



# **mental ray Release notes**

Version 3.2



© 2003 Alias, a division of Silicon Graphics Limited. All rights reserved.

Printed in the USA. Revised August 2003.

Inside cover image credit: Danny Mousses, Alias.

This documentation ("Documentation") is copyrighted © 2003 Alias a division of Silicon Graphics Limited ("Alias") and contains proprietary and confidential information of Alias. The Documentation is protected by national and international intellectual property laws and treaties. All rights reserved. Use of the Documentation is subject to the terms of the license agreement, beta agreement or other agreement with Alias that governs the use of the software product to which the Documentation pertains (the "Software"). If the Documentation is provided in electronic form only, the authorized licensee of the Software is hereby authorized to print no more than one (1) hardcopy of the Documentation per valid license of the Software held by such licensee, which hardcopy must be immediately destroyed upon the termination or expiry of such license(s). Except for the foregoing, the Documentation may not be translated, copied or duplicated in any form (physically or electronically), in whole or in part, without the express written permission of Alias. THE DOCUMENTATION IS PROVIDED "AS IS" AND ALIAS DOES NOT WARRANT THAT THE DOCUMENTATION WILL MEET YOUR REQUIREMENTS OR THAT THE CONTENTS OF THE DOCUMENTATION ARE ACCURATE OR COMPLETE. CONTENTS ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALIAS MAKES NO WARRANTIES, EXPRESS, IMPLIED OR ARISING BY CUSTOM OR TRADE USAGE RELATING TO THE DOCUMENTATION, AND WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER ALIAS NOR ITS AFFILIATES, PARENT COMPANIES, LICENSORS OR SUPPLIERS SHALL HAVE ANY LIABILITY IN CONTRACT, TORT OR OTHERWISE ARISING OUT OF OR IN CONNECTION WITH THE DOCUMENTATION. mental ray and mental images are registered trademarks of mental images GmbH & Co. KG in the U.S.A. and/or other countries. Alias, the Alias logo and Can You Imagine are trademarks of Alias, a division of Silicon Graphics Limited. Maya and the Maya logo are registered trademarks of Silicon Graphics, Inc., exclusively used by Alias, a division of Silicon Graphics Limited. Windows, Windows NT, Windows 2000 Professional and Explorer are trademarks of Microsoft Corporation, registered in the United States and/or other countries. IRIX is a registered trademark of Silicon Graphics, Inc. Netscape is a registered trademark of Netscape Communications Corporation. Pentium is a registered trademark of Intel Corporation. Linux is a registered trademark of Linus Torvalds. Mac is a registered trademark of Apple Computer, Inc. All other trademarks, trade names, service marks, or product names are property of their respective owners.



# Table of Contents

<b>1</b>	<b>Release notes</b>	<b>7</b>
	Memory and disk management improvements	7
	Memory issues	7
	Panic mode	7
	Swap directory	7
	Disk swapping statistics	7
	Memory Allocator	7
	Heap allocator	8
	Disk swapping cleanup	8
	Network rendering improvements	8
	General texture improvements	9
	Color texture files	9
	Parsing when a texture file exists	9
	Texture cache statistics	9
	mi_query	9
	Images with one bit per pixel	9
	General shader improvements	9
	Geometry shaders	9
	Image data created by output shaders	10
	Complex Phenomena	10
	Phenomena with photon volume shaders	10
	NaN results	10
	miQ_PIXEL_SAMPLE mode	10
	Shader return codes	10
	Duplicate parameters	10
	Shader interface function mi_finalgather_store	11
	mi_api_lightprofile_lookup shader API function	11
	Material shader definitions and Rapid Motion	11
	Rendering with no image file or output shader statements in the camera	11
	A shader returns a scalar, color, vector, or matrix containing a NaN value	11
	Side-effect tags	11

## Table of Contents

Hair rendering improvements . . . . .	11
Hair shadow flag . . . . .	12
Raytracing hair . . . . .	12
Barycentric coordinates . . . . .	12
Hair rendered in Rapid Motion . . . . .	12
Invalid barycentric coordinates . . . . .	12
Rapid Motion rendering with hair geometry . . . . .	12
Hair rendering in Rapid Motion mode . . . . .	12
Radius values attached to individual hairs . . . . .	12
Hair rendering performance . . . . .	12
Lighting improvements . . . . .	13
Rendering with light profile light sources . . . . .	13
Directional light photon emission . . . . .	13
Photon emission for cylindrical area light sources . . . . .	13
IES light profiles . . . . .	13
Shadow and light map improvements . . . . .	13
Shadowmaps rendered with OpenGL . . . . .	13
Motion-blurred shadowmap shadows . . . . .	13
Shadowmaps . . . . .	14
Shadowmap merging . . . . .	14
Tessellation improvements . . . . .	14
Fine displacement . . . . .	14
Tessellator . . . . .	14
Immediate-mode tessellation . . . . .	14
Tessellation results cached from frame to frame . . . . .	15
Miscellaneous improvements . . . . .	15
echo command . . . . .	15
Hyperthreading . . . . .	15
Filter and Rapid Motion . . . . .	15
Degenerate triangles . . . . .	15
Elliptical filtering problems . . . . .	15
Elliptical filtering performance . . . . .	16
Real-time image streams . . . . .	16
Tagged objects . . . . .	16

## Table of Contents

-motion on off command . . . . .	16
Multipass rendering pass files . . . . .	16
Multipass rendering . . . . .	16
Rapid Motion rendering statistics . . . . .	16
Subdivision surface object referencing a non-existing vertex. . . . .	17
Incremental changes to an instance. . . . .	17
db summary command . . . . .	17
Messages such as wallclock/CPU timing . . . . .	17
Missing displacement . . . . .	17
-h command . . . . .	17

## **Table of Contents**

mental ray 3.2 Release notes	
6	

# 1

# Release notes

This document lists fixes made for this release and any new limitations and workarounds for mental ray standalone version 3.2.6. This version fixes a number of reported issues and does not introduce new features.

## Memory and disk management improvements

The following lists improvements, limitations and workarounds you should know about memory and disk management.

### Memory issues

If memory exceeded available limits, a garbage collection process could have deleted data that was simultaneously accessed, which crashed mental ray. This was observed during Rapid Motion rendering. This now works as expected.

### Panic mode

Normally, when mental ray needs more memory, it evicts data from the geometry cache. If that fails to make enough memory available, it performs global memory collection. If that fails as well, it goes into “panic mode” and deletes all kinds of acceleration data structures that are not absolutely needed to continue, regardless of the performance impact. The panic mode code could crash if the panic situation happened twice in a row. This case is now handled properly.

### Swap directory

The swap directory is no longer cleared between frames.

### Disk swapping statistics

Added disk swapping statistics, reporting the amount of data written to and read from disk. (See `-swap_*` on the command line.)

### Memory Allocator

Improvements have been made to the new memory allocator. In particular bad memory statistics showing NaN values, and random early aborts are now resolved.

## 1 | Release notes

### > Heap allocator

#### Heap allocator

General improvements have been made to the heap allocator allowing very small scenes use more memory due to larger blocking, but larger scenes use less memory due to reduced fragmentation and better paging. Performance has also improved slightly due to this.

Previous incorrect memory statistics messages have been corrected.

#### Disk swapping cleanup

Exposed disk swapping cleanup for dead processes on Windows. When mental ray starts up and shuts down, it searches the swap directory specified with the `-swap_dir` command line option for files that belong to processes that no longer run and deletes them. (Different machines should never share a swap directory.)

#### Network rendering improvements

The following lists improvements, limitations and workarounds you should know about network rendering.

- Slave data now bypasses the master as much as possible, reducing memory usage on the master. This fixed a situation when the network master ran out of memory and the data created by the slave was unavailable.
- Improved the communication between slaves that could get interrupted by an out-of-memory condition on the master.
- mental ray negotiates version compatibility when establishing a connection. While these issues are not considered so severe that mismatched slave versions are rejected, master/slave version mismatches are not recommended in general.
- If a slave version 3.2.6 (and later) is connected to a pre-3.2.6 master, risk of failure can actually increase.
- When rendering with network slaves and hosts, overloading the master host with local compute requests causing it to run out of resources should no longer occur.
- A networking problem upon mental ray start up has been fixed: when a race condition could cause slaves to attempt to load shader libraries before the slave had completed the rendezvous handshake with the master, this would sometimes cause misdirected network messages. This matter has now been resolved.
- The amount of licensing network traffic has been reduced to allow a license server to manage a much larger number of clients.



## General texture improvements

The following lists improvements, limitations and workarounds you should know about textures.

### Color texture files

Color texture files containing high dynamic range (RGBE) data were clipped to the [0, 1] range. They are no longer clipped.

### Parsing when a texture file exists

mental ray could abort if a texture file existed when parsing the scene, but no longer existed or was unreadable during rendering. These cases are now handled properly.

### Texture cache statistics

The texture cache statistics printout has been improved, listing megabytes read and written.

### mi\_query

When checking the number of textures on a subdivision surface object with mi\_query, an incorrect number could be returned for certain combinations of texture and motion spaces. The correct value is now returned.

### Images with one bit per pixel

Images with one bit per pixel, as used by the pixel previewing facility that re-renders only pixels that involved shaders that changed, were handled incorrectly because their size was miscalculated. This now works as expected.

## General shader improvements

The following lists improvements, limitations and workarounds you should know about shaders.

### Geometry shaders

Incremental changes to geometry shaders that generated geometry during a previous frame could crash mental ray. This is now handled properly.

## 1 | Release notes

### > Image data created by output shaders

#### Image data created by output shaders

Exposed a feature that sends image data created by output shaders after rendering to real-time streams, such as imf\_disp viewer connections.

#### Complex Phenomena

Complex Phenomena render more efficiently now.

#### Phenomena with photon volume shaders

Phenomena with photon volume shaders that use interface assignments could fail. They are now stable.

#### NaN results

Shader results are no longer checked for not-a-number (NaN) values by default. This check can reduce performance noticeably in scenes that call large numbers of shaders, such as some Phenomena, and once a shader is debugged the test is not normally necessary. The test can be explicitly enabled with the command line option `-message phen debug`. (This option can also be put into the `MI_RAY_OPTIONS` environment variable to permanently re-enable NaN checks.)

#### miQ\_PIXEL\_SAMPLE mode

Exposed new `miQ_PIXEL_SAMPLE` mode to the `mi_query` shader interface function. It returns an array of two QMC sequence numbers, and installs them in the state. This is useful in lightmap shaders to achieve low-discrepancy sampling.

#### Shader return codes

Shader return codes are checked for not-a-number (NaN) values, but not all shaders returning colors set all four RGBA components. For example, light shaders only return RGB. The not-a-number check now checks only those components that actually get used to avoid false error messages.

#### Duplicate parameters

Duplicate parameters in shader or phenomenon declarations are now ignored and print an error message. Duplicate parameters confused later shader instance definitions where only the first was accessible.

**> Shader interface function `mi_finalgather_store`****Shader interface function `mi_finalgather_store`**

The shader interface function `mi_finalgather_store` could crash mental ray. This now works as expected.

**`mi_api_lightprofile_lookup` shader API function**

The `mi_api_lightprofile_lookup` shader API function failed when the name referenced a non-existing profile after printing the error message. This now works as expected.

**Material shader definitions and Rapid Motion**

In Rapid Motion mode, rendering could abort if a material shader definition failed, for example, due to a shader library that could not be found. Now it renders black instead.

**Rendering with no image file or output shader statements in the camera**

In situations when rendering with no image file or output shader statements in the camera, mental ray would interpolate the color frame buffer. It now defaults to “+rgba” instead of “-rgba”.

**A shader returns a scalar, color, vector, or matrix containing a NaN value**

A warning message is printed if a shader returns a scalar, color, vector, or matrix containing a NaN (not-a-number) value. Other types, in particular structs, are not checked. Such values are usually the result of an incorrect mathematical operation such as taking the square root of a negative scalar number, and typically cause flat white or black pixels in the rendered image. The new warning may help diagnose scenes more easily.

**Side-effect tags**

In some cases geometry shaders created extra side-effect tags, that were deleted at the end of the frame, while the geometry object itself may have been cached for the next frame. This was a problem when the object attempted to reference the extra tags, causing access errors. This has been resolved and the same delete rules apply for both object and their tags.

**Hair rendering improvements**

The following lists improvements, limitations and workarounds you should know about rendering hair.

## **1 | Release notes**

### **> Hair shadow flag**

#### **Hair shadow flag**

Hair could cast shadowmap shadows even if the object or instance shadow flag was turned off. The flag is now respected.

#### **Raytracing hair**

In cases where a ray traced by a shader using `mi_trace_probe` hit a hair object, the hair's material shader information was not put into the state for the caller to evaluate. The information can now be evaluated.

#### **Barycentric coordinates**

Barycentric coordinates supplied to hair material shaders in Rapid Motion mode were not correct. They are now correct.

#### **Hair rendered in Rapid Motion**

mental ray could fail if hair was rendered in Rapid Motion mode with many processors, if multiple processors happen to render the same hair. This case is now handled properly.

#### **Invalid barycentric coordinates**

Downgraded the warning “invalid barycentric coordinates” to a debug message. This situation is harmless; it’s caused by easily recoverable rounding errors.

#### **Rapid Motion rendering with hair geometry**

Fixed Rapid Motion rendering with hair geometry containing textures and user data.

#### **Hair rendering in Rapid Motion mode**

Implemented limited hair rendering in Rapid Motion mode, by creating triangles on the fly and passing them to Rapid Motion-style material shaders. Access to the original hair cylinders is not available.

#### **Radius values attached to individual hairs**

Radius values attached to individual hairs did not work and could crash mental ray. Also, hair objects did not properly inherit visible, trace, and shadow flags from the instance tree. These issues are both working correctly now.

#### **Hair rendering performance**

Significantly improved hair rendering performance.

## Lighting improvements

The following lists improvements, limitations and workarounds you should know about lighting.

### Rendering with light profile light sources

Rendering with light profile light sources produced artifacts at the angle boundaries. They also resulted in wrong coefficients for non-equidistant grids and higher order approximation. The default resolution has been increased for higher-order approximation. These issues have been resolved.

### Directional light photon emission

Fixed a crash in directional light photon emission in the photon autovolume mode.

### Photon emission for cylindrical area light sources

Fixed photon emission for cylindrical area light sources.

### IES light profiles

IES light profiles with tilt could create out-of-bounds data and abort rendering. This now works as expected.

## Shadow and light map improvements

The following lists improvements, limitations and workarounds you should know about shadow and light maps.

### Shadowmaps rendered with OpenGL

Shadowmaps rendered with OpenGL with a nonzero softness had self-shadowing problems. They are now correct.

Tagged objects require tag integers in the geometry definition. If they were missing, an error was caused. If the object material contained a lightmap, the error caused a crash instead of recovering. This case is now handled properly.

### Motion-blurred shadowmap shadows

The sample filtering code often caught not the front-most but the next shadow depth, causing the shadow to shrink or disappear. This code now behaves as expected.

## 1 | Release notes

### > Shadowmaps

#### Shadowmaps

Shadowmaps that did not include hair geometry now behave correctly.

#### Shadowmap merging

Shadowmap merging has been exposed. If the options contain the statement shadowmap rebuild merge and a light source contains the statement shadowmap merge on and a file statement, the shadow map will be read in from disk as usual, but whenever it is consulted, the point will be considered in shadow if either the loaded shadow map or the rebuilt shadow map shows that it is in shadow. This is useful to import shadow maps from other passes with extra shadowcasting objects that are not present in the current pass.

#### Tessellation improvements

The following lists improvements, limitations and workarounds you should know about tessellation.

##### Fine displacement

Fine displacement of view-dependent used a heuristic to reduce the subdivision recursion limits, as specified at the end of approximation statements. This is no longer the case because the limit caused nearby objects to show artifacts. For distant objects this change can cause higher tessellation. If triangle counts go up significantly, check approximation subdivision limits.

##### Tessellator

The tessellator could get motion vectors and derivatives mixed up for NURBS objects that requested derivatives and had motion. The incorrect result of objects with incorrect or seemingly missing motion blur has been resolved.

##### Immediate-mode tessellation

Immediate-mode tessellation (mi\_geoshader\_tessellate shader interface function) did not handle instance approximation overrides correctly. This now works as expected.

> Tessellation results cached from frame to frame

### Tessellation results cached from frame to frame

mental ray caches tessellation results from frame to frame. Sometimes it is useful to discard the tessellation of an object, for example, if it is known that the displacement shader would make different decisions in the next frame. This can now be done with the touch “objectname” command in .mi scene files.

### Miscellaneous improvements

The following lists miscellaneous improvements.

#### echo command

The `-echo` command line option did not echo pass merge statements in camera blocks correctly; the square brackets were missing. They are now present.

Files created by the `-echo` incremental command line option no longer print comments for omitted elements into the file. These comments could make the file much larger than necessary. The comments are now optional and need to be enabled with the `- message echo debug` - command line option.

#### Hyperthreading

In some cases when an odd number of threads were requested on a hyperthreaded host, undesirable results would occur. (Hyperthreading means that each CPU runs exactly two threads.) This matter has been resolved.

#### Filter and Rapid Motion

In Rapid Motion mode with Z buffer output and filtering enabled, the filter could pick the wrong Z value. This usually caused inappropriate zero depths and incorrect compositing orders. This now works as expected.

#### Degenerate triangles

Degenerate triangles no longer print large numbers of error messages.

#### Elliptical filtering problems

Problems observed in elliptical filtering on Pentium 4 machines have been resolved. This involved upgrading to the Intel C compiler.

## 1 | Release notes

### > Elliptical filtering performance

#### Elliptical filtering performance

Elliptical filtering performance has improved. The new algorithm introduced was designed to make filtered textures look much sharper, including support for filter values less than 1.0. Texture lookups are now weighted with a Gaussian filter, which makes it somewhat slower than before. Filtering can still be turned off with the bilinear flag.

#### Real-time image streams

Real-time image streams directed to callbacks installed with `mi_disp_stream_cb_begin` cause two callbacks per rendered tile, one when the tile starts and one when it finishes. Contrary to the documentation, the first call provided a black image rectangle instead of no rectangle at all. This now works as described in the documentation.

#### Tagged objects

Tagged objects got incorrect contours in contour rendering mode. This now works as expected.

#### -motion on|off command

The `-motion on|off` command line option did not work. This now works as expected.

#### Multipass rendering pass files

Multipass rendering pass files are now saved correctly in Rapid Motion (-scanline rapid) mode if the filter size is greater than 1. Previously, the filter size was misinterpreted when reading the file later.

#### Multipass rendering

The multipass rendering component now supports variable filter sizes and Rapid Motion.

#### Rapid Motion rendering statistics

Rapid Motion (-scanline rapid) now prints rendering statistics. This includes tessellated triangles, shading candidates (points selected on triangles for shading), shading samples (points that actually did get shaded because they were visible and got hit by an eye ray), and collect samples (the number of points picked up during collection when filtering samples to pixels).



### > Subdivision surface object referencing a non-existing vertex

If the sampling parameters are very low, collection can cause extra eye rays to fill in gaps. This causes rendering performance to become highly dependent on the collection rate (-samples\_collect). Normally, if there are sufficient shading points to begin with as a result of higher sampling parameters, rendering performance is not affected much by the collection rate.

### Subdivision surface object referencing a non-existing vertex

If a subdivision surface object references a nonexistent vertex, mental ray now prints an “invalid space for parametrization” error message instead of failing as it did before.

### Incremental changes to an instance

Incremental changes to an instance lost the approximation list in the instance. This now works as expected.

### db summary command

The problems with the debug “db summary” command in .mi scene files have been resolved.

### Messages such as wallclock/CPU timing

When mental ray terminates, it sometimes failed to print the last few messages such as wallclock/CPU timing messages. This now works as expected.

### Missing displacement

Fixed problem with missing displacement for certain motion and texture vector combinations in subdivision surface objects.

### -h command

Fixed the -h (short form of -help) command line option.

## **1 | Release notes**

> -h command