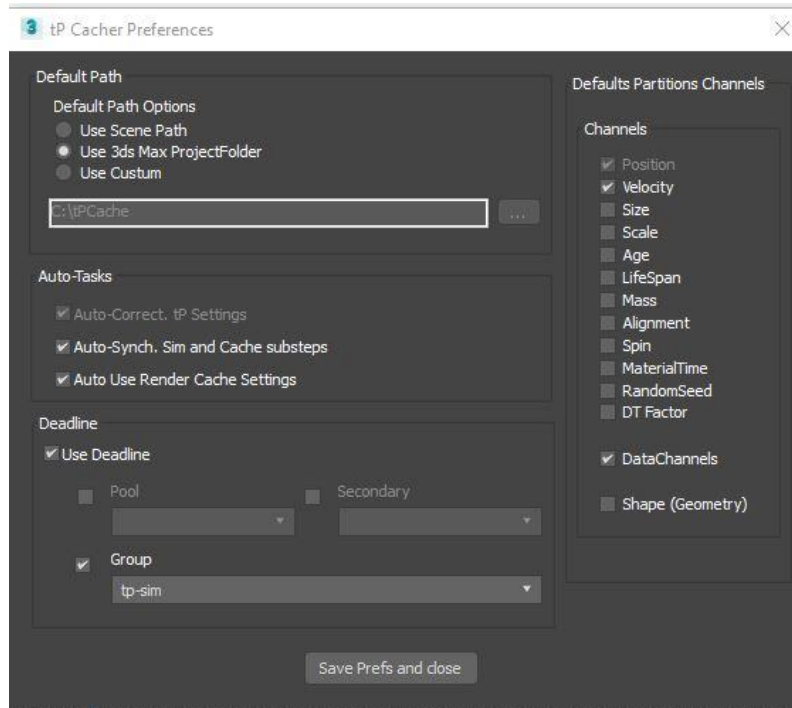


TP CACHER V2.10

The Preferences Dialog.



This is the preferences Dialog, which will store your default settings to apply as a default whenever the main tool is launched. It will appear when you first launch the application, and new options for the preferences will be updated by me, as updates come.

1. Default path : The default path is where the Cache have to be saved.

- ⇒ Scene Path : the tool will create the cache folder structure to store the cache inside the Scene folder.
- ⇒ Use 3ds Max ProjectFolder : uses the default 3ds max project folder structure and adds a 'ParticlesCache' folder to the root of your project folder before storing, and use it as starting point to create the cache folders hierarchically
- ⇒ Use Custom : to use a custom path for hierarchical directories upon creation .

Tips : Even if you are not using Custom as the main path option, you can still set a path and it will be stored as the Root path when you switch to 'custom' inside the main tool.

Autotasks :

- ⇒ Auto-correct tp settings (fixed 'on') : this will correct your substep setting based on scene frame rate, for example, if you are @ 25fps, and have entered 102 samples per second for simulation inside thinkingParticles, upon opening the tool Autocorrect will correct the samples to 100 (a multiple of 25), helps to avoid having wrong settings.
- ⇒ Auto-synch sim and cache substeps : it synchronises the Substeps between simulation and Cache (.tpc) by default, this can be change inside the UI if required.
- ⇒ Auto Use Render Setting : If thinkingParticles have Render Setting option for sim, it will serve to enforce cache to use those settings as default cache method. Settings can be changed on the main tool.

Use Deadline : Activate the deadline module and connect to deadline (deadline launcher has to be running and be connected to a repository) .

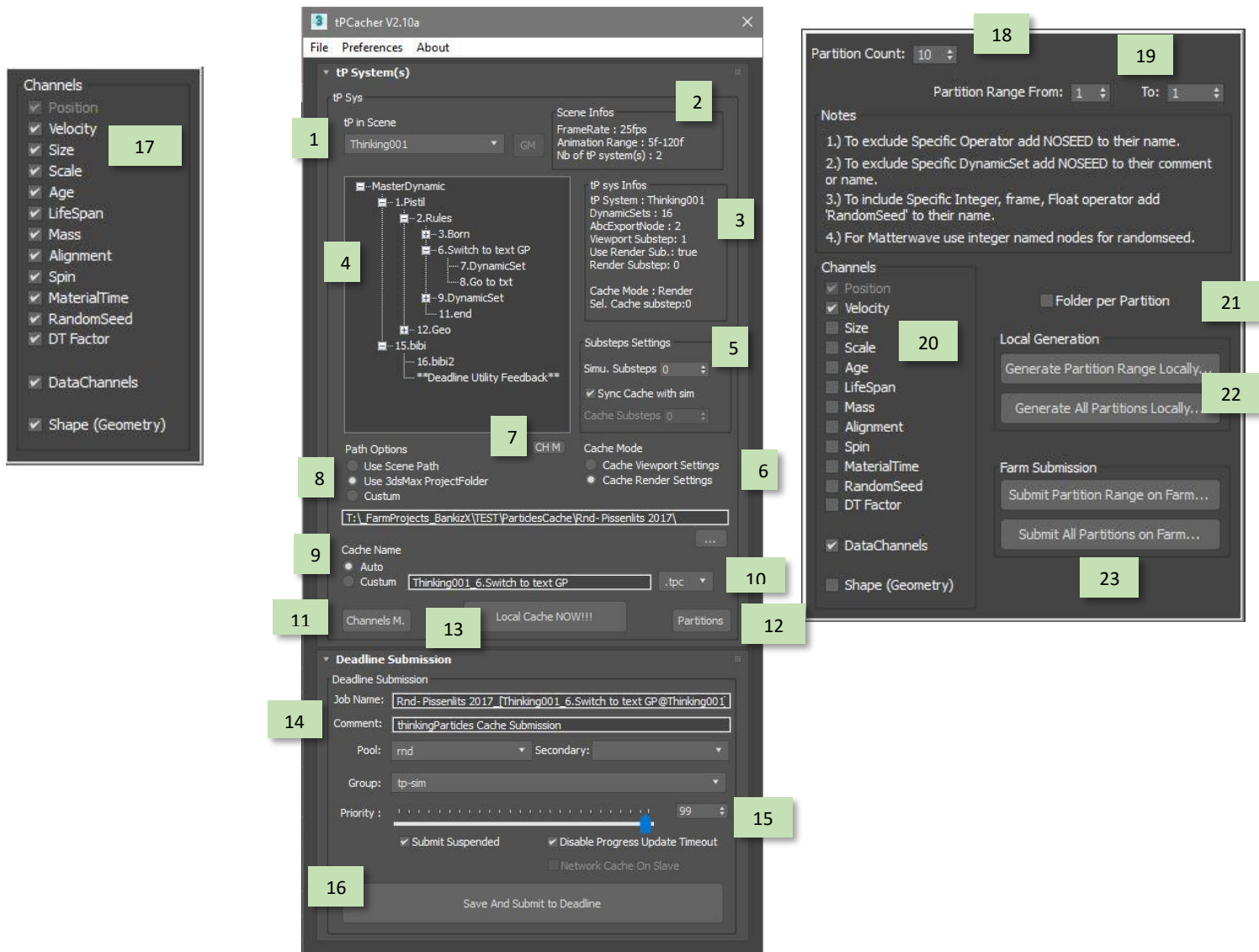
- ⇒ Pool/Secondary/Group Checkbox : allows the setting of a pool/secondary/group as default for deadline submission. Again, can be changed in the module.
- ⇒ Pool/Secondary/Group : auto-populates based on your Repository settings, you can set the default to use, and modifications can be changed later inside the main tool.
- ⇒ All settings that are not set as default inside the preferences dialog the preferences dialog will be auto-populated with the last SMTD ones, on selection they will overwrite the SMTD entries... For those set in the Preference Dialog, settings can be changed in the main tool, however any changes will not overwrite the SMTD settings

Due to this, try to avoid opening this tool with SMTD open, and vice-versa.

Default Partitions Channels : Default Channels to cache during partitioning can be set in the main tool.

Save Prefs and close : Save the new preferences and then open the main tool.

Main UI



Main Dialog : All settings are Auto-populated based on selection. The tool will correct thinkingParticles if you have wrong settings such as Substeps (multiplier of framerate)

- 1 Choose the thinkingParticles system you want to manage.
- 2 : Visual generic Informations about the scene (Frame Rate, Frame Range, Number of thinkingParticles systems in the scene, etc.).
- 3 : thinkingParticles and dynamicTree selection information (Name, Number of DynamicSets, number of ABC export Nodes, Simulation Viewport substeps, Use Render Substeps « ON or OFF », Simulation Render Substeps, Cache method inside the thinkingParticles system (cache viewport setting, or cache render settings)).
- 4 : tP selected Dynamic Tree (based on Treeview). Select the dynamicSet you want and choose your settings or cache.
 - 5 : Substeps Settings
 - ⇒ Simu. Substeps : Set your simulation Substeps setting.
 - The same rule as used for .tpc cache substeps- this means you have to understand « How many substep(s) more to add between frames ».
 - 0 mean => 0 in between = per frame setting
 - 1 mean => 1 more between each frame => Half per frame setting
 - 2 mean => 2 more between each frame => (@25fps = 75 samples per second, @24fps = 72 sample per second)
 - ...Etc.
 - ⇒ Sync Cache with sim : This will synchronized your Cache Substeps setting with your simulation settings (works best for re-timing cache, or when performing layered cache).
 - ⇒ Cache Substeps : Set your Cache file Substeps.

6 : Cache Mode : Load/Set setting namely the substeps settings, and the cache method to use (choose Viewport setting or Render setting) when performing the cache.

7 : Channels Memo Button : Will open the DataChannel Memo dialog.

8 : Path Option => Same as the preference dialog, this is the base path to start the folder hierarchy creation for your cache...

9 : Cache Name : this will set the name of your cache , each time you select a new dynamicSet, an automatic naming option based on selection results. If a Farm module is opened, it will then set the Job name based on cache name automatically.

10 : Choose your extension : .tpc => works best for new cache, and partitions tasks. / .tps => works best when using animated geo instancer, or animated material with geoinstancer.

11 : Channel Manager Button : Open the Channel Manager dialog.

12 : Partitions Button : Open the Partitions dialog.

13 : Local Cache NOW !!! : this will save your scene, then perform the cache locally.

The Deadline Module : Send your cache task to your Deadline Farm => Deadline luncher must be up and running and connected to a repository for it be working...

14 Standard settings for Job submission on Deadline: all settings are auto-populated based on the settings, selection, task performed, SMTD settings of the last job, and the set Preferences.

15 : Disable Progress Update timeout => see deadline SMTD documentation for more information on this. Best checked.

14 : Network Cache on slave / Local cache on slave : Allows you to perform the cache on slave temp directory before copying files to final path, makes for better performance (.tps only, and not working with partitions tasks).

16 : Save and submit to deadline: saves your max file, and performs the routine to send a single cache only to Farm based on deadline management.

The Channel Manager :

17 : Load / Set the Channel: Write settings for your thinkingParticles dynamic set selection.

The Partitions Dialog : This will perform the same simulation again and again with different random seed...and the PDialog will also keep the actual Krakatoa workflow as closest as possible .

The Partitions tasks will work for the MasterDynamic, as for Dynamic Tree selected DynamicSet.

Tips: Since Partitions thinkingParticles features, it allow for the storage of Geometries meshes as well. So you can perform the same cache with different random values by naming a pointed node setting correctly to automate the multiple generation of the different versions.

18 : Partitions Count => Maximum number of partitions wanted (however many times you want to regenerate the cache with different random values). This action will influence the job farm name, as the final cache name.

19 : Partition Range From / To : Set the partition range you want to perform.

20 : Channels Setting : Let you set the Channels you want to store inside the cache file(s).

21 : Folder Per partitions : used to store each partition inside its own folder to keep your Folder project cleaner.

22 : Perform Partitions task locally (range or all partitions)

23 : Perform Partitions Task on Farm (use the farm manager whichever module opened.) - Note: for Deadline only at the moment.

The script comes with a sub-script called « thinkingParticles_Loader », which allows you to create a brand new particle system and load it all Partitioned inside the respective dynamic Sets.

This is not a perfect method, but better than nothing... You will then be free to render your partitions using the given Script « thinkingParticles-Loader » and use the thinkingParticles smoke render node to render the scenes, or generate your own script which fits your needs.

You also need to be aware that the « thinkingParticles loader script » will destroy your old version of « thinkingParticles loader system » (if it already exist).

The Toolkit comes with a BlackBox, which is required to enable the continuous update of the ProgressBar inside of Deadline. To get proper updates of the progress, just put the BlackBox inside of the DynamicSet that you are going to cache.