere), and in this case, it will just sit there and be 'hard'. The other item will be the dynamic-actor, cloth in this case, and it will be simulated, resulting in it falling onto and draping over the sphere.

Note: please use the previous tutorial as a guide for most of the early setup of your scene. I will be using DAZ Studio in this tutorial, but the concepts and steps are almost identical.

- **Load the sample scene-file:**

- **DAZ Studio:** Look for the simple scene-file located in your installation kit (in your base VWD installation folder's **Working_Files** folder:

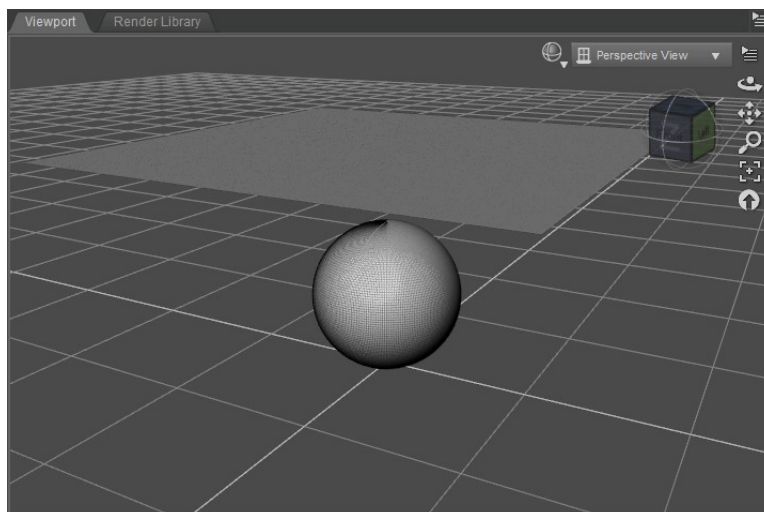
(e.g. C:\VWD\Working_Files\Scenes\DS_PlaneSphere.duf)

- **Poser:** Look for the simple scene-file located in your installation kit (in your base VWD installation folder's **Working_Files** folder:

(e.g. C:\VWD\Working_Files\Scenes\Psr_PlaneSphere.pz3)

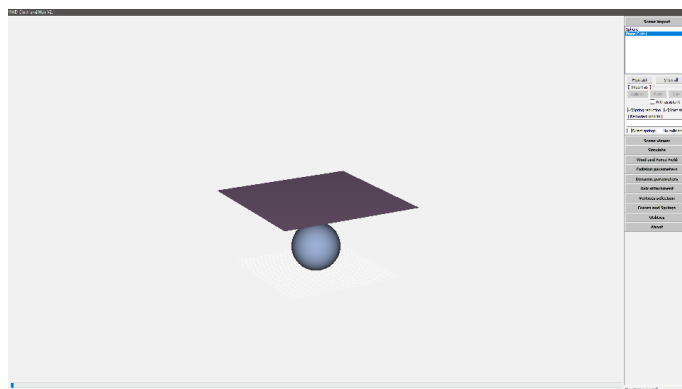
- **Or... Build your scene:** Create a sphere and a plane of usable mesh-density in VWD in your host-application:

- Create or import your sphere (in DS = 1f x 45segments x 90sides), or import from the installation package in Poser (e.g. C:\VWD\Working_Files\OBJ Meshes\Psr_Sphere.obj).
- Create or import your plane (in DS = 3f x 100divisions), or import the plane from the installation package in Poser (e.g. C:\VWD\Working_Files\OBJ Meshes\Psr_Plane.obj).
- Arrange the items near the middle of the scene with the sphere lifted a bit off the ground-plane, and the plane lifted above the sphere by some amount (not much, but not too much)
- In frame 1 of your animation timeline (no, we're not going to animate here, but this is to be sure we don't have things going on in your scene), raise the plane to some amount above (not intersecting) the sphere.

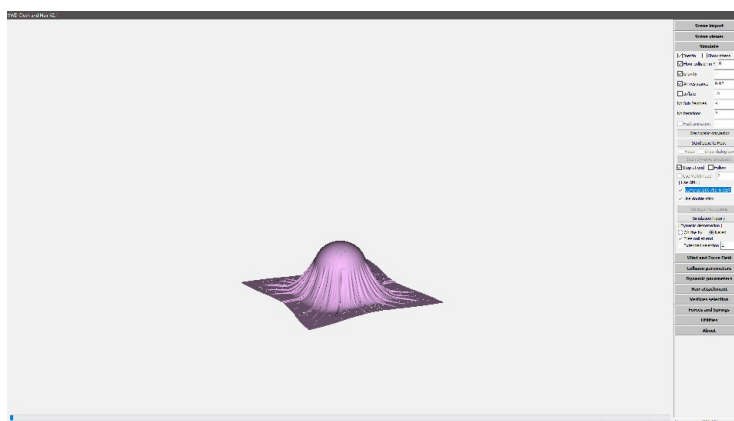


- **Start VWD...**
- **Import the two items, with their new VWD roles:** In the *Scene import* tab, find the Plane and Sphere items:
 - Import the Sphere as a *collision-item without the assistant settings* by selecting it in the listbox, unchecking the *Use asst* checkbox, then clicking the ***Collision*** button, followed by the new, bigger ***Collision*** button. This will import the sphere using the VWD default collision item defaults on the sphere. The scene will now have a 'hard' sphere for the plane (next) to simulate against.
 - Import the Plane as a dynamic-actor without the assistant settings by selecting it in the listbox, unchecking the *Use asst* checkbox, then clicking the ***Cloth*** button, followed by the new, bigger ***Cloth*** button.

Note: Like the previous tutorial, all of the settings in these tabs have uses and should be considered for each real project, but the defaults are usable and don't need to be adjusted to keep this simple lesson on track. See the descriptions of theses settings for details on their usage.
- **Adjust the VWD viewport** so you can see both items in a way that works for you.
 - Click on the ***Scene viewer*** tab to expand that to full-screen, the right-click on the viewport background area and drag your mouse around to adjust the scene view. You can use the mouse wheel (center roller button) to zoom in and out as well. If you get 'lost' while navigating, just double-click on the viewport background to reset the view to a decent centered view of the current *active-item* (the item selected in the top *Scene import* tab's listbox).



- **Run the simulation:** Drape the (dynamic cloth actor) plane onto the (collision-item) sphere:
 - Open the *Simulate* tab
 - Set the simulation to run on CPU by clicking on the *Use Multithread* checkbox (optional)
 - Click on the *Start static simulation* button... (watch the fun...)
 - When settled, press the same button, which is now *Stop dynamic simulation*



- **Send the result back to your host-application:**
 - Press the *Send pose to Host* button
 - Pull up your host-application and review, save, render, etc. the result.
- **Optional – Reset the simulation:** by pressing the **Alt** key while clicking the *Start dynamic simulation* button.
- **Optional – Interactively simulate in 'bursts':** by pressing and releasing the **Shift** key while the mouse cursor is within *Scene viewer* viewport.
- **Optional – Save your import list and their VWD properties** using the *Simulation history* tab in the *Simulate* tab. While there's not much to save here, there will be times when this is very useful.
- **Exit VWD:** Close VWD

Things to try: This process is almost exactly like the previous tutorial, except that the simulated dynamic-actor (plane) is interacting with the newly introduced collision-item. Of note, possible questions you might have at this point, and some things to try:

- You can have many/multiple collision-items in a VWD session, and they can be animated (from your host-application), but you can only have one dynamic-actor (plane, cloth in this case) in that VWD session.
- To layer clothing items, you can run a series of distinct simulation sessions, then import the results each of those previous simulations (**xxxx_VWD** items) as collision-items that the new dynamic-actors (clothes or hair) can then interact with. This is how you would do layered clothes: tshirt on a figure, jacket on the t-shirt on the figure, scarf on the jacket on the t-shirt on the figure. All static or animated.
- You can stiffen or soften the dynamic-actor as you import it, or after you import it, using VWD's **Dynamic parameters** tab and the **Forces and springs** tab.

Note: some settings can only be set during import, but most can be reset or adjusted anytime after the import process.

- You can make certain changes while the simulation is running (gravity, inertia, inflation, wind and force-fields, etc.) by simply clicking the checkboxes and/or adjusting the numeric values as the simulation runs. Try and reset then restart the simulation, and while running:
 - Change **Gravity** to -1 while the simulation is running!
 - Then set **Gravity** back to 1...
 - Turn off **Floor collision**...
 - While the mesh is moving quickly, turn off the **Inertia** checkbox, then back on...
- You can pause/stop a simulation, and adjust the **Forces and springs** tab settings on parts or all of the dynamic-actor, then continue the simulation, pausing to adjust or **Send pose to host**, when you like the state of the dynamic-actor.
- If you wish to play with the **Wind and Force Field** tab settings while the simulation is running, you must first go to the **Forces and Springs** tab, select some or all of the dynamic-actor mesh, and press the **Apply react to wind** button, and/or (select again) **Apply force Field**, then (re)start the simulation, and open the **Wind and Force Field** tab and adjust the settings to taste as the simulation runs – pausing and saving the results as you progress.

9 Additional Resources:

After you've perused this VWD Program Manual to get a sense of the program's available controls and functions, you will probably want to follow some typical usage examples and view some video tutorials. Here's a list of the resources that are available at this time:

- The official program documentation and usage guides are located in the main VWD program's **Documentation** folder. (e.g. **C:\VWD\Documentation**) and version-specific notes and issues are also located in the main VWD program's **_V1.2.333.4444** directory (where the version number is real, e.g. **C:\VWD_V2.1.892.6290**).
- **The VWD website resources page(s):** This should always be the best and most current source of the available VWD resources. If there's information about the status, versions, help-files, tutorials, test-files, or any other VWD-related support assets, this should always be the first place to look.

<https://www.virtualworldldynamics.net/support/>

- **The VWD youtube page(s):** (demos, tests, and tutorials)

<https://www.youtube.com/channel/UCOpGh2GKgryXyMlbcTQ45dg/feed>

- **The VWD Patreon page:** This is where we'll share information, presets, and other VWD-related tidbits, news, and resources.

<https://www.patreon.com/VWD>

- **The VWD Usage Guides:** Usage guides that show real-world examples of the techniques that can be used to produce various effects using VWD's various capabilities. While these should be included in your installation package, it is likely that we might update them (without notice) with corrections and new tricks and techniques. Follow the notes on the website, Patreon site, and/or Renderosity VWD forums to get word of such updates.

(See the VWD website support page up above for the latest versions of our Usage Guides)

- **The Renderosity VWD forums:** Our main product and information discussion users forum. User issues, product help and announcements, etc., are all consolidated here.

https://www.renderosity.com/mod/forumpro/?forum_id=12511

- **Availability:** The available suite of VWD products are currently offered at renderosity.com:

<https://www.renderosity.com/mod/bcs/vendor/VirtualWorldDynamics>

Any other resources that we add or any changes we make can be found via the **support** page on our website (<https://www.virtualworldldynamics.net/support/>).