

SMEDGE

# What's New in Smedge

Smedge 2015 Update 2

© 2004 - 2015 Überware™

# Table of Contents

## INTRODUCTION 3

## UPDATE 2 4

- [Automatic Latest Executable improved functionality](#)
- [Archive Jobs on delete](#)
- [Engine command line shell commands to get active work count and lds](#)
- [Improved Cinema 4D submit script](#)
- [New Parameter commands](#)
- [Job History can open captured output logs when grouping by engine](#)
- [Job Failures for deleted jobs are now removed when reset](#)
- [Bug fixes and performance optimizations](#)

## UPDATE 1 6

- [Pixar RenderMan module automatic path translation](#)
- [Maya now ignores 'UI commands' errors by default](#)
- [No longer force processor affinity on Windows by default](#)
- [3DS Max 2016 output shows one character per line](#)
- [Products that used Choice parameters got display value in the command line](#)
- [Linux no longer spits out crazy gtk "fatal error" messages](#)
- [Some UI elements were not displaying and updating correctly](#)

## SYSTEM CHANGES 7

- [Smedge now works on up to 4 nodes with no license installed](#)
- [Offline Engine Control](#)
- [Engine History](#)
- [Improved Failure system](#)
- [Work limits can be set by work count or by active engine count](#)
- [Priority Boosting](#)
- [Jobs can be assigned to multiple Pools](#)

- [Success Message detection](#)
- [User and Machine are logged when work is aborted](#)
- [Elapsed process time and real time are accumulated by the parent Job](#)
- [Staggering no longer "reserves" engines](#)
- [Mirrors correctly handle the free license](#)
- [Default Master Data folder no longer hidden on Unix](#)
- [Improved scalability](#)
- [Jobs that generate a huge number of errors no longer consume all available memory](#)
- [General fixes](#)

## GRAPHICAL INTERFACE 10

- [New Dark Interface](#)
- [Job History Grouping by Engine](#)
- [Full screen mode on Mac](#)
- [View > Windows > Collect All works better](#)
- [End of work processing was failing for some types of jobs or could be run twice](#)
- [Upgraded to wxWidgets 3](#)

## COMMAND LINE INTERFACE 11

- [Submit now understands the -IncludeEngines and -ExcludeEngines flags](#)
- [Submit command line recognizes 0 and 1 for BoolOverride type parameters](#)

## MODULES 12

- [Houdini Hrender script Prouct](#)
- [Fixed Arnold for Maya flag for skipping the license check](#)
- [Max module was not correctly parsing the output for detecting error strings](#)
- [Thea command line generation fixed](#)
- [Some Maya extra parameters were not being enquoted if they contained spaces](#)
- [Linux distribution was missing Indigo renderer module](#)

*Welcome to Smedge 2015!*

---



The latest version of Smedge provides a whole new level of performance and reliability, and gives you more control over your rendering workflow than ever before. Production tested on networks over 1,000 nodes, scalability is no problem.

---

Smedge 2015 does not communicate with prior versions of Smedge. We strongly recommend that you update your whole network at one time to avoid strange any problems with old versions left running.

Because some of the low-level API elements that make the system operate have been modified, we also recommend that you back up any old data before upgrading. While the data can be upgraded automatically, if you ever want to go back to the old version of Smedge, the upgraded data will not be successfully read by the old versions.

As always, if you have any questions about new features or our future development plans, we encourage you to contact us.

---

*Thanks, and see you in Smedge!*

## Automatic Latest Executable improved functionality

The system for finding the latest executable has been improved. It will now only look for the latest executable if the supplied executable (in the Job, the Engine Production Options or the Master Defaults) cannot be found when work is started. At that point, it will hunt for the latest executable, allowing it to automatically work without configuration. The new system will also detect if a new version is installed, even while Smedge is running, and can automatically update to the latest version. It also fixes an issue introduced in Smedge 2015 that could cause the path to the executable to be unable to be correctly configured if you had multiple versions installed and did not want to use the latest one.

## Archive Jobs on delete

There is a new Master option that can automatically have job data and history archived when a job is deleted. The files can be set to be archived to a specific folder, or left in the Master Data folder. The file extension is `zjob`, but the files are just ZIP archive files. You can expand them with any standard ZIP expander back into the Data folder to recover the data. Further interface to manage these inside of Smedge will be coming.

## Engine command line shell commands to get active work count and Ids

Engine command line shell adds two new commands: `Active` shows the count of active work units, and `Work` will list the work Ids for every all active work on the selected Engine(s).

## Improved Cinema 4D submit script

The Cinema 4D submit Python script now is a little smarter about finding the Smedge executable and gives a basic report of success or failure of actually submitting the job to Smedge.

## New Parameter commands

Smedge can now do conditional substitution of parameters using the new parameter command `?`. The format is:

```
$ (Name.?:ValueIfTrue|ValueIfFalse)
```

`ValueIfTrue` and `ValueIfFalse` can be parameters themselves, undergoing full expansion including parameter commands. If they do not start with a parameter name that can be found in the Job or Options, then the value will be passed through as is. The separator between the “true” and “false” values is the vertical bar character: `|`. The Name parameter will be evaluated based on Smedge string boolean conversion. The values `true`, `yes`, `on`, or a non-zero number will evaluate as “true” while `false`, `no`, `off`, `0`, or empty or non-numeric values will evaluate as “false”.

There are also several new math commands. `i/` (or `/i`) will do integer division (dropping fractional results). `Abs` will return the absolute value of a number. `Ciel` will round a fraction up to the nearest integer. `Floor` will round a fraction down to the nearest integer. `Trunc` will drop the fractional portion of a floating point value.

## **Job History can open captured output logs when grouping by engine**

When you group the Job History by Engine, you can now right click on a specific work unit under the engine and use the View Process Output command to see the output from that work unit.

## **Job Failures for deleted jobs are now removed when reset**

The new failure table system will now remove the failure counts for jobs that have been deleted when the count is reset for that job or for an Engine that had failed on that job.

## **Bug fixes and performance optimizations**

There are several other bug fixes, mostly small issues in the GUI, but including a few possible conditions that could cause the Master to hang in unusual circumstances.

This update is mostly a maintenance update to fix some issues and clean up some loose ends.

## Pixar RenderMan module automatic path translation

The PRMan virtual module file was modified to allow automatic path translations of the contents of the RIB files being passed to the renderer. This allows use of prman cross-platform using the Smedge translation system automatically. It also makes it easier to customize the RIB file sequence formatting using a new parameter available at the Job level.

## Maya now ignores 'UI commands' errors by default

Some scripts generate errors at render time if they try to do something to the Maya UI. This error can usually be safely ignored, so the keyword 'UI commands' has been added to the default list of error ignore keywords.

## No longer force processor affinity on Windows by default

32 bit Smedge requires to force the affinity to all cores in order to make use of more than 32 cores on a machine. 64 bit Smedge no longer has this requirement, so the default is to no longer force the affinity in Smedge. If you have any problem with renders not using all cores, try using the 64 bit version of Smedge, or you can set an environment variable or command line flag. See the Administrator Manual for details.

## 3DS Max 2016 output shows one character per line

Autodesk changed how Max outputs its standard output to force it to always use wide characters. Smedge saw the extra characters as invalid and terminated the line. It now will correctly work for any process that uses narrow or wide character output, and can adjust to changes in the character output on a line by line basis, so you can use scripts that mix and match processes with different output semantics.

## Products that used Choice parameters got display value in the command line

Most commonly reported with mental ray for Maya showing an error with an incorrect value for the verbosity flag. This issue was caused by a bug in the GUI, so command line submitted jobs were not affected. The GUI now uses the correct value, instead of the display value for the parameter.

## Linux no longer spits out crazy gtk “fatal error” messages

These messages were harmless, despite the drastic sounding name, but the underlying causes have been corrected to avoid seeing them at all.

## Some UI elements were not displaying and updating correctly

Linux had issues with the dialog box buttons and all platforms could show issues with the toolbar buttons not always lighting up correctly.

## Smedge now works on up to 4 nodes with no license installed

In the spirit of the original Smedge from the late '90s, Smedge will now run on up to 4 computers with no license installed. If you have only a couple machines you want to hook together, or if you are a student or just learning about animation, or if you want to try it out without dealing with a demo license, just start it up! Gain the power of rendering on up to 4 machines without any worry about licenses, support, expiration dates or anything. Go ahead, take those dollars and spend them on something much more fun than render farm management software, like a nice pair of sunglasses or a trip to Six Flags!

## Offline Engine Control

You can now configure any setting for any Engine in the system, even if that Engine is offline. Any changes you make to offline Engines will be applied automatically the next time the Engine connects. The new Engine system is also more efficient, uses less memory, and the default system works more consistently.

## Engine History

The Master now keeps a history of the events that occur on each engine, which is available both in the GUI, and by the command line Engine shell. You can see every time a change is made to the engine settings, including the user and machine that made the change.

## Improved Failure system

The Master has a new system for keeping track of the failures, making it easier to find issues, and more clear about what is going on when work starts failing. Besides keeping track of both the total and the “active” failures (the count that actually affects distribution and can be reset), the new system also allows you to see which engines have failed on a job and which jobs an engine has failed on. Failures are persisted to disk by the Master and restored when it restarts or if a mirror takes over as the master, making for more consistent operation.

## Work limits can be set by work count or by active engine count

Any place where you could previously set a limit on work, you can now choose if that limit is applied to the number of active workers, or the number of active engines. This means that you can now use products that use a single license for multiple simultaneous workers on a machine with the correct type of limit to actually respect the capabilities of the renderer and your licenses. This change applies to the limit you can set for individual Jobs, as well as the product, pool and user limits you can set on the master.

## Priority Boosting

You can now configure priority boosting for new jobs. This allows you to ensure that a few frames from a new job get rendered, even if there is a long queue of work before that job would normally start. This is very useful for ensuring that every render in a long queue is properly set up before

you leave for the night or weekend, so you can rest assured that the renders will complete as you expect. You can set up a boost for individual jobs, for each type of product, or a system global limit. The job's boost setting will override the product boost setting, which will override the global boost setting. The number you specify is the number of workers that should be boosted ahead of the normal priority. The priority boosting works essentially by adding 100 to the priority value of the job until the specified number of workers have completed.

## **Jobs can be assigned to multiple Pools**

Using the custom pool configuration system, available both in the GUI and by the Submit command line component, you can specify multiple Pools for a Job to use for distribution. This system only works if Pool Prioritization has been disabled in the Master. Using multiple pools this way can make it easier to manage large groups of nodes, for example to assign work to racks of machines for optimizing network or pipeline usage on larger networks.

## **Success Message detection**

You can now specify a single “success string” for Smedge to look for. If you specify this value (either in a Job or for a whole Product), Smedge will only consider a work unit successful if the message is detected in the output. This test is in addition to the other failure tests Smedge performs, so you may want to disable checking the process exit code or other error detection system if you specify a success string.

## **User and Machine are logged when work is aborted**

To make it easier to see track down who messed with your Job, the system now logs the user and machine when work is manually aborted. Now you can know who to blame for stopping your progress.

## **Elapsed process time and real time are accumulated by the parent Job**

Before, these parameters were only available for the work unit that just finished. This means that the values were useful in event commands, but any events related to the parent job always showed 0. Now the Master accumulates the reported time from each work unit into the parent job. These values will now report correctly the amount of time elapsed (either processor usage or clock time) as each work unit finishes. Note that the times are only updated at the end of each successful work unit.

## **Staggering no longer “reserves” engines**

The original work staggering system would reserve an engine, leaving it sitting idle as the work was assigned, until the scheduled stagger time had elapsed. When the global staggering was added to the Master, that system did not reserve engines but simply did not distribute work from jobs that hit their stagger limit, and allowed distribution of work from other jobs to available engines, for more efficient use of the machines. Now, the job staggering works the same as the Master global staggering, so that operation is more consistent and use of your machines is maximally efficient.

## **Mirrors correctly handle the free license**

The mirror system now correctly handles the default free license so that if you switch masters when running without a license installed, you don't have to remove the invalid license to get it working as expected.



## Default Master Data folder no longer hidden on Unix

The default data folder was renamed to “Data” instead of starting with a dot, which means it's no longer hidden by default on Unix based platforms (Mac and Linux). If Smedge finds a “.data” folder on startup, it will try to rename it to the new name, and it will fall back to using the current location if it cannot rename the folder for some reason.

## Improved scalability

Run on networks up to 4,000+ nodes with no problems, just as easily as running on 4 machines.

## Jobs that generate a huge number of errors no longer consume all available memory

When Smedge is set to report about errors that get ignored, scenes that generated a huge number of errors could cause the Master to use up all available memory and possibly crash. Smedge now only reports any given error a single time, so multiple instances of the same error message no longer get repeated in the history. It also limits the total number of ignored errors it reports so that scenes that generate a huge number of ignored errors cannot use up all of the RAM on the Master, since if you are not fixing these errors, there is no point in being pedantic about reporting them all anyway.

## General fixes

There were a few conditions that could lead to memory leaks, or to the Master hanging or crashing. These uncommon problems have now been fixed, making this the most reliable version of Smedge yet.

## New Dark Interface

Smedge has a new dark interface to match the direction that all of the 3D and compositing applications it supports have gone in recent years. This means that you should feel a little more comfortable in your dark and moody workspace. The color scheme has been tweaked to be less glaring and easy to read. And if you want to, you can still choose a lighter color scheme that is more in line with the old Smedge colors, so you don't feel too uncomfortable with the changes.

## Job History Grouping by Engine

Job history can now be grouped by the Engine as well as by the work, making it easier to see how Engines are performing on a job, including average work time statistics, and easier to find problematic Engines having trouble with a job.

## Full screen mode on Mac

Smedge can take advantage of the OS-X full screen API and works seamlessly in full screen mode like any other Mac application.

## View > Windows > Collect All works better

When the interface gets crazy and lost because of a lot of open windows, and you start getting lost, the Collect All command will now raise windows better and will no longer lose the list of open windows. It will also scale up windows that get sized down too small, in case they get shrunk too much to be visible even when raised on the window stack.

## End of work processing was failing for some types of jobs or could be run twice

In some uncommon situations, bugs in the way the Master was processing the end of certain work units could lead to incorrectly determining that failed work was complete, and possible duplicated processing of the finish of a job, which could cause confusion or possible corruptions. These problems have been addressed so end of work processing is always consistent and reliable.

## Upgraded to wxWidgets 3

On all platforms, Smedge now uses version 3 of the wxWidgets GUI library. This means better performance and functionality and bug fixes in the GUI libraries that are out of our direct control.

## **Submit now understands the -IncludeEngines and -ExcludeEngines flags**

You can set the custom pool by command line using the Submit tool in the command line flag mode now, to specify multiple pools or customize the pool by including or excluding machines directly from a job without having to write out a .SJ job file.

## **Submit command line recognizes 0 and 1 for BoolOverride type parameters**

Previously, to specify the value of a BoolOverride type parameter when submitting a Job by command line you had to specify the exact string text as shown in the GUI. Now Submit will recognize 0 for “no” or “false” and 1 for “yes” or “true” regardless of what the exact display string is for that specific parameter, making it easier to assemble a command line to submit a job with a value for this type of parameter.

## **Houdini Hrender script Prouct**

Smedge includes a new virtual module to control Houdini's hrender script, allowing you direct rendering from your Houdini file without using mantra directly.

## **Fixed Arnold for Maya flag for skipping the license check**

## **Max module was not correctly parsing the output for detecting error strings**

## **Thea command line generation fixed**

## **Some Maya extra parameters were not being enquoted if they contained spaces.**

## **Linux distribution was missing Indigo renderer module**