Copyright (c) 2003, 2005, Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions

are met:

- Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

- Neither the name of Oracle nor the names of its

contributors may be used to endorse or promote products derived

from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS

IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Header for -agentlib:hprof (or -Xrunhprof) ASCII Output (JDK 5.0 JVMTI based)

WARNING! This file format is under development, and is subject to

change without notice.

This file contains the following types of records:

THREAD START

THREAD END mark the lifetime of Java threads

TRACE represents a Java stack trace. Each trace consists

of a series of stack frames. Other records refer to

TRACEs to identify (1) where object allocations have

taken place, (2) the frames in which GC roots were

found, and (3) frequently executed methods.

HEAP DUMP is a complete snapshot of all live objects in the Java

heap. Following distinctions are made:

ROOT root set as determined by GC

CLS classes

OBJ instances

ARR arrays

SITES is a sorted list of allocation sites. This identifies

the most heavily allocated object types, and the TRACE

at which those allocations occurred.

CPU SAMPLES is a statistical profile of program execution. The VM

periodically samples all running threads, and assigns

a quantum to active TRACEs in those threads. Entries

in this record are TRACEs ranked by the percentage of

total quanta they consumed; top-ranked TRACEs are

typically hot spots in the program.

CPU TIME is a profile of program execution obtained by measuring

the time spent in individual methods (excluding the time

spent in callees), as well as by counting the number of

times each method is called. Entries in this record are

TRACEs ranked by the percentage of total CPU time. The

"count" field indicates the number of times each TRACE

is invoked.

MONITOR TIME is a profile of monitor contention obtained by measuring

the time spent by a thread waiting to enter a monitor.

Entries in this record are TRACEs ranked by the percentage

of total monitor contention time and a brief description

of the monitor. The "count" field indicates the number of

times the monitor was contended at that TRACE.

MONITOR DUMP is a complete snapshot of all the monitors and threads in

the System.

HEAP DUMP, SITES, CPU SAMPLES|TIME and MONITOR DUMP|TIME records are generated

at program exit. They can also be obtained during program execution by typing

Ctrl-\ (on Solaris) or by typing Ctrl-Break (on Win32).