Tractor 2 Features

Tractor-2.0 added significant new features to the core Tractor system established in the 1.0 family of releases. There are a broad range of new additions and improvements, from productive new command line and scripting interfaces for wranglers, to simple user interface changes; from upgrading the entire job database to a new high-performance, high-capability format, to new studio-wide resource sharing and allocation controls.

Here are some highlights:

- **Improved Tractor Product Layout** -- *Single Release Directory, Single Download per platform, Bundled Subsystem Updates* -- The new Tractor 2.0 packaging and installation layout places a matched set engine, blade, spooler, user interaces, and scripting components all in one place, along with only one copy of shared resources including pre-built versions of several third-party subsystems.

- **Tractor Query Tools** -- Introducing tq the *tractor query* command line tool and modules. Based on proven Pixar studio tools, tq is the best way to query live Tractor data from your terminal shell, from your Python scripts, or from a new tab in the Dashboard.

- **Adaptive Farm Allocations** -- A new way to dynamically allocate abstract resources between people or projects using Tractor's flexible *Limits* system. If two films are in production, 60% of the farm can be allocated to one of them, 30% to the other, leaving the remaining 10% for other projects. If one show is idle the others can be allowed to temporarily expand their shares, then they shrink back to the nominal levels when all projects are active.

- **Dispatching Tiers** -- A simple way to organize broad sets of jobs into a descending set of site-defined priority groups. The default tiers are named: **admin, rush, default, batch.** Create your own!

- **Custom Menu Actions** -- Add site-defined Dashboard menu items that can invoke your centralized scripts, parameterized by the user's current list selection.

- **Job Authoring API** -- A new tractor.api.author module allows your Python scripts to easily create Job, Task, and Command objects linked together according to your dependency requirements. The Job.spool() method then sends the resulting job to the tractor-engine job queue.

- **Simple Engine Discovery** -- A simple new "zero-config" announcement capability for small studios allows tractor-blades and other Tractor tools to find tractor-engine on the local network without requiring manual nameserver (DNS/LDAP) configuration changes. Tractor-engine will automatically disable this SSDP-style traffic at studios where the hostname alias "tractor-engine" has already been created by an administrator in the site nameserver database.

- **Checkpoint-Resume Support** -- New extensions to job scripting, dispatching, and the Dashboard add interesting new capabilities related to incremental computation. Tractor also supports a general "retry from last checkpoint" scheme. Both features integrate with the new RenderMan 19 RIS checkpoint and incremental rendering features.

- **Blade Auto-Update** -- A new tractor-blade patch management system allows administrators to "pin" the farm to a particular blade patch version, and automatically push out a new version to the entire farm. Out of date blades restart themselves using the new module version.

- **Pluggable Authentication Module (PAM) support** -- The engine's optional new built-in PAM support delegates password validation directly to the operating system on the engine host. This alternative makes it simple to enable password support at studios where the LAN already provides adequate credential transport security.

- **Privilege Insulation** -- A new *EngineOwner* setting in tractor.config specifies the login identity under which tractor-engine should operate. This setting is important because it allows the engine to drop root privileges immediately after it has acquired any protected network ports that it may need. The engine's normal day-after-day operations will then occur under the restrictions associated with the specified login name.

- **Dynamic Service Key Advertisement** -- Several new blade profile **"Provides"** modes have been added to support some advanced usage cases. For example, blades can dynamically advertise a different set of service key capabilities depending on which keys have already been "consumed" by previously launched commands.

- **Resource Usage Tracking** -- The CPU, RSS, and VSZ *rusage* process metrics are now recorded into the job database for each launched command. Currently supported on Linux and OSX tractor-blades.

- **Command Retry History** -- A unique tracking record is now created in the job database for every command launch attempt. So the history of retries on a given task can be reviewed using the tq tool, for example.

- **Configuration File Loading** -- A streamlined override system can help to reduce clutter and improve clarity about which files have been modified from their original "factory settings" at your studio.

- **Task Concurrency Throttle** -- Each job has a new "maxActive" attribute that constrains the maximum number of tasks allowed to run concurrently from that job. This optional control over the "footprint" size of the job on the farm may provide useful temporary wrangling control over a few jobs in cases where changing site-wide Limits settings might not be appropriate.

- **Automatic Blade Error Throttle** -- A new blade profile setting that can prevent blades from picking up new work if they encounter too many errors within a given time interval.

- **Job Spooling Improvements** -- Job processing upgrades include faster processing, better error checking, and bundling of required subsystems. A parallelized job intake and database staging scheme can dramatically reduce backlogs when many jobs are spooled simultaneously, or when many "expand" tasks are running in parallel. A self-contained Tcl interpreter bundled with the spooler simplifies site install requirements and can perform client-side error checking prior to job delivery to the engine. A new JSON job spooing format will also supported (but is currently unavailable pending changes).