

Smedge 2.37.3 Manual

Installation and Setup

- How to set up your network for Smedge
- Installing Smedge
- Configuring Smedge
 - Specifying a dedicated server
 - Advanced port configuration
 - Behind a firewall
- Uninstalling Smedge

Using Smedge

- **The Way of Smedge**
 - The Smedge Interface
 - Clients and Servers
 - Jobs
 - Packets
- **The Smedge Client**
 - Client Settings
 - Smedge Options
 - Configuring Smedge across your network
 - Other Smedge configurations
- **The Smedge Job**
 - Submitting a Job
 - Extra Parameters
 - Loading/Saving a job and Repeats
 - Command Line Interface
 - Drag and Drop
 - Submitting a bunch of jobs with the same settings at one time.
 - The Job Lifespan
 - Viewing the a job's history
 - The Check Frames Window
 - Changing job priority
 - Changing job pool
 - Stopping a job
 - Saved Job Information
 - After a Packet
 - After a Job
 - Other Cool Stuff
 - Job IDs
 - Job Files
- **The Smedge Packet**

- The Packet Lifespan
- Stopping Packets
- Viewing the Rendering Frame
- Viewing the Output
- Packet Options.
- **Pools**
- **Smedge Globals**
 - Overview
 - Maximum Failure Counts
 - Defaults
 - Maximum Product Usage
 - Factory Defaults
- **Instant Messages**

Products Available To Render

- Generic Scripts
- Maya (6.0+)
- MentalRay for Maya (6.0+)
- Maya Vector Renderer (6.0+)
- Maya Hardware Renderer (6.0+)
- Maya (5.0 or earlier)
- MentalRay for Maya (5.0 or earlier)
- Maya Vector Renderer (5.0 or earlier)
- Lightwave
- 3ds max (6.0+)
- 3ds max (via SFRender)
- XSI
- SoftImage
- MentalRay
- Houdini
- Jig
- Alias Renderer
- Alias Raytracer
- Alias Powercaster
- Alias Powertracer
- After Effects
- Shake
- Digital Fusion
- Chalice
- Rayz
- Air
- MayaMan
- imgcvt

Interface Reference

- Queue Menu

- Host Menu
- View Menu
- Help Menu
- The System Tray Icon
- The Commandline Interface

Secrets and Security

- Feature Locking
- Client Locking
- Forced Job Settings
- License Broadcasting
- Startup Sound
- Check for Updates
- Confirm Exit
- Automatic Hyperthread checking

Installation and Setup

- [How to set up your network for Smedge](#)
 - [Installing Smedge](#)
 - [Configuring Smedge](#)
 - [Specifying a dedicated server](#)
 - [Advanced port configuration](#)
 - [Behind a firewall](#)
 - [Uninstalling Smedge](#)
-

How to set up your network for Smedge

Welcome to the world of Smedge.

Before installing Smedge, there are some considerations about your network setup that you should be aware of. At its heart, Smedge is nothing more than a shell program for launching command line renders. In order for it to work, you need to actually be able to render from any machine. Smedge makes one basic assumption:

The files being used for a job are in the same place on every render client.

Implicit in this assumption is the idea that every machine has access to the same files. If MyScene.mb on machine A is different than MyScene.mb on machine B, then the two machines are going to render different images.

One way to set the clients to be the same is to use Windows NT drive letters. If you have a file server, make sure that every machine is mounting the shared volume at the same drive letter. If you are using Windows peer-to-peer networking, you can change the drive letter of your shared disk and mount that shared disk at the same drive letter on the other machines.

Smedge has no problem with UNC pathnames (e.g. \\MyServer\SharedDisk\Project\scenes\scene.mb). However, Windows does not support UNC paths as a startup directory. For products that allow you to set the startup directory, if you provide a UNC path, Windows will ignore it and use the Windows directory by default.

Pre-Smedge checklist

The following steps provide a simple way of setting up your network to allow Smedge to render. For this example, we concentrate on Maya, but the same theories apply for any product you want to render through Smedge. *Please Note:* These are generic instructions. Your network setup may be different. Contact your network administrator for more information.

1. **Install Maya on your Render Clients** Follow the instructions that come with Maya to

- install and configure Maya to render on each computer you want to use as a render client
2. **Set-up your shared project space**
Make sure that every client is set up with the same access points to your scenes. For example, if you use a file server, make sure that every client mounts that file server at the same drive letter.
3. **Install Smedge on your Render Clients**
See the [next section](#) for more information.
4. **Configure Smedge to render**
Make sure that every render client has Maya enabled in its [Client Settings](#) dialog box. Also make sure that every client is enabled and has checked out a license.
5. **Test it out**
Try submitting a render. If it doesn't work, take a look at the [FAQ](#) for more help, or try contacting technical support at support@uberware.net.

Installing Smedge

Smedge comes with an installer program. However, Smedge can also install itself just by running the Smedge.exe program on any computer. There are five files that are necessary to run Smedge:

Smedge.exe	The executable itself
Watcher.exe	A supporting application for crash protection
Platform.dll	A library of Win32 network functions
Smedge.hlp	The help file
Smedge.cnt	The contents file for the help system

The other files that come in the Smedge distribution are files that provide information about the product, or examples of using the product. You do not need these files to run Smedge.

After you have run Smedge you will see a file with the name of your computer in the same directory as the Smedge.exe file. These are the debugging log files. The name of the file tells you which computer generated this log information. You can delete these files at any time. If you are having a problem, though, it may be useful to send the log file to technical support. It provides information about what is happening to Smedge and what it is trying to do.

An easy way to configure Smedge is to install it onto a shared directory on your file server. Then just make a shortcut to the executable on each computer you want to run Smedge. It is often a good idea to put a shortcut to Smedge in the Startup folder of the Start menu for "All Users." This way, whenever someone logs into the computer, Smedge will start automatically. And to update Smedge, you only have to update it in one place.

Configuring Smedge

Smedge stores its configuration in the Windows NT registry. Some settings are user specific, such as the window position and the sound Smedge plays when a render completes, and others are machine specific, such as the number of processors available. You need to make sure that the user running Smedge has permission to modify registry settings under the keys:

HKEY_CURRENT_USER\Software and
HKEY_LOCAL_MACHINE\Software

By default, Smedge will configure a client in the following manner:

- Smedge will assume that all the renderer executables are in the user's PATH environment variable
- Smedge will try to check out a license to render on the client
- Smedge will be enabled for Maya and Generic scripts
- There will be no time restrictions
- All processors on the machine will be made available to jobs
- Smedge will try to save output in the local TEMP folder, and will limit that output to 5 MB

All of these options, and more, are configurable. You can even configure multiple machines at the same time, and change the configuration of any machine running Smedge on your entire network. For more information, see the chapter on [The Smedge Client](#).

Specifying a dedicated server

Smedge #31 and later allow you to specify a dedicated server machine. Any client that you tell about the server will bypass the UDP server location system. Obviously, if the client sees that it is supposed to be the dedicated server, it will start up the server process automatically.

There are two ways to configure the dedicated server, with a **server.txt** file or with the command line.

You can create a text file in the same folder as the `smedge.exe` program file called **server.txt**. This file should have one line, and that one line should contain the name or the IP address of the computer you want to act as the server, and optionally the TCP port to use for communication. The format of the line is:

```
server:port
```

`server` can be a computer's name (for example, EDGAR), or its IP address (192.168.0.3). `port` should be an unsigned integer less than 65536. You shouldn't use a port that is being used for any other service on your computer, as conflicts will result and Smedge may not work correctly. If you leave the port off, Smedge will use the default port **4566**.

If you use the command line switch **-server** it will override the **server.txt** file, if it exists. The syntax for the command line is:

```
Smedge -server server:port
```

the `server` and `port` parameters work the same as in the file. The port is optional.

If you dedicate a server machine, that server will still listen on the UDP port as well, and will respond correctly to server requests from the original floating server location system. This means that if you do not provide either the file or the commandline switch to a client, it will still find the dedicated server. However, this means that you still may need to be aware of the [Advanced port configuration](#) settings for both the dedicated server and the client that doesn't know about the dedicated server.

Note that you should use the same `server.txt` file for every client on your network that you want to connect to that server. If you do not, other clients could start up their own server, and you could end up with multiple servers on your network. One easy way to do this is to setup Smedge to run from a common shared folder, as described [above](#).

Advanced port configuration

Smedge #30 and later have the ability to configure the port used for server location. Clients use this port to locate the server, through which all other communication is routed. If you change this port, you need to make sure that every machine you want to connect uses the same port number. If two clients are started with different server ports, those two machines will never know about each other, and will work independently.

Smedge uses the following method to determine the port to use to locate a server:

1. If you provide the command line switch `-port port_number`, Smedge will use that port to look for a server.
2. Next, Smedge will look at the registry key
`HKEY_CURRENT_USER\Software\Überware\Smedge\Options\Port`, which should be a **DWORD** value.
3. Next, Smedge will look at the registry key
`HKEY_LOCAL_MACHINE\Software\Überware\Smedge\Host\Port`, which should also be a **DWORD** value.
4. Finally, if none of the above options are set, Smedge will use the default port, **4566**

Note: Smedge #29 and earlier *cannot* change this port. The port they use will always be 4566.

Behind a firewall

There is only one configuration that should have to be changed to use Smedge in a protected network. Smedge uses a UDP port to locate the server. You need to make sure that your firewall allows UPD network broadcasts on the chosen port in order for Smedge to find the network. All other communication is sent through TCP ports that are determined at run-time and broadcast through that UDP port.

Alternatively, you can now dedicate a machine to be the server machine, and no UDP messages will be used to locate the server. See [Specifying a dedicated server](#) for more information.

See [Advanced port configuration](#) for more information on how Smedge chooses which port to use. Smedge #29 and earlier *cannot* change the port, and always use the default port **4566**.

Uninstalling Smedge

If you use the installer to install Smedge, it provides an option for removing the program.

If you didn't use the installer, you can just delete the Smedge files.

You can also delete the "Smedge" keys in the registry in the following places:

HKEY_CURRENT_USER\Software\Überware

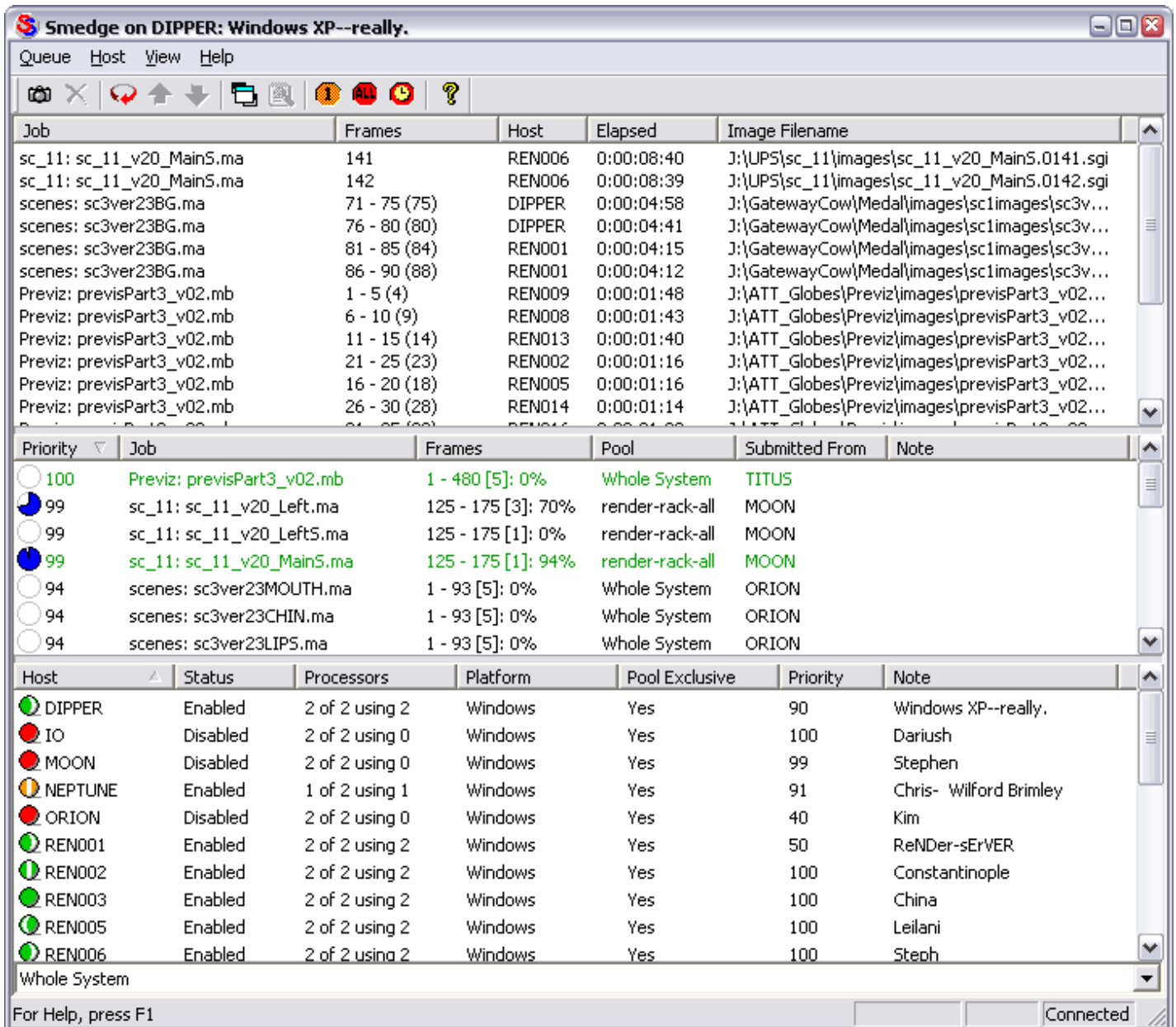
and

HKEY_LOCAL_MACHINE\Software\Überware

The Way of Smedge

- [The Smedge Interface](#)
- [Clients and Servers](#)
- [Jobs](#)
- [Packets](#)

The Smedge Interface



The Smedge window in action.

The main window is divided into three areas. In the top area, you can choose to look at either a list of all the packets currently rendering or a list of all the clients that are

connected together. The middle section shows a list of jobs that have active packets rendering. The bottom section shows the queue of jobs. You can sort the lists by any of the columns in the lists.

Every aspect of Smedge has been designed to be easy to use. Most elements are self explanatory, and almost every dialog box has context sensitive help. Click on the little question mark icon in the corner, then click on the item you want information about.

Smedge is really designed to be left running all the time. Design decisions made early in the development of Smedge caused the program to be written as a single easy to use application rather than a Windows NT based "Service." These considerations were both ease of setup and use by the end user and limitations in the multi-user design of the Windows operating system. While Smedge will continue to work if you open and close it all the time, it will be more efficient and maintain and display better information if you leave it running.

Clients and Servers

Smedge is a "Peer to Peer" networking application. This means that there is no central Smedge Server that must be running at all times. (Note that you can [specify a dedicated server](#) in Smedge #31 and later.) However, the notions of "Client" and "Server" are used in a specific manner inside of Smedge.

A **Client** is any computer that is running Smedge. The Client portion of the program does the real work of Smedge: the rendering of [packets](#).

A **Server** is one specific instance of Smedge that acts as a dispatcher for all of the Smedges on your network. This is how Smedge can make sure that two computers don't try to work on the same [packet](#) at the same time.

Any Smedge client can act as the server. You can see the IP address of the server by looking in the about box of any Smedge that is connected. You can tell if Smedge is connected by looking in the bottom corner of the window for the "Connected" text. Should the server machine shut down (if the user closes Smedge on that machine, shuts that machine down, or that machine crashes), one of the other Smedge clients on the network will take over as server. Because every client maintains a list of all the [job](#) information, there should be no loss of data when this happens. It is normal for Smedge to take a few seconds to connect to the new server and resume normal operation.

With Smedge #31 and later, you can dedicate a specific machine to act as server. This process is a little more complex to setup, but can provide more stability. For more information see [Specifying a dedicated server](#).

For more information see [The Smedge Client](#).

Jobs

The central piece of information that Smedge deals with is a **Job**. The job contains the information that tells the [client](#) what to do. It contains:

- What [product](#) to use
- A job name (*for informational purposes only*)
- The scene file to render
- A project for that scene (*in some cases*)
- The range of frames to render
- The size of the [packets](#) to use
- Any extra parameters to pass to the renderer
- A priority
- and A Pool

The specifics of jobs can be dependent on the product. For more information see [The Smedge Job](#).

The last two items listed are used by smedge to know how to distribute a job to the [clients](#). The priority of a job determines what order jobs are rendered in. Higher priority jobs are rendered before lower priority jobs. If a job has a priority of zero, [packets](#) for that job are not sent out to be rendered. This is like a pause option.

Groups of [clients](#) can be assembled into [Pools](#). The job is assigned to a pool, and will have priority on machines in that pool over jobs not assigned to the pool. You can decide if a client is allowed to render jobs in pools that the client does not belong to (see [The Smedge Client](#) for more information).

Note that pools override priorities. A machine will render an "in-pool" job before an "out-of-pool" job regardless of the priorities of the two jobs.

If the pool and priority of jobs are the same, they will be rendered on a first come, first served basis.

Packets

A **Packet** is a portion of a scene that will actually be rendered by a [client](#). Packets are determined by the **Packet Size** parameter of the [job](#), and are in units of frames.

For example, if you have 50 frames to render and set the packet size to 5, the scene will be broken into work units of 5 frame each. The first packet will contain frames 1-5, The second packet will contain frames 6-10, and so on. Each client asks for a packet to render. The [server](#) will start with the first packet. As the client finishes a packet, the status of that packet is updated, and the next packet will be dispatched. Packets are dispatched in this manner until every packet has been sent.

The packet method of distribution has advantages and disadvantages. Because the granularity of packets is smaller than if you just divided up the total frames by the total number of clients, you can more efficiently use your resources if you have scenes that can vary in time from frame to frame. However, there is overhead incurred in starting the rendering application for each packet, so it is not always wise to simply use single frame packets.

A compromise is set your packet size for your particular needs. If you have a job that renders really fast, use a larger number. If you have a really slow render, use smaller packet sizes, even single frame packets. You will have to experiment to determine the optimum packet size for a different jobs based on your rendering clients and network.

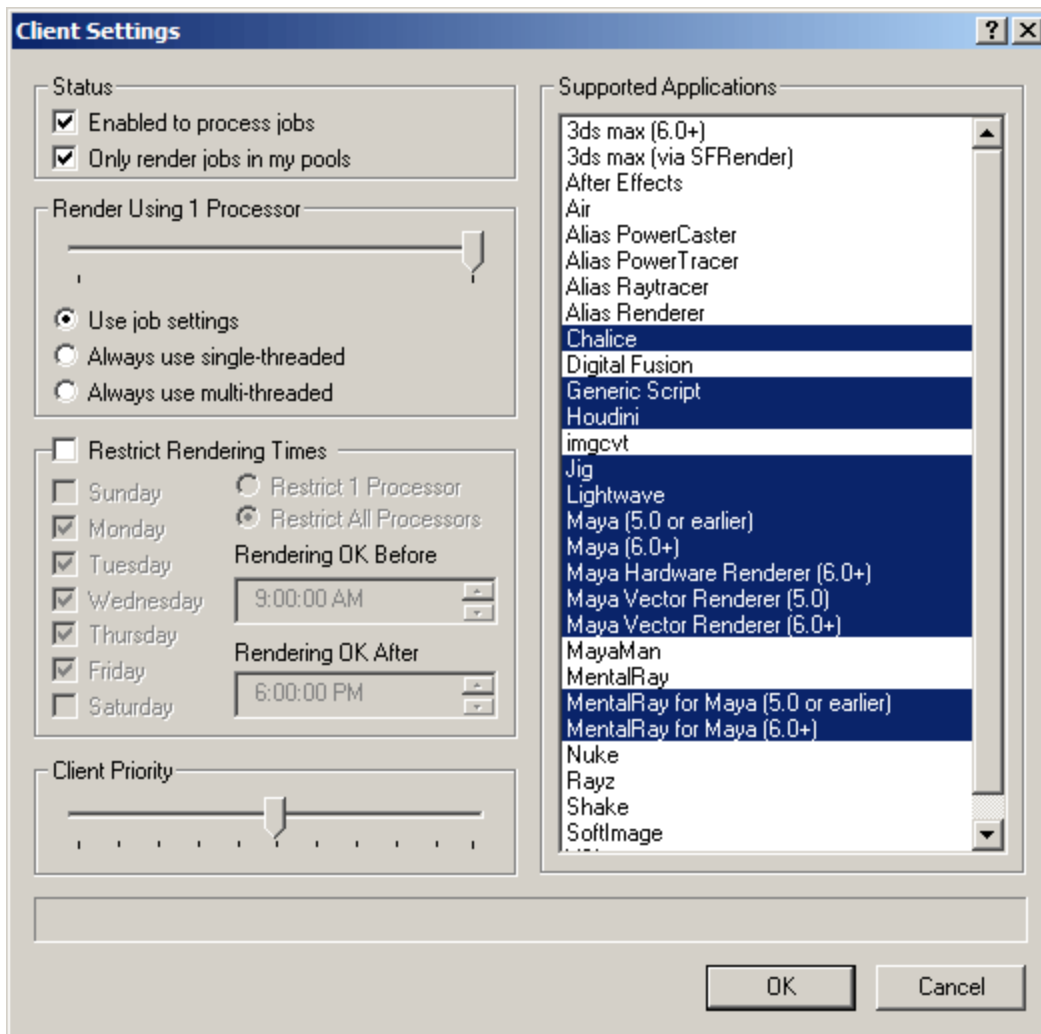
For more information see [The Smedge Packet](#).

The Smedge Client

- [Client Settings](#)
- [Smedge Options](#)
- [Configuring Smedge across your network](#)
- [Other Smedge configurations](#)

Client Settings

This dialog allows you to configure how Smedge will act as a render client.



The slider in the top left can limit the number of processors available for Smedge to use on the client. The radio buttons under that slider tell smedge how to use those processors for jobs.

- **Use Job Settings** means to distribute processors according to the settings in the [job](#).

- **Always use single-threaded** tells Smedge that this client should always launch a single threaded render process, and to launch as many of these processes as necessary to utilize available processors.
- **Always use multi-threaded** tells Smedge to launch a single render process that uses the available processors.

Some renderers cannot be launched multi-threaded and these settings will not apply to those render processes.

In the top right is a list of the render types that Smedge currently supports. By selecting items in this list, you tell Smedge which renderers this particular client can support. In the example above, the client supports the Generic Script and Maya renderers only.

Restrict Render Times tells Smedge to not allow this client to perform renders during specific times. You can use the day check boxes to pick the days that are restricted.

- The **Rendering OK Before** and **Rendering OK After** controls tell Smedge what times during the restricted days to allow rendering. If a day is not restricted, these times will be ignored and rendering will be allowed.
- The radio buttons tell Smedge how restrictive to be: **Restrict 1 Processor** will tell Smedge to always make sure at least one processor is unused during the restricted times; **Restrict All Processors** will tell Smedge to make sure all processors are unused during restricted times.

The check boxes in the Status section are global controls for the render client.

- **Enabled to process jobs** is a master switch to turn on or off the client. If unchecked, Smedge will not render on this client at any time.
- **Only render jobs in my pools** tells Smedge if this client is allowed to render "out of pool" jobs: i.e., jobs that are assigned to a pool in which this client is not a member.

The **Client Priority** is another parameter that Smedge uses to prioritize the distribution of work. Higher priority clients will be assigned work first, assuming all other things are equal, and that more than a single client idle. For the purposes of this sort of prioritization, Smedge uses a granularity of about 1 second. You can use this control to try to make sure that your faster machines will pick up work first if your system is completely idle, or to make sure that you only uses workstations to render if all the render boxes are already in use.

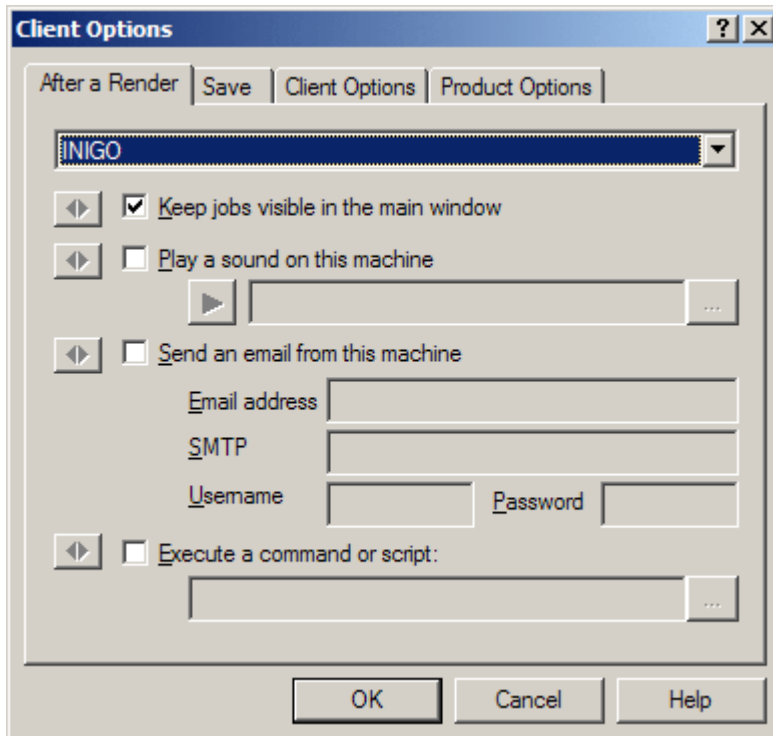
The space at the bottom will display a note that is attached to the client, if there is one.

Smedge Options

This dialog allows you to configure how Smedge will interact with you on the desktop. The default settings are minimally intrusive and work well for render-only clients, such as machines in a render farm. For a workstation, you may want to set things up in your own way.

You can configure the options for any client from any other client. You can also configure more than one client at a time. Along the top of each page of the settings dialog box is a drop-down list of all the clients you are currently configuring. By selecting a different client from the list, you can configure the settings for that particular client, without affecting the settings for any other client.

Next to each option is a button with two red triangles on it. Pressing this button will copy the current client's value for the option to every other client you are currently editing. This provides an easy way to change a single setting on a large number of clients without affecting other settings that have already been specifically configured.



The first page of the options dialog box allows you to configure how this client behaves after a render is finished successfully.

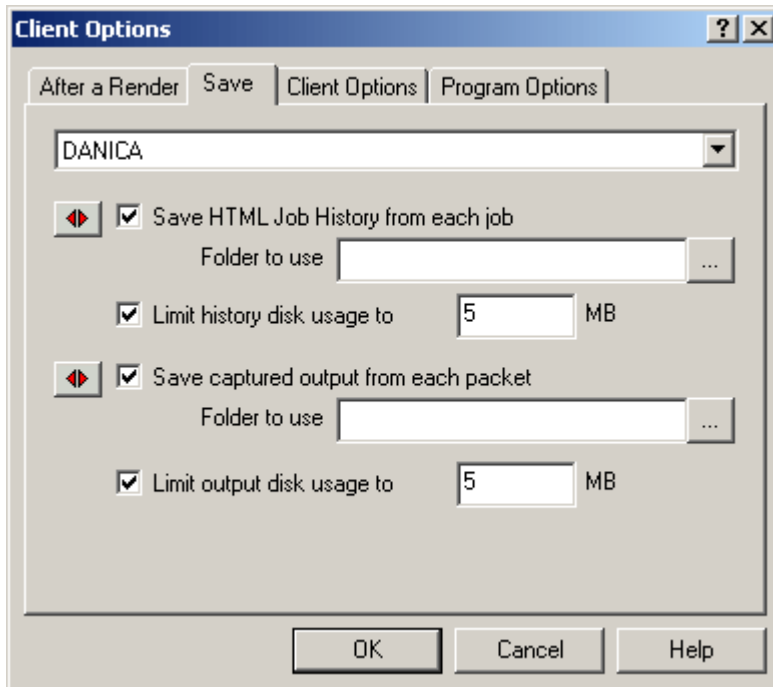
If you check **Keep jobs visible in the main window**, jobs will be left in the job list on this client when they finish. Completed jobs appear in a different color, and will not be resubmitted after a recovered crash. All of the job history is also kept, again, on this client only. To get rid of a completed job, select it and use the [Delete Job](#) command in the [Queue](#) menu.

If you check **Play a sound on this machine**, the computer will attempt to play a WAV file when jobs successfully complete. You can press the blue Play button to hear a preview of the sound file. If you leave the sound file field blank, Smedge will play a default sound (A gong). You can select any WAV file you want to play. Please note, if you do not have a sound card installed, you will not hear anything.

If you check **Send an email from this machine**, the computer will try to send the HTML job history file to the specified email address. You must fill in the **Email address** and

SMTP Server fields with your email information. Note that the "From" and "To" fields of the email will both be the email address you supply. Some SMTP servers require authorization to send emails. If you need to provide a username and password, enter them in the fields provided. Note, however, that some SMTP servers do not use this kind of authorization to send emails, in which case you should leave both of the fields blank. Because Smedge is not designed as an email client, it does not support every possible configuration of SMTP server authorization. Please consult with your system administrator if you need to discuss SMTP server setups in your system.

If you check **Execute a command or script**, the computer will try to execute the command or script file that you put into the text field. You can browse for batch files or scripts.

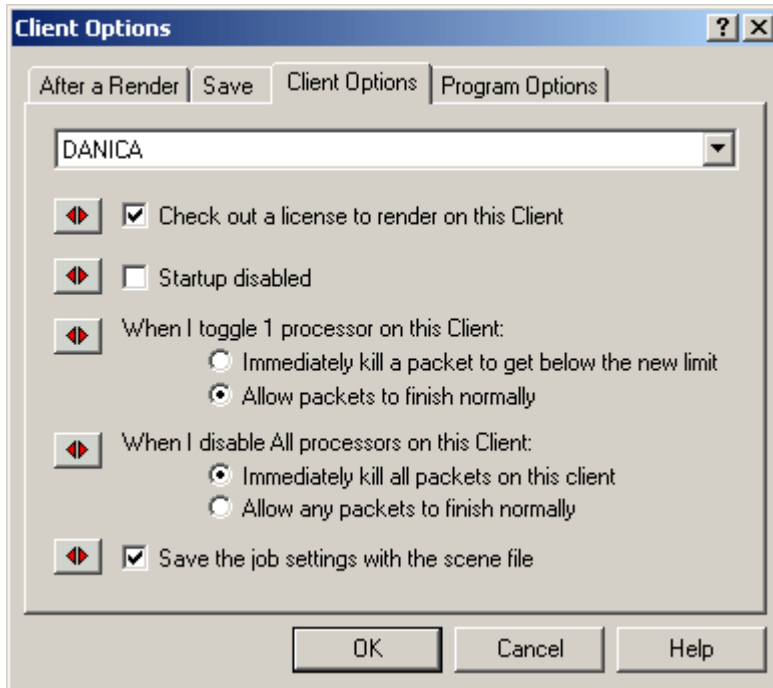


The second page of the options dialog box configures how and where Smedge stores information from the renders.

After each successful render, Smedge can generate an HTML file with the history of the execution of that job. Check the **Save HTML Job History from each job** check box to enable this behavior. If you have checked this box, you have the option of selecting the folder in which the files will be saved. If you leave the folder blank, Smedge will use its default location: a subdirectory under your TEMP directory called "Smedge." Because these files can fill up your hard disk if you don't delete them, Smedge has the option to limit the total size these files can consume. Every hour, Smedge will clean up files (older files first) to get under the limit you set.

Smedge captures the output from the renderer. To save this output, check the **Save captured output from each packet** check box. Smedge will dump the entire output to a file. You can configure the folder where Smedge will save these files. If you leave the folder blank, Smedge will use its default location: a subdirectory off your TEMP

directory called "Smedge." Inside of this directory, the file will have the scene name, the first frame of the packet, and a ".log" extension. Because these files can fill up your hard disk if you don't delete them, Smedge has an option to limit the size that the folder can consume on your hard drive. Every hour, Smedge will clean up files (older files first) to get under the limit you set.



The third page of the option dialog box allows some general configuration options.

Check out a license to render on this Client controls whether this client will request a license from the server to allow it to render. Remember that Smedge does not need to be licensed to control the queue. You only need to check out a license if you want to use this client to render.

Startup disabled tells Smedge to ignore the last saved state for this client and force it to startup disabled.

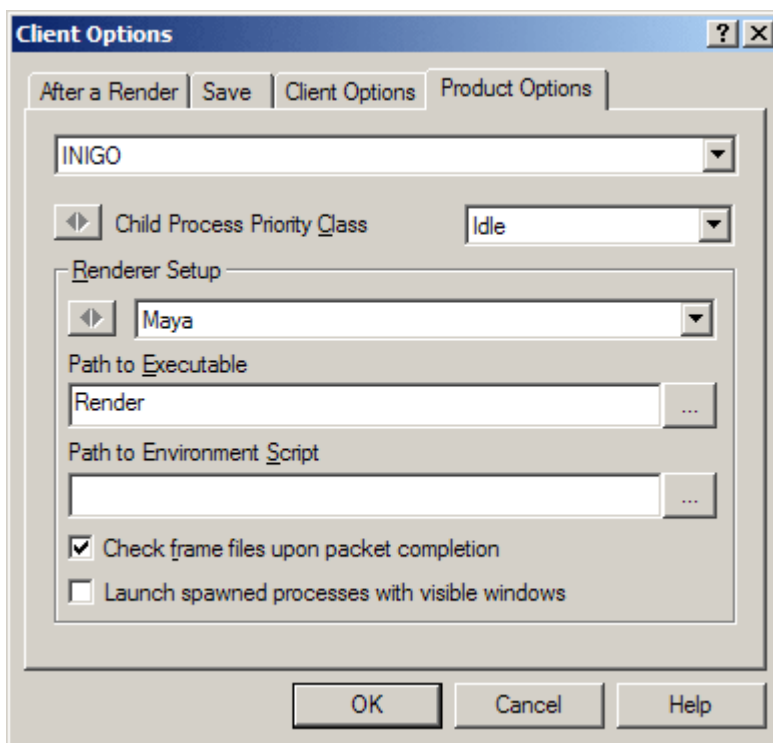
When you change the number of available processors to Smedge, using either the **Disable One** or the **Disable All** controls, your selections on this window control what will happen. For the **Disable One** command your options are:

- **Immediately kill a packet to get below the limit.** This tells Smedge to abort enough running packets to get below the new processor limit. If you had two single threaded packets running on the client, Smedge would abort one of them. That packet would then get re-queued. If you had a multi-threaded packet running, Smedge would abort the multi-threaded packet, then start a new packet single threaded.
- **Allow the packet to finish normally.** This tells Smedge not to mess with any packets, but to use the new processor limit after any running packets finish.

For the **Disable All** command your options are:

- **Immediately kill all packets on this client.** This tells Smedge that when you disable this client, you want to immediately stop any work on it. Any packets that are running when you disable the client will be aborted and re-queued.
- **Allow any packets to finish normally.** This tells Smedge that when you disable this client any work that is currently running on it should be allowed to finish, but no new work will be started.

Save the job settings with the scene file tells Smedge to save a `.smr` file with all the settings needed to re-submit this job. The file will be saved in the same directory as your scene file and will have the name of your scene file plus the `".smr"` file extension. You can later double click this file to re-queue the whole job. You can also use the **Load** menu option or button to load the `.smr` file into the Smedge queue.



The last page of the options dialog configures how Smedge operates with the specific products it can use.

The **Child Process Priority Class** selection allows you to control the priority that Smedge will give to the actual renders it launches. You can select from "Normal," meaning that the processes will have equal priority with other processes on the machine, or "Idle," meaning that the processes will be running at a lower priority than other user processes. If you chose Idle, you may find your machine more responsive even while a render is executing, but the renders will usually take longer to complete.

In the "Renderer Setup" area, you can configure details for each individual product that Smedge can support. First, select the product you wish to configure from the drop-down list (Maya has been selected in the screenshot). You have up to three options you can

configure for each product. The **Path to Executable** has the name and optionally the full path to the executable for the renderer. Note that you generally should not change this unless your product is not in your PATH. Make sure to select the correct EXE file if you do provide a full path, or your jobs will not render.

Please note that the MentalRay for Maya product works slightly differently than all the rest. It uses the path from the normal Maya product to find the actual program file to be executed. In the **Path to Executable** field for the MentalRay for Maya type, you should put the actual MEL command that starts the render. This should **not** be a path to an executable, the batch rendering command that comes with the program, or a .MEL script file. If you put anything other than the actual MEL command to start the render, the rendering will fail.

For more information, see the [MentalRay for Maya](#) section.

In the **Path to Environment Script** field you can put the full path to a file that Smedge will read for additional environment variables that you may want or need for the renderer to function. Smedge will read this file as a text file, but will not execute it (for example if it is a batch file). Smedge is pretty good at parsing these files, and can even handle expanding environment variables that use other environment variables. However, it is not a perfect system, and if you have complex automated batch systems generating environment variables, they may not work as expected.

For some products, Smedge can extract the filenames generated by the renderer. For these products, Smedge checks that every file that the renderer said it created was actually created and that the files are not empty. If the files do not exist, or are empty, Smedge assumes the renderer failed, and marks the packet to be requeued. There may be times when you want to disable the file check (for example if you have a post-render MEL command that moves the frames around in a Maya scene). You can uncheck the **Check frame files upon packet completion** box to disable this check.

Check the **Launch spawned processes with visible windows** button if you want the output from this type of render to always be visible on the screen. When a job of this type starts, Smedge will open an Output Window for the job, and you can monitor the status. The window will automatically close when the render finishes.

Configuring Smedge across your network

Any of the Client configurations you can do through the Client Settings and Smedge Options dialog boxes, you can change on any client currently connected to your Smedge network. Simply select the client you want to change in the "All Clients" tab of the main window, then choose the **Edit Selected Client Settings** or **Selected Client Smedge Options** items from either the **Host** menu or from the right button context menu.

If you have a single client selected and choose **Edit Selected Client Settings**, Smedge will open the Client Settings dialog box with the current settings from the client that you have selected. When you press **OK**, Smedge will notify that client of its new settings, and that client will save those settings and update its display if needed. If you have multiple clients selected, Smedge will open the Client Settings dialog with a combined average Client. When you press **OK**, Smedge will update every client you have selected with the

settings you have chosen. Each client will save those settings and update its display if needed.

If you choose **Selected Client Smedge Options** and you have the local client selected, Smedge opens the Smedge Options dialog box with the current local options. If you have a remote client selected, that client's settings will be displayed, along with the name of the client in the list. If you have multiple clients selected, the dialog box can show you each client's settings, and you can use the buttons with the red arrows to transfer a setting from the current client to all of the other clients you are editing. When you press **OK** Smedge will notify all the selected clients of the options you have set. Every client will get the settings that exist for that client, and save these options for itself.

Other Smedge configurations

There are other settings you can configure about how Smedge will run on your system. Mostly, these are little things about the how the GUI behaves.

Frame Viewer

You can configure which application Smedge uses to view rendered images. Select [Select Frame Viewer](#) from the [Host](#) menu. Smedge passes the filename to the application in a command line style (e.g., fcheck.exe "filename.iff"). This setting only applies to the current user.

Job name prefix

Smedge can automatically append some text to the front of all job names submitted from this machine. You can set this text by selecting [Set Job Name Prefix](#) from the [Host](#) menu. This setting only applies to the current user.

Debug Level

You can adjust the amount of information stored in the debugging log file. This file is useful for when things go wrong. For more information see [Debug Level](#) in the [Host](#) menu.

Job list colors

You can configure the colors used to display the status of Jobs in the list on the main window. Select [Colors...](#) from the [View](#) menu. This setting only applies to the current user.

Window properties

You can adjust several properties about how the Smedge main window behaves. These commands are all in the [View](#) menu:

[Always On Top](#) Allows you to set the Smedge main window to always stay on

top of other windows, even when it is not the active application.

Hide on Minimize Allows you to set the Smedge window to disappear from the task bar when you press the minimize button. To restore the window you can either use the **System Tray** or you can launch another instance of the Smedge.exe program, and your window will be restored.

Start Hidden Allows you to set Smedge to always start up hidden. To restore the window you can either use the **System Tray** or you can launch another instance of the Smedge.exe program, and your window will be restored.

Toolbar Toggles visibility of the **Toolbar**.

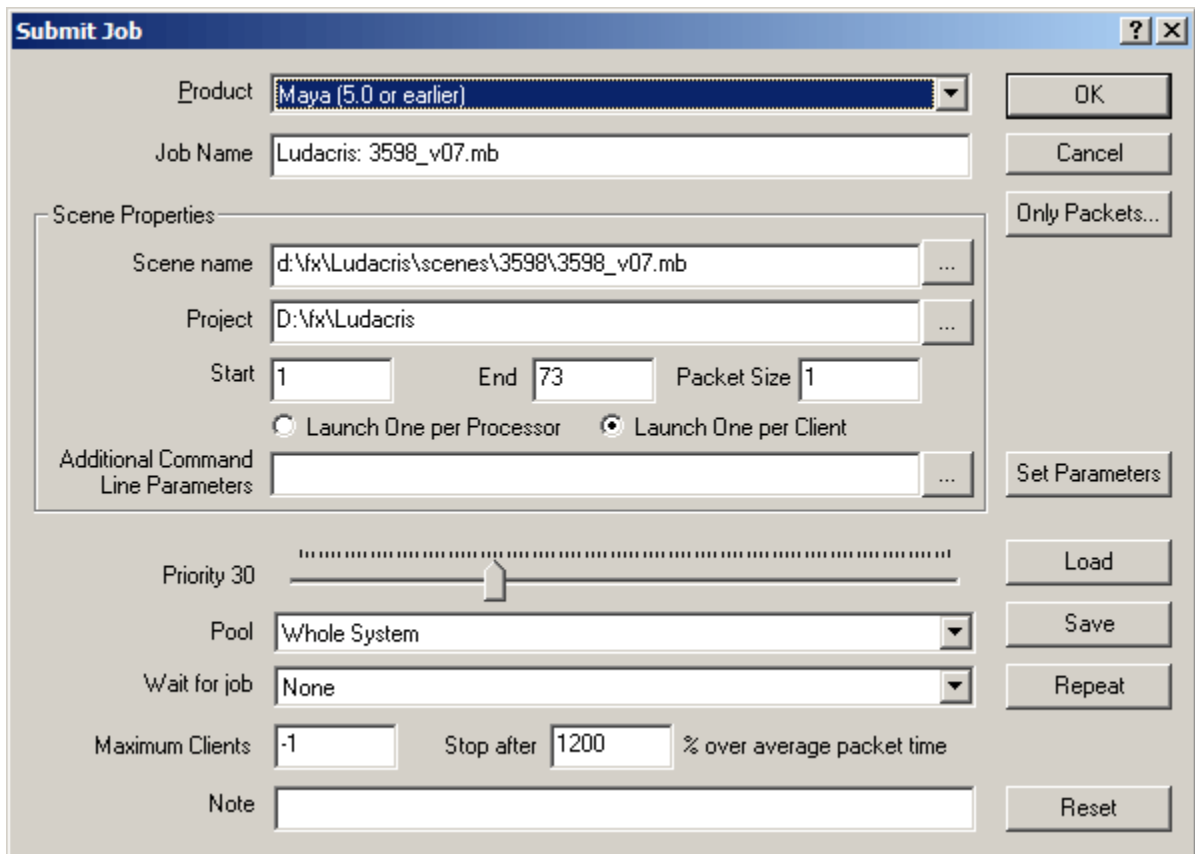
Status Bar Toggles visibility of the status bar on the main window.

The Smedge Job

- Submitting a Job
 - Extra Parameters
 - Loading/Saving a job and Repeats
 - Command Line Interface
 - Drag and Drop
 - Submitting a bunch of jobs with the same settings at one time.
 - The Job Lifespan
 - Viewing the a job's history.
 - The Check Frames Window.
 - Changing job priority.
 - Changing job pool
 - Stopping a job.
 - Saved Job Information
 - After a Packet
 - After a Job
 - Other Cool Stuff
 - Job IDs.
 - Job Files.
-

Submitting a Job

Select Queue|New Job to bring up the Submit Job dialog box. By default, the dialog box will be filled in with the last job that you submitted.



The information at the top and bottom of the dialog box is used by Smedge to control the dispatching of the job. The stuff in the middle, surrounded by the "Scene Properties" box is specific to each renderer, and contains the information needed by the renderer to do the job.

Product

A list of renderers that Smedge supports. By using a renderer from this list, you don't have to worry about formatting a correct command line for distribution. You also gain access to customized features for each renderer. However, by choosing [Generic Script](#) you have complete control to format a command line for any program in the world.

For a complete list of the internally supported products, usage of the Submit Job dialog box specific to each product and possible options for those products, see the [Products Available to Render](#) section.

Job Name

A name that you supply to identify a job in the Smedge interface. What you type here has no implications on the actual render or any filenames. You can use any characters you want in this name. The default action is to design a name based on the project and scene filenames, but the format varies by product.

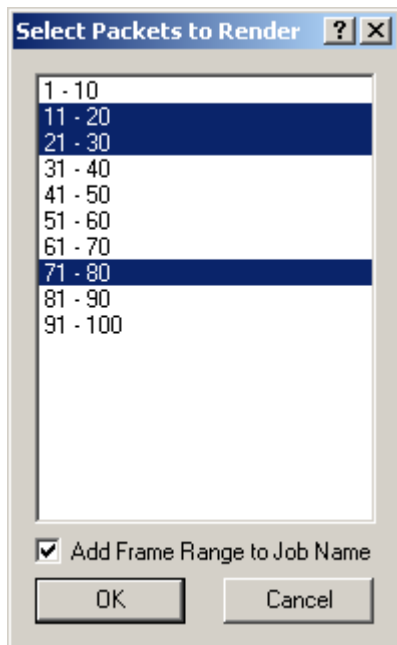
Priority

Tells Smedge what priority to assign to this job. Priorities range from 0 (lowest) to 100 (highest). A job with priority set to zero is essentially paused. No packets from that job will be sent out until the priority is increased. Higher priority jobs are executed before lower

priority jobs.

- Pool** Tells Smedge which pool to assign to this job. Clients that are members of this pool will have priority rendering this job. You can force clients that are not members of this pool to never work on this job. See [The Smedge Client](#) for more information.
- Wait for Job** You can make this job dependent on the completion of another Smedge job. Select a job from this list. No packets from this job will be started until every packet from the selected job has been completed.
- Maximum Clients** You can limit the number of machines that will be allowed to work on this job. Set this to -1 to disable this restriction.
- Stop after...** Smedge tracks how long every packet takes to render. If a packet takes longer than this given percentage of time than the average packet time, Smedge will Stop and Requeue the packet, on the assumption that it has probably failed somehow. Use caution setting this number too low, as normal variation in packet time may be incorrectly interpreted as a failure.
- If you set this value to zero, this check will be disabled for the job.
- Note** You can attach a note to every job. This note will display in the Jobs pain of the main window.

You can submit the job by pressing the **OK** button. You can also choose to select only certain packets from the job by pressing the **Only Packets** button. This will bring up the Select packets dialog. You can select which packets from this job to submit. Each packet will be submitted as its own job, one packet long.



Extra Parameters

Most job types have a space where you can type in extra parameters to send directly to the renderer via its command line. For some renderers, Smedge provides a Common Options dialog box that has a few of the more common parameters that people override on the command line.

You do **not** have to use this dialog (if it exists) to supply command line options. You can simply type any command line parameters into the field on the Submit Job window as you would in a command line.

An even more powerful option is to provide a parameters file. If you put the full pathname to such a file into the field, then Smedge will parse that file at the time of job submission. The file can contain line breaks and comments to make life easy for you. Line breaks are turned into whitespace, and any line that starts with the # character will be ignored. Empty lines are also perfectly acceptable.

The file will be read when you press the **OK** button. This means that you do not have to have this file accessible by every node on the network. It only has to be readable from your computer when you submit the job. Smedge will parse the parameters, and the job will be saved as if you had typed the parameters into the field by hand.

Loading/Saving a job and Repeats

You can fill in the fields of this dialog in many ways. Obviously, you can type things in by hand. You can also use the buttons to the right of the edit fields to bring up dialog boxes that help you fill in fields. For example, for the scene field, it brings up a file dialog box. On the far right of the dialog, you will also see a couple buttons that you may find helpful.

Repeat brings up a dialog that show you the last 20 renders that you submitted. You can then double click on one of those renders and all of its settings will be filled in. If you submit an identical render, it won't make another copy in this list. However, if you change any parameter of the render, Smedge considers it a new render and adds it to this list.

Load allows you to select a **.smr** file. These files are a file on your disk that Smedge uses to save the settings of a job. These files can be generated either automatically by Smedge (see [The Smedge Client](#)) or by using the **Save** button on this dialog box.

Command Line Interface

You can submit jobs into Smedge via a commandline. This is a powerful feature for scripting Smedge job submission, or for integrating Smedge directly into your pipeline via scripts and extensions to the products you use. For full documentation of the Smedge command line interface, see [The Smedge Command Line](#).

For an example of integrating Smedge directly into an animation package, take a look at the `smedgeRenderSession.mel` script included in the distribution. This script shows one way to integrate Smedge into Maya by using Maya's `system` MEL command to call the program with the appropriate settings from your scene.

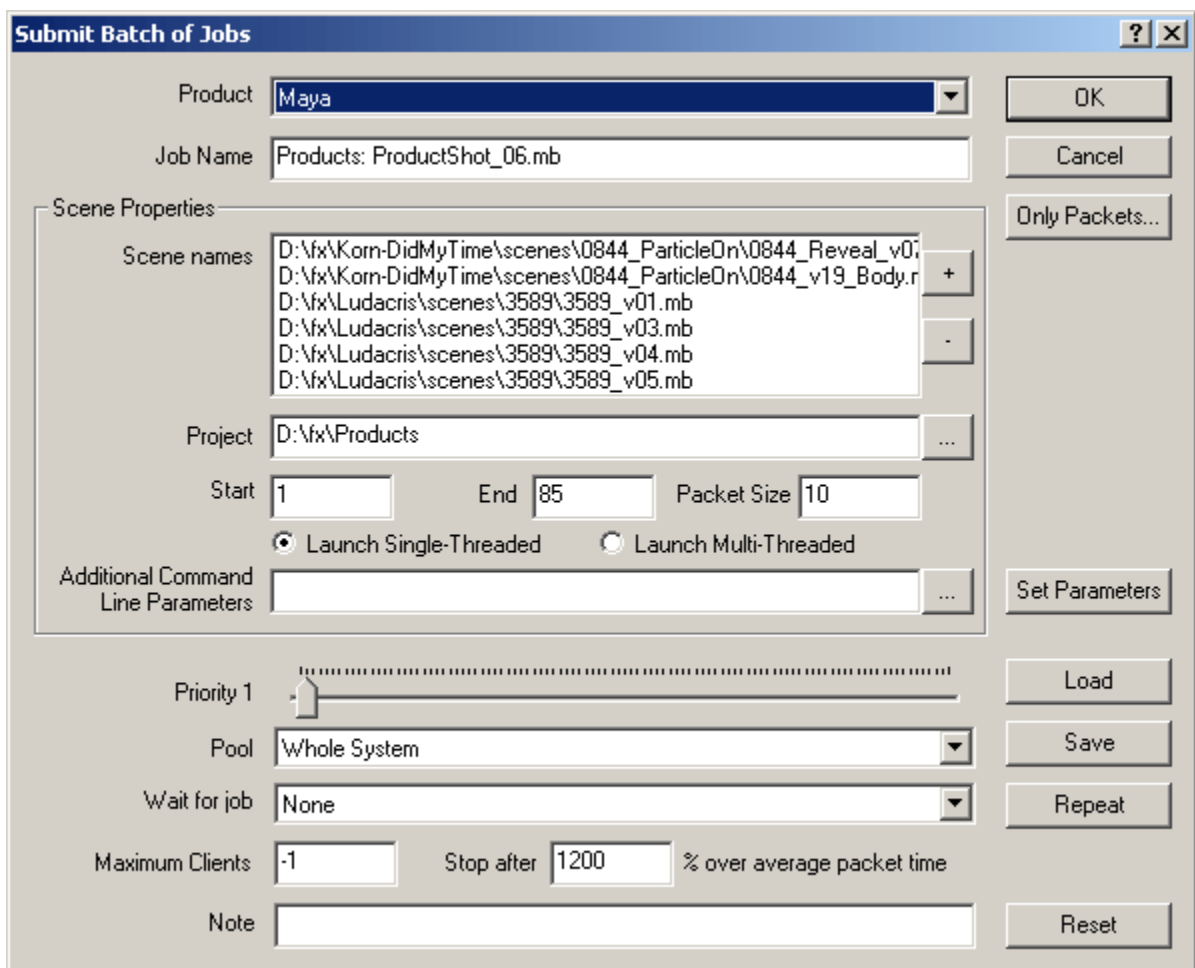
Drag and Drop

You can drag and drop files from the Windows Explorer onto Smedge. For each file dropped, Smedge will open a Submit Job dialog box. You will still have to fill in the frame range and other job specific parameters.

You can also drag and drop a file onto the Submit Job dialog box itself. In this case, the scene (and, for some render types, the project) will be automatically filled in with the file you drop.

Submitting a bunch of jobs with the same settings at one time

Smedge can submit a whole bunch of scenes to be rendered with the same parameters using the [New Batch of Jobs](#) command. This will bring up the Batch Submit Job dialog, which works almost identically to the normal Submit Job dialog box.



The main difference is that instead of a single scene file, you can create a list of scene files. Use the + and - buttons to add or remove scene files. All the other settings in the dialog will be constant for every scene file. Each scene file listed will be submitted as a new job. You can even drag and drop onto the Batch Submit Job dialog to extract a bunch of scenes at one time.

The Job Lifespan

Once a job has been submitted into the system, it will live until every packet in that job has finished, or until the job is explicitly killed by a user. You can see the progress made on a job in the main window. The "Frames" column of the Job pane tells you the frame information about the job (start, end, and packet size) and the percentage of completed packets.

You can also open the Job History window, and see more detailed information about the job. To open the Job History window, you can double click the job, you can choose "Job Statistics" from the menu, or you can select the job, then press Alt + Enter.

Viewing the job's history

Smedge maintains a detailed history of every job as it's being executed. You can access this history in several ways:

- Double click a job in the middle pane of the main window.
- Right click a job and select **Job Statistics** from the **context menu**.
- Select one or more jobs and choose **Job Statistics** from the **View** menu.

Packet	Status	Machine	Time	Errors	Filename
1	Completed	GEORGE	0:00:00:19	0	D:\fx\Products\images\ProductShot_06.iff.1
2	Completed	GEORGE	0:00:00:17	0	D:\fx\Products\images\ProductShot_06.iff.2
3	Completed	GEORGE	0:00:00:13	1	D:\fx\Products\images\ProductShot_06.iff.3
3	Failed	GEORGE	0:00:00:11	1	D:\fx\Products\images\ProductShot_06.iff.3
4	Completed	GEORGE	0:00:00:14	0	D:\fx\Products\images\ProductShot_06.iff.4
5	Completed	GEORGE	0:00:00:13	0	D:\fx\Products\images\ProductShot_06.iff.5
6	Completed	GEORGE	0:00:00:12	0	D:\fx\Products\images\ProductShot_06.iff.6
7	Completed	GEORGE	0:00:00:13	0	D:\fx\Products\images\ProductShot_06.iff.7
8	Completed	GEORGE	0:00:00:12	0	D:\fx\Products\images\ProductShot_06.iff.8
9	Completed	GEORGE	0:00:00:13	0	D:\fx\Products\images\ProductShot_06.iff.9
10	In Progress	GEORGE	0:00:00:04	0	D:\fx\Products\images\ProductShot_06.iff.10
11	In Progress	GEORGE	0:00:00:01	0	
12	Pending				
13	Pending				

The history window shows you a bunch of useful information about the job. The first field shows you the number of finished packets out of the total job. The second shows you the number of active packets out of the remaining unfinished packets. The third is the total amount of CPU time taken for all rendering on this job so far. This number includes packets that are in progress, finished, and failed.

The fourth field is an average amount of time for the packets rendered so far, and the fifth is an estimate of the time left to render based on that average. Note that the average is simply the average of finished packets, so if there are no finished packets, it cannot calculate an estimate, and if the later packets render at a vastly different speed than the early ones, the estimate can be significantly off. Finally, if Smedge is able to detect the location of the image files, or if it's supplied from the user, it will be shown in the last field.

The history of a job is specific to that job's life in Smedge only. Smedge does not maintain history for specific scenes across jobs. So if you resubmit a job, the old history will not be accounted for in the new job's history.

The spreadsheet shows a detailed history of every packet from a job. If the **Show all failed packets** option is checked, there will be one entry for every attempt to render a packet, including all failures and killed packets. If the box is unchecked, the list will only list the most recent history element for each packet. You can sort this list by any column, but ascending and descending. Click on the column headers to change how the list is sorted.

The history window comes with its own set of menus. From these menus you can do lots of things with the job, its packets and the individual frames.

File menu

Save	Saves the job history to an HTML file. This is the same sort of HTML file that Smedge can generate automatically at job completion
Explore to scene directory	Open a Windows Explorer window in the directory with the scene file
Explore to images directory	If the image folder has been found or set, open a Windows Explorer window in the image directory
Check all frames...	Opens the Check Frames window for this job

Job menu

Job settings...	Opens a Job Settings dialog box to view the job as it was submitted, and change the pool, priority, note, or scheduling options for this job.
Set images directory...	Manually set the image directory for this job. This will override any previously detected image directory, and won't be changed again automatically. Changing this directory after Smedge automatically detects it may cause the frame checking features of Smedge not to work correctly.
Delete job	Deletes this job from the Smedge interface. If the job is still running, all packets will be canceled first.
Resubmit this job	Resubmits this job as a new job. The new job will have all settings identical to the current job. However, any related history of this job will not be carried forward to the new job.

Packet menu

The **Packet** menu allows you to control individual packets from this job. The behavior of this menu depends on your selections in the list of packets. You can also access this menu by right clicking in the list of packets.

Stop packet and requeue	These are the commands to stop a currently executing packet with various requeueing options. They are the same as the main interface versions of the commands. You can read more about how they work in the Stopping a Packet section of the Smedge Packet chapter.
Stop packet permanently	
Stop packet and submit as new job	
Resubmit packet as new job	This command will resubmit the selected packet or packets as a new job. This submission will happen regardless of the status of the selected packets. All job parameters will be the same as the original job, except for the range, which will be that of the packet.
View frame	This will view the last known filename associated with this packet. This will be the filename that's shown in the

Filename column of the packet list.

Copy filename to clipboard

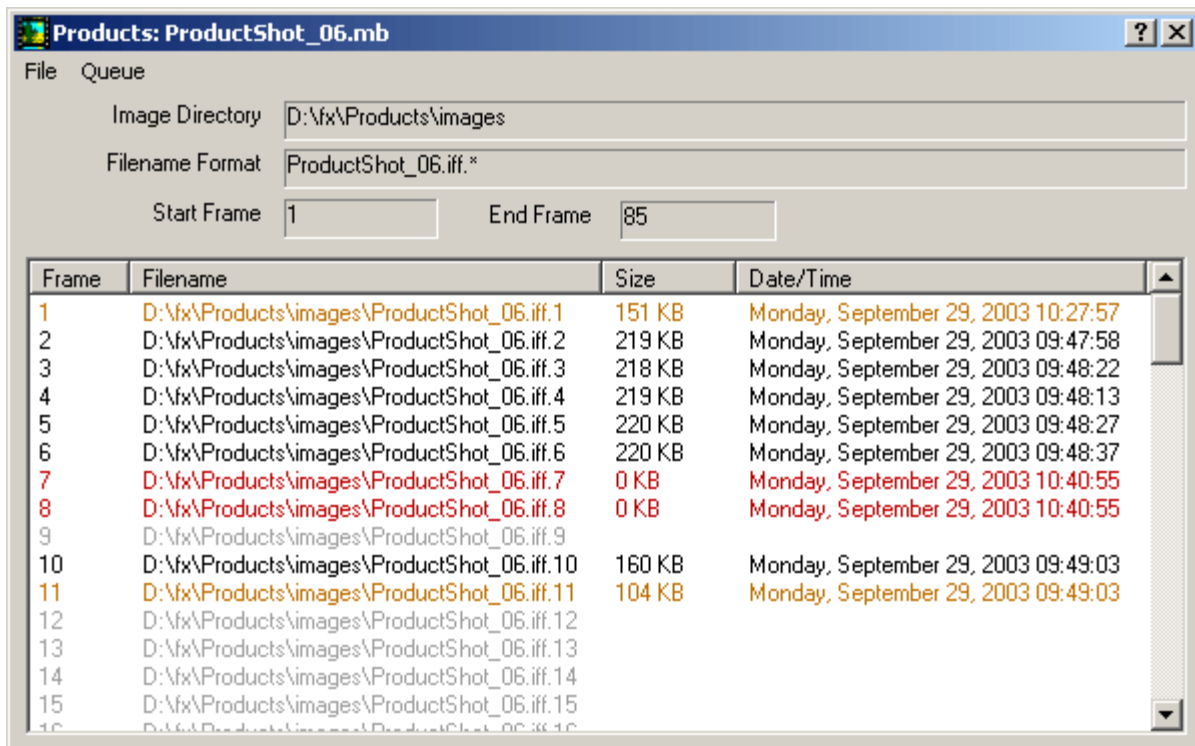
This will copy the last known filename associated with this packet to the clipboard. This will be the filename that's shown in the **Filename** column of the packet list.

View output

This will try to view the captured output from the selected packet. If the packet is still in progress, Smedge will try to connect to the rendering client, and view the captured output live, as it streams from the renderer. For any other packet, Smedge will try to find the saved captured output file for that packet. If the file can be found, Smedge will show that file. This option works best when all clients are directed to save their captured output in a shared folder on the network.

The Check Frames Window

Smedge can automatically check the rendered frames from a job. The Check Frames window allows you to see every single frame file that should exist from a job, along with its size and date. Smedge tries to determine bad frames in the list, and displays them in different colors for easy identification. From this list you can easily requeue some or all of the bad frames from a job.



Frame	Filename	Size	Date/Time
1	D:\fx\Products\images\ProductShot_06.iff.1	151 KB	Monday, September 29, 2003 10:27:57
2	D:\fx\Products\images\ProductShot_06.iff.2	219 KB	Monday, September 29, 2003 09:47:58
3	D:\fx\Products\images\ProductShot_06.iff.3	218 KB	Monday, September 29, 2003 09:48:22
4	D:\fx\Products\images\ProductShot_06.iff.4	219 KB	Monday, September 29, 2003 09:48:13
5	D:\fx\Products\images\ProductShot_06.iff.5	220 KB	Monday, September 29, 2003 09:48:27
6	D:\fx\Products\images\ProductShot_06.iff.6	220 KB	Monday, September 29, 2003 09:48:37
7	D:\fx\Products\images\ProductShot_06.iff.7	0 KB	Monday, September 29, 2003 10:40:55
8	D:\fx\Products\images\ProductShot_06.iff.8	0 KB	Monday, September 29, 2003 10:40:55
9	D:\fx\Products\images\ProductShot_06.iff.9		
10	D:\fx\Products\images\ProductShot_06.iff.10	160 KB	Monday, September 29, 2003 09:49:03
11	D:\fx\Products\images\ProductShot_06.iff.11	104 KB	Monday, September 29, 2003 09:49:03
12	D:\fx\Products\images\ProductShot_06.iff.12		
13	D:\fx\Products\images\ProductShot_06.iff.13		
14	D:\fx\Products\images\ProductShot_06.iff.14		
15	D:\fx\Products\images\ProductShot_06.iff.15		
16	D:\fx\Products\images\ProductShot_06.iff.16		

Smedge will try to automatically determine the filename and full path from the renderer. However, even if Smedge cannot automatically determine this information, you can actually enter it yourself. You can also override the range Smedge will check. Because this checking can be processor intensive, Smedge has some shortcuts built in to help cut down on processor use, at the expense of accuracy. Specifically, you may notice that if

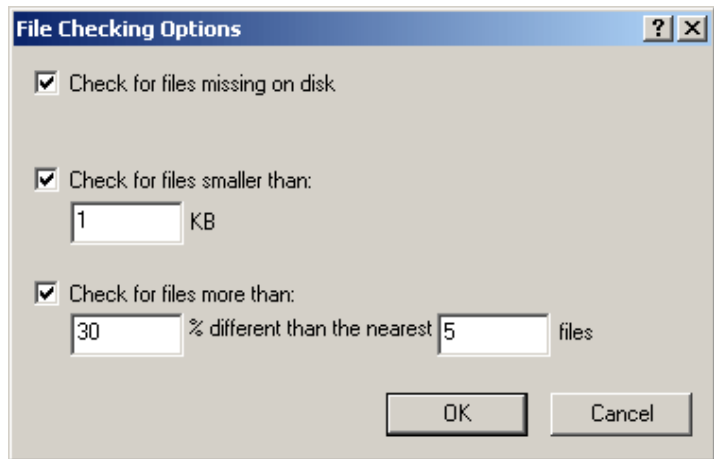
you move or delete files early in a sequence, those changes won't show up unless you manually force the list to update (available in the **File menu**). The processor usage will only be significant on sequences longer than a couple hundred frames.

File menu

View frame	This will view the currently selected frame or frames.
Explore to images folder	This will open a Windows Explorer window in the folder where the images are located
Update list	This will force Smedge to refresh the entire file list. This is useful if a render is partially finished, and you want to verify the file status of earlier rendered frames
Set job image folder	This allows you to set or override the image directory for the job. Once you set the folder, any frame detection will take place in that folder from then on. If Smedge normally detects the folder automatically, it will not reset the folder after you set it manually.
Set image filename format	This allows you to set or override the image filename format for images rendered by this job. Once you set this formatting, it will be used from then on. If Smedge normally detects the format automatically, it will not reset the format after you set it manually. Use the wildcard character (*) to signify an unpadded frame number, or a series of question marks (?) to signify a padded number, where the number of question marks indicates the number of padded digits to use.
Set frame range	This allows you to set the range of frames for Smedge to check. Normally, the range is the full range of the job, but you can override it here. If you override the range, Smedge will only show the range you set, but this check only affects the Check Frames window, not the job. The job will always render the full range it has been set to. If you close the Check Frames window, the next time you open it, Smedge will revert to checking the entire frame range of the job.

Queue menu

Select bad and missing frames	This will select the frames in the list that Smedge has detected as either bad or missing. No other action is taken other than to select the frames in the list. The criteria used to judge "bad" or "missing" are customizable.
Configure bad frame selection	This allows you to customize the criteria used to judge bad frames.



If you select **Check for files missing on disk** then frames that are not found at all on the disk will be shown in grey.

If you select **Check for files smaller than** then frames that are below the minimum threshold size you set (in kilobytes) will be shown in red. The default is 1 KB, generally much smaller than any normally complete image file will be, even if it's a compressed format and a completely black image.

If you select **Check for files more than** then frames that vary in size by the specified criteria will be shown in orange. The percentage is a difference in size percentage of an average of the number of files specified in each direction from the file. The default settings look for a 30% variation in size compared to the average of the 5 files before and 5 files after the current file. These settings work for many animations, but you may need to tweak them for highly dynamic scenes.

Requeue all bad and missing frames This will cause Smedge to requeue all currently detected bad or missing frames. The frames will be queued as new jobs. Be aware that this command ignores the current selection of the list, and requeues the frames based on the detection state. If the detection criteria are too narrow, Smedge may miss some bad frames, and if they are too loose, extra frames that had successfully rendered may get requeued.

Requeue selected frames This will cause Smedge to requeue any frames currently selected in the list. This command ignores the detection criteria altogether.

Changing job priority

You can change the priority of a job after it has been submitted. To increment or decrement the priority by 1, you can use the Up and Down arrows, or their corresponding menu commands. The keyboard shortcuts for these actions are Control + Up Arrow or Down Arrow.

You can also change the priority of a job by opening the Job Settings window. This window looks just like the Submit Job window, only many of the options are read-only. However, you can change the priority of a job to any valid priority, and then press OK, and Smedge will update the job's priority.

Changing job pool

As with priority, you can change the pool that a job is assigned to after the job has been submitted. You can choose a pool from the menu, or you can open the Job Settings window and select a pool from the list.

Stopping a job

You can stop a job at any time in one of two ways. If you set the priority of a job to zero, no new packets will be launched from that job. The job is essentially paused where it is. Note, however, that any currently processing packets will finish naturally.

If you want to permanently halt a job, you can simply delete the job from the queue. When you delete a job, all work currently in progress will be aborted.

Saved Job Information

Smedge saves two different types of information about the jobs it renders.

After a packet

The text output that you normally see on a commandline render is captured by Smedge for every packet. Besides being able to see the output in a window while renders are going, Smedge can also save this output in a text file for you to view later. To view this output, you can choose **View Saved Captured Output** from the **View** menu. The files are named with the job name plus the first frame of the packet, plus the **.log** extension.

By default these files are saved in the %TEMP%\Smedge folder on your local machine. However, you can configure the folder in the **Smedge Options** dialog box. One useful idea is to have all of your clients save their captured output into a shared folder on your file server. This way you can find the output from any packet that is rendered on your system, if you want.

You can also use the **Smedge Options** dialog box to configure a limit for how much disk space these files will take up. Note that if you chose to direct every machine's output to the same folder, you will probably want to increase the default limit of 5 MB of files.

After a job

After any successful job completion, Smedge can generate an HTML file with a complete history of that job. This file is useful for gathering statistics about your renders, and for having a log of the complete history of packet execution for that job. You can view these files by choosing [View HTML Job History](#) from the [View](#) menu.

By default these files are saved in the %TEMP%\Smedge folder on your local machine. However, you can configure the folder in the [Smedge Options](#) dialog box. Also, just as with the captured output files, you can ask Smedge to limit the amount of disk space that these files can possibly consume.

These files are named by the job name. If more than one file with the same job name exists, Smedge will attach a suffix number to keep each iteration of the same job separate.

Other Cool Stuff

Some other cool features of jobs that you may find useful

Job IDs

Every job has a unique ID code. For normal GUI operation, you should never need to know anything about this. However, it does come in handy for commandline operations. Specifically, this ID is needed if you want to do any commandline control of a job, or to submit a new job that is to wait for an existing job.

You can find the ID for a job in the GUI by selecting the [Show Job ID](#) command in the [Queue](#) menu. The ID is also displayed as a return output from the `submit.exe` program.

For more information, see the [Command Line Interface](#) reference.

Job Files

You can load and save jobs on disk as a Smedge `.smr` file. A single `.smr` file can contain zero or more jobs. Though the file is saved in an INI format, it is not recommended to modify this file yourself.

You can load or save a single job in the [Submit Job](#) dialog box. If you load a file that has multiple jobs in it from the **Load** button, only the first job in the file will be loaded. If you use the **Load** button in the Submit Batch of Jobs dialog box, all job scenes will be loaded, but only the last job's settings will be used.

You can also load jobs from the [Load Job](#) command in the [Queue](#) menu. Every job in the file will be loaded in this case.

You can use the [Save Job](#) command in the [Queue](#) menu to save selected jobs from the main window. Select the job or jobs you want to save, and they will all be saved in a

single **.smr** file. Alternatively, you can save the entire job queue in a single file by selecting the [Queue Snapshot](#) command.

The Smedge Packet

- [The Packet Lifespan](#)
 - [Stopping Packets](#)
 - [Viewing the Rendering Frame](#)
 - [Viewing the Output](#)
 - [Packet Options](#)
-

The Packet Lifespan

Packets are started automatically by Smedge as machines become available to perform work and work is queued for the machines. Which job will start first is based on the job priorities. You can get more information about job priorities in [The Smedge Job](#).

Packets will live as long as the rendering program is executing. How the rendering program finishes is one of the ways that Smedge uses to determine if a packet finished successfully. Every program returns some kind of result code, and Smedge can use this code to determine if the renderer is reporting some kind of error. Smedge also uses other checks to make sure that the packet finished successfully, even if the renderer is not reporting an error.

Any packet that finishes successfully will be finished forever in Smedge and will never be requeued. If a packet finishes in failure, Smedge will attempt to requeue the packet. A packet will retry on the same machine only a certain number of times, before restricting that machine from rendering that packet again. If a packet finishes early due to a user aborting it, what happens depends on how the user stops the packet. A packet can be aborted any number of times and will always be able to be requeued on any machine.

Stopping a Packet

There are three ways to stop a currently going packet early:

- | | |
|--|--|
| Stop packet and requeue | This command means that the packet will be immediately stopped and requeued as part of the same job. The job history will show the packet as canceled until it starts again, when it will go back to a normal status. The packet will not count as complete in the progress of a job |
| Stop packet permanently | This command means that the packet will be immediately stopped and never requeued again. The job history will show this packet as canceled, and it will be counted as complete in the progress of a job. |
| Stop packet and submit as new job | This command means that the packet will be immediately stopped and not requeued as part of the job, like the Stop Packet Permanently command, however a new job with only the one packet will be created. |

The preferred method of stopping and requeuing a packet is to use the **Stop packet and requeue** command. The **Stop packet and submit as new job** command is for those pesky packets that don't seem to be responding correctly.

Viewing the Rendering Frame

For some products, Smedge can determine the filename of the frame as it is being rendered. For these products, you have the option of viewing the frame, even as it is still being rendered. Smedge uses the frame viewer that you have configured to display the image. By default, Smedge is configured to try to use fcheck, the image viewer that is included with Maya, but you can configure it using the **Select Frame Viewer** command in the **Host** menu.

Smedge will add the filename as a parameter to the viewer command you select. If the filename contains relative paths, or contains a path to a disk that is not mounted on the local machine, the viewer program may fail to find the file. Also, sometimes the rendering program may determine the name well before the file is actually created on disk. Smedge will report this sort of failure and fail gracefully.

Viewing the Output

Smedge captures the output from the rendering program and processes it as part of its error detection system. Processing the output is also how Smedge can determine the current frame inside a packet, and how Smedge can detect the currently rendering filename. Smedge also saves the captured output for later perusal.

To view the output while a packet is rendering, you can use the **View Output** command. Your Smedge client will connect directly to the client that is rendering, which will spit out all of the captured output from the render so far. As more output is generated by the packet, it will continue to be communicated to your machine, and displayed in the window as long as the window is open. The text in the window can be copied and pasted using the standard Windows commands and keyboard shortcuts.

Packet Options

Besides the above options, you can also use the **Copy filename to clipboard** command to copy the string with the filename into the Windows clipboard. This functionality will only be enabled for products that can determine the filename.

You can also use the **Job Settings** command to open the read-only Submit Job dialog box. This allows you to adjust properties of the job to which this packet belongs. These changes will not affect the packet itself, but will affect future packet generation.

Pools

Overview

Pools in Smedge can be used to further configure how jobs get distributed to machines on your network. You can group machines into pools. Each job has a single pool to which it is assigned. Machines that belong to that pool will have priority in rendering that job.

The pool priority rules can get complex. A client will always do work on jobs that are assigned to pools to which that client belongs over any other work, even if the job has a lower priority than a job assigned to a pool to which the client does not belong.

Clients can be set up to allow them to render work that is assigned to a pool to which they do **not** belong as well. In the [Client Information](#) dialog box, you can uncheck **Only render jobs in my pools**. When this setting is checked, the client is "Pool Exclusive," meaning that if a job is assigned to a pool to which the client does not belong, that job will never be worked on by that client. When unchecked, the client is allowed to do work from other pools, even though it does not belong to them. In this case, the client will do "in-pool" work first, then start working on any "out-of-pool" jobs that are available.

There are two pools to which every client belongs, and which you cannot modify. First, there is the pool "Whole System" which is, strangely enough, a pool that covers every client that is connected. Second, each client's name can be used as a pool to force jobs to render only on that client. This simplifies situations where you just want to render on a single machine using Smedge.

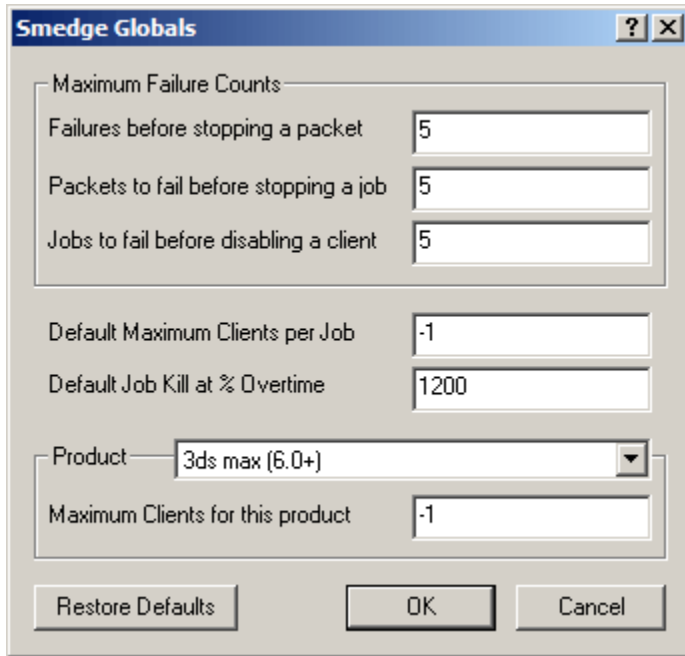
Note that the pool "Whole System" is available through the main window's menus, but the individual client names will only appear in the **Pool** drop down list of the [Submit Job](#) window.

Smedge Globals

- [Overview](#)
- [Maximum Failure Counts](#)
- [Defaults](#)
- [Maximum Product Usage](#)
- [Factory Defaults](#)

Overview

Smedge can configure behaviors that affect all Smedge clients. These settings are called the Smedge Globals, and can be accessed by selecting [Smedge Globals...](#) from the [View](#) menu. When you change these settings, the changes are automatically propagated to every other client on your network, and will affect how every client behaves.



Maximum Failure Counts

Smedge will stop working on things when too many failures occur. You can configure these limits here, using these three fields.

Failures before stopping a packet

A single packet can fail this many times on a single machine before that machine will no longer try to execute this packet any longer. This failure count is reset for a particular packet when you select either [Reset Job Failure Counts](#) on the job for which it's a part, or [Reset Client Failure Counts](#) on the client that failed.

Packets to fail before stopping a job This many packets can fail as above from a single job

on a single machine before that machine will no longer try to start new packets from that job. This failure count is reset for a particular job when you select either [Reset Job Failure Counts](#) on the job, or [Reset Client Failure Counts](#) on the client that failed.

Jobs to fail before disabling a client This many jobs of a particular render type can fail as above before that machine will disable itself from rendering that type of job. The client can be re-enabled using the [Client Information](#) dialog box.

Defaults

Here you can configure the defaults that will be used for new jobs. Note that once you have submitted a job once, Smedge will always fill in the Submit Job dialog box with the same settings as the last submitted job, regardless of what the defaults are. So, essentially, these settings will only affect the very first job a user ever submits.

Default Maximum Clients per Job The default value for the **Maximum Clients** setting in the [Submit Job](#) dialog box.

Default Job Kill at % Overtime The default value for the **Stop After...** setting in the [Submit Job](#) dialog box.

Maximum Product Usage

This allows you to limit the total number of packets for a given type of job on the entire system. This is useful, for example, in situations where the number of licenses for a product is less than the number of machines available, and a [Pool](#) is not desirable.

To configure the limit for a particular product, select that product from the list. Then enter the desired limit for that product in the **Maximum Clients for this product** field. Setting this value to -1 will mean that there is no limit to the number of packets for this product type.

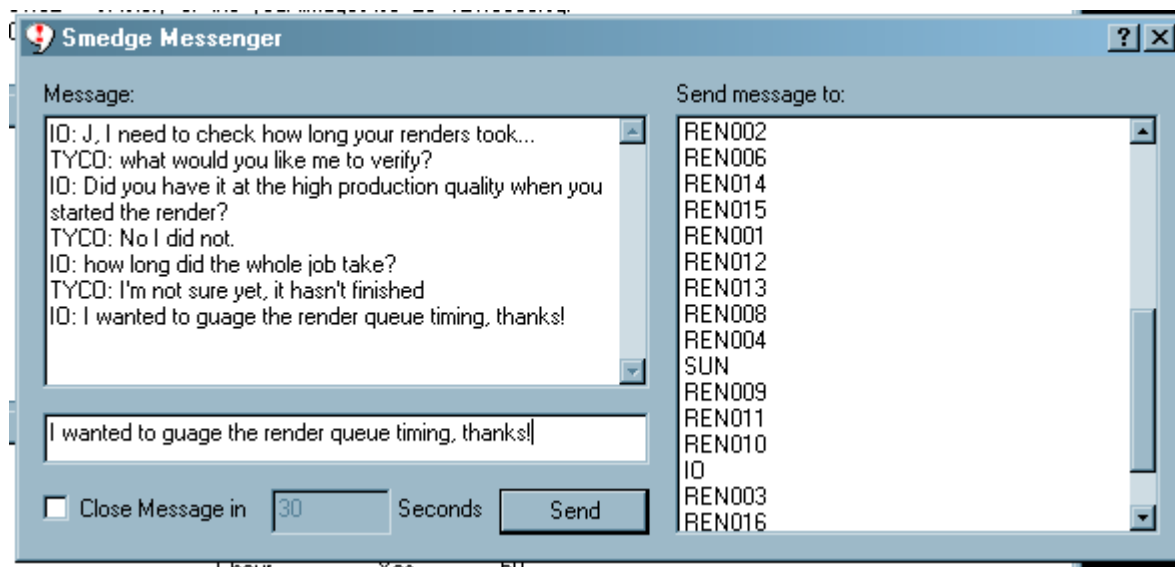
Factory Defaults

Clicking the **Restore Defaults** button will restore the Smedge Globals to their factory defaults:

Failures before stopping a packet	5
Packets to fail before stopping a job	5
Jobs to fail before disabling a client	5
Default Maximum Clients per Job	-1
Default Job Kill at % Overtime	1200
Maximum Clients for all products	-1

Instant Messages

Smedge has an instant message facility built into it for communicating to other Smedge clients. It's not terribly sophisticated at this time, but it does provide a simple, integrated interface for easily sending messages back and forth between clients, or to groups of clients, or even the whole system.



To send a message, select [Send Message](#) from the [Host](#) menu. The Smedge Messenger dialog box opens. This window is a modeless child window, and you can actually open several of these at the same time, allowing several different simultaneous conversations.

Select one or more Clients or Pools from the list on the right side of the window. Selecting a client will, obviously, send the message to that client. Selecting a Pool will send the message to every client in that pool.

To send your message, simply type the message and press the **Send** button. You can also specify an optional timeout period for the message. By selecting the **Close Message in** box and filling out the number, you can tell your message to close on the target machines automatically in the given number of seconds. This is useful when you send messages to a render farm where the GUI may not be used very often. Note that if a user starts to type a response, the timeout will be canceled on their machine.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

After Effects

The screenshot shows the 'Submit Job' dialog box with the following settings:

- Product: After Effects
- Job Name: Opening_v08.aep
- Scene Properties:
 - Scene name: D:\fx\ALIEN\alienAE\Opening_v08.aep
 - Comp name: Final Comp
 - Start: 0, End: 89, Packet Size: 3
 - Launch One per Processor (selected)
- Additional Command Line Parameters: (empty)
- Priority 1: (slider at minimum)
- Pool: Whole System
- Wait for job: None
- Maximum Clients: -1, Stop after: 1200 % over average packet time
- Note: (empty)

This generates a job specifically for [After Effects](#).

To create an After Effects job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

AfterEffects
AE

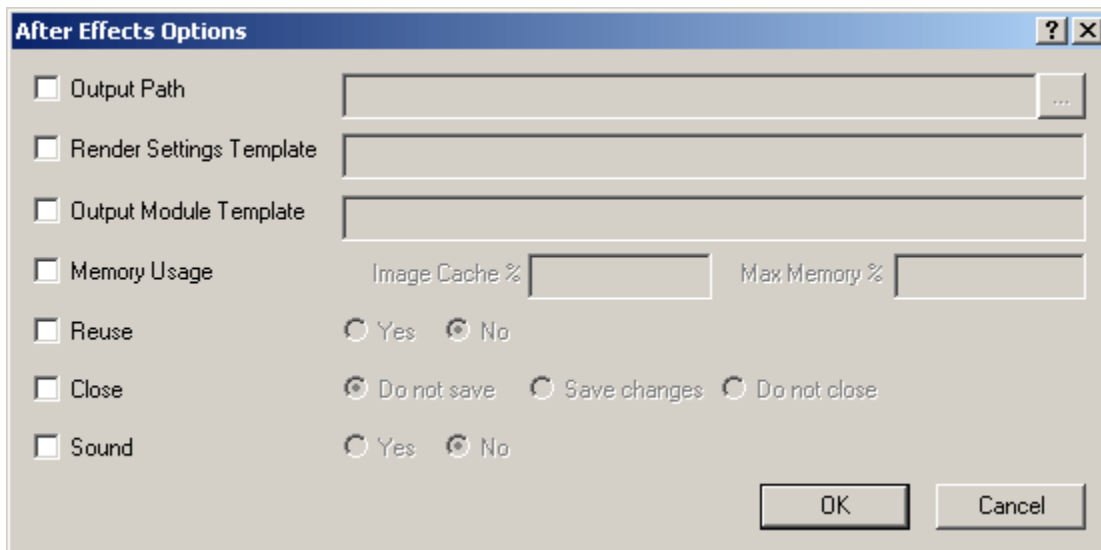
After Effects has a few *quirks* in how it renders:

- ***You must provide a comp name for Smedge to be able to render After Effects projects.*** It won't work without a name.
- You must check the **Use Comp Frame Number** check box in the Output Module settings of your project. If you don't check this, After Effects will start counting at

1 for each packet, and you won't get your expected results. **This is a new feature in AE 6.** To render a 5.5 project, you will need to open it in AE 6, check this setting, and save out a new file. The new file **will not be compatible with 5.5 anymore**, so be sure to use a new filename if you need to use the file with AE 5.5.

- The start and end frames of your job are related to the After Effects timeline, not the rendered frame numbers. If you have set the **Starting Frame** to anything other than zero, Smedge will attempt to determine the offset when it is processing the output. However, the renderer always renders based on the zero based timeline from inside the application.
- If you have checked the **Skip Existing Frames** check box in your After Effects file, this option cannot be overridden at render time from the commandline. This means that your render may not behave as you expect if there are any previously rendered image files when you start the job. One option is to have two comps in your project file, one set to skip existing frames, and the other set to overwrite frames. You can then select which comp to render in the Submit Job dialog box.
- If you have checked the **Use Storage Overflow**, After Effects may report successful renders even when it fails, for example from a full disk. It's probably best to simply leave this option unchecked.

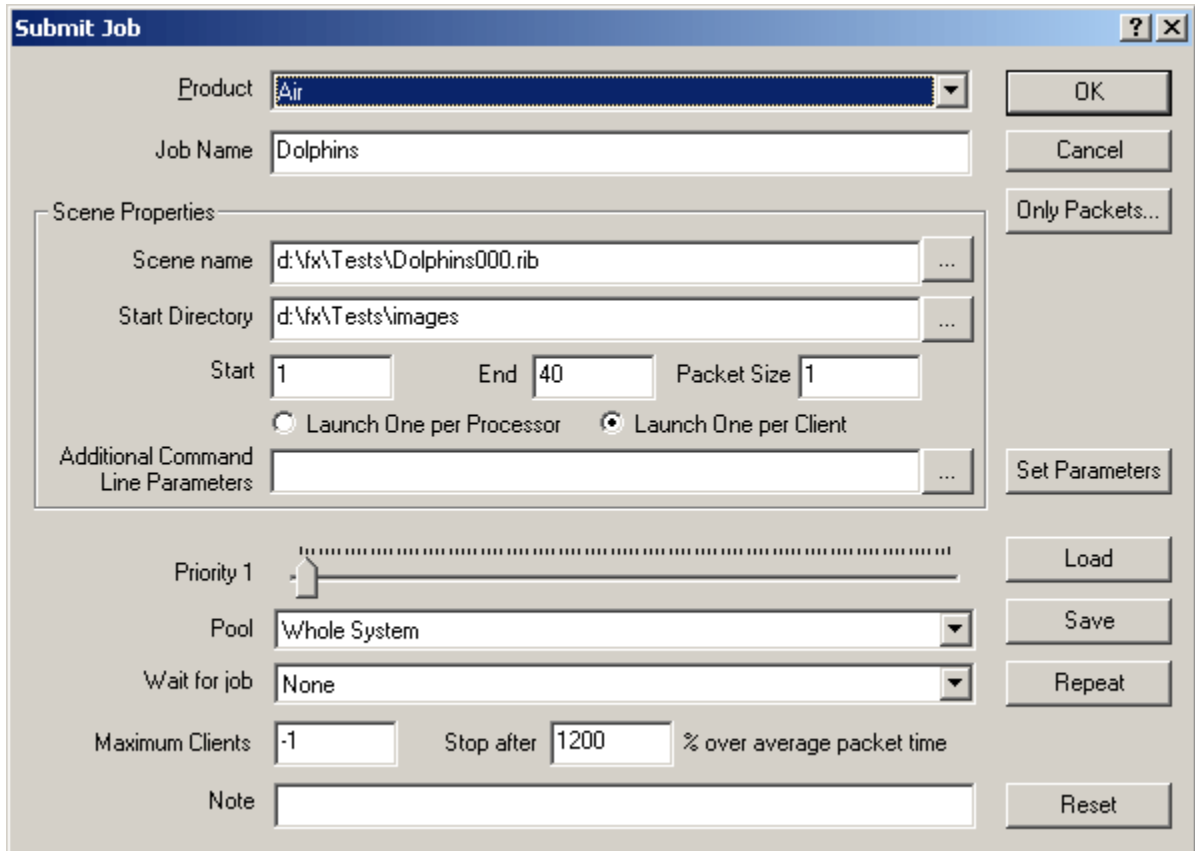
There is a Common Render Options dialog box for After Effects.



Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Air



This generates a job specifically for [Air](#).

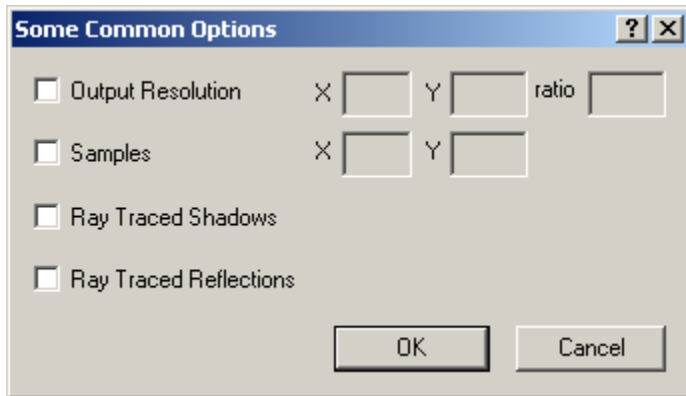
To create an Air job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Air

With Air, you can select any of the .rib files in your frame sequence. Smedge will try to determine the frame range from the available files. It uses intelligence to determine how the frames are actually rendered:

Scene file selected	Example command line
gears.rib	air -frames 1 3 gears.rib
car001.rib	air car001.rib car002.rib car003.rib

Smedge searches for this frame number from the rear of the filename. This means, if you have other numbers in your filenames, you need to make sure that they are before the frame number in the filename.



Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Alias

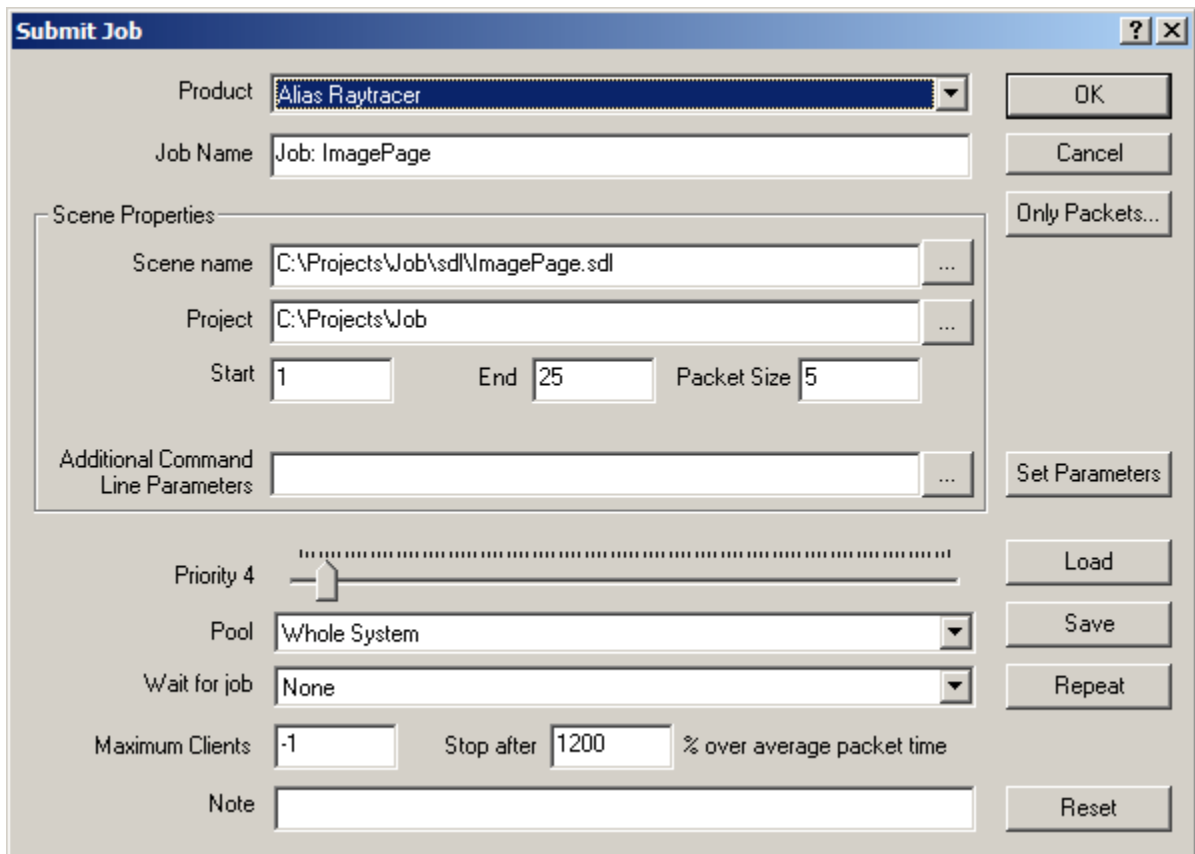
Alias has four different rendering executables depending on which type of rendering you wish to do. Because they are different executables, to fit in the Smedge design, you must select the type of rendering you want to do as the actual product. You can select the **Renderer**, **Raytracer**, **PowerCaster**, or **PowerTracer** products.

The screenshot shows the 'Submit Job' dialog box with the following configuration:

- Product:** Alias Renderer
- Job Name:** Job: ImagePage
- Scene Properties:**
 - Scene name: C:\Projects\Job\sdl\ImagePage.sdl
 - Project: C:\Projects\Job
 - Start: 1, End: 25, Packet Size: 5
 - Additional Command Line Parameters: (empty)
- Priority:** 4
- Pool:** Whole System
- Wait for job:** None
- Maximum Clients:** -1, **Stop after:** 1200 % over average packet time
- Note:** (empty)

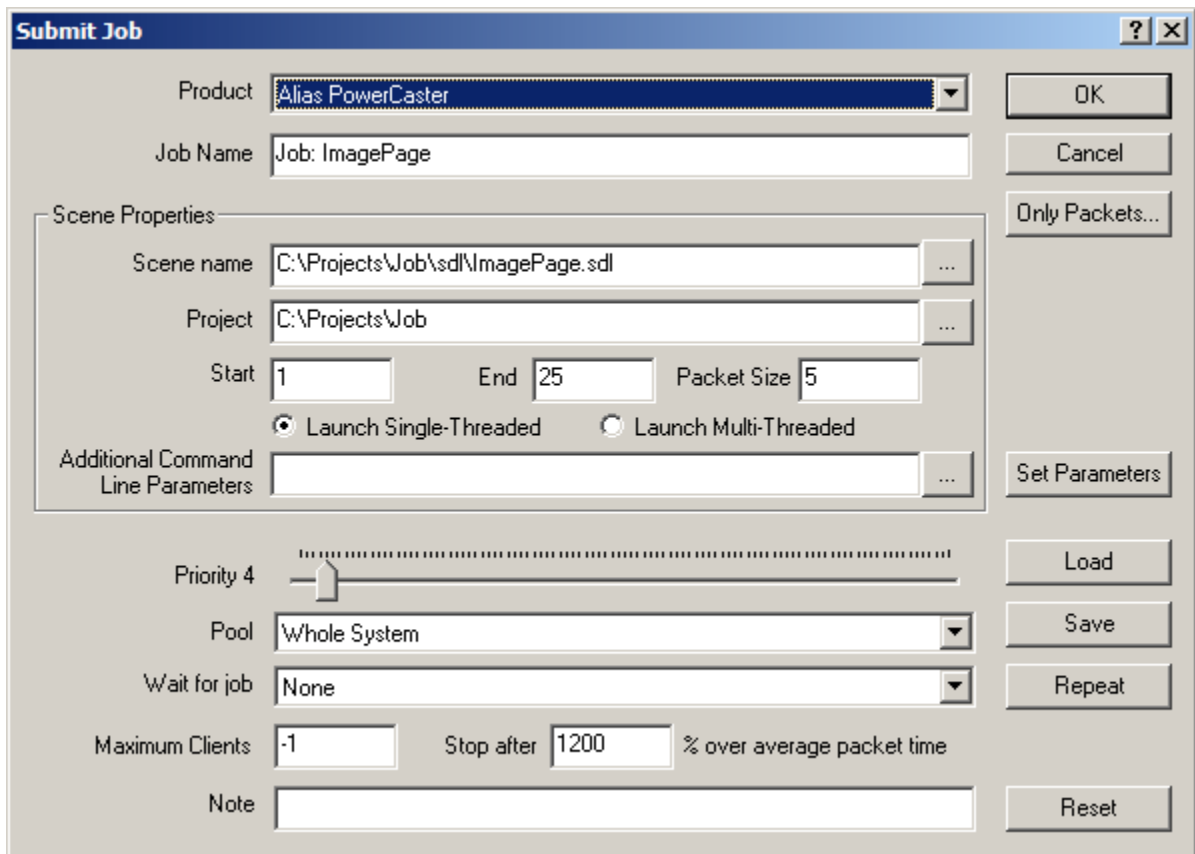
To create a Renderer job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Renderer



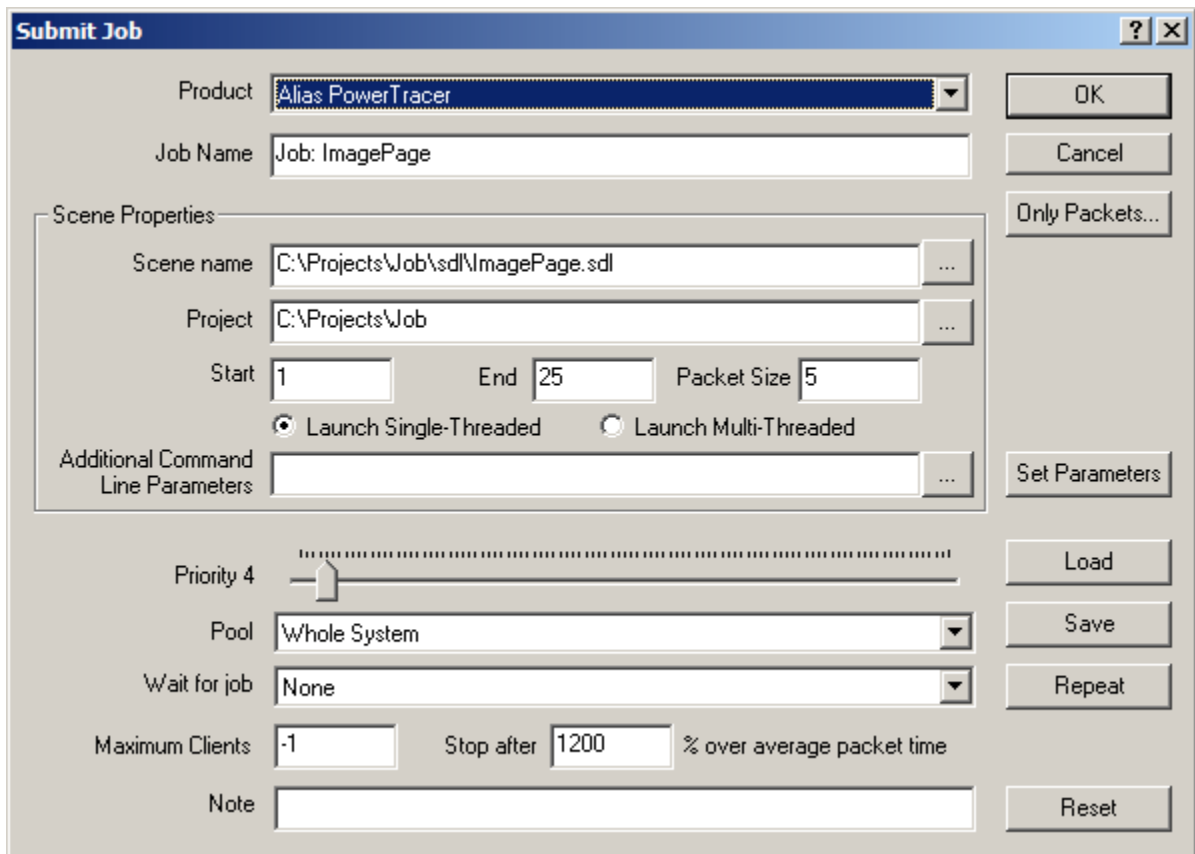
To create a Raytracer job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Raytracer



To create a PowerCaster job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

PowerCaster



To create a PowerTracer job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

PowerTracer

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Chalice

The 'Submit Job' dialog box is shown with the following settings:

- Product: Chalice
- Job Name: Shot32_Test_v02.grail
- Scene Properties:
 - Scene name: D:\fx\Episode3\comps\Shot32_Test_v02.grail
 - Exec Input: execNode
 - Start: 1, End: 50, Packet Size: 5
 - Launch Single-Threaded (selected), Launch Multi-Threaded
 - Additional Command Line Parameters: -e node -j -k -r -s 4 -M 255 0 0 -P h 50
- Priority: 4
- Pool: Whole System
- Wait for job: None
- Maximum Clients: -1, Stop after: 1200 % over average packet time
- Note: (empty)

Buttons on the right side include: OK, Cancel, Only Packets..., Set Parameters, Load, Save, Repeat, and Reset.

This generates a job specifically for [Chalice](#).

To create a Chalice job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Chalice
Chal

With a Chalice job, you can use the Common Render Options dialog for Chalice.

Some Common Options [?] [X]

Scan disk to update all Disk Input nodes

Exec Node

Rendering Type Interleaved Sequential

Do not abort rendering on errors

Create temporary restart option (may not work correctly with single threaded renders)

Slices

Replace Missing Frames

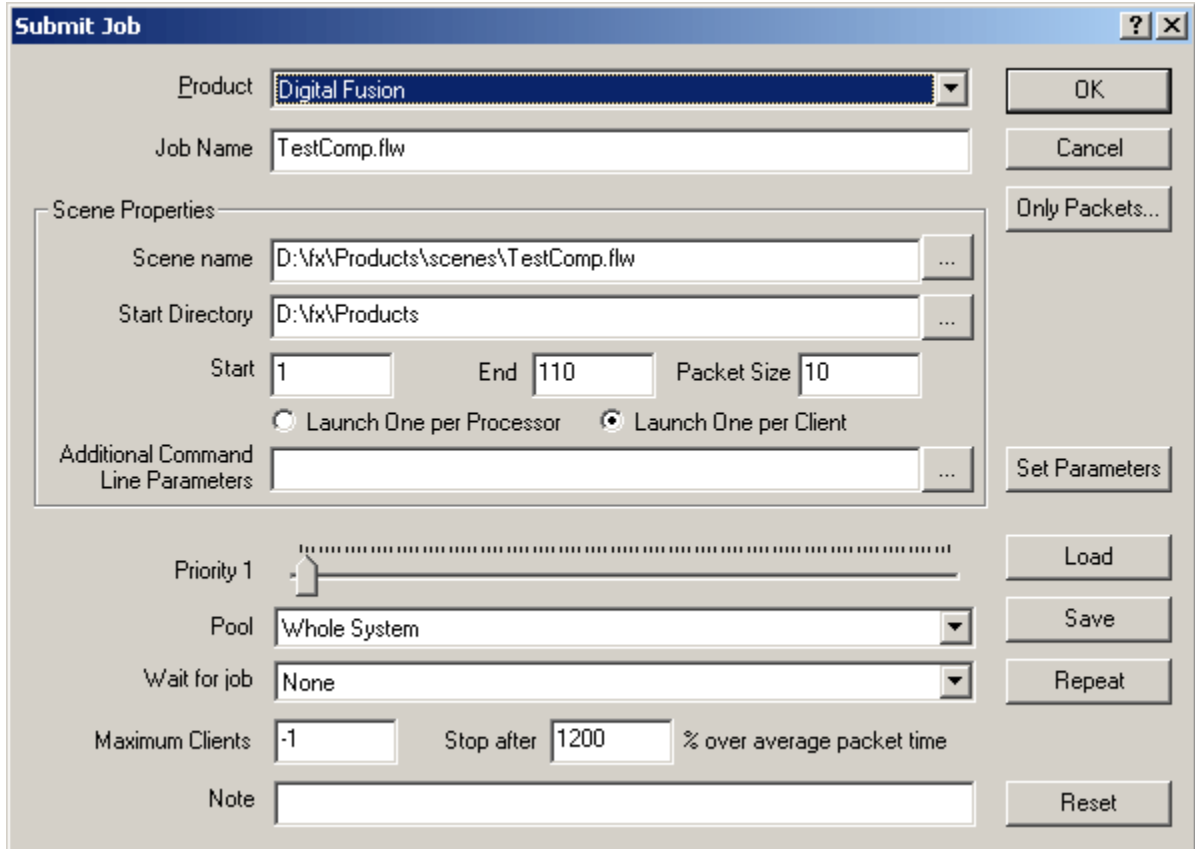
Quality and Scale

OK Cancel

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Digital Fusion



This generates a job specifically for [Digital Fusion](#).

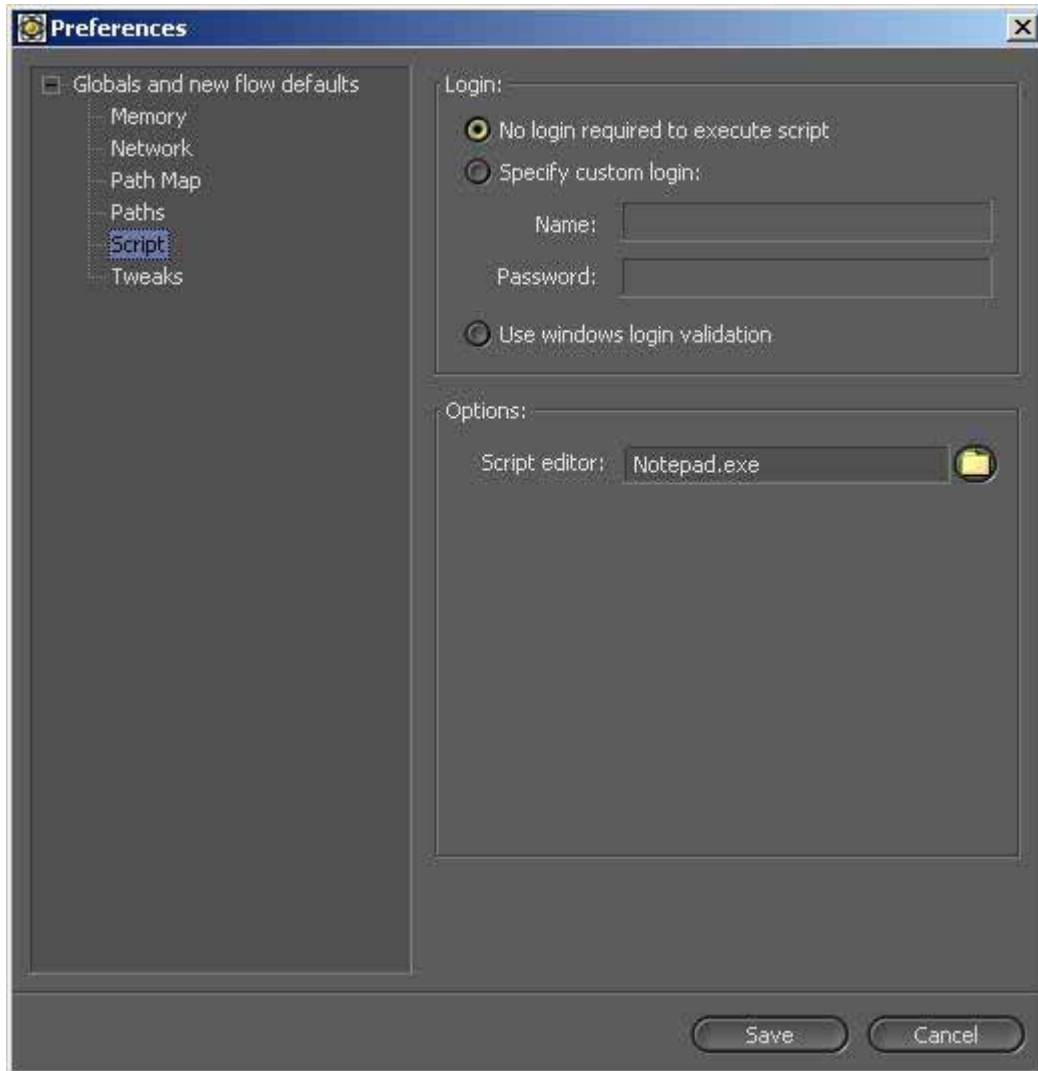
To create a Digital Fusion job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

DigitalFusion
Fusion
DF

Digital Fusion works by executing a script using the DFScript.exe program. By default, Smedge will generate a script for each packet and pass the scene information to that generated script. However, you can override this behavior with your own rendering script. Create a Digital Fusion script file called **FusionRender.dfscript** and put it in the Smedge folder on the rendering client.

Smedge will pass three parameters to the script: The scene file, the start frame and the end frame. The parameters will be passed in that order.

Clients have reported success with Fusion and Smedge if you open up the Fusion security. Specify the **No login required to execute script** option in the preferences:

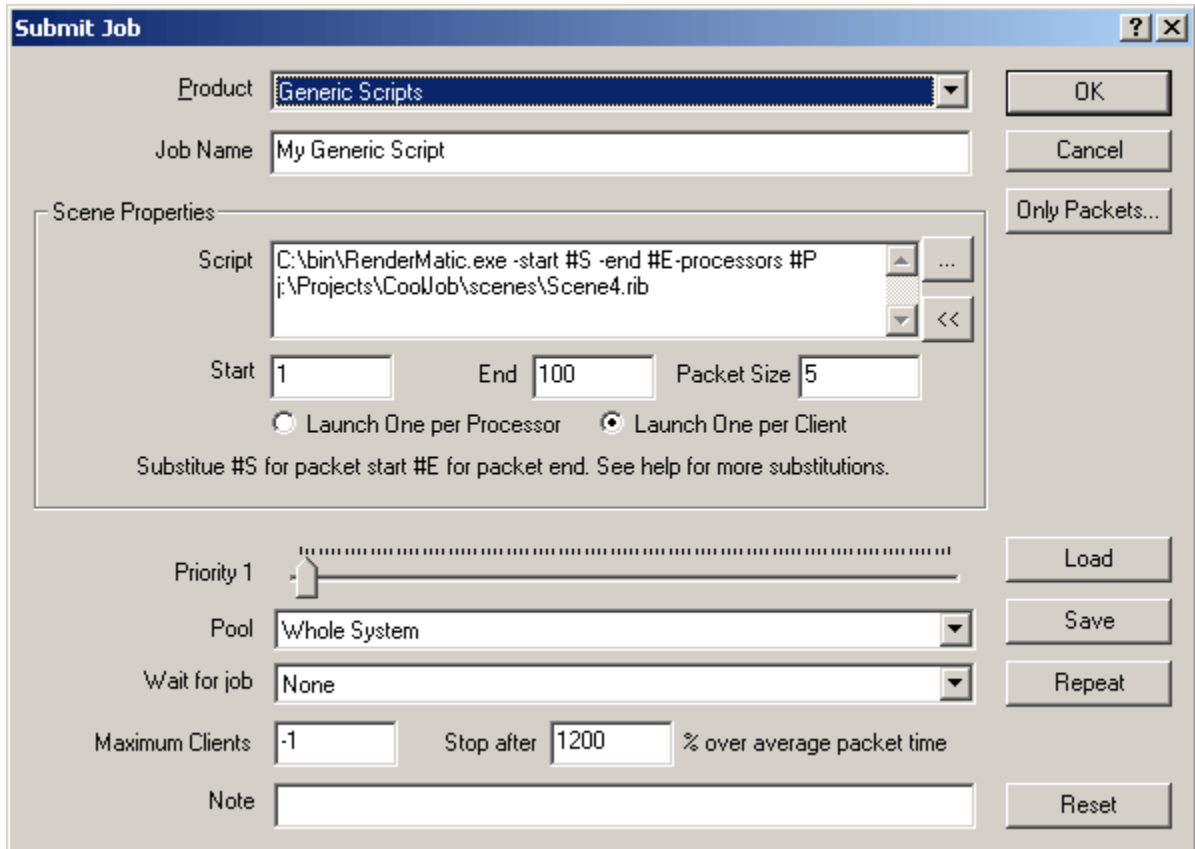


Please note that this will mean looser security on your network. Be sure to adequately protect your network from misuse if you do this.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Generic Scripts



Generic scripting provides the most powerful feature of Smedge.

To create a Generic Script job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Script Generic

Essentially, Smedge is a giant system that does nothing but distribute command lines. As long as the client knows how to interpret the command line, you can distribute any command line based application. Smedge provides some shortcuts that you can put into your commandlines that get replaced in order to make it more useful.

#s and **#0s** Provide a way to give the start frame of the packet to the command line. **#S** is replaced by the frame number, and **#0S** (that's a zero in front of it) is replaced by a

four digit padded version of the frame number.

- #E** and **#0E** Provide a way to give the end frame of the packet to the command line. **#E** is replaced by the frame number, and **#0E** (that's a zero in front of it) is replaced by a four digit padded version of the frame number.
- #B** and **#0B** Provide a way to give the packet size to the command line. **#B** is replaced by the packet size, and **#0B** (that's a zero in front of it) is replaced by a four digit padded version of the packet size.
- #A** and **#0A** Provide a way to give the first frame of the job to the command line. **#A** is replaced by the first frame. **#0A** (that's a zero in front of it) is replaced by a four digit padded version of the first frame.
- #Z** and **#0Z** Provide a way to give the last frame of the job to the command line. **#Z** is replaced by the last frame. **#0Z** (that's a zero in front of it) is replaced by a four digit padded version of the last frame.
- #P** Provides a way to give the number of processors Smedge is assigning to this task to the command line. Generic scripts still allow you to set the job to either single or multi-threaded. However, since Smedge doesn't know anything about the program being called, it is up to you to make sure that you set up both your command line and your Smedge job to optimize processor usage. Essentially, if a job is set to single threaded, this value will be one, but if a job is set to multi-threaded, this value will be the number of processors available to Smedge on the client.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Houdini

The screenshot shows the 'Submit Job' dialog box with the following fields and controls:

- Product:** Houdini (selected in a dropdown menu)
- Job Name:** TestScene.ifd
- Scene Properties:**
 - .IFD File:** C:\Projects\TestProject\scenes\TestScene.ifd
 - Start Directory:** (empty)
 - Start:** 1
 - End:** 50
 - Packet Size:** 5
 - Launch Single-Threaded:** (selected radio button)
 - Launch Multi-Threaded:** (unselected radio button)
 - Additional Command Line Parameters:** (empty)
- Priority 1:** (slider set to 1)
- Pool:** Whole System
- Wait for job:** None
- Maximum Clients:** -1
- Stop after:** 1200 % over average packet time
- Note:** (empty text box)

Buttons on the right side of the dialog include: OK, Cancel, Only Packets..., Set Parameters, Load, Save, Repeat, and Reset.

Use this to submit a Houdini job

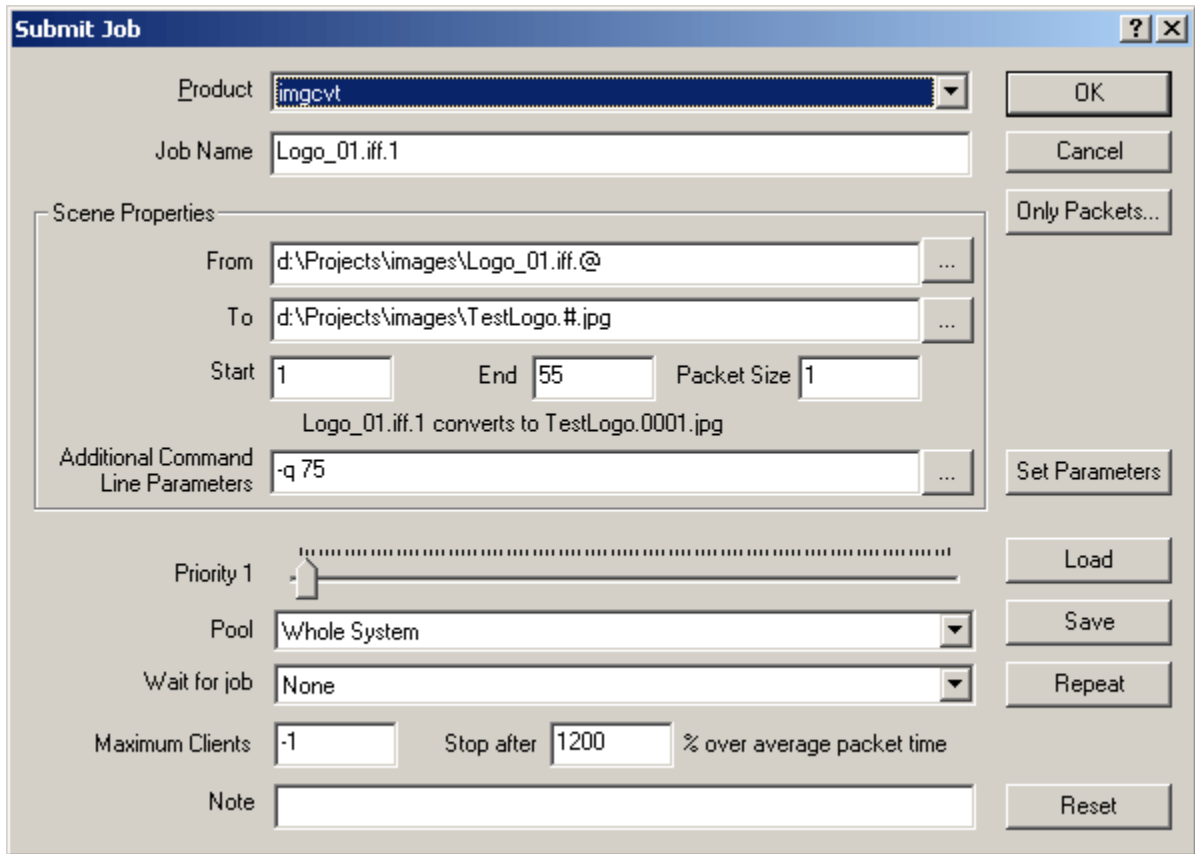
To create a Houdini job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Houdini
VMantra

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

imgcvt



This generates a job specifically for `imgcvt`.

To create an `imgcvt` job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

imgcvt

The `imgcvt` page is a little different from the other renderers. Instead of typing in an exact filename, you need to type in the exact parameters for the `imgcvt` program just as if you were running the program from a command line. You can type any formatting characters that `imgcvt` accepts. Smedge will attempt to provide an example of the change, however, this example may not necessarily display correctly.

Smedge uses the project for the destination parameter. This means command line operation is slightly different for `imgcvt`. For example:

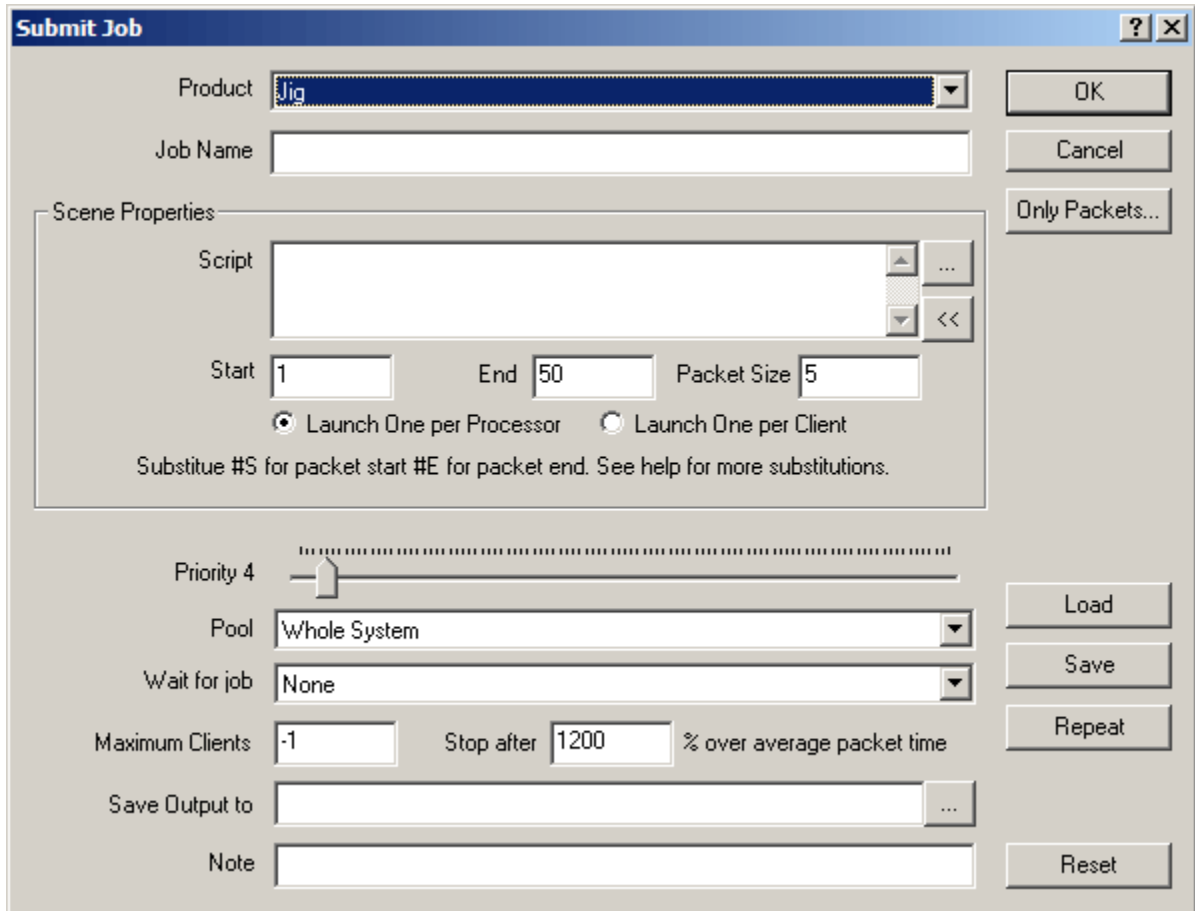
```
Smedge -product imgcvt -j "D:\images\source.iff.@" -proj  
"D:\images\destination.#.tga" -n 1 20 1
```

Some of `imgcvt`'s tricks cannot be performed with Smedge. Most notably, you are not going to be able to reverse sequences. Adding to a frame range should work just fine, though it will not display correctly in the preview on the Submit Job dialog box.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Jig



Jig is a copy of the Generic Script interface, and has the same features and abilities. It was created this way because the user that requested it needed this flexibility. Be sure to type the entire commandline you want to render just as if you were using the [Generic Script](#) render type.

To create a Jig job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Jig

Essentially, Smedge is a giant system that does nothing but distribute command lines. As long as the client knows how to interpret the command line, you can distribute any

command line based application. Smedge provides some shortcuts that you can put into your commandlines that get replaced in order to make it more useful.

#S and **#0S** Provide a way to give the start frame of the packet to the command line. **#S** is replaced by the frame number, and **#0S** (that's a zero in front of it) is replaced by a four digit padded version of the frame number.

#E and **#0E** Provide a way to give the end frame of the packet to the command line. **#E** is replaced by the frame number, and **#0E** (that's a zero in front of it) is replaced by a four digit padded version of the frame number.

#B and **#0B** Provide a way to give the packet size to the command line. **#B** is replaced by the packet size, and **#0B** (that's a zero in front of it) is replaced by a four digit padded version of the packet size.

#A and **#0A** Provide a way to give the first frame of the job to the command line. **#A** is replaced by the first frame. **#0A** (that's a zero in front of it) is replaced by a four digit padded version of the first frame.

#Z and **#0Z** Provide a way to give the last frame of the job to the command line. **#Z** is replaced by the last frame. **#0Z** (that's a zero in front of it) is replaced by a four digit padded version of the last frame.

#P Provides a way to give the number of processors Smedge is assigning to this task to the command line. Generic scripts still allow you to set the job to either single or multi-threaded. However, since Smedge doesn't know anything about the program being called, it is up to you to make sure that you set up both your command line and your Smedge job to optimize processor usage. Essentially, if a job is set to single threaded, this value will be one, but if a job is set to multi-threaded, this value will be the number of processors available to Smedge on the client.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Lightwave

The screenshot shows the 'Submit Job' dialog box for Lightwave. The 'Product' dropdown is set to 'Lightwave'. The 'Job Name' is 'TestJob: TestScene.lws'. Under 'Scene Properties', the 'Scene name' is 'L:\Projects\TestJob\scenes\TestScene.lws', the 'Content Directory' is 'L:\Projects\TestJob', and the 'Lightwave Config Directory' is 'L:\Lightwave\Config'. The 'Start' field is '1', 'End' is '75', and 'Packet Size' is '3'. The 'Launch One per Client' radio button is selected. The 'Priority' slider is at '1', 'Pool' is 'Whole System', and 'Wait for job' is 'None'. The 'Maximum Clients' is '-1' and 'Stop after' is '1200 % over average packet time'. The 'Note' field is empty. Buttons on the right include 'OK', 'Cancel', 'Only Packets...', 'Set Parameters', 'Load', 'Save', 'Repeat', and 'Reset'.

This generates a job specifically for Lightwave.

To create a Lightwave job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Lightwave **LW**

Lightwave requires a third parameter, the Config directory, which Smedge allows you to specify where you would normally specify additional command line parameters. This means you cannot use any other commandline parameters. This limitation is in the design of Smedge, and will be changed for Smedge 3.x's modular design.

You **must** specify the scene, content directory and config directory for every scene.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

3ds max (6.0+)

The screenshot shows the 'Submit Job' dialog box with the following settings:

- Product: 3ds max (6.0+)
- Job Name: TestScene.max
- Scene name: c:\Projects\MjJob\Scenes\TestScene.max
- Start: 1, End: 200, Packet Size: 5
- Additional Command Line Parameters: (empty)
- Priority: 50
- Pool: Whole System
- Wait for job: None
- Maximum Clients: -1, Stop after: 1200 % over average packet time
- Note: (empty)

3ds max, version 6 or later, has its own commandline rendering system, which Smedge uses directly. Use this module to create jobs for this version of max.

To create a 3ds max (6.0+) job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Max
3dsMax
3ds

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

3ds max (via SFRender)

The screenshot shows the 'Submit Job' dialog box with the following settings:

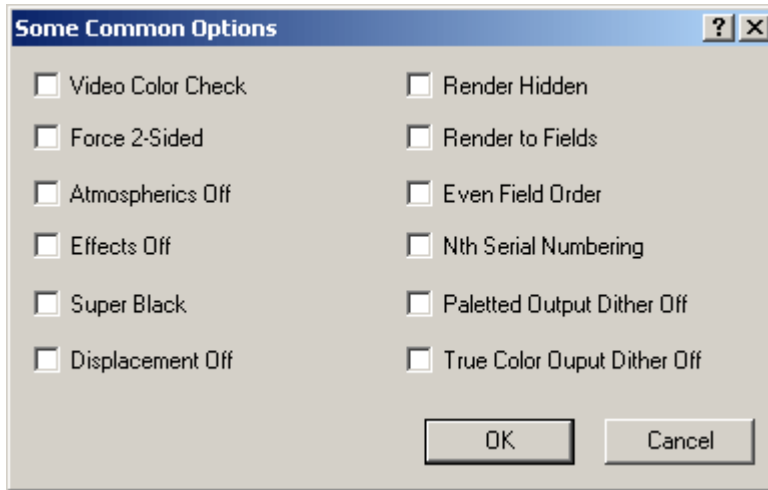
- Product: 3ds max (via SFRender)
- Job Name: TestScene.max
- Scene name: c:\Projects\MyJob\Scenes\TestScene.max
- Start: 1, End: 200, Packet Size: 5
- Additional Command Line Parameters: -xo -xv
- Priority: 4
- Pool: Whole System
- Wait for job: None
- Maximum Clients: -1, Stop after: 1200 % over average packet time
- Note: (empty)

3ds max, version 5 or earlier, is controlled via a third party commandline rendering controller written by [SplutterFish](#). You must install this program in your 3ds max folder. More details of its installation and usage are included in the distribution package, and can also be viewed by [clicking here](#). [Click here](#) to download the commandline program.

To create a 3ds max (via SFRender) job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

SFRender
Max5
Max4
Max3

You can use the Common Render Options dialog for choosing command line parameters for customizing the renderer.



Each option in the dialog alters the default behavior of the max renderer by toggling the option on or off as labeled with each check box. For more information about what each option does, please see the max and the SFRender documentation.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Maya (6.0+)

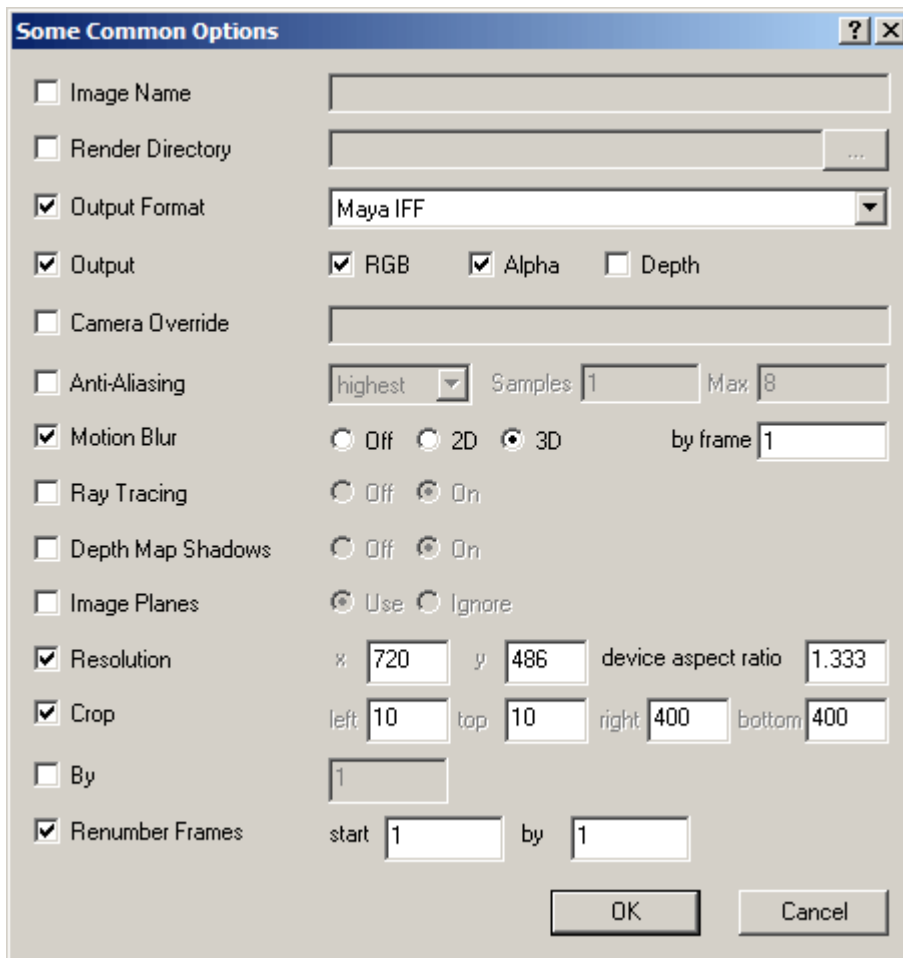
The screenshot shows the 'Submit Job' dialog box for Maya (6.0+). The 'Product' dropdown is set to 'Maya (6.0+)'. The 'Job Name' field contains 'Ludacris: 3598_v07.mb'. Under 'Scene Properties', the 'Scene name' is 'D:\fx\Ludacris\scenes\3589\3589_v07.mb', 'Project' is 'D:\fx\Ludacris', 'Start' is '1', 'End' is '73', and 'Packet Size' is '1'. The 'Launch Multi-Threaded' option is selected. The 'Additional Command Line Parameters' field is empty. The 'Priority' slider is set to 30, 'Pool' is 'Whole System', and 'Wait for job' is 'None'. 'Maximum Clients' is '-1' and 'Stop after' is '1200 % over average packet time'. The 'Note' field is empty. Buttons on the right include 'OK', 'Cancel', 'Only Packets...', 'Set Parameters', 'Load', 'Save', 'Repeat', and 'Reset'.

This generates a job specifically for Maya versions from 6.0 onwards. To render with a previous version of Maya (1.0 to 5.0), use the [Maya \(5.0 or earlier\)](#) Product.

To create a Maya job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Maya

With a Maya (6.0+) job, you can use the Common Render Options dialog for Maya.



Checking on a box on the left of the window will add the appropriate command line switches with the informatin from the right side of that box. A wide variety of options are available, but if you want to add other commandline parameters, you are free to do so by hand in the **Extra Parameters** field of the **Submit Job** dialog. See [Extra Parameters](#) in the [Jobs](#) manual page for more information.

Please note that there is a bug in Maya 6.0 that stops Smedge from capturing the output. The output capture is vital for Smedge to determine the rendered frame files, and to verify successful completion of a packet. This bug has been fixed in Maya 6.0.1. Be sure to upgrade to the latest version to use Smedge correctly.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Maya Hardware Renderer (6.0+)

The screenshot shows the 'Submit Job' dialog box for the Maya Hardware Renderer (6.0+). The 'Product' dropdown is set to 'Maya Hardware Renderer (6.0+)'. The 'Job Name' is 'Ludacris: 3598_v07.mb'. Under 'Scene Properties', the 'Scene name' is 'd:\fx\Ludacris\scenes\3598\3598_v07.mb', 'Project' is 'D:\fx\Ludacris', 'Start' is '1', 'End' is '73', and 'Packet Size' is '1'. The 'Additional Command Line Parameters' field is empty. The 'Priority' is set to 30, 'Pool' is 'Whole System', and 'Wait for job' is 'None'. 'Maximum Clients' is '-1' and 'Stop after' is '1200 % over average packet time'. There is a 'Note' field at the bottom. On the right side, there are buttons for 'OK', 'Cancel', 'Only Packets...', 'Set Parameters', 'Load', 'Save', 'Repeat', and 'Reset'.

This submits a job for the Maya Hardware Renderer that comes with Maya versions 6.0 or later. Smedge cannot control the hardware renderer built into Maya 5.0.

To create a Maya Hardware Renderer (6.0+) job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

MayaHardware
MHardware
Hardware
hw

Please note that there the Maya 6.0 Hardware Renderer does not output the necessary text for Smedge's automatic parsing.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Maya Vector Renderer (6.0+)

The screenshot shows the 'Submit Job' dialog box for the Maya Vector Renderer (6.0+). The 'Product' dropdown is set to 'Maya Vector Renderer (6.0+)'. The 'Job Name' field contains 'Ludacris: 3598_v07.mb'. Under 'Scene Properties', the 'Scene name' is 'd:\fx\Ludacris\scenes\3598\3598_v07.mb', 'Project' is 'D:\fx\Ludacris', 'Start' is '1', 'End' is '73', and 'Packet Size' is '1'. The 'Additional Command Line Parameters' field is empty. The 'Priority' is set to 30, 'Pool' is 'Whole System', and 'Wait for job' is 'None'. 'Maximum Clients' is '-1' and 'Stop after' is '1200 % over average packet time'. There is a 'Note' field at the bottom. On the right side, there are buttons for 'OK', 'Cancel', 'Only Packets...', 'Set Parameters', 'Load', 'Save', 'Repeat', and 'Reset'.

This submits a job for the Maya Vector Renderer that comes with Maya versions 6.0 or later. To submit a job for the Maya Vector Renderer that comes with Maya 5.0, use the [Maya Vector Renderer \(5.0\)](#) product.

To create a Maya Vector Renderer (6.0+) job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

MayaVector
MVector
Vector
mv

Please note that there the Maya 6.0 Vector Renderer does not output the necessary text for Smedge's automatic parsing.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Maya (5.0 or earlier)

The screenshot shows the 'Submit Job' dialog box with the following settings:

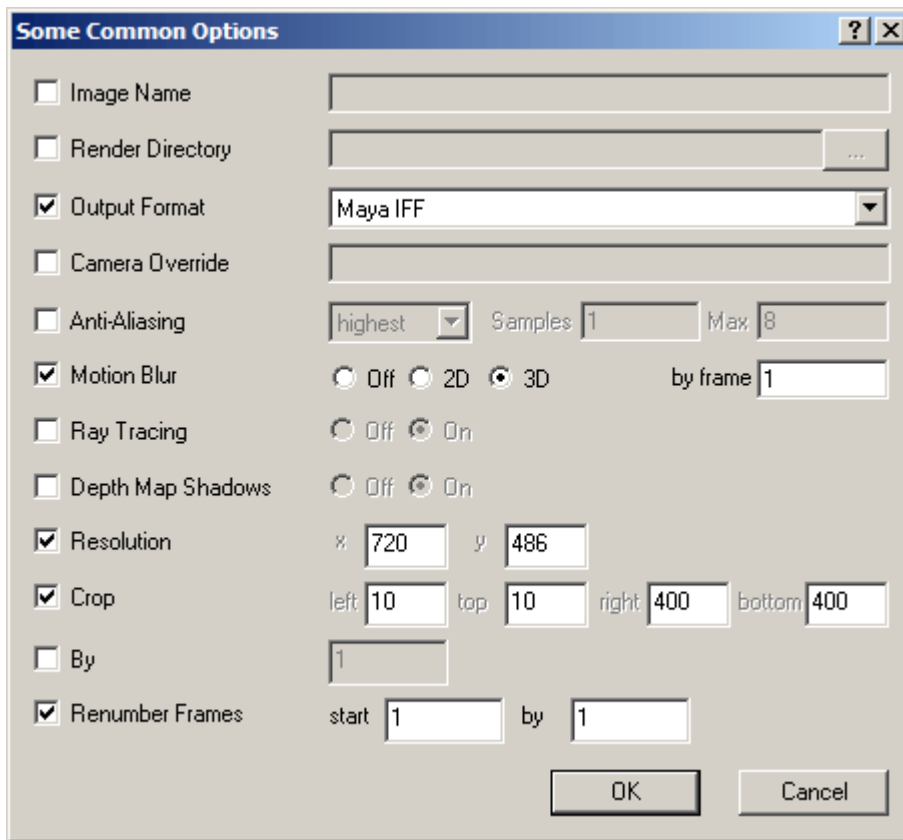
- Product:** Maya (5.0 or earlier)
- Job Name:** Ludacris: 3598_v07.mb
- Scene Properties:**
 - Scene name: d:\fx\Ludacris\scenes\3598\3598_v07.mb
 - Project: D:\fx\Ludacris
 - Start: 1, End: 73, Packet Size: 1
 - Launch One per Processor (unselected), Launch One per Client (selected)
- Additional Command Line Parameters:** (empty)
- Priority:** 30
- Pool:** Whole System
- Wait for job:** None
- Maximum Clients:** -1, **Stop after:** 1200 % over average packet time
- Note:** (empty)

This generates a job specifically for Maya versions from 1.0 to 5.0. To render with Maya 6.0 or later, use the [Maya \(6.0+\)](#) Product.

To create a Maya job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Maya5

With a Maya job, you can use the Common Render Options dialog for Maya.



Checking on a box on the left of the window will add the appropriate command line switches with the information from the right side of that box. A wide variety of options are available, but if you want to add other commandline parameters, you are free to do so by hand in the **Extra Parameters** field of the **Submit Job** dialog. See [Extra Parameters](#) in the [Jobs](#) manual page for more information.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

MayaMan

Submit Job

Product: MayaMan

Job Name: Products: Coke_Box_v22.mb

Scene Properties

Scene name: D:\fx\Products\scenes\Coke_Box_v22.mb

Project: D:\fx\Products

Start: 1 End: 240 Packet Size: 5

Launch One per Processor Launch One per Client

Additional Command Line Parameters: -resolution 720 486 -ribstyle delayed -binaryrib -nodel

Priority 1

Pool: Whole System

Wait for job: None

Maximum Clients: -1 Stop after: 1200 % over average packet time

Note:

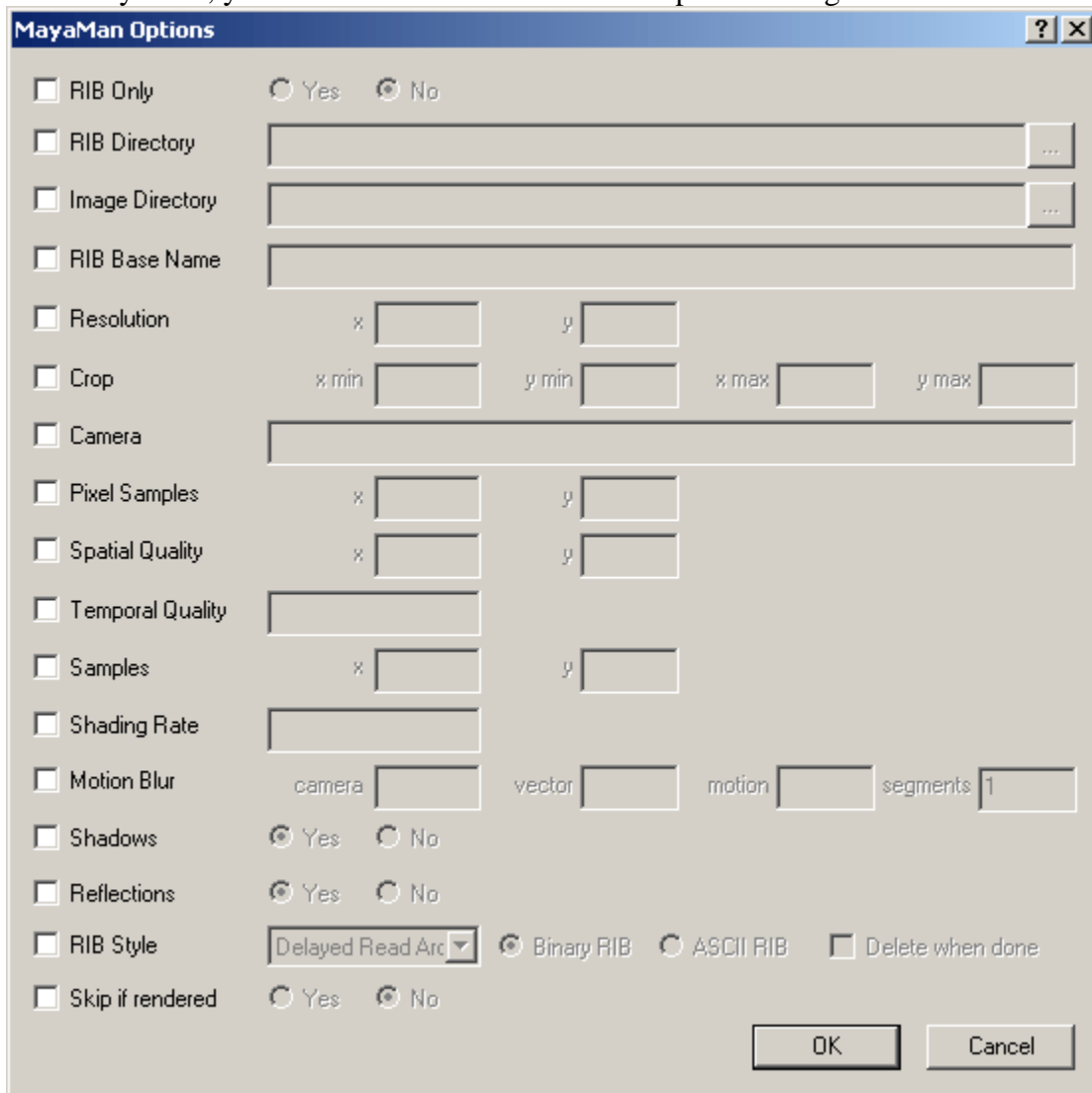
Buttons: OK, Cancel, Only Packets..., Set Parameters, Load, Save, Repeat, Reset

This generates a job specifically for [MayaMan](#).

To create a MayaMan job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Mayaman
MMan
MM

With MayaMan, you can use the Common Render Options dialog box.



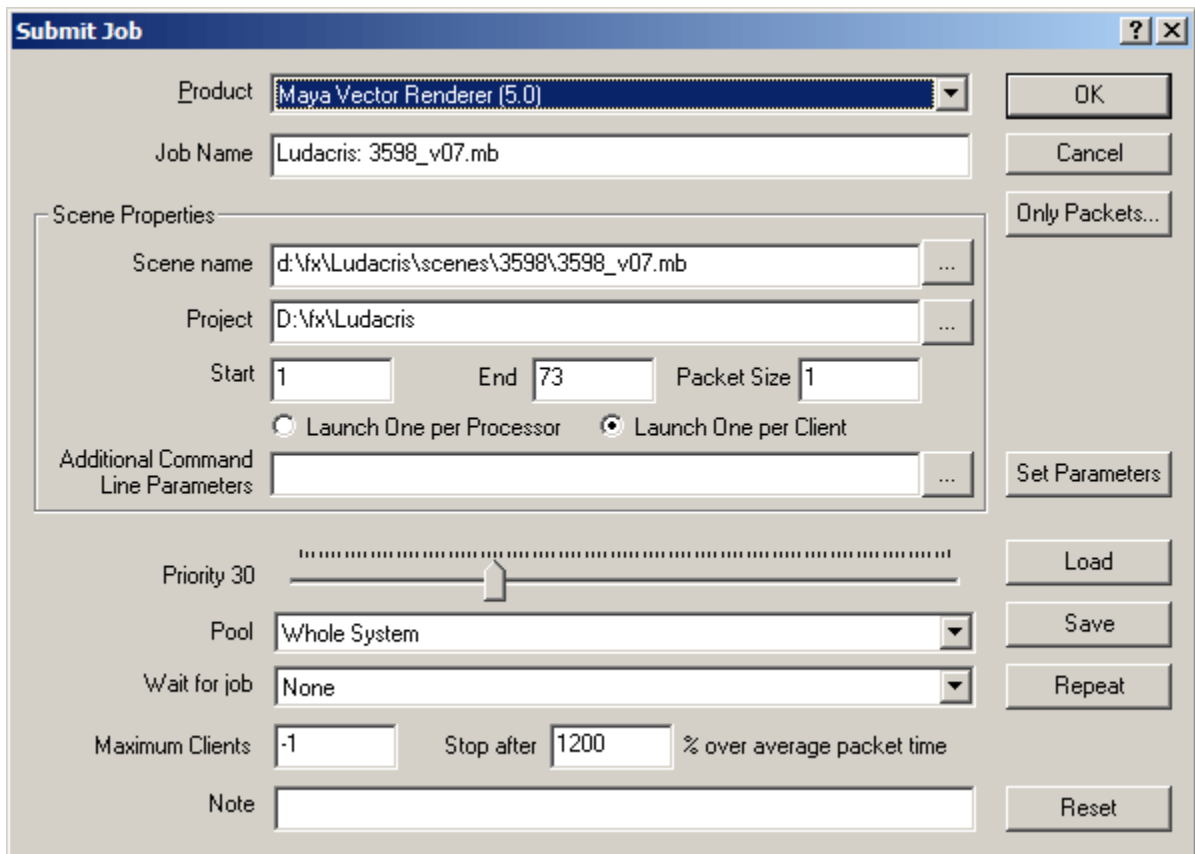
The image shows a software dialog box titled "MayaMan Options". It contains various settings for rendering, organized into a list of checkboxes on the left and corresponding controls on the right. The controls include radio buttons for Yes/No, text input fields, and a dropdown menu. At the bottom right, there are "OK" and "Cancel" buttons.

Option	Control
<input type="checkbox"/> RIB Only	<input type="radio"/> Yes <input checked="" type="radio"/> No
<input type="checkbox"/> RIB Directory	Text field with browse button (...)
<input type="checkbox"/> Image Directory	Text field with browse button (...)
<input type="checkbox"/> RIB Base Name	Text field
<input type="checkbox"/> Resolution	x: [] y: []
<input type="checkbox"/> Crop	x.min: [] y.min: [] x.max: [] y.max: []
<input type="checkbox"/> Camera	Text field
<input type="checkbox"/> Pixel Samples	x: [] y: []
<input type="checkbox"/> Spatial Quality	x: [] y: []
<input type="checkbox"/> Temporal Quality	Text field
<input type="checkbox"/> Samples	x: [] y: []
<input type="checkbox"/> Shading Rate	Text field
<input type="checkbox"/> Motion Blur	camera: [] vector: [] motion: [] segments: [1]
<input type="checkbox"/> Shadows	<input checked="" type="radio"/> Yes <input type="radio"/> No
<input type="checkbox"/> Reflections	<input checked="" type="radio"/> Yes <input type="radio"/> No
<input type="checkbox"/> RIB Style	Delayed Read Arc [v] <input checked="" type="radio"/> Binary RIB <input type="radio"/> ASCII RIB <input type="checkbox"/> Delete when done
<input type="checkbox"/> Skip if rendered	<input type="radio"/> Yes <input checked="" type="radio"/> No

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Maya Vector Renderer (5.0)



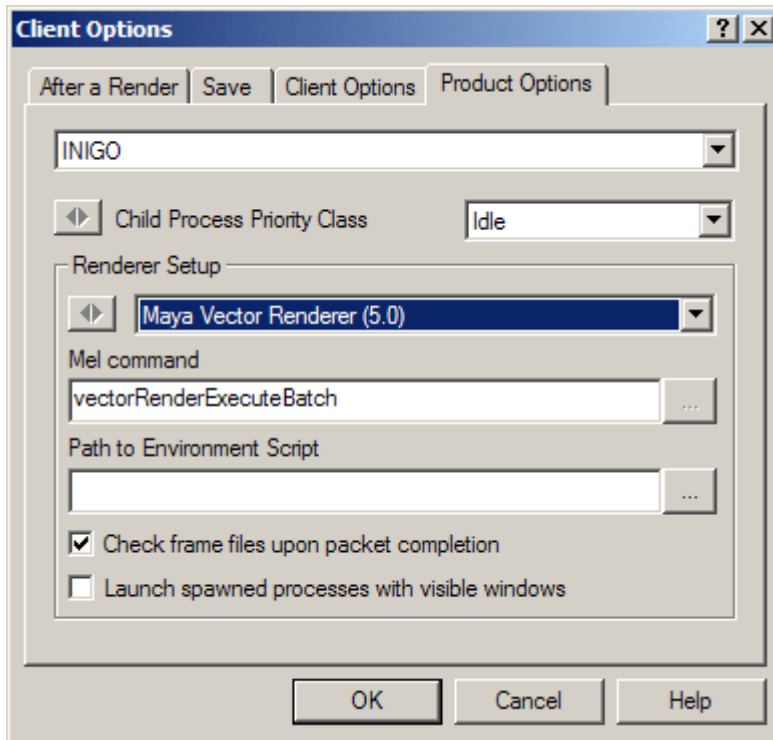
This submits a job for the Maya Vector Renderer that came with Maya version 5.0. To submit a job for the Maya Vector Renderer that works with Maya version 6.0 or later, use the [Maya Vector renderer \(6.0+\)](#) product.

To create a Maya Vector Renderer (5.0 or earlier) job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

MayaVector5
MVector5
Vector5
mv5

The Maya Vector Renderer in Maya version 5.0 works slightly differently than most of the other products. There is no direct commandline program that executes the render, but

instead it is launched by sending some MEL commands directly to the mayabatch program. This has implications for the configuration of your rendering clients. Specifically, Smedge uses the executable file path from the standard Maya render type, substituting the `Render.exe` program file with the `mayabatch.exe` program file. In the **Smedge Options** dialog box, with the MentalRay for Maya program selected, the text displayed in the **Path to Executable** field is actually the MEL command that is used to start the render.



You should probably not change the default, unless you have customized the Maya 5.0 vector renderer or written your own render command. Please note: This is **not** a script file or batch file that comes with the MentalRay for Maya program. This should be the actual MEL command to start the render, picked from the following table. **Do not use the browse button**, because you cannot browse to this command. If you have a path to any script or executable, the render will not work. If you do not type one of the following MEL commands into this field, the render will not work.

Please note also that the **Extra Commandline Parameters** field works differently. These are additional MEL commands that will be inserted just before the call to actually start the render. If you use this field, be sure to use it with caution, and have a good understanding of MEL.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

MentalRay for Maya (6.0+)

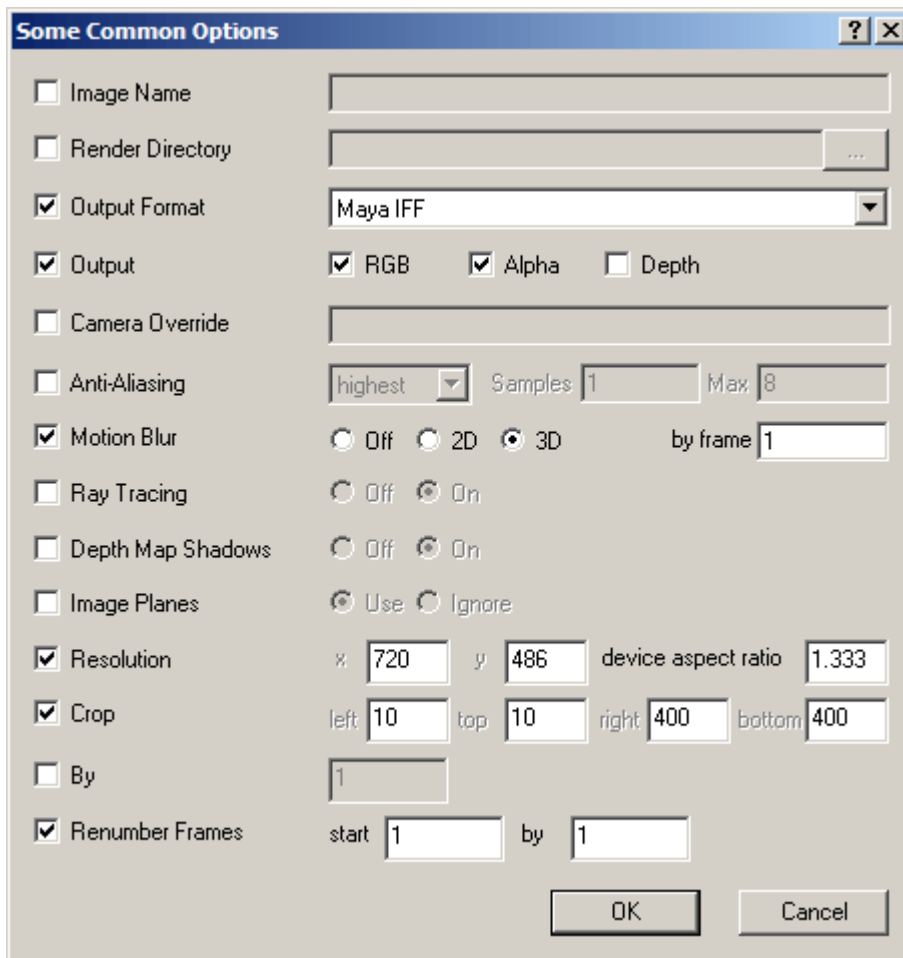
The screenshot shows the 'Submit Job' dialog box for MentalRay for Maya (6.0+). The 'Product' dropdown is set to 'MentalRay for Maya (6.0+)'. The 'Job Name' field contains 'Ludacris: 3598_v07.mb'. Under 'Scene Properties', the 'Scene name' is 'd:\fx\Ludacris\scenes\3598\3598_v07.mb', 'Project' is 'D:\fx\Ludacris', 'Start' is '1', 'End' is '73', and 'Packet Size' is '1'. The 'Launch One per Client' radio button is selected. The 'Additional Command Line Parameters' field is empty. The 'Priority' slider is set to 30, 'Pool' is 'Whole System', and 'Wait for job' is 'None'. 'Maximum Clients' is '-1' and 'Stop after' is '1200 % over average packet time'. The 'Note' field is empty. Buttons for 'OK', 'Cancel', 'Only Packets...', 'Set Parameters', 'Load', 'Save', 'Repeat', and 'Reset' are visible on the right side.

This submits a job for MentalRay for Maya that works with Maya version 6.0 or later. To submit a job for MentalRay for Maya that works with earlier versions of Maya (4.5 or 5.0), use the [MentalRay for Maya \(5.0 or earlier\)](#) product.

To create a MentalRay for Maya job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

MentalRayForMaya
MentalRay4Maya
MR4M
MMR

With a MentalRay for Maya (6.0+) job, you can use the Common Render Options dialog for Maya.



Checking on a box on the left of the window will add the appropriate command line switches with the information from the right side of that box. A wide variety of options are available, but if you want to add other commandline parameters, you are free to do so by hand in the **Extra Parameters** field of the **Submit Job** dialog. See [Extra Parameters](#) in the [Jobs](#) manual page for more information.

Please note that in Maya 6, the frame renumbering flag **-rfs** which is supposed to change the output frame number does not work. I have contacted Alias about this, but have no further information at this time. This bug is not fixed in 6.0.1.

Please note that there is a bug in Maya 6.0 that stops Smedge from capturing the output. The output capture is vital for Smedge to determine the rendered frame files, and to verify successful completion of a packet. This bug was fixed in Maya 6.0.1, so you should download the fix from Alias.

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

MentalRay for Maya (6.0+)

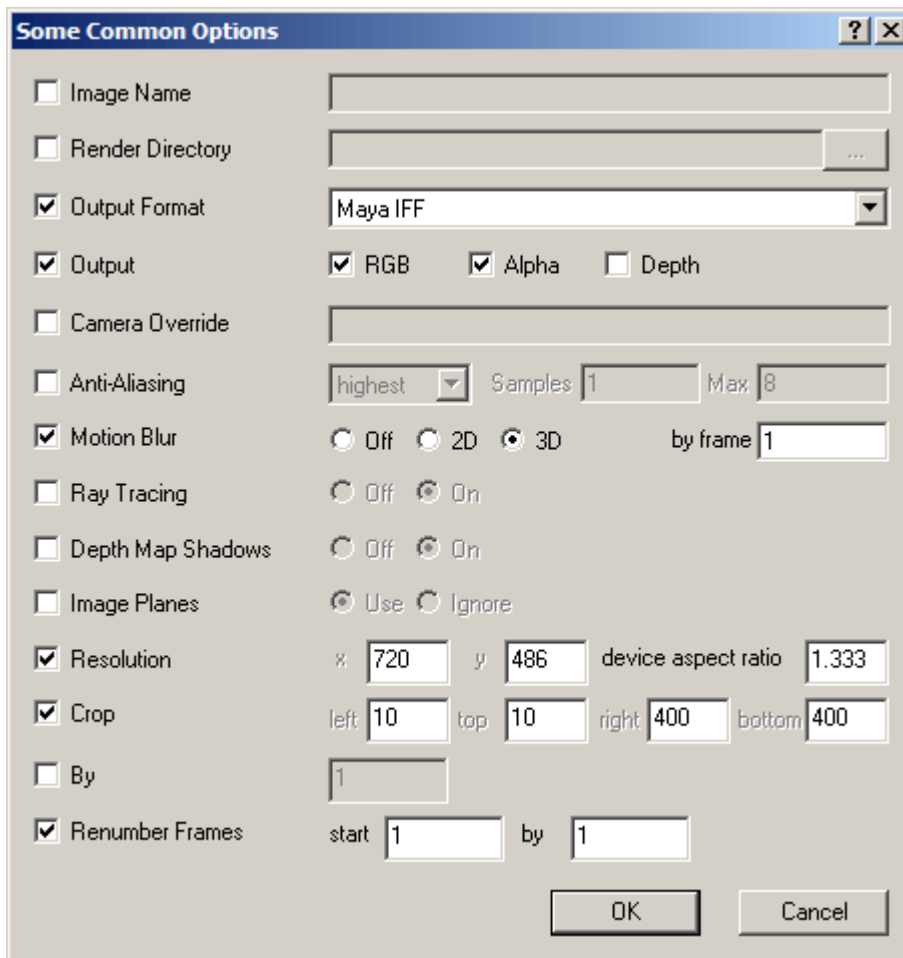
The screenshot shows the 'Submit Job' dialog box for MentalRay for Maya (6.0+). The 'Product' dropdown is set to 'MentalRay for Maya (6.0+)'. The 'Job Name' field contains 'Ludacris: 3598_v07.mb'. Under 'Scene Properties', the 'Scene name' is 'd:\fx\Ludacris\scenes\3598\3598_v07.mb', 'Project' is 'D:\fx\Ludacris', 'Start' is '1', 'End' is '73', and 'Packet Size' is '1'. The 'Launch One per Client' radio button is selected. The 'Additional Command Line Parameters' field is empty. The 'Priority' slider is set to 30, 'Pool' is 'Whole System', and 'Wait for job' is 'None'. 'Maximum Clients' is '-1' and 'Stop after' is '1200 % over average packet time'. The 'Note' field is empty. Buttons for 'OK', 'Cancel', 'Only Packets...', 'Set Parameters', 'Load', 'Save', 'Repeat', and 'Reset' are visible on the right side.

This submits a job for MentalRay for Maya that works with Maya version 6.0 or later. To submit a job for MentalRay for Maya that works with earlier versions of Maya (4.5 or 5.0), use the [MentalRay for Maya \(5.0 or earlier\)](#) product.

To create a MentalRay for Maya job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

MentalRayForMaya
MentalRay4Maya
MR4M
MMR

With a MentalRay for Maya (6.0+) job, you can use the Common Render Options dialog for Maya.



Checking on a box on the left of the window will add the appropriate command line switches with the information from the right side of that box. A wide variety of options are available, but if you want to add other commandline parameters, you are free to do so by hand in the **Extra Parameters** field of the **Submit Job** dialog. See [Extra Parameters](#) in the [Jobs](#) manual page for more information.

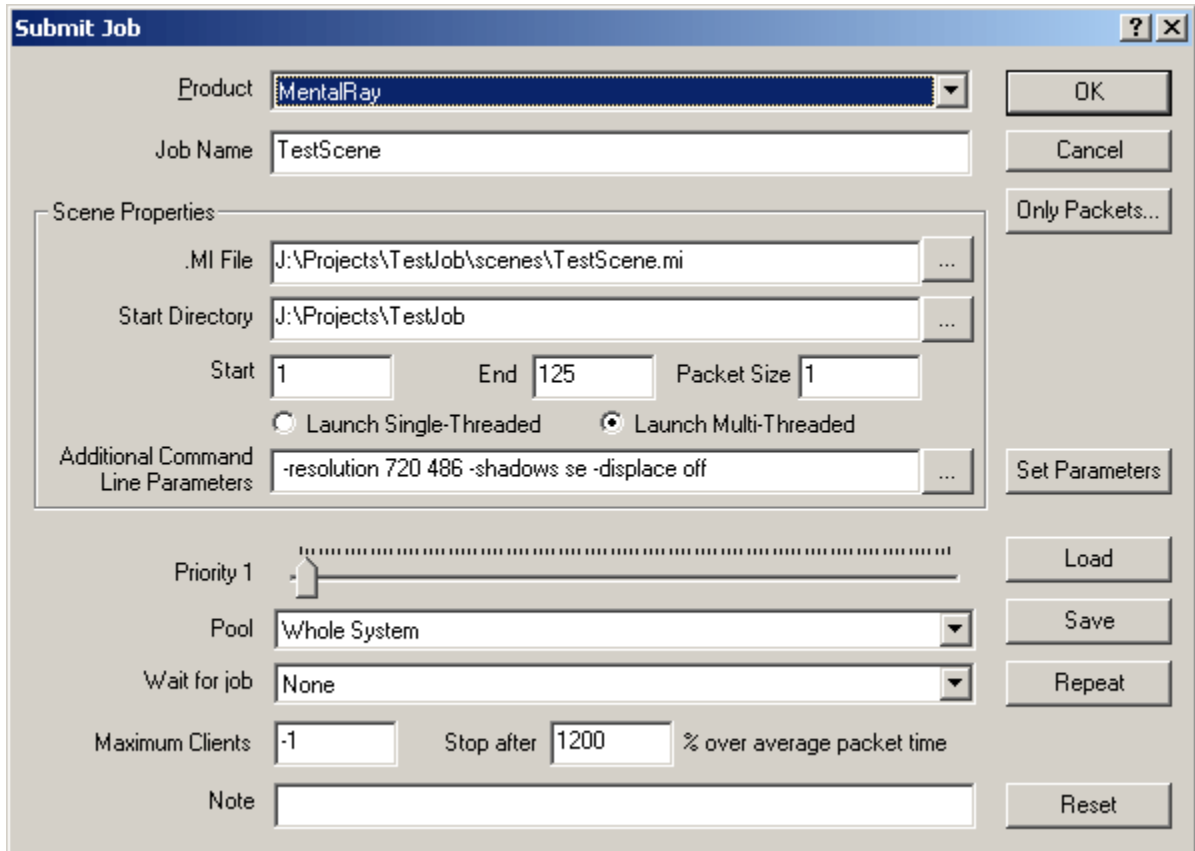
Please note that in Maya 6, the frame renumbering flag **-rfs** which is supposed to change the output frame number does not work. I have contacted Alias about this, but have no further information at this time. This bug is not fixed in 6.0.1.

Please note that there is a bug in Maya 6.0 that stops Smedge from capturing the output. The output capture is vital for Smedge to determine the rendered frame files, and to verify successful completion of a packet. This bug was fixed in Maya 6.0.1, so you should download the fix from Alias.

Products Available To Render

3ds max (6.0+)	Alias Renderer.	Lightwave	MentalRay.
3ds max (via SFRender)	Chalice.	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects.	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air.	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster.	Houdini.	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer.	imgcvt.	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer.	Jig.	MayaMan	XSI

MentalRay

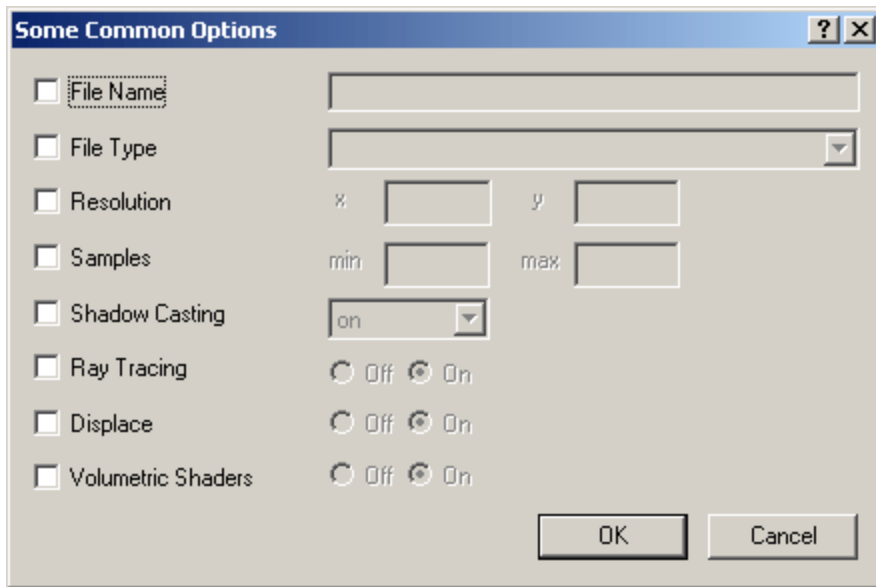


This generates a job specifically for MentalRay.

To create a MentalRay job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

MentalRay
Ray
MR

With MentalRay, you can use the Common Render Options dialog box.



- | | |
|----------------------|---|
| File Name | Will add the -file_name switch and the text that you type for the rendered images' filenames |
| File Type | Will add the -file_type switch and the text that you type for the rendered images' file type. You can chose from a list of the file types that MentalRay supports. |
| Resolution | Will add the -resolution switch and the numbers you type into the fields for the X and Y pixel sizes of the rendered images. |
| Samples | Will add the -samples switch and the numbers for the minimum and maximum sample sizes for the renderer to use |
| Shadow Casting | Will add the -shadows switch and the option you select from the list. You can select "on" "off" "sort" or "selected". |
| Ray Tracing | Will add the -trace switch with the appropriate option to turn on or off ray tracing. |
| Displace | Will add the -displace switch with the appropriate option to turn on or off displacement rendering. |
| Volumetric Shaders | Will add the -volume switch with the appropriate option to turn on or off the rendering of volumetric shaders. |

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Rayz

The screenshot shows the 'Submit Job' dialog box for Rayz. The 'Product' dropdown is set to 'Rayz'. The 'Job Name' is 'RayzTest_01.rayz'. Under 'Scene Properties', the 'Scene name' is 'D:\fx\Products\scenes\RayzTest_01.rayz' and the 'Start Directory' is 'D:\fx\Products\images'. The 'Start' field is '1', 'End' is '110', and 'Packet Size' is '5'. The 'Launch Single-Threaded' radio button is selected. The 'Additional Command Line Parameters' field is empty. On the right side, there are buttons for 'OK', 'Cancel', 'Only Packets...', 'Set Parameters', 'Load', 'Save', 'Repeat', and 'Reset'. At the bottom, there is a 'Priority 1' slider, a 'Pool' dropdown set to 'Whole System', a 'Wait for job' dropdown set to 'None', 'Maximum Clients' set to '-1', and 'Stop after' set to '1200' % over average packet time. A 'Note' field is also present.

This generates a job specifically for **Rayz**.

To create a Rayz job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Rayz

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

Shake

Submit Job

Product: Shake

Job Name: TestComp.shk

Scene Properties:

Scene name: D:\fx\Products\scenes\TestComp.shk

Start Directory: D:\fx\Products

Start: 1 End: 110 Packet Size: 10

Launch Single-Threaded Launch Multi-Threaded

Additional Command Line Parameters:

Priority 1: [Slider]

Pool: Whole System

Wait for job: None

Maximum Clients: -1 Stop after: 1200 % over average packet time

Note:

Buttons: OK, Cancel, Only Packets..., Set Parameters, Load, Save, Repeat, Reset

This generates a job specifically for **Shake**.

To create a Shake job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

Shake

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

SoftImage

The screenshot shows the 'Submit Job' dialog box with the following settings:

- Product: SoftImage
- Job Name: TestScene
- Scene Properties:
 - Scene name: J:\Projects\TestJob\scenes\TestScene.dsc
 - Project: J:\Projects\TestJob
 - Start: 1, End: 125, Packet Size: 3
 - Launch One per Processor (unselected), Launch One per Client (selected)
 - Additional Command Line Parameters: (empty)
- Priority 1: (slider)
- Pool: Whole System
- Wait for job: None
- Maximum Clients: -1, Stop after: 1200 % over average packet time
- Note: (empty)

Buttons on the right side include: OK, Cancel, Only Packets..., Set Parameters, Load, Save, Repeat, and Reset.

This generates a job specifically for SoftImage.

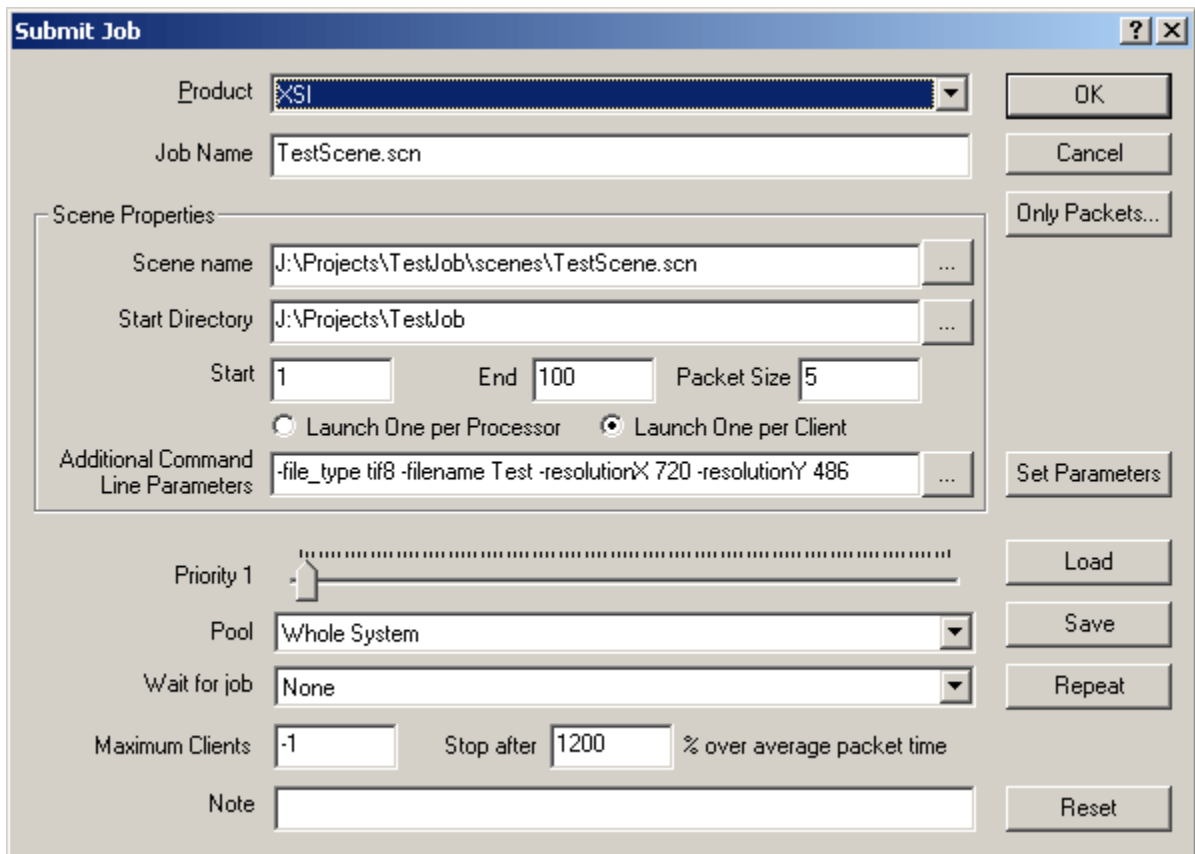
To create a SoftImage job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

SoftImage
Soft
Si

Products Available To Render

3ds max (6.0+)	Alias Renderer	Lightwave	MentalRay
3ds max (via SFRender)	Chalice	Maya(5.0 or earlier)	MentalRay for Maya (5.0 or earlier)
After Effects	Digital Fusion	Maya (6.0+)	MentalRay for Maya (6.0+)
Air	Generic Scripts	Maya Vector Renderer (5.0)	Rayz
Alias PowerCaster	Houdini	Maya Vector Renderer (6.0+)	Shake
Alias PowerTracer	imgcvt	Maya Hardware Renderer (6.0+)	SoftImage
Alias Raytracer	Jig	MayaMan	XSI

XSI

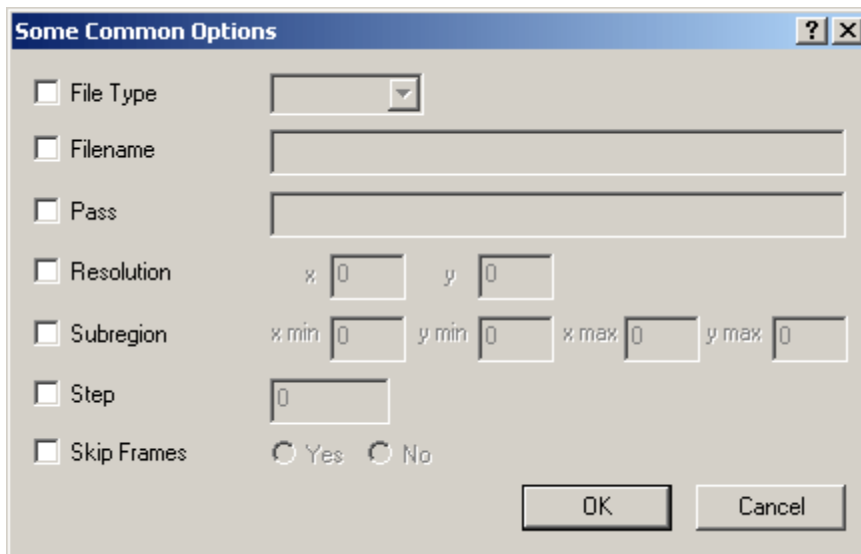


This generates a job specifically for XSI.

To create an XSI job via commandline, you can use the following parameters for the **-product** switch (case is ignored):

XSI

With XSI, you can use the Common Render Options dialog box



- File Type Will add the **-file_type** flag followed by the text you type as the filename. There is a premade list of the supported file types the XSI renderer supports that you can chose from
- Filename Will add the **-filename** switch followed by the text you type as the rendered images' filename.
- Pass Will add the **-pass** switch followed by the text you type as the render pass to render.
- Resolution Will add the **-resolutionX** and **-resolutionY** switches followed by the numbers you enter for the rendered images' pixel dimensions.
- Subregion Will add the **-subregion** switch followed by the dimensions of the subregion you wish to render of the full image
- Step Will add the **-step** switch followed by the number you enter in the text box.
- Skip Frames Will add the **-skip** switch with the appropriate option for allowing you to skip rendered frames

The Queue Menu

- New Job
 - New Batch of Jobs
 - Load Job
 - Save Job
 - Queue Snapshot
 - Show Last 20 Jobs
 - Increase Priority
 - Decrease Priority
 - Pause
 - Set Pool
 - Show Job ID
 - Reset Job Failure Counts
 - Delete Job
 - Stop Packet and Requeue
 - Stop Packet Permanently
 - Stop Packet and Submit as New Job
 - Restart
 - Super Restart
 - Exit
 - Super Exit
-

New Job

Brings up the [Submit Job](#) dialog box, allowing you to submit a new job into the Smedge job queue.

New Batch of Jobs

Brings up the [Submit Batch of Jobs](#) dialog box, allowing you to submit a bunch of new jobs with the same parameters into the Smedge job queue.

Load Job

Allows you to use load a `.smr` file to load a job from disk. The [Submit Job](#) dialog box will be opened with the settings from the job you have selected.

Save Job

Allows you to save a `.smr` file with all of the jobs currently selected in the Job list on the main window. Note that jobs are saved in their original state, so the file will not be aware of any finished or currently executing packet from the selected jobs.

Queue Snapshot

Allows you to save a `.smr` file with every job currently in the queue. Note that the jobs are saved in their original state. This file will not be aware of any finished or currently executing packets for the jobs in the queue. Jobs that are complete will not be included in this file.

Show Last 20 Jobs

Opens up a the window with the last 20 jobs that you have submitted from this system. This allows you to easily bring up a [Submit Job](#) dialog box with all the settings from a previous job.

Increase Priority

Increases the priority of every selected job by 1. The maximum priority for any job is 100.

Decrease Priority

Decreases the priority of every selected job by 1. The minimum priority for any job is 0. If a job is set to priority 0, it will not have any new packets executed.

Pause

Decreases the priority of every selected job to 0. No new packets will be executed from any of these jobs until you increase their priority back above 0.

Set Pool

Select an item from the sub-menu to change the pool that every selected job is assigned to.

Show Job ID

Opens a window with the string form of the code that uniquely identifies this job in the Smedge system. This code can be used for commandline control of jobs, including deleting a job or setting a new job to wait for this job. The window allows for you to copy this code to the clipboard.

Reset Job Failure Counts

Forces Smedge to reset the failure counts for the selected jobs. This is useful if you have had a temporary problem that has caused jobs to fail repeatedly. Smedge normally keeps a count of the failures, and stops letting clients try (and fail) on a job infinitely. If you have fixed the problem, you can reset the failure counts for a job using this command. To reset failure counts by client instead of by job, use the [Reset Client Failure Counts](#) command in the [Host](#) menu.

Delete Job

Immediately removes all the selected jobs from the queue. Any active packets from those jobs will be aborted.

Stop Packet and Requeue

Immediately stops the selected packets from rendering and tells Smedge to re-queue those packets.

Stop Packet Permanently

Immediately stops the selected packets from rendering and tells Smedge to never try to render those packets again.

Stop Packet and Submit as New Job

Immediately stops the selected packets from rendering. The packets are marked as permanently aborted in the job to which they belong, but they are re-queued as new jobs independent of the original job from which they came.

Restart

Tells Smedge to try to restart itself. This is useful when things start to go wrong in heavy network situations.

Super Restart

Tells the entire Smedge network to try to restart itself. **This will stop all renders on your network.** Smedge will try to recover the jobs when it restarts, but any current progress may be lost, Note that clients that have lost their connection to the Smedge master may not receive the Super Restart command from the server, and therefore may not restart correctly.

Exit

Closes the Smedge application on your computer. **This will make your computer unavailable for rendering and unable to control your render queue.** Any renders currently going on your machine will be aborted and re-queued.

Super Exit

Closes the Smedge application on your computer and on every computer that is currently running Smedge on your network. **This will stop all renders on your network and remove all of your queued renders. There is no way to recover the state of your render queue after you select this menu item.**

Host Menu

- Toggle 1 (Local) Processor
 - Toggle (Local) Client Availability
 - Toggle (Local) Time Restricted
 - Pool Manager
 - Edit Local/Selected Client Settings
 - Local/Selected Client Smedge Options
 - Add Client Note
 - Reset Client Failure Counts
 - Send Message
 - Select Frame Viewer
 - Set Job Name Prefix
 - Enter A New License
 - Request A License
 - Prioritize Clients
 - Debug Level
-

Toggle 1 (local) Processor

This toggles the availability of Smedge to use all of the installed processors on a machine. This option is only available on machines with more than one CPU installed.

If you have no clients selected in the Client section of the main window, this menu option will control the local client. If you have one or more clients selected, this option will attempt to perform the action on all of the selected clients.

You can see the state of the computer by looking at the Clients pane of the main window. If the Client has one processor disabled, the icon will be orange, and the "Processors Available" column will show something like "1 of 2." You can also see the state by looking at the toolbar:



All of the client's processors are available to be used



Smedge will leave one of the client's processors available for user use



Note: If you have no clients selected, this menu option will display as **Disable 1 Local Processor**.

Toggle (Local) Client Availability



This toggles the availability of this client to render jobs. This is the same as checking or unchecking the **Enabled to process jobs** check box in the [Client Settings](#) dialog box.

If you have no clients selected in the Client section of the main window, this menu option will control the local client. If you have one or more clients selected, this option will attempt to perform the action on all of the selected clients.

You can see the state of the computer by looking at the Clients pane of the main window. If the Client is disabled, the icon will be red, and the "Status" column will display "Disabled." You can also see the state by looking at the toolbar:

-  This client is enabled to execute renders
-  This client cannot execute renders

or by looking at the System Tray icon:

-  This client is enabled to execute renders
-  This client cannot execute renders



Note: If you have no clients selected, this menu option will display as **Disable Local Client**.

Toggle (Local) Time Restricted



This toggles the time restrictions on this client's availability of this client to render jobs. This is the same as checking or unchecking the **Restrict Rendering Times** check box in the **Client Settings** dialog box. Note that you need to have set up the time and day fields previously in this dialog box in order for the time restrictions to actually restrict anything.

If you have no clients selected in the Client section of the main window, this menu option will control the local client. If you have one or more clients selected, this option will attempt to perform the action on all of the selected clients.

The icon on the toolbar will reflect the state of the check box, regardless of whether that client is actually in a restricted time or not:

-  This client will always be available for rendering
-  This client may be subject to time restrictions based on the settings in the Client Settings dialog

You can see whether the computer is in a restricted time by looking at the Clients pane of the main window. If the Client is disabled, the icon will be magenta, and the "Status" column will display "Restricted." You can also see the state by looking at the tray icon:

-  This client is allowed to execute renders
-  This client cannot execute renders

Note: If you have no clients selected, this menu option will display as **Local Time Restricted**.

Pool Manager

This opens up the Pool Manager window, which allows for the creation and modification of pools. Note that this window is not modal, so you can continue to do work even if it is open.

Edit Selected/Local Client Settings

This opens up the [Client Settings](#) dialog box. If you have no clients selected in the Client pane of the main window, this will open a dialog box for the local client. If you have one or more clients selected in the list, the dialog box will reflect the settings of those clients, and any changes you make will be broadcast to all the clients you have selected.

Local/Selected Client Smedge Options

This opens up the [Smedge Options](#) dialog box. If you have no clients selected in the Client pane of the main window, this will open a dialog box for the local client. If you have one or more clients selected in the list, the dialog box will fill with default options, any changes you make will be broadcast to all the clients you have selected.

Add Client Note

Clients can have a note attached to them, where you can put information you find useful for everyone to be aware of about the client. The note text can be anything you want. Selecting this command will open the Client Note dialog box, with your local client selected in the list. You can select any client you wish in the dialog box, and the note text you type will be attached to the selected client.

Reset Client Failure Counts

Forces Smedge to reset the failure counts for the selected clients. This is useful if you have had a temporary problem that has caused jobs to fail repeatedly. Smedge normally keeps a count of the failures, and stops letting clients try (and fail) on a job infinitely. If you have fixed the problem, you can reset the failure counts for a client using this command. To reset failure counts by job instead of by client, use the [Reset Job Failure Counts](#) command in the [Queue](#) menu.

Send Message

Allows you to send an instant message to any or all Smedge clients currently running on your network. For more information, see [Instant Messages](#).

Select Frame Viewer

This allows you to select which program will be used by Smedge to display the currently rendering frame (for products that support this functionality). Smedge passes the filename to the application in a command line style (e.g., `fcheck.exe "filename.iff"`).

Set Job Name Prefix

This opens a window that allows you to set text that will be prepended to the name of every job that is submitted from this client.

Enter A New License

This is how you can enter the license code that you received either to test Smedge on your network, or if you have purchased it.

Request A License

This brings up a form you can use to request a license code for Smedge on your network. Fill out the dialog box and press "Send" to request your license. If your computer is not directly connected to the Internet, press "View" and copy and paste the contents of the message to an email sent to license@uberware.net.

Prioritize Clients

Tells the server to automatically prioritize clients based on their relative speed. This process takes only about a second, and will change the [Client Priority](#) of every client currently running on your network.

Debug Level

This sets the level of detail for the information that Smedge saves in its debugging log file. This file can be useful for troubleshooting, or for sending to technical support if you are having problems. To see where this file is saved, see the FAQ [Where are details about a job saved when that job finishes?](#)

View Menu

- Frame
 - Check All Frames
 - Job History...
 - View Truncated Output
 - Job Settings...
 - Explore to scene
 - Explore to images
 - View Saved HTML Job History
 - View Saved Captured Output
 - Smedge Globals...
 - Colors...
 - Always On Top
 - Hide On Minimize
 - Start Hidden
 - Toolbar
 - Status Bar
-

Frame

If you have one or more Active Packets selected, and Smedge has discovered a frame filename from the renderer, this will bring up the frame viewer to look at that frame. The viewer can be set using [Select Frame Viewer](#) from the [Host Menu](#). The default is to use `fcheck`, the image viewing program that comes with Maya.

Check All Frames

This brings up the [Check Frames](#) window. From this window you can view every expected frame file from a job, and look for or even requeue bad or missing frames.

Job History...

If you have one or more Active Packets selected, this will open a window for each packet in which you can watch the output of the renderer. If you have one or more Jobs selected, this will open a Job History window for each selected job.

View Truncated Output

If you have one or more Active Packets selected, this will open a window for each packet in which you can watch the output of the renderer. The difference between this and the View Output command is that any previously captured output will not be sent when you use this command. Instead, the window will only start displaying any output generated from the moment the window connects to the process on. This is useful when a render creates copious amounts of output, and you don't want to use network resources and waste time waiting for the whole previously generated output to download.

Job Settings...

This will open a read-only Submit Job dialog for the job that is selected, or the job to which the selected packet belongs. In this dialog you can view the settings that were used to create the job, but most of these settings cannot be changed. You can still change the Priority, Pool, Waiting for Job, or Note fields of this job. You can also re-submit packets from the job.

Explore to Scene

This will open a Windows Explorer window in the folder with the scene file.

Explore to Images

This will open a Windows Explorer window in the folder with the rendered image files, if known. If Smedge cannot detect this automatically, you can set it in the [Job History](#) window.

View Saved HTML Job History

This will open the Saved History window and look at all available HTML Job History files on this system. These are the files created after a job has finished that contain a full breakdown of the history of the job. Every time a job is executed, a new file is created, based on the scene name. If the scene name has already been used, Smedge appends a number after it. You can sort this list by the name or by the date that the file was last modified. Double clicking one of the files will open the HTML file into a new window for you to browse.

View Saved Captured Output

This will open the Saved History window and look at all available captured output files in the folder Smedge is currently using to save output. Smedge saves the output for a packet into a file based on the scene name and the first frame of the packet. Note that Smedge will always append to this file if it already exists. You can sort this list by the name or by the date that the file was last modified. Double clicking one of the files will open the file into a new window for you to browse. Note that Smedge uses the file semantics for the extension ".log" to launch the file viewer.

Smedge Globals...

Opens the [Smedge Globals](#) window. This allows you to configure system wide settings for how Smedge operates.

Colors...

Opens a dialog box that allows you to select the display colors for jobs.

Always On Top

Allows you to set the Smedge main window to always stay on top of other windows, even when it is not the active application.

Hide On Minimize

Allows you to set the Smedge window to disappear from the task bar when you press the minimize button. To restore the window you can either use the [System Tray](#) or you can launch another instance of the Smedge.exe program, and your window will be restored.

Start Hidden

Allows you to set Smedge to always start up hidden. To restore the window you can either use the [System Tray](#) or you can launch another instance of the Smedge.exe program, and your window will be restored.

Toolbar

Toggles visibility of the [Toolbar](#).

Status Bar

Toggles visibility of the status bar on the main window.

Help Menu

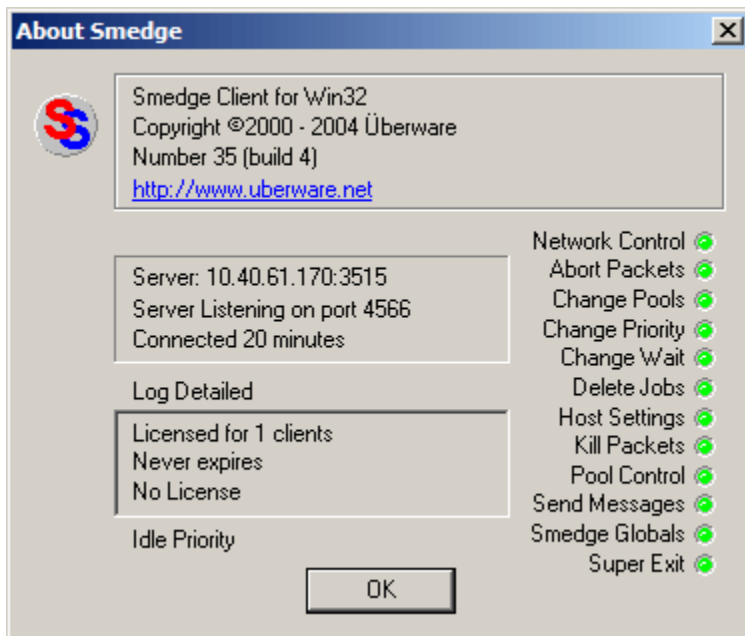
- [Help Topics](#)
 - [About Smedge...](#)
-

Help Topics

Brings up the Windows Help system for the Smedge application.

About Smedge...

Opens the About dialog box.



In this box you can see:

Product Information The version of Smedge you are running, copyright notices, and notification if an upgrade is available

System Information The IP address and port number for the Smedge client that is acting as the server. It also tells you the UDP port where the server is listening for clients who are looking for it. See [The Way of Smedge](#) for more information about Clients and Servers.

License Information The number of clients you are licensed to render on, the date the license expires, and the date the license was created.

Security Information Each green light means that this client is allowed to perform the action listed. A dark light means that you do not have permission from this client to perform the action.

Client Information You can also see the debugging log level and the priority at which this client will launch child processes

The System Tray Icon

- The state of the icon as a status display.
 - Double clicking the icon
 - Right clicking the icon
 - Smedge GUI
 - Version...
 - New Job
 - Show Last 20 Jobs
 - Disable 1 Processor
 - Disable This Machine
 - Time Restricted
-

The state of the icon as a status display

The icon will display whether this client is allowed to render anything or not. If the icon is brightly colored, the client is enabled to render. If it is dull, one or more of the following conditions may apply:

- The Client is not Enabled to render
- The Client could not check out a license, or was asked not to check out a license
- The Client is in a Restricted time period

Double clicking the icon

Double clicking the icon will restore the Smedge window on your screen. If it has been minimized, the window will be restored. If it has been hidden, the window will be shown. It will also be brought to the top of your window stack.

This is the same as selecting [Smedge GUI](#) from the right click menu.

Right clicking the icon

Right clicking the icon brings up a menu with the most common functions in Smedge

Smedge GUI

This will restore the Smedge window on your screen. If it has been minimized, the window will be restored. If it has been hidden, the window will be shown. It will also be brought to the top of your window stack.

This is the same as [double clicking the icon](#).

Smedge Version...

This brings up the Smedge About dialog box. This dialog box provides a lot of information about the Smedge application, including License, Client, and Server information.

This is the same as selecting [About Smedge...](#) from the [Help Menu](#). For more information see that command.

New Hob

This opens a new [Submit Job](#) dialog box. Note that the main Smedge window does not have to be visible to access this functionality. Also note that this dialog box is not modal, so you can have more than one of them open at the same time.

This is the same as selecting [New Job](#) from the [Queue Menu](#). You can get more information about jobs in the sections on [The Smedge Job](#) and [Products Available to Render](#).

Show Last 20 Jobs

This opens up a the window with the last 20 jobs that you have submitted from this system. This allows you to easily bring up a [Submit Job](#) dialog box with all the settings from a previous job. Note that the main Smedge window does not have to be visible to access this functionality. Also note that this dialog box is not modal, so you can still access other Smedge commands when it is open.

This is the same as selecting [Show Last 20 Jobs](#) from the [Queue Menu](#).

Disable 1 Processor

Toggles the number of available processors on the local machine. This is the same as the [Toggle 1 Local Processor](#) command in the [Host](#) menu.

Disable All Processors

This toggles the availability of this client to render jobs. This is the same as checking or unchecking the **Enabled to process jobs** check box in the [Client Settings](#) dialog box, or as selecting the [Toggle Local Client Availability](#) command in the [Host](#) menu.

Time Restricted

This toggles the time restrictions on this client's availability of this client to render jobs. This is the same as checking or unchecking the **Restrict Rendering Times** check box in the [Client Settings](#) dialog box, or selecting the [Toggle Local Time Restricted](#) command in the [Host](#) menu. Note that you need to have set up the time and day fields previously in this dialog box in order for the time restrictions to actually restrict anything.

The Smedge Commandline Interface

The Command line ability allows you to integrate Smedge into your workflow and production pipelines by scripting it. For an example of integrating Smedge into Maya, see the `smedgeRenderSession.mel` script.

You can also access the job submission abilities of the commandline interface using the **Submit.exe** program file. This has the identical commandline syntax as the **Smedge.exe** program file, but has a few differences in behavior. First, when used to submit jobs, it will return the result job ID. You can save this for later use in your scripts. It also removes the possibility of opening multiple instances of Smedge. Note that multiple instances of Smedge should, in theory, be impossible. But, just to be sure, you may want to use **Submit.exe**.

The Smedge syntax is:

```
Smedge
```

Launches Smedge or restores your Smedge window if it's already running.

```
Smedge -h
```

Shows the command line options for Smedge.

```
Smedge [-nosplash] [-hide] [-port port] [-server server[:port]]  
[-noQueueFile]
```

These are global options that control the initial appearance of the Smedge window and the Server settings. You can use these settings any time you use the Smedge command line, even for submitting jobs. If Smedge is already running, they will be ignored.

- | | |
|-------------------------------------|--|
| -nosplash | Disables the Smedge splash screen at startup. |
| -hide | Forces Smedge to start hidden. This implies the -nosplash flag. |
| -port <i>port</i> | Tells Smedge which UDP port to use for broadcast server location. The port number must be the same for every client you want to communicate with. Switch overrides any registry settings that control the Server UDP port. |
| -server <i>server[:port]</i> | Tells Smedge the name or IP address (and optionally the TCP port) where of the machine that will act as server. If it is the local machine, it will become the server. Overrides the <code>server.txt</code> file in the Smedge directory. Note that the server still listens and responds on the UDP port (above) for clients not given explicit server info. |
| -noQueueFile | Tells Smedge not to use the queue file on disk. This is just in case this file causes problems. It has never actually caused any problems, but this switch exists just in case it ever does for anyone. |

```
Smedge -submit file [-gui] [-uid]
```

Tells Smedge to load job settings from the saved .SMR file. You can also double click a file and it will open the dialog box in Smedge automatically. The .SMR file can contain a single render, or it can contain a complete list of many renders.

-gui Force smedge to open a Submit Job dialog box instead of dropping the job directly into the queue. Note that if Smedge has any problems with the command line options it will open the dialog box no matter what. (optional: default is to directly submit jobs)

-uid Tells Smedge to load the saved ID instead of creating a new one. If the job already exists, it will change the existing job, which may cause problems if you change the frame range or packet size.

```
Smedge -s scene -n start end packet [-gui]
      [-j job name] [-product product] [-proj proj]
      [-single|multi] [-p priority] [-pool pool name]
      [-note note] [-maxClients max] [-killOvertime percent]
      [-wait jobID] [-imageDir dir] [-imageFmt format]
      [-extra extra parameters]
```

-s scene	The scene to render (Required)
-n start end packet	First Frame, Last Frame, Packet Size (Required)
-gui	Force smedge to open a Submit Job dialog box instead of dropping the job directly into the queue. Note that if Smedge has any problems with the command line options it will open the dialog box no matter what. (optional: default is to directly submit jobs)
-j job name	The name of the job (optional). If the name has spaces in it, it must be surrounded in quote marks.
-product product	The software product this job uses (optional: default=Maya)
-proj proj	The project or start directory for this job (optional: default depends on product)
-single multi	Request to execute this job single- (one per processor) or multi- (one per client) threaded (default=single)
-p priority	The priority to submit this job at (0-100). 0 means paused, 1 is low, 100 is high. (optional: default=1)
-pool pool name	The Pool to submit this job to. If the name has a space in it, it must be surrounded by quote marks. (optional: default is "Whole System")
-note note	Attach a note to this job. If the note has a space in it, it must be surrounded by quote marks.
-maxClients max	Set the Maximum Clients for this job. (optional: default is the default in the Smedge Globals)
-killOvertime percent	Set the Kill After... value for this job. (optional: default is the default in the Smedge Globals)
-wait jobID	The ID of a job that this new job will wait for before

execution. The ID is a string format of the unique identifier of an existing job. You can find this value in the interface by selecting a job and using the [Show Job ID](#) command in the [Queue](#) menu. This value is also returned from the output of the `submit.exe` program when used to submit jobs via the commandline.

- `-imageDir` *directory* The folder where the rendered images will be saved for this job. If you leave this blank (the default), Smedge will try to autodetect this from the output of the renderer, if possible for the job type.
- `-imageFmt` *format* The format of the rendered image filenames for this job. If you leave this blank (the default), Smedge will try to autodetect this from the output of the renderer, if possible for the job type.
- `-extra` *extra parameters* Anything after the `-extra` flag will be placed verbatim into the Extra Commandline Parameters of the job. This flag must be the last Smedge flag on the command line (optional: default=none)

```
Smedge -JobCommand command jobID [parameter]
```

Allows Smedge to control jobs via the commandline.

- command** The command you want to perform on the job. One of:
 - decrease** Decreases the job priority by 1
 - delete** Delete the job (cancel any running packets)
 - increase** Increases the job priority by 1
 - pause** Sets the job priority to 0
 - pool** *parameter* Assign the job to a specific pool. The pool must be specified as the *parameter*
 - priority** *parameter* Set the job priority. The priority must be specified as the *parameter*
 - requeue** Requeues the job (if it can be found in the GUI) as a new job
- jobID** The ID string of the job you want to control. This can come from the interface by selecting [Show Job ID](#) from the [Queue](#) menu, or as the result from using the `submit.exe` program.
- parameter* The optional parameter for certain commands. See **commands** above.

```
Smedge -disable | -disable1 | -enable | -enable1 | -toggle | -toggle1  
| -restrict | -restart | -kill
```

This allows commandline control of some basic client functionality. These commands have no effect if Smedge is not already running on the machine.

- `-disable` Disables the client
- `-disable1` Forces Smedge to leave 1 physical processor free.

- enable** Enables the client.
- enable1** Forces Smedge to use all physical processors on the client.
- toggle** Toggles the client availability.
- toggle1** Toggles the availability of 1 processor of the client.
- restrict** Toggles the time restrictions on the client.
- restart** Tells Smedge to try to restart itself.
- kill** Causes Smedge to shut down on the client

Secrets and Security

- [Feature Locking](#)
 - [Client Locking](#)
 - [Forced Job Settings](#)
 - [License Broadcasting](#)
 - [Startup Sound](#)
 - [Check for Updates](#)
 - [Confirm Exit](#)
 - [Automatic Hyperthread checking](#)
-

Feature Locking

By default, Smedge allows full control of all Smedge functionality. However, sometimes you may want to limit the amount of control that some clients have. In Smedge number 22 or later, you can accomplish this through a set of registry settings that will lock out features.

The base registry key for these settings is:

`HKEY_LOCAL_MACHINE\Software\Überware\Smedge\Security`

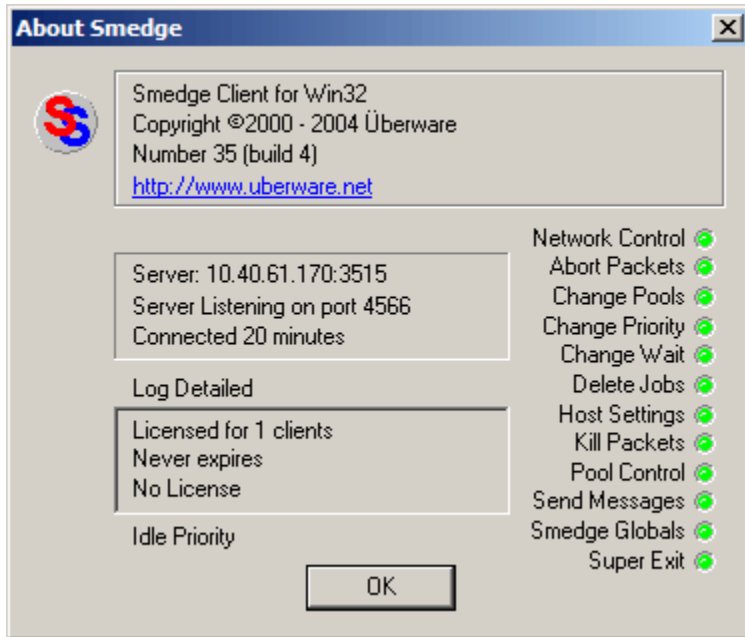
Under this registry key you can create **DWORD** values to block certain features. Set the value to zero in order to block the feature, and one to allow it. If the value does not exist or cannot be read, Smedge will assume that feature to be allowed.

Obviously, this security is only as strong as the security to access this section of the Registry. You can password protect registry keys, and you will probably want to if you want to enhance this security, but be aware that it's not a foolproof system.

<code>AbortPacket</code>	Controls whether the client can access the Abort Packet and command.
<code>ChangePool</code>	Controls whether the client is allowed to change a job's pool after the job has already been submitted.
<code>ChangePriority</code>	Controls whether the client is allowed to change a job's priority after the job has already been submitted.
<code>ChangeWait</code>	Controls whether the client is allowed to change the job that another job is waiting for after the waiting job has already been submitted.
<code>DeleteJob</code>	Controls whether the client can delete a job after it has been submitted.
<code>HostSettings</code>	Controls whether the client can change any host settings. Access to the host settings dialog and the shortcuts to toggle processor, enabled and time-restricted states are all included with this control.
<code>KillPacket</code>	Controls whether the client can access the Kill Packet and command.
<code>PoolControl</code>	Controls whether the client can create, modify or remove pools.
<code>SendMessage</code>	Controls whether the client can send instant messages.
<code>SmedgeGlobals</code>	Controls whether the client is allowed to change the Smedge Globals .
<code>SubmitJob</code>	Controls whether the client is allowed to submit new jobs

- SuperExit Controls whether the client can initiate a Super Exit request and shut down the whole system.
- SuperRestart Controls whether the client can initiate a Super Restart request and attempt to restart Smedge on the whole system

Note that you must create these registry keys by hand. There is no interface element that allows control of these values.



In the About Smedge... window, you can see the current status of your security settings. A green light means that this client is allowed to perform the action listed. If the light is off, the client will not be able to perform the requested action.

Client Locking

In Smedge 26 and later, you can now set the client to only allow it to control things on that client. To enable this, you need to manually set the following registry value:

HKEY_LOCAL_MACHINE\Software\Überware\Smedge\Security\

Create a **DWORD** value called `ClientOnly` and set the value to 1.

This value is read at Smedge startup. If you change this value, you will need to exit and restart Smedge on the client in order to read the change.

When the client is locked, it will only be able to affect local changes:

- You can submit jobs, but you can only modify or delete jobs that you have submitted.
- You can only abort or kill packets that are on your machine.
- You can only modify client settings for the local client.

Any additional feature level security as described [above](#) will also still affect the client.

Forced Job Settings

You can also force some job settings to be at fixed values. You can force the **Priority** and the **Pool** with the following registry values, in this registry key:

HKEY_LOCAL_MACHINE\Software\Überware\Smedge\Security\

ForcePool A **DWORD** value that will be used as the **Priority** of all jobs created on this client.

ForcePriority A **DWORD** value that will be used as the **Pool** of all jobs created on this client.

If you use these forced values, you will probably also want to lock out the client from being able to change the **Pool** or **Priority** of jobs using the [Feature Locking](#) abilities of Smedge.

License Broadcasting

As noted before, Smedge is a system without a central server. While this simplifies day to day use, issues like licensing become more complicated and harder to control. The default action for Smedge is to broadcast the license to your entire network, thus assuring that no matter which machine is acting as server, your license will be there.

Say, however, you have a laptop that you take to another facility. You don't want to broadcast your license to their network, or have your license overwritten by their broadcast. You can use this setting to disable the license broadcasting:

HKEY_CURRENT_USER\Software\Überware\Smedge\Options

Change the **DWORD** value called `BroadcastLicense`. When this value is zero, this machine will not broadcast its license. When the value is 1, it will.

This value is also controlled at the time of license entry. On the Enter License dialog box there is a check box. When checked, the license you are entering will be broadcast. When unchecked, it will not.

Startup Sound

The first time you ever start Smedge on a computer, it plays that god-awful sound. Thankfully, it never plays it again.

Unless you ask for it! That's right, there's a registry key that controls what happens.

Under the registry key:

HKEY_CURRENT_USER\Software\Überware\Smedge\Options

You will see a **DWORD** value called `BionicMan`. When this value is zero, the sound never plays. When this value is 1, the sound always plays. If you set the value to greater than 1, the sound will randomly play every x times you start Smedge. For example, if you set the value to 10, your computer has a 10% chance of playing the sound when you start Smedge. This can keep things interesting.

Check for Updates

On startup, Smedge tries to check for an update on the internet. If you want to disable this feature, you can now do so. Go to the registry key:

HKEY_CURRENT_USER\Software\Überware\Smedge\Options

You will see a **DWORD** value called `updateCheck`. If this value is zero, Smedge will not attempt to check for updates. If this value is one, it will.

Confirm Exit

By default, Smedge will put up a message box when you try to exit the program asking you to confirm the action. You can disable this with a registry key:

HKEY_CURRENT_USER\Software\Überware\Smedge\Security

You can create a **DWORD** value called `checkExit`. When this value is zero, the check is disabled, and Smedge will exit without confirmation when you select the [Exit](#) command from the [Queue](#) menu, or use the system menu or close box to close the main Smedge window.

Automatic Hyperthread checking

Smedge tries to determine if you are running on a machine with Hyperthreading processors. That way, when you disable 1 processor, it actually disables 1 physical processor, not 1 logical processor. If you have problems with this detection system, you can disable the check with a registry key:

HKEY_CURRENT_USER\Software\Überware\Smedge\Options

Create a **DWORD** value called `hyperthreadCheck`. When this value is zero, the check is disabled, and Smedge will simply assume that 1 logical processor is 1 physical processor.

Note: Smedge version 2.37.0 and earlier looked in the wrong registry key:

HKEY_LOCAL_MACHINE\Software\Überware\Smedge\Security

Be sure to use the correct location for Smedge 2.37.1 and later.