MemoryScape New Features Guide

version 3.0

TotalView Technologies

Contents

New Features: Version 3.0

New Features: Version 3.0

This document describes changes made to MemoryScape for version 3.0. While this information lets you know what changes have occurred, it doesn't describe these changes. Instead, you'll find descriptions for these changes within the *Debugging Memory Problems With MemoryScape Guide*.

New Platforms and Compilers

We have added support for new versions of operating systems and compilers. You'll find a complete list of supported platforms and compilers in the *MemoryScape Platforms and System Requirements Guide*.

New and Changed Features

Interoperability with TotalView

MemoryScape now offers greatly increased interoperability with the TotalView debugger. You can launch TotalView from within an already running MemoryScape session, if you want to examine specific variables or exert more precise control over programs that you are debugging. When you enable memory debugging within a TotalView session it will bring up the MemoryScape GUI.

Support for heterogeneous debugging

MemoryScape supports several forms of heterogeneous debugging, where the operating system and/or architecture differ. For example, from a Linux x86-64 session you can debug remote processes on Linux Cell.

This table shows the supported combinations:

Host System	Target System
Linux x86-64	Linux x86 Linux x86-64 Linux Power 32 Linux Power 64 / Cell SiCortex Cray XT
Linux x86	Linux x86 Linux Power 32 Linux Power 64 / Cell
Linux Power 64 (including Linux Cell)	Linux Power 32 Linux Power 64 / Cell Blue Gene
SiCortex Linux x86-64	Linux MIPS 64

Support for Power PC32 cross debugging

MemoryScape now supports debugging PowerPC32 embedded applications. Support is delivered through a cross debugger. The host system (where MemoryScape is running) must be one of the following platforms:

- x86-64 Linux
- x86 Linux
- Power64 Linux
- Cell Linux

Red Zones for Linux

The Red Zones feature added to MemoryScape 3.0 provides instant array bounds detection for Linux systems. MemoryScape can immediately detect when a program tries to access memory beyond the allocation bounds.

- Red Zone events: MemoryScape uses Red Zones to detect access violations before and after allocated memory bounds. It can also detect when a program accesses memory that has been deallocated. MemoryScape will stop the program's execution and raise an event alerting you to the illegal access and allowing you to see exactly where the code overstepped the bounds.
- Red Zone allocation size range controls: Red Zones will increase the memory consumption of your program. To reduce this impact, special controls have been added to give you full control over how and when Red Zones are applied to allocated memory. You can restrict Red Zones to allocations in several user-defined size ranges and easily turn Red Zones on or off at any time during a program's execution.
- Red Zone support in the CLI, TVScript, and MemScript: Red Zones are supported in the CLI and TVScript and MemScript via new commands and command qualifiers. The appropriate product documentation provides the details.

Support for malloc zones on Mac OS X

Mac OS X provides a mechanism for multiple pools of memory called malloc zones. MemoryScape now tracks both the allocator and owner of all heap allocations. These properties can be displayed and used for filtering.

Hoard low memory detection

When you ask MemoryScape to hoard deallocated memory, you may increase the risk of running out of available memory earlier than expected. MemoryScape now has the capability to reduce this risk and alert you when you are at risk.

- Hoard low memory controls: you can tell the hoard to automatically release its memory when available memory gets low, allowing the program to run longer.
- Hoard low memory events: MemoryScape can stop execution and notify you when the hoard drops below a defined threshold, so that you know the program is getting close to running out of memory.
- Hoard low memory support in the CLI, TVScript, and MemScript: this feature is supported in the CLI, TVScript, and MemScript via new commands and command qualifiers. The appropriate product documentation provides the details.