

SMEDGE

Installation Guide

Smedge 2026

© 2004 - 2025 Überware™

Table of Contents

<u>WELCOME TO SMEDGE!</u>	<u>3</u>	<u>RUNNING SMEDGE AUTOMATICALLY</u>	<u>16</u>
SOME DETAILS ABOUT SMEDGE	<u>4</u>	MACHINE ROLES	<u>16</u>
WINDOWS	<u>5</u>	INSTALLING SERVICES USING SMEDGEGUI	<u>17</u>
OS-X	<u>6</u>	INSTALLING SERVICES USING THE SYSTEM TOOLS	<u>18</u>
LINUX	<u>7</u>		
INSTALLING THE LICENSE	<u>8</u>	<u>UPGRADING SMEDGE</u>	<u>19</u>
GETTING OTHER PROGRAMS TO WORK WITH SMEDGE	<u>9</u>	FROM SMEDGE 2010 OR LATER	<u>19</u>
		UPGRADING FROM SMEDGE 3 OR EARLIER	<u>19</u>
<u>WORKING CROSS-PLATFORM</u>	<u>10</u>		
SOME NOTES ABOUT WORKING CROSS PLATFORM	<u>11</u>	<u>TROUBLESHOOTING</u>	<u>20</u>
<u>ADDITIONAL INSTALL INFORMATION</u>	<u>12</u>	<u>BUILDING SMEDGE FROM SOURCE</u>	<u>22</u>
AUTOMATIC ENGINE MODE	<u>12</u>	<u>UNINSTALLING SMEDGE</u>	<u>23</u>
ALTERNATE COMPONENT CONFIGURATIONS	<u>13</u>		
ENCRYPTION	<u>14</u>	<u>GETTING HELP</u>	<u>24</u>
PYTHON	<u>15</u>		

Welcome to Smedge!



Smedge is a system that allows you to control a wide variety of other software packages on multiple machines at the same time. It provides accurate and reliable queuing and distribution of work on multiple devices. It is completely platform independent and flexible enough to support any type of work process you can imagine.

Some details about Smedge

Smedge works in a “client/server” architecture. Note that every machine can be both client and server, and that the details of that architecture are handled automatically by the program. However, there are some fundamental prerequisites for Smedge to operate correctly. Specifically, machines must be connected to a network, and machines must be able to connect to each other directly through TCP/IP. The machines do not have to be able to find each other by name for Smedge to work, but some operations may be limited if they cannot. Ask your network administrator if you have any questions about your network.

If a machine has a software firewall installed, that firewall must either be disabled or configured to allow Smedge component applications to communicate. By default, the clients let the OS pick an interface and port to use for communication, so it is generally easiest to open up any firewall by executable instead of by port. However, the ports used by the clients are configurable as well.

Please note that the products that Smedge can control are distributed and licensed separately from Smedge. For installation or licensing of these products, please contact your support agent or the technical support facilities for that software product.

We recommend using a file server to use for storing your projects and associated source, texture, data and render files. Configuring a file server is beyond the scope of this manual. There are many resources on the Internet for setting up file servers, or you can consult an IT professional or value added reseller to configure your file server and networking optimally for your needs and budget. By default, Smedge does not copy any files around as part of its processing, though it is possible to configure the system to copy files if you want to set up that kind of pipeline.

All trademarks are the property of their respective owners.

Windows

Follow these steps to install Smedge on a computer running Windows Vista or later

1. Download the MSI file.
2. If it is not opened automatically, double click the MSI file to start the installation process.
3. Run the **Smedge** program from the Start menu or the desktop icon to start Smedge.

More Details

Smedge is a 64 bit Windows application. If you want to install run Smedge from a network location, be sure to install the Microsoft Visual C++ runtime library (x64). You can download this from the Microsoft web site:

<http://www.microsoft.com/en-us/download/details.aspx?id=30679>

You can also download Smedge as a ZIP archive. This archive does not include the Microsoft Visual C++ runtime library. Download from the link above if you need it.

Smedge is no longer officially supported on Windows XP, as that operating system is end-of-life. We can provide a 32 bit build, if you need, but recommend using the 64 bit build if possible, as some operating system features are limited and limit Smedge operation and performance.

OS-X

Follow these steps to install Smedge on an Intel based Macintosh computer running OS-X 10.6 or later

1. Download the DMG file.
2. If it is not automatically mounted, double click the DMG file to mount the distribution volume.
3. Drag the Smedge application icon to the folder where you want to install it (for example to /Applications)
4. Double click the Smedge icon to start Smedge.

More Details

Smedge is a 64 bit Intel Mac application. The Smedge program files and libraries on Macs are entirely self contained in the Smedge.app bundle (the application icon you dragged). You can access the individual components, including the command line utilities, virtual modules and module INI files using Terminal or by using the Finder's **View Package Contents** command. You can run the application from any folder.

You can also download Smedge as a ZIP archive. Unzip it with the Finder or another Mac based tool to ensure that the internal structure is preserved correctly. Once uncompressed, the app icon can be dragged to any location you wish. Double click it to start it up.

Linux

Follow these steps to install Smedge on an Intel compatible computer running Linux

1. Download the RPM file
2. If it is not opened automatically, you can use a graphical package manager or the rpm command line tool to install it.
3. Start Smedge by typing the path to where it is installed:

```
/opt/smedge/SmedgeGui
```

More Details

You can create an alias, add a symbolic link into your binary path, add the Smedge program folder to the default PATH environment variable, and so forth. Smedge currently does not automatically integrate with the Linux distributions' graphic front ends at this time, though the GUI components should work normally on any recent distribution.

You can also download Smedge as a compressed tar archive. This method allows you to place the software wherever you wish on your system. However, it does not automatically create the “Machine Folder” (`/etc/smedge`) or give it appropriate permissions. Without the Machine folder, some machine wide-settings may not be saved correctly between sessions. You should create this folder and open its permissions to any users that will be using Smedge for best results. The tar archive also will not install any required dependencies automatically, so you may need to install these yourself manually.

The Linux distribution is built on CentOS 7 x86_64. Other custom builds are possible, for other distributions, 32 bit versions, or other architectures. Contact support@uberware.net if you have a specific distribution need that is not met by one of the public builds.

Installing the License

Once you have installed at least the single Master machine, you can install the license code. You can install the code from any machine, on any platform, that is currently connected to the Master. You will only need to install a single code no matter how many machines you have.

Follow these steps to install the license:

1. Start **SmedgeGui**. It does not matter which machine you use, as long as it can connect to the Master.
2. Make sure you are in **Administrator Mode**
3. Choose **System > System Commands > Submit License...**
4. Copy the **Master ID** code you see into the field on your Uberware.net account [generate license page](#).
5. Paste the resulting code back into the **License Code** field.
6. Click **OK**.

Getting other programs to work with Smedge

Smedge can control many third-party applications if they provide a command line based interface. For example, the Maya Product supports rendering of Maya scenes using a common command line interface. Some of these Products may require additional support to get operating. Besides basic issues such as installation and licensing of the third party application, one of the most common issues is ensuring that Smedge is able to find the command line executable to use the system.

Many products in Smedge can try to find the executable in a known default installation location, and can often find the latest version of an application if multiple versions are installed. When Smedge is started on a machine, it checks the paths it knows about, looking for new installations. Note, however, that once it has detected an installed version, it will not update that version if you later install an updated version of that product. You will need to manually update the version like below.

If a Smedge Product uses a command line executable, that Product will have a way to set the path to the executable manually. The most common way is to set this as an option for that Product on the Engine, by following these steps:

1. Select all of your Engines in **SmedgeGui**
2. Choose **Engine > Configure Product Options...**
3. Select the Product you wish to configure, for example “Maya”.
4. Put the desired path into the **Path to the Executable** field

For example, the path to the Maya 2014 executable on OS-X is usually:
`/Applications/Autodesk/maya2014/Maya.app/Contents/bin/Render`

5. If desired, repeat the steps 3 and 4 for any other Products you may need.
6. You may want to save a full configuration as a default you can apply to have all settings configured from that default automatically by pressing the **Save as Default** button. Alternatively, you can save the full configuration as a Preset, which you can the apply to new machines to set them up quickly with standardized settings.
7. Click **OK** to have the change committed to every Engine you had selected.

Working Cross-Platform

When you are mixing multiple operating system platforms together, any Smedge component application can be run on any platform and they all interconnect together. However, because each platform handles access to network resources differently, there are some additional steps that are required for completely transparent multi-platform operation.

The key to making Smedge work across multiple platforms is to tell the system how to translate the different access systems that you are using on your site. For example, if your project files are generally kept on a file server that is mounted on a Windows drive letter, you would tell Smedge the drive letter that you are using on Windows, and the mount point on a Unix file system that a Linux or OS-X machine uses to access that same file server.

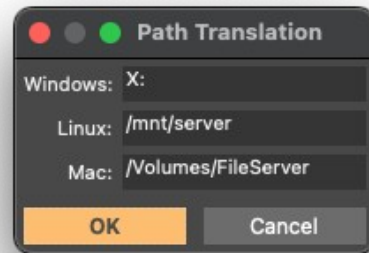
To configure path translations, follow these steps:

1. Start **SmedgeGui**.
It does not matter which platform you use, or which machine you use. Any machine that has Smedge installed can allow you access to configure the system. However, you must be in **Administrator Mode**. This command is available in the **System** menu.
2. Choose **System > System Commands > Configure Master...**
This opens the Master configuration dialog box. You can get more information about this dialog box in the *User Manual* included in the Smedge distribution, or available on our website:
<http://www.uberware.net/smedge3/docs.php>
3. Click on the **Path Translations** tab.
4. Click the **Add** Button
5. The three fields listed here are the “root” part of the file path, or the leftmost part, which indicates which physical volume on which the file resides. For example, on Windows, the root is usually a drive letter or UNC path:
X: \Projects\MyJob\scenes\MyScene.fil
\\Server\Drive \Projects\MyJob\scenes\MyScene.fil

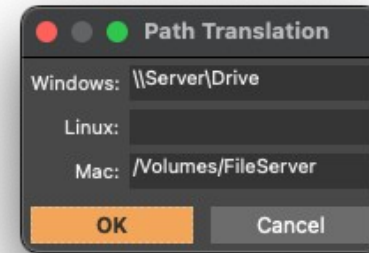
Each platform has its own syntax for identifying file systems. For example, on a mac, this same file may be accessed with this path:

/Volumes/FileServer /Projects/MyJob/scenes/MyScene.fil

The Path translations are the parts of the two names above that are different, the underlined part, starting at the leftmost character. See the following examples:



In this example, the mapped drive X: on Windows will be translated to /mnt/Server on Linux and /Volumes/FileServer on OS-X.



You can leave the field for any platform you do not use blank. This example is how it can look if you only use Windows and OS-X machines. It also shows a UNC path instead of a drive letter.

Although these screen captures were made on Mac, you could use any platform to make these changes.

6. Click **OK** to add the translation to the list of translations and close the small Path Translations adding dialog.
7. Click **OK** again to close the Configure Master dialog and submit the changes to the Master.

Some notes about working cross platform

Smedge will always translate the paths that it actually operates with. This means, paths that are submitted by the user as part of the Job information, or paths that the Smedge software itself was able to detect from the products it controls. However, any paths that may be embedded into the work's source files (like texture files referenced in a scene or source clips from a project) may not be accessible to Smedge.

Smedge includes a feature to try to translate the scene contents. However this will only work reliably with rendering products that use a text file format for their scene data. Every Product currently supported in Smedge that uses only a text file format for its scene data already has this system enabled by default. For example, Nuke, mental ray Standalone, Pixar RenderMan, and V-Ray Standalone all use this system by default.

You can try enabling the system for other Products. If you only ever use Maya ASCII format scenes, this will work great with all Maya products, too. However, note that the translation Smedge does is a simple text file translation, so if the file is in binary format, it may get corrupted by the attempts to translate the files, and will probably not work correctly. Many of these products (like Maya) can be somewhat smart about translating their own scene data as needed. If you experience unexpected results from cross-platform renders, check that your source files are accessible by all nodes when accessed through the file server.

Automatic Engine Mode

To reduce the Smedge overhead when a machine is primarily being used for rendering and not as an active interface to the system, Smedge includes a feature called “Engine Mode” where the GUI disconnects itself from the Master, and shuts down any redundant backup Master running on the machine. In this mode, the machine runs with about the same overhead as if you ran just the SmedgeEngine component manually.

Automatic Engine Mode kicks in after a certain amount of time without any interaction in the GUI. The amount of time is configurable at the system level in the Administrator Options dialog box, and can also be overridden for a specific machine in the SmedgeGui options. You can also manually enter Engine Mode using a menu command in the system menu.

If your computer's name includes one of the words: “render”, “node”, “blade”, or “smedge” and a number, or consists only of numbers, Smedge assumes that this machine is primarily meant to run as an Engine, and the GUI will start in Engine Mode by default the first time you start Smedge. You can always use the Connect menu command to run in normal mode.

For successive runs, GUI will run in the same mode as whatever the last mode it was in when it shut down. Note that the Master component is also stopped in Engine mode, unless your machine is the primary Master machine. When you restart a machine that was in Engine Mode, the Master component is not started with the GUI, and is only restarted when you try connecting if your machine was previously the Master when it last shut down.

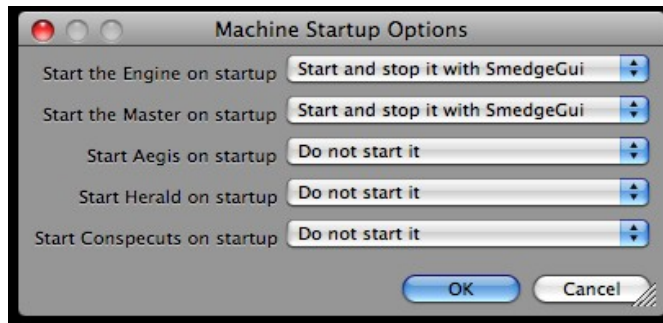
In normal usage, this is not a problem. If you generally start the same machines up in the same modes, things will work consistently. If you end up with no Master on your network, you will need to manually start the Master on at least one machine. If that machine was not the master or an up-to-date mirror the last time it was running, you may have out of date job information. If you find this to be the case, you can stop that master and try starting it on a different machine that may have been up-to-date.

Alternate Component Configurations

By default, Smedge starts with a standard component configuration when you start SmedgeGui (the primary access interface to the system). Normally, each instance of the GUI tries to start an instance of the Master and the Engine on the machine when it starts, all running under the same user account as the original GUI process. This allows every machine to be able to be used for rendering, and allows any machine to take over as the Master for your system at any time.

This configuration is useful in small installations, but may not be optimal for larger setups. You may not want to have certain machines check out Smedge licenses because they are never used for rendering, and you may find that too many Masters on your network can degrade performance. You can configure which components start with the GUI by default for the system using the menu command **System > Components >**

Set System Default Component Startup (available in Administrator Mode).



This dialog allows you to set how any instance of SmedgeGui that connects to this Master will try to operate, by default. If you set the “Master” not to be started by default, then only the current Master machine will be allowed to start the Master on startup.

You can also specify that the GUI also start some of the extra graphical shell components included with Smedge: Aegis, Herald and Conspectus. It can be useful, for example, to set Aegis to start automatically with the GUI, so that users have a very simple 3 button control to get their machine enabled or disabled for rendering.

You can override these defaults in the SmedgeGui options dialog, or by the command line to start the program. These options are stored per user on the local machine only, and the command line options will only apply to that instance of the process running on the machine.

Encryption

Smedge communication is encrypted using perfect forward encryption included with the networking library used internally (ZeroMQ). This security uses a public/private key encryption mechanism. By default, the Master uses a built-in private key, so every user gets encryption automatically by default. However, since this private key is shared by all installations of Smedge, it's not truly secure unless you generate your own key pair. You can use the SmedgeMaster command line to generate a new truly random private key, which gives you complete security.

1. Use this command to generate the public and private key file:

```
SmedgeMaster -keygen
```

This will generate 2 files: **smedge_public.key** and **smedge_private.key**.

2. Move the private key file into a folder where the Master can read it on startup. Commonly you can use the Master “Machine Folder” or the Smedge program folder
3. Copy the public key file into a place your clients will read it. The Smedge program folder is usually the best place.

As long as you use the matched public and private key files on your network, any client on any OS will be able to connect and communicate correctly. Any mismatch in the keys will result in the client not being able to connect to the system at all.

Python

The Python API lives inside the Smedge distribution. The most common way to make it available is to set your PYTHONPATH environment:

```
Windows    set PYTHONPATH=C:\Program Files\Smedge\python
Mac        export PYTHONPATH=/Applications/Smedge.app/Contents/MacOS/python
Linux      export PYTHONPATH=/opt/smedge/python
```

Some environment variables that can configure the Python API:

Variable	Meaning
SMEDGE	Override the Smedge program folder location (default assumes the code is in the default deployment location).
SMEDGE_MASTER	Override the Master location. You can specify the master hostname or IP and optionally the port.
SMEDGE_MASTER_PORT	Override the Master port (default is 6870).
SMEDGE_LOG_LEVEL	Set the log file level (default is 5, debug level is 6).
SMEDGE_CONNECT_TIMEOUT	Set number of seconds the CLI tool will wait trying to connect before returning a failure (default is 10)

Further documentation can be found online:

<https://www.uberware.net/support/python-api/>

Running Smedge Automatically

Machine Roles

Smedge defines 3 roles for a computer:

Master	This is the machine that will manage the system. This machine should not also be both an Engine and a File Server for your network if you can avoid it.
Engine	This is a machine that does the work.
Workstation	This is any machine that has Smedge installed, if it is not performing one of the other roles.

When you run Smedge normally, it automatically starts the Master and the Engine on your machine when you start the GUI, and stops them when you stop the GUI, making each machine able to fulfill any role at any time. This is a simplistic setup that makes it easy to use, but can degrade performance in large deployments by increasing the amount of network overhead that Smedge uses. Additionally, many high performance rendering installations want to optimize the amount of system overhead, running as few processes as necessary.

[Automatic Engine Mode](#) can help, and you can configure which components get started automatically with SmedgeGui both by default, and customized for a specific machine (see [Alternate Component Configurations](#), above), but these still require that the GUI is started in order to start the other components, whereas running each component automatically can be done without the GUI running at all, and even without logging into the console at all.

Note, however, that running Smedge automatically has disadvantages as well. The most common concerns are the user account that is used to run the process and how the process accesses the shared resources that are required for doing the work. You should pick a user account that is appropriate for running the SmedgeEngine daemon. **Using a system administrator or root user account, which is the default, can be dangerous to your network security and stability if your system is accessed by someone with malicious intent.** The details of making your file server available to a service or daemon is beyond the scope of this document. See your network administrator or IT department.

At an absolute minimum, to make a working Smedge network you need only a single machine running SmedgeMaster, and you only need the SmedgeEngine process on each machine you want to do work. You can run the other shell components on as many or as few machines as you wish. If you have redundant Master systems, they will “mirror” each other, and can take over as Master automatically if needed.

Installing services using SmedgeGui

The easiest way to install Smedge components as services is to use the commands for doing so from the SmedgeGui System menu:

System > Components > Engine Service > Install the Engine Service

This will install the Engine component, making the machine available to do work automatically from the moment it boots until the moment it shuts down, without a user needing to be logged in. There are minor differences on how this works on each platform:

- | | |
|----------------|--|
| Mac | You will have the opportunity to set the user account and any extra parameters that you want, and you will have the opportunity to authenticate as an administrator in order to install and start the service. |
| Windows | You will have the opportunity to set the user account and any extra parameters that you want, but you must be running SmedgeGui as an administrator to successfully install the service. |
| Linux | You will not have an opportunity to set the user account or extra parameters and you must be running SmedgeGui as a superuser. |

You may see your machine go offline for a bit in the SmedgeGui while you do this. It should come back within a minute or two.

System > Components > Master Service > Install the Master Service

This will install the Master component, making the machine available to administer the system automatically from the moment it boots until the moment it shuts down, without a user needing to be logged in. There are minor differences on how this works on each platform:

- | | |
|----------------|---|
| Mac | You will have the opportunity to set any extra parameters that you want, and you will have the opportunity to authenticate as an administrator in order to install and start the service, but you cannot override the user account. |
| Windows | You will have the opportunity to set any extra parameters that you want, but you must be running SmedgeGui as an administrator to successfully install the service and you cannot override the user account. |
| Linux | You will not have an opportunity to set the user account or extra parameters and you must be running SmedgeGui as a superuser. |

You may see your machine get disconnected from the Master while you do this. It should come back within a minute or two.

Installing services using the system tools

On Macs, there are a series of utilities in the Utilities folder of the distribution for installing, removing, starting and stopping the daemon components. These scripts are self descriptive about what they do, and take care of 100% of the work, including being able to set the user account, extra parameters and authenticate as a system administrator.

On Windows, there are several batch files in the Utilities folder, along with some supporting programs and data files that are used by the batch files to do their work. The batch files will do all of the work, including optionally creating a default Smedge local user account for your system to use. You must be run the batch files as an administrator for them to work correctly.

On Linux, there are two shell scripts in the Utilities folder that do the work. Run these scripts as a superuser to install the daemons. If you don't supply an argument, it will install (or remove) the engine daemon by default. Otherwise, supply the argument **master** to install or remove the Master as a daemon or the argument **engine** to install or remove the Engine as a daemon.

Other capabilities

Each platform has its own peculiarities about running applications automatically in the background. For more information, see your system documentation, your system administrator or your IT department. You can also contact your software or hardware support vendor or Überware technical support

Upgrading Smedge

We always recommend that you install and use the latest version of Smedge, in order to take advantage of improved reliability and new features that may be included. Always be sure to read any release notes or additional information Überware supplies with each new release. There may be changes that would require more complicated upgrade procedures.

From Smedge 2010 or later

Upgrading from Smedge 2010 or later basically only requires that you overwrite the program files with the new version. In order to allow this, you should shut down all Smedge component processes that are started from the files on disk, so that the files can be overwritten. (If any component is running when you try to do this, the attempt to overwrite the files may fail.)

Smedge 2023 does not allow earlier versions to connect to it. This can make it difficult to know if a machine is still running an old version. We strongly suggest you shut down all Smedge components with a system-wide exit, and upgrade your whole system at one time.

Note that the Windows installer will automatically upgrade Smedge 2010 or later to Smedge 2023, as it is designed to allow upgrades without uninstallation first. It will also try to notify you if any components are still running when you try to run the installation, and can configure your system to finish the upgrade on reboot, if needed.

Upgrading from Smedge 3 or earlier

When upgrading to a new Smedge version, it is best to perform a system wide shutdown to make sure that the old version does not try to use the machine at the same time. The easiest way to do this is to use the **System > System Commands > System-Wide Exit** command (you must be in **Administrator Mode** to access this command).

Smedge 2010 and onward is a significant operational change from earlier versions. You may want to uninstall any older versions to avoid confusion, especially because the versions do not interconnect.

Forcing clients to connect to a specific Master

The best way to manually configure where to find the Master is to use the Connection.ini file. This file can live in any of the default INI file locations. You can configure it in the Smedge Gui using the menu command System > Configure Connection. You can set which machine will be the master, using that machine's name or IP address, and you can also (optionally) change the port to use for communication.

Note that it is also possible to configure the master name and port by command line flags to the processes. See the Administrator manual for more information on INI file loading and on the common command line flags that configure communication.

Services

If your system has unusual security or control systems for installing and configuring these service or daemons, then the installer may fail to get the processes to actually start. Check for the actual daemon or Service process executable, which will have some variation of the process executable “SmedgeMaster” or “SmedgeEngine” for the Master and Engine, respectively. If you do not see the process executable running on your system, check your system’s Service or daemon configuration utility.

Firewalls

You must either disable firewall or port blocking software, or open an exception for all of the Smedge executables. The Smedge Master listens on port 6870 by default, but all other ports are dynamically assigned by the system. If you cannot get SmedgeGui to connect to the Master even with the processes running on the same machine, this is the most likely culprit.

Subnets

Smedge's automatic location system uses UDP broadcasts, which are generally restricted to the local subnet. To make Smedge able to work across multiple subnets, you will need to force the clients to find the Master manually, following the instructions above.

Smedge no longer runs after upgrading

The most likely cause of this problem is that the Smedge program files were still in use when you ran the installer. This can happen if you have a shared install and not every client that is using the shared install was shut down when you actually ran the installer. If the installer asks you to restart the computer, make sure that you perform the restart. The restart will allow the OS to finish the installation process after no more clients are using the program files. See the [After Installation](#) section for more information.

Smedge seems to lag, or seems to lose connections regularly

If you are using your Master as an Engine or as a File Server, or both, it may be overloaded. Try disabling one or more of the processes on this machine, or using another machine to take over one of those tasks. If the Master machine gets overloaded, delays in communication can cause these kinds of behaviors. It's impossible to give an exact number of machines that any specific combination of hardware and software can support as a Master and still also operate as an Engine.

The Master running on my Mac is having errors about 'too many open files'

On some Macs, the default process file limits are quite low (we've seen as low as 256 open files on machines in the wild with default settings from Apple). Because Smedge can have many communication ports open when working on medium to large networks, and network ports count as open files to the operating system, this resource can get exhausted easily. To resolve the problem, increase the maximum open file limit. The exact details depend on how you are running Smedge, but you can use the `ulimit` tool, change the settings in the `launchd.conf` file, and/or configure the limit in the Smedge component's daemon plist file. See the OS-X docs for more information on setting the maximum file limit.

Jobs have errors about missing resources

If you use multiple mapped network drives as part of your work flow, Smedge may not be able to automatically detect all of the required mapped drives. Try using the **Mapped Drive Manager** in SmedgeGui to easily configure the drives that Smedge will need to remap in order to successfully process work.

I cannot see anything in SmedgeGui, and it never connects to the Master

Make sure that the SmedgeMaster process is running on the Master machine and that there is no firewall software (or hardware) blocking communication (even on the same machine). Also, try giving the components a few seconds to try connecting. If heavy machine load or network traffic causes a connection attempt to time-out, the client will automatically try again every 15 seconds. If none of this helps, please send the MachineLogs folder from the client and from the Master to Uberware technical support, and we will be glad to help diagnose the problem to get you up and running quickly.

Building Smedge from Source

Because Smedge uses some open source products, we are required by license to make our own source code available to customers. To gain access to the source code, please contact support@uberware.net. You will be required to sign a confidentiality agreement, to ensure that our own proprietary code is protected.

Building Smedge from source requires building the following dependencies:

- **boost** Boost C++ libraries, requires building the **regex**, **system**, and **thread** libraries
- **hwloc** Portable Hardware Locality library, part of OpenMPI
- **wxWidgets** Cross-Platform GUI platform
- **ZeroMQ** Communication library. Version 4.3.2 currently.

There are supporting build scripts to help build most of these dependencies on most platforms, included in the platform sub-folder of the **dist** folder in the Smedge code distribution. The build system depends on some specific environment variables:

SMEDGELIB	The root folder of the Smedge source code	<code>/usr/dev/Smedge</code>
WXWIN	The path to the wxWidgets base folder to use	<code>/usr/dev/wxWidgets-3.1</code>
WXVER	The major and minor version of wxWidgets	<code>31</code>
BOOST	The path to the boost base folder to use	<code>/usr/dev/boost_1_65_0</code>
HWLOC	The path to the hwloc base folder	<code>/usr/dev/hwloc_1.11.6</code>
ZERMQ	The path to the ZeroMQ base folder	<code>/usr/dev/zeromq-4.1.6</code>

Smedge can be built with Visual Studio Express 2017 on Windows, Xcode 7 on Mac, or with any recent GCC version on Linux.

Contact support if you have any trouble getting a dependencies built correctly for Smedge, or are interested in a custom build for alternate hardware or operating systems.

Uninstalling Smedge

On Windows, Smedge integrates with the programs control panel and includes an uninstall script, which can be accessed through the standard control panel, or via the shortcut in the Start menu.

On Linux, use the standard RPM management tools to remove Smedge.

On Mac, you can drag the Smedge app icon to the Trash to remove Smedge.

Uninstalling Smedge cleans up all of the program files and common system data. It will not clean up log files that were generated with your scene files during renders, and it may not clean up log files and preferences in each user's home folder (or equivalent) on each platform.

Smedge only touches the file system, so you can always remove Smedge simply by deleting all of its files.

If you have any questions about removing Smedge, you should contact Überware technical support.

Although we strive to make setting up and managing a server farm as easy as possible, it is still a complicated task and there are a lot of options. If things aren't going well for you, there are several places you can get help.

Documentation

The documentation provides a lot of information about working with the product. Make sure to check the Administrator Manual or any custom Module manuals (like the “Using Smedge with After Effects” manual) for more details about advanced operation or specific product operation. The manuals are also available online on our web site, where you will always find the most up-to-date version:

<http://www.uberware.net/smedge/docs.php>

Frequently Asked Questions

There is a list of answers to common questions and basic problems available on our web site, in the list of Frequently Asked Questions. You can browse and search through it here:

<http://www.uberware.net/faq.php>

Remote Support

Uberware provides remote support and installation services. You can get more information here:

http://www.uberware.net/remote_support.php

Contact Us

You can contact Uberware technical support. Find the latest contact information on our web site here:

<http://www.uberware.net/contact.php>