OmniObjectMeter 2.5 (v24)

Relative to 2.5 beta 2, current as of 80333

Known Issues

- Using a symbol ordering file during linking will confuse OmniObjectMeter. Setting your Debug build style to not have a symbol ordering file will avoid this issue.

General

- Minor updates to online Help.
- Turned off some debug logging for release builds.

Performance

- Caches demangled C++ symbol names.
- Fixed some memory leaks in OOM itself. OOM still can't run on itself, sadly, or this would have been trivial.

Interface

- Zombies are again selected correctly in the UI when hit (with zombie support enabled in preferences).
- The restart button no longer warns if the target isn't currently running. Use 'Restart' in the warning instead of 'Terminate' if we we are restarting. Also, use sheets for both the restart and terminate warnings.
- Multiple clicks on the app icon no longer create multiple open panels.
- Added a 'Send Feedback' menu item.
- Updated to latest About panel.

OmniObjectMeter 2.5 beta 2 (v23)

Relative to 2.5 beta 1, current as of r71515

Known Issues

- Using a symbol ordering file during linking will confuse OmniObjectMeter. Setting your Debug build style to not have a symbol ordering file will avoid this issue.

General

- OmniObjectMeter now requires Mac OS X 10.4. If you still need to test on 10.2 or 10.3, older versions of OmniObjectMeter can be download from ftp.omnigroup.com.

Intel Support

- OmniObjectMeter now (again!) supports running on Intel hardware.

Interface

- The stack outline panel now remembers what settings you had selected on your last usage and restores them. For example, if you'd chosen to see Autorelease events on Live and Dead objects, counting whole instances, these settings will be the default for future stack outline panels. This saves time when you are trying to track down similar problems across multiple classes or runs of the application.

OmniObjectMeter 2.5 beta 1

Relative to 2.0, Current as of r69929

Known Issues

- Using a symbol ordering file during linking will confuse OmniObjectMeter. Setting your Debug build style to not have a symbol ordering file will avoid this issue.

General

- OmniObjectMeter now requires Mac OS X 10.4. If you still need to test on 10.2 or 10.3, older versions of OmniObjectMeter can be download from ftp.omnigroup.com.

Interface

- Unified toolbar for 10.4
- New default toolbar layout that makes more sense
- Fixed labels and shortcuts on various menu items to be more HIG compliant.
- Flipped restart toolbar image.
- If you launch OOM and connect to an application that is directly linked to the OOM framework and is paused due to hitting a zombie, the zombie object will be automatically shown in OOM.

Stability

- Fixed a hard-to-duplicate crasher in the communication channel between OOM and the target process.

Symbol Gathering

- C++ symbol names now get demangled.
- OmniObjectMeter can now deal with the radically different dyld implementation introduced with Mac OS X 10.4.
- Reenabled & fixed support for reading Mach-O files that begin with a fat_header. Who thought we'd ever need that again?
- Fixed reading symbols out of frameworks that have been relocated by dyld.
- Fixed cases where an unknown symbol would be reported as a symbol from some other module.

Xcode integration

- No special plugin is needed to integrate OmniObjectMeter with Xcode as of this release. OOM now uses AppleScript to communicate with Xcode 2.x. Please make sure you are using the latest version of Xcode you can (Xcode 2.2 preview 1 is available to ADC members on connect.apple.com).

AppleScript

- OmniObjectMeter now uses sdef as its native scripting definition format. As part of this, we made the script suite act a little more like a normal document-based application's suite.
  - The application has a 'documents' array instead of 'targets'
  - Documents have an 'id' property
  - Documents have a 'name' property
  - Documents have a 'mark range' property that can be used to set the currently selected mark range.
  - The application represented by a document can be started, stopped via AppleScript
  - New marks can be created using the 'begin mark' command.
-

Event Collection

- If zombies are enabled in OmniObjectMeter's preferences, OOM will now set the CFZombieLevel environment variable such that CoreFoundation objects should be scribbled over on deallocation and the underlying memory shouldn't be freed.
- OOM no longer clobbers DYLD environment variables in the launched target other than DYLD_INSERT_LIBRARIES.
- Objects that seem to appear out of thin air as far as OOM is concerned (it never saw an allocation event for them) are listed in a new top level '(Unknown)' group.
- OOM now deals better with objects with custom CoreFoundation allocators, resulting in fewer objects appearing out of thin air!
- The environment that OmniObjectMeter is launched with will be used as the basis for the environment used to launch the target. In particular, settings from your ~/.MacOS/environment.plist will be passed along.

Public API

- The OOMPublic.h header now has C++ header guards.
- Added new public API points

  __OOMNumberOfCategories
  __OOMGetCategoryName
  __OOMGetCategorySummary