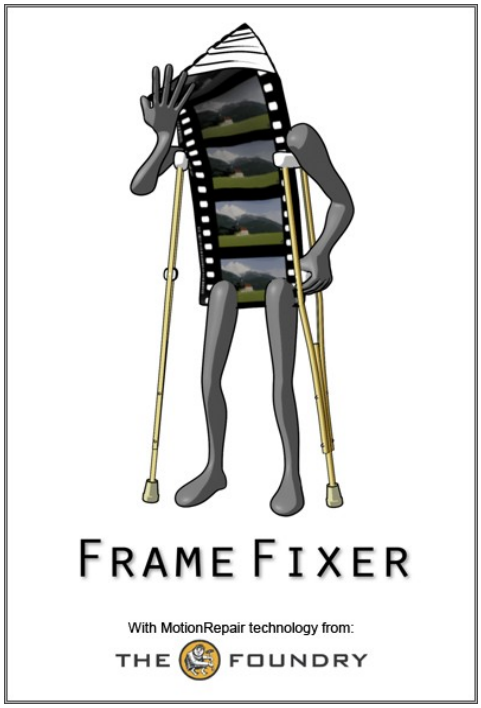


FRAME FIXER
USER DOCUMENTATION

Table of Contents

Introduction

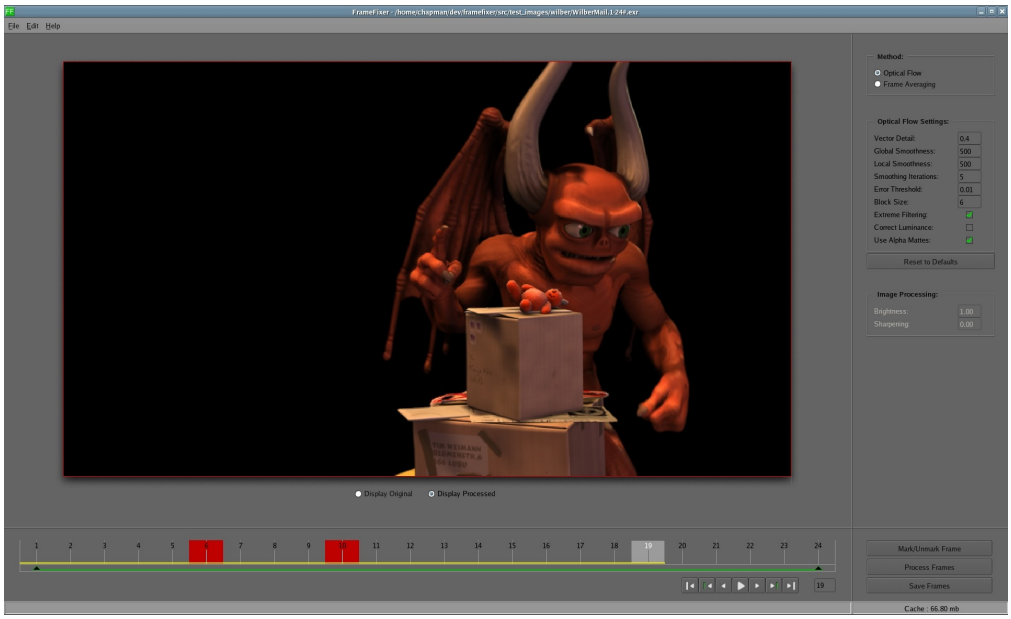
FrameFixer aims to smooth the work flow and reduce stress in busy production environments by removing the problem of corrupt or missing frames from 3D rendering.

FrameFixer works by analyzing only the output images: no scene files or other data is required to reproduce the missing frames, and unlike simple frame averaging or other quick fixes, the results match how the true frame would look, and in many cases the generated frames are indistinguishable from the adjacent frames.

FrameFixer enables quicker turnaround for dailies and other review processes, so that decisions and further iterations can be made instead of waiting for the missing frames to be re-rendered before any further progress can be made.

Using FrameFixer you can easily work through any missing frames encountered during your production, only needing to correct them before your final deliveries. The amount of time this saves over the duration of a typical production can be significant, not to mention the peace of mind in knowing you'll be able to reliably deliver and review your work each day regardless of render glitches.

The high quality interpolation used by FrameFixer is achieved by using advanced optical flow motion analysis technology from The Foundry (www.thefoundry.co.uk), makers of acclaimed high-end compositing plugins such as the Tinder and Furnace suites.



Quickstart Guide

This quick start guide describes the graphical work flow - for command line use please see the section entitled Running from the Command Line.

To correct one or more frames in a sequence, first launch FrameFixer and select **Open Sequence** from the **File** menu (or double-click the empty viewing area). This brings up the File Sequence Browser where each image sequence is listed as a single entry. When you double-click a sequence it will then be shown in the main image display area, and the playback timeline will be updated to show the frame range you have loaded.

Next, using the timeline, simply move to a frame you wish to repair, and press the **[SPACE]** key on your keyboard (or press the **Mark/Unmark Current Frame** button in the lower right corner of the window). That frame will now be drawn in red in the timeline and is marked for processing.

When you have finished marking the corrupt frames, press the **[ENTER]** key (or click the **Process Frames** button in the lower right corner) and the frames will be processed. When complete, the processed frames will be marked in green on the timeline, and if you move to that frame you can see the corrected image.

If you are happy with the results you can save the processed frames back to disk by pressing **[CTRL+S]** (or click the **Save Frames** button). If you want to discard any of the processed frames before saving, move to the frame and press **[SPACE]** again.

After processing and saving the synthesized frames you will have a complete image sequence. Any existing frames you have replaced will be saved with a .bak file extension, so you can always review or revert to them later.

That's all there is to it!

Installation

The FrameFixer application looks and functions much the same under all the supported operating systems. Therefore the majority of the documentation is the same for all platforms except for these instructions on installing and launching the application.

The application is self-contained, and doesn't require any changes to be made to your system, so installation is just a matter of extracting the downloaded archive to your preferred location.

Linux

Extract the FrameFixer archive to your preferred location (e.g. **/usr/local/framefixer**) and simply run the **framefixer** script located in that directory. Of course, you may want to setup an alias or modify your path to be able to run it without specifying the full path.

Please ensure you have recent OpenGL and display drivers installed, and you may also need to install some additional Linux libraries if you have a non-standard installation. FrameFixer has been tested on the following Linux installations: Fedora Core 2, Fedora Core 4, Debian 3.1, SUSE 9.2, Ubuntu 6.06, Ubuntu 6.10.

Microsoft Windows

Extract the FrameFixer zip archive (using Winzip or a similar utility) to your preferred location (e.g. **C:\Program Files\FrameFixer** or **N:\Apps\FrameFixer**).

You can then run the program by double clicking the **FrameFixer.exe** application file within that directory using the Windows Explorer, or execute it in the command shell. You can optionally right-click and drag the application into your Start Menu or Desktop to create a shortcut for quicker access (be sure to make a shortcut, not move the file).

FrameFixer has been written for Windows XP or later. It may run on earlier versions of Windows, but this is not supported.

Apple Mac OS X

Double click the FrameFixer disk image (.dmg) file to mount it, then simply drag **FrameFixer.app** into your **/Applications** folder (or any other location if you prefer).

You can then run the program by double clicking the **FrameFixer** application file within that directory using the Finder.

FrameFixer has been written to run under OS X v10.4 or later. It may run under earlier versions of OS X, though it is not supported.

Licensing

The FrameFixer application will run without a license, but you will need to install one before you can process any frames. FrameFixer's image processing uses the industry standard FLEXlm licensing system. You can purchase node-locked or floating licenses and request evaluation licenses from the FrameFixer web site: www.framefixer.com

If you are using a node-locked license, all you need to do is save your license to the **license.dat** file in the FrameFixer installation directory (there will be an empty file with that name already there). The Windows version of FrameFixer is hard-coded to look for the license file there, and under Linux and OS X the FrameFixer launch script sets the **LM_LICENSE_FILE** environment variable to this location by default.

If you have a floating license, a license server needs to be installed to a machine on your network. There is a FLEXlm directory within the FrameFixer installation which contains the license server daemon, some diagnostic tools and a **readme.txt** with more specific instructions on getting the license server running. After setting up the license daemon, you will then need to save your license file over **license.dat** in the FrameFixer installation directory, or set the **LM_LICENSE_FILE** environment variable to point to the license server (e.g. 27000@yourserver).

If you are having trouble getting the licensing to work, you can set the environment variable **FRAMEFIXER_IGNORE_NO_LICENSE** in a command shell, then launch FrameFixer and attempt to fix a sequence. At the start of processing, instead of a dialog box informing you that there is no license, you will instead get a FLEXlm license error message, which might point you to the problem, or you can send the output to the support email address for help.

Loading Image Sequences

Of course the first step to getting your image sequence repaired is to load the sequence into FrameFixer. You can load a sequence by selecting the **Open Sequence** option in the **File** menu. You could also press **[CTRL+O]**, or double-click on the empty image area that initially says 'no images loaded'.

This will open a File Sequence Browser where each image sequence is listed as a single entry (e.g. **sequence_005.1-30#.tif**). The sequence format shown in this dialog (and used in the command line options) uses the same syntax made popular by Apple's Shake compositing package. For a description of the sequence syntax see Appendix B.

The File Sequence Browser is normally set to ignore any missing frames, so if you have a sequence with frames 1 to 100, but frame 33 is missing, it would be listed as something like **sequence.1-100#.exr** rather than **sequence.1-32,34-100#.exr**.

The following image formats are currently supported by FrameFixer (please let us know any others that would be helpful to you):

- EXR
- TIFF
- Cineon
- DPX
- SGI
- Maya IFF
- Softimage PIC
- TGA
- JPEG
- PNG

FrameFixer does not immediately load all the images into memory, but loads each frame only when it is first viewed. In the timeline you can see a yellow bar at the bottom of any frames that have had their images loaded. You will be able to quickly move around the timeline and play back your sequence at full speed within the cached range. For long sequences it is quickest to only cache the few frames around those being corrected, so that you can scrub back and forth around those areas to check the processed results.

However, you can also tell FrameFixer to cache all the frames in the playback range, by selecting the **Cache Playback Range** option from the **File** menu or specifying the **-c** option when loading a sequence from the command line. The maximum amount of cache memory to be used is set in the Preferences dialog.

A color lookup table (LUT) can be applied to the preview images, which is often needed when dealing with film images. See the section on Display LUTs under Preferences later in this document.

Marking, Processing and Saving Frames

After loading in a sequence you can move around the timeline or play the clip. Navigate to the frame you wish to repair, and press the **[SPACE]** key to mark it as a bad frame (or press the **Mark/Unmark Current Frame** button in the lower right corner of the window). That frame will then be colored red in the timeline, showing it is marked for processing.

When you have marked the frames you want to process, press **[ENTER]** (or click the **Process Frames** button in the lower right corner) and FrameFixer will generate those images using the current interpolation method and quality settings.

When the processing is complete, the generated frames will be colored in green on the timeline, and if you move to that frame you will see the generated result. You can press the **[S]** key to toggle the display between the original frame (if there is one) and the generated frame, or press **[A]** to toggle the display between the RGB and alpha channels. Scrub back and forth over the frame and its surrounding frames to check the result.

If you are happy with the results press **[CTRL+S]** (or click the **Save Frames** button) to save the results back to disk in the same format as the rest of the sequence. You should then have a complete sequence that will play back smoothly.

If you are not happy with a generated frame, press **[SPACE]** on that frame to unmark it and discard the results. You might then re-process it using different settings (see Interpolation Method and Settings below).

Auto-tagging

Any missing or zero-byte files will be automatically marked when you import the sequence (though this behavior can be turned off in the Preferences dialog).

This simple test is the best that can be done in the general case across all image formats and rendering pipelines. However, in certain situations more sophisticated checking is possible, and you can extend FrameFixer's auto-tagging functionality in the Preferences dialog by specifying a command or script to run on each frame to test if it should be tagged. For example, perhaps the 3D rendering package you use places a tag in the header when it starts writing the image, and clears it when it is complete - you could then have a small utility to check for this flag and report the result back to FrameFixer.

Your custom auto-tagging script will be run once per frame and passed the full path name of the image as a command line argument. It should return status 0 if the image is complete, or any other value if it should be tagged as a bad frame. If a custom script is specified then the default behavior of tagging missing and zero-byte files will be skipped, so your script should also check for these.

To use a custom script, specify its full path in the FrameFixer Preferences dialog.

Interpolation Method and Settings

FrameFixer should produce good results using the default settings, but in some cases you might benefit from adjusting the interpolation method or optical flow settings. These controls can be found in the panel to the right of the image display area.

Interpolation Method

Optical Flow is the usual method of interpolation. This attempts to track the movement of features between the frames before and after the one being processed, and interpolates them to produce the frame in between. In most cases this method gives excellent results, but it can occasionally give poor results with sequences featuring chaotic movement such as lots of small object moving around.

As an alternative to optical flow, FrameFixer can also use simple frame averaging, which mixes together the frames either side of the one being replaced. This will normally show a double vision effect in areas of movement, and can also produce a slight jump when playing back. So although it isn't ideal, frame averaging is a good fall back option for the rare cases when the optical flow results aren't satisfactory.

Original Frame

Optical Flow Interpolation

Frame Averaging

Optical Flow Settings

These controls can be used to fine tune the optical flow interpolation. The default values work well in most cases, but if it is not working well there are plenty of things to adjust here.

These controls match those found in The Foundry's Furnace plugin suite, so if you're familiar with those you should feel right at home here.

Vector Detail: Controls the density of the optical flow field - or how many positions to track across the neighboring frames. A value of 1.0 tracks every pixel in the image, and 0.5 tracks every second pixel. The default value gives a good result for high resolution images (film and HD), but you should probably go closer to 1.0 when working at PAL/NTSC video resolution

Global Smoothness: After tracking features between the neighboring frames, a smoothing process is applied to the optical flow field. The Global Smoothness parameter controls the amount of smoothing applied to larger areas of movement in the image

Local Smoothness: Like Global Smoothness, but this parameter changes how the smoothing affects the finer details

Smoothing Iterations: Controls how many smoothing passes are run over the flow field's motion vectors. A larger value will usually produce a better result, but will take longer to compute

Error Threshold: Differences in pixel values below this threshold are ignored to prevent noise or grain in the images from being interpreted as motion. You may need to raise this value for very noisy or grainy images

Block Size: This parameter controls how the optical flow algorithm finds features in the image. If you are getting poor results for a certain sequence you can try setting this value up or down by a few pixels.

Extreme Filtering: With this option off, bilinear filtering is used to reconstruct the interpolated images. This is fast but can produce soft results when there is a lot of motion. Turn on Extreme Filtering for a much sharper result, but this will take a lot longer to compute

Correct Luminance: If there are luminance changes between the neighboring frames (i.e. the sequence is getting darker or lighter at that point) then you should turn on this option which will equalize the brightness of the two images before calculating the optical flow field

Use Alpha Mattes: FrameFixer will use the alpha channel of the images to help lock on to the edges of objects when calculating the optical flow field. You may need to turn this off if your sequence has an alpha channel containing something other than a matte of your foreground (or all) objects

If you hover the mouse pointer over these controls in the interface, a brief tool tip will be shown describing that parameter.

Image Processing Controls

In situations where the optical flow hasn't been able to lock on to the image well (such as lots of small objects moving around quickly) it may produce a darker or softer result. This is because the optical flow field in this case is quite erratic, and causes a lot of blending to occur within the image.

In this case you can adjust the brightness and sharpness values to more closely match the image to the rest of your sequence. If this is necessary then the final result isn't likely to be particularly good, as it means the interpolation has not gone well, but it can help make a poor result more satisfactory.

To make an adjustment, move to a processed frame and you will notice the Brightness and Sharpness controls become active. Type in a value and press **[ENTER]**. After a brief computation you will see the adjusted result in the image preview. Adjustments are non-destructive, so you can change these values back and forth with no loss of quality.

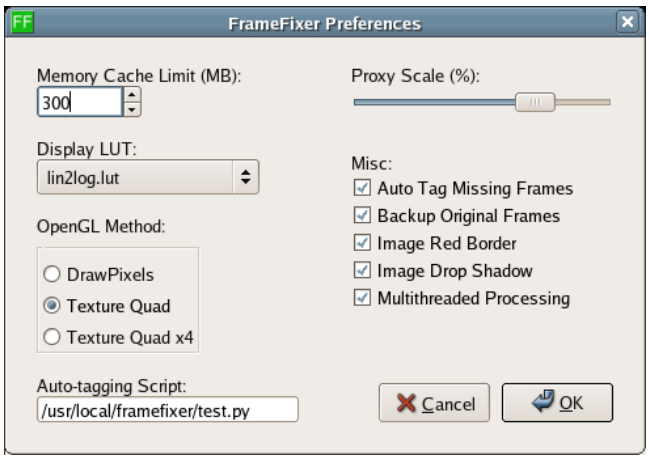
If you get a slightly soft but otherwise good result from the optical flow interpolation, then you should get a better result by reprocessing the frame with the **Extreme Filtering** parameter turned on, rather than sharpening as a post-process.

The default brightness value of 1.0, and sharpness value of 0.0 make no change to the images.

Preferences

You can open the Preferences dialog by selecting the **Preferences** menu item in the **File** menu (relocated under the main **FrameFixer** menu in OS X).

The preferences are saved to a **.framefixer** file in your home directory (which is a hidden file under Linux). A file called **default.prefs** in the FrameFixer installation directory defines the initial preferences.



Memory Cache Limit

Controls how much memory the application will use to cache the proxy display images.

This is initially set to 200MB, but you can set it lower if you're running short on memory, or higher to cache longer sequences for playback. FrameFixer will not always use the full cache limit, shorter sequences can be cached in much less memory. When the cache limit is reached, frames that have not been accessed recently will be cleared from the cache, and the yellow bar will disappear from those frames in the timeline.

Display LUT

FrameFixer can use a custom lookup table (LUT) to modify how colors are displayed. The most common need for this is when dealing with film images, which often have their data encoded in a logarithmic format.

The FrameFixer installation contains a **luts** directory, and any files found there with a **.lut** extension are available in the LUT drop-down menu in the Preferences dialog. The LUT files use the same format as used by Apple's Shake - more details can be found in the **readme.txt** file located in FrameFixer's LUTs directory, along with some example files.

The LUTs are only used for displaying the preview proxy images - no color lookups are ever saved back to your sequences.

OpenGL Method

If you are having trouble with the image display area, or if you have extremely slow playback speed you can try the different options here. This is normally only needed to get around problems with display driver setup under Linux.

Auto Tagging Script

FrameFixer can tag missing or empty files during loading. You can extend this functionality by specifying the full path to a custom auto-tagging script here that is tailored to your own pipeline. See the section on Auto-tagging earlier in this document for more details.

Proxy Scale

FrameFixer can display proxy images at a lower resolution than your sequence. This allows you to scrub and play back much longer sequences that if you were loading the full resolution, and may also speed up loading (particularly if using display LUTs). You can specify the proxy scale here, which defaults to 1.0 (full resolution).

Misc Options

Auto Tag: you can disable FrameFixer's auto tagging functionality here (see the section on Auto-tagging for more information)

Backup Original Frames: When you save a processed frame, FrameFixer will backup the existing frame with the extension **.bak** (or subsequently **.bak02**, **.bak03**, etc)

Image Red Border / Image Drop Shadow: controls whether these are drawn around the image display area

Multithreaded Processing: with this option turned on, FrameFixer will multi-thread the image processing using all local CPUs. It also controls multi-threaded image loading of formats that support it (currently only EXR)

Running from the Command Line

FrameFixer fully supports execution from the command line. As well as being more convenient for some uses, it also allows FrameFixer to be used by other tools within a complex production pipeline. For example, you might configure your render farm to automatically run FrameFixer over the sequence when a render task completes.

To get a full rundown of the available command line options, simply run FrameFixer with **--help** (or **-help**, or **-h**) to display the following usage information:

```
Usage: framefixer [options] [sequence]

  -f,   --fix=<str>            fix the specified frame(s) (e.g. -f 10,32,92) (CMD)
  -m,   --mark=<str>           mark the specified frames(s) (e.g. -m 10,32,92) (GUI)
  -a,   --auto                 fix all auto-tagged frames (CMD)
  -c,   --cache                cache the image sequence being loaded (GUI)
  -g,   --gui                  show GUI even if all parameters are specified
  -ng,  --nogui                dont show GUI, require all parameters on command line
  -t,   --threads=<int>        number of CPUs to use, 0 meaning all (CMD)
  -avg, --average              use simple frame averaging instead of optical flow interpolation
  -vd,  --vectorDetail=<float> optical flow: vector detail
  -gs,  --globalSmooth=<float> optical flow: global smoothness
  -ls,  --localSmooth=<float>  optical flow: local smoothness
  -si,  --smoothIter=<int>     optical flow: smoothing iterations
  -et,  --errorThresh=<float>  optical flow: error threshold
  -bs,  --blockSize=<int>      optical flow: block size
  -ef,  --extremeFilter=<bool> optical flow: extreme filtering
  -cl,  --correctLum=<bool>    optical flow: correct luminance
  -ua,  --useAlpha=<bool>      optical flow: use alpha mattes
  -h,   --help                 show this help message
```

Some of the options only affect the program when running in command line mode, and others are only for graphical mode. These are marked **(CMD)** and **(GUI)** respectively.

If sufficient options are specified on the command line then FrameFixer will process the frames without showing the graphical interface. You can force it to open with interface with the **–gui** flag, or force it to require all options on the command line with the **–nogui** flag.

Appendix A: Keyboard Shortcuts

Key	*Function*
[SPACE]	Mark (or Unmark) the current frame
[ENTER]	Process all marked frames
[CTRL+O]	Open the File Sequence Browser
[CTRL+S]	Save any processed frames back to disk
[CTRL+U]	Unmark All Frames
[LEFT]	Step back one frame
[RIGHT]	Step forward one frame
[CTRL-LEFT]	Jump to the start of the playback range (the in-point)
[CTRL-RIGHT]	Jump to the end of the playback range (the out-point)
[SHIFT-LEFT]	Jump to the previous marked or fixed frame
[SHIFT-RIGHT]	Jump to the next marked or fixed frame
[>] or [P]	Play the sequence (it will read/cache the frames as it goes)
[ESC]	Stop playback
[S]	Toggle between the processed and original frames
[A]	Toggle between displaying the Alpha and RGB channels
[CTRL+L]	Show/Hide the Message Log Window
[CTRL+G]	Set the input focus to the 'goto frame' field
[F1]	Display the Online Documentation

You can view this table from within FrameFixer by clicking **Keyboard Shortcuts** from under the **Help** menu.

If you find that the shortcut keys aren't working, you may need to click on the image area to get the mouse focus back (usually after adjusting the interpolation controls).

Appendix B: File Sequence Syntax

For command line use, and within the file sequence browser, FrameFixer uses the same sequence syntax made popular by Apple's Shake compositing package, which is:

```
BASENAME FRAME_NUMBERS [EXTENSION]
```

Where **FRAME_NUMBERS** can be any number of sub-sequences or single frames, comma-delimited, and the start/end/step of a sub-sequence is specified with **START-ENDxSTEP** (e.g. 1-100x1). **STEP** is optional when its value is one.

Padding is specified by following the sub-sequence with a **#** character for 4 digit padding (e.g. **sequence.0001.exr**), a **@** character for no padding (e.g. **sequence.1.exr**). For other padding widths you repeat the **@** character (e.g. **sequence.@@@.exr** for 3 digit padding).

Although this format is extremely flexible, in most cases you only have a single range of continuous frames, and it is very simple – here are a few common examples:

`sequence.1-50@.exr`	- frames 1 to 50, no padding
`sequence.1-50@@.exr`	- frames 1 to 50, padded 2 digits
`sequence.1-100#.exr`	- frames 1 to 100, padded 4 digits
`sequence.20-40x2@.exr`	- frames 20 to 40, with a step of 2, no padding

Appendix C: End User License Agreement

In using the FrameFixer software you agree to the following terms:

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL TECHNOLUMIERE LIMITED OR ANDREW CHAPMAN BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY OR NEGLIGENCE, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

YOU AGREE TO ABIDE BY THE SOFTWARE LICENSING RESTRICTIONS APPLIED TO THE SOFTWARE AND AGREE TO MAKE NO ATTEMPT TO CIRCUMVENT THESE MECHANISMS. IF YOU HAVE PURCHASED A 'NODE-LOCKED' LICENSE YOU AGREE TO USE THE SOFTWARE EXCLUSIVELY ON THE COMPUTER FOR WHICH THE LICENSE WAS PURCHASED. IF YOU HAVE PURCHASED A 'FLOATING' LICENSE YOU AGREE TO CONCURRENTLY USING AT MOST THE NUMBER OF LICENSES PURCHASED, AND FOR NO LONGER THAN THE PERIOD OF LICENSE VALIDITY PURCHASED. YOU MAY NOT ALLOW THE SOFTWARE TO BE USED BY ANY THIRD PARTY HAVING ACCESS TO YOUR LICENSE SERVER.

YOU MAY NOT SELL, RENT, LEASE OR TRANSFER YOUR LICENSE TO A THIRD PARTY.

NO MODIFICATION IS PERMITTED TO THE 'COMPILED' OR 'BINARY' COMPONENTS OF THE SOFTWARE, AND LIKEWISE NO REVERSE ENGINEERING, DECOMPILATION OR DISASSEMBLY OF THE SOFTWARE IS PERMITTED FOR ANY PURPOSE.