



Overview

Deadline is a hassle-free administration and compute management toolkit for Windows, Linux, and macOS based render farms. It supports over 80 different content creation applications out of the box. It offers a world of flexibility and a wide-range of management options for render farms and compute clusters of all sizes, and allows users the freedom to easily access any combination of on-premise or cloud-based resources for their rendering and processing needs.

Deadline 10 is the latest update of our easy to use compute management solution. This update adds native integration to the AWS Cloud to enable customers to expand their render farms - whether it exists on-premise, purely on the cloud, or a hybrid of both - simply and securely.

To ensure that all the appropriate files are available in the cloud, Deadline 10 synchronizes with local asset servers and manages the data transfer before rendering begins, tagging accounts and instances for cost tracking. With flexible licensing options, customers can purchase software licenses from the Thinkbox Marketplace, deploy existing licenses, or leverage a combination of the two to grow render farms elastically from the AWS Cloud.

The Deadline Slaves can now automatically switch between traditional Floating Licensing and Usage Based licensing. In addition, there is now an option for unlimited 3rd party Usage Based Licensing when configuring a 3rd party Limit.

The new Deadline Remote Connection Server application is a drop-in replacement for the Deadline Proxy Server application and offers better performance and scalability.

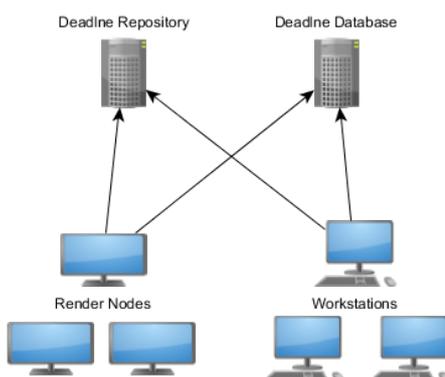
All settings in the Deadline configuration file can now be overridden using environment variables for added flexibility.

To enhance security, a new Execute Command whitelist option has been added to Remote Administration, allowing administrators to pick from a list of pre-defined commands to execute remotely, while removing the ability to type in arbitrary commands. This new feature is optional, and enabled by default.

Components

The Deadline Render Farm Management System is built up of 3 components:

- A single Deadline Database
- A single Deadline Repository
- One or more Deadline Clients



The Database and Repository together act as a global system where all of Deadline's data is stored. The Clients (workstations and render nodes) then connect to this system to submit, render, and monitor jobs. It is important to

note that while the Database and Repository work together, they are still separate components, and therefore can be installed on separate machines if desired.

Database

The Database is the global database component of the Deadline Render Farm Management System. It stores the jobs, settings, and Slave configurations. The Clients access the Database via a direct socket connection over the network. It only needs to be installed on one machine (preferably a server), and does not require a license.

Repository

The Repository is the global file system component of the Deadline Render Farm Management System. It stores the plugins, scripts, logs, and any auxiliary files (like scene files) that are submitted with the jobs. The Clients access the Repository via a shared network path. It only needs to be installed on one machine (preferably a server), and does not require a license.

Client

The Client should be installed on your render nodes, workstations, and any other machines you wish to participate in submitting, rendering, or monitoring jobs. The Client consists of the following applications:

- Launcher: Acts as a launch point for the Deadline applications on workstations, and facilitates remote communication on render nodes.
- Monitor: An all-in-one application that artists can use to monitor their jobs and administrators can use to monitor the farm.
- Slave: Controls the rendering applications on the render nodes.
- Command: A command line tool that can submit jobs to the farm and query for information about the farm.
- Pulse: An optional mini server application that performs maintenance operations on the farm, and manages more advanced features like Auto Configuration, Power Management, Slave Throttling, Statistics Gathering, and the Web Service. If you choose to run Pulse, it only needs to be running on one machine.
- Balancer: An optional Cloud-controller application that can create and terminate Cloud instances based on things like available jobs and budget settings.
- Remote Connection Server: An optional application that handles HTTP or HTTPS connections to a remote repository for use by Deadline applications.
- License Forwarder: An optional application that acts as a license server for third party applications when using Third Party Usage Based Licensing.
- Web Service: An optional command line application that allows you to get query information from Deadline over an Internet connection such as via Deadline Mobile.

Note that the Slaves and the Balancer applications are the only Client applications that require a license.

Jobs

A Deadline job typically represents one of the following:

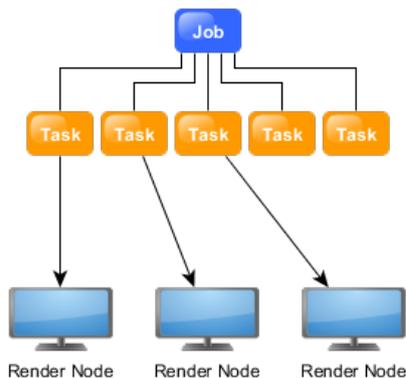
- The rendering of an animation sequence from a 3D scene.
- The rendering of a frame sequence from a composition. It could represent a single write node, or multiple write nodes with the same frame range.
- The generation of a Quicktime movie from an existing image sequence.
- A simulation.

These are just some common cases. Since a job simply represents some form of processing, a plug-in can be created for Deadline to do almost anything you can think of.

Job Breakdown

A job can be broken down into one or more tasks, where each task is an individual unit that can be rendered by the Slave application. Each task can then consist of a single frame or a sequence of frames. Here are some examples:

- When rendering an animation with 3ds Max where each frame can take hours to render, each frame can be rendered as a separate task.
- When rendering a compositing job with After Effects where each frame can take seconds to render, each task could consist of 20 frames.
- When rendering a Quicktime job to create a movie from an existing sequence of images, the job would consist of a single task, and that task would consist of the entire image sequence.



Job Scheduling

Use numeric job priorities, machine groups and pools, and job-specific machine lists to explicitly control distribution of rendering resources among multiple departments. Limits allow you to handle both limited license plug-ins and render packages, while job dependencies and scheduling allow you to control when your jobs will begin rendering.

The Slave applications are fully responsible for figuring out which job they should render next, and they do this by connecting directly to the Database. In other words, there is no central server application that controls which jobs the Slaves are working on. The benefit to this is that as long as your Database and Repository are online, Deadline will be fully operational.

