GemStone®

Visual Statistics Display (VSD) User's Guide

Version 4.0

February 2015

INTELLECTUAL PROPERTY OWNERSHIP

COPYRIGHTS

PATENTS

GemStone software is covered by U.S. Patent Number 6,256,637 "Transactional virtual machine architecture", Patent Number 6,360,219 "Object queues with concurrent updating", Patent Number 6,567,905 "Generational garbage collector with persistent object cache", and Patent Number 6,681,226 "Selective pessimistic locking for a concurrently updateable database". GemStone software may also be covered by one or more pending United States patent applications.

TRADEMARKS

GemTalk Systems
15220 NW Greenbrier Parkway
Suite 240
Beaverton, OR 97006

Preface

About This Documentation

This manual describes how to use the graphical tool, Visual Statistics Display (VSD), to examine the statistics generated by a GemStone/S 64 Bit or 32-bit GemStone/S repository, or by GemBuilder for Smalltalk (GBS). This information can be used for understanding the performance of your system and diagnosing problems, and for tuning your GemStone configuration. This manual does not describe the underlying processes, or how to tune your system; that information is specific to the GemStone product and version, and described in *System Administration Guide* for your server product and version.

Terminology Conventions

The term "GemStone" is used to refer to the server products GemStone/S 64 Bit and GemStone/S, and the GemStone family of products; the GemStone Smalltalk programming language; and may also be used to refer to the company, now GemTalk Systems, previously GemStone Systems, Inc. and a division of VMware, Inc.

Technical Support

Website

http://gemtalksystems.com

GemTalk's website provides a variety of resources to help you use GemStone products:

- **Documentation** for the current and for previous released versions of all GemTalk products, in PDF form.
- **Product download** for the current and selected recent versions of GemTalk software.
- **Bugnotes**, identifying performance issues or error conditions that you may encounter when using a GemTalk product.
- **TechTips**, providing information and instructions that are not in the documentation.

- **Compatibility matrices**, listing supported platforms for GemTalk product versions.

This material is updated regularly; we recommend checking this site on a regular basis.

Help Requests

You may need to contact Technical Support directly, if your questions are not answered in the documentation or by other material on the Technical Support site. Technical Support is available to customers with current support contracts.

Requests for technical assistance may be submitted online, by email, or by telephone. We recommend you use telephone contact only for more serious requests that require immediate evaluation, such as a production system down. The support website is the preferred way to contact Technical Support.

Website: http://techsupport.gemtalksystems.com

Email: techsupport@gemtalksystems.com

Telephone: (800) 243-4772 or (503) 766-4702

When submitting a request, please include the following information:

- Your name and company name.
- The versions of all related GemTalk products, and of any other related products, such as client Smalltalk products.
- The operating system, version, and operating system platform you are using.
- A description of the problem or request.
- Exact error message(s) received, if any, including log files if appropriate.

Technical Support is available from 8am to 5pm Pacific Time, Monday through Friday, excluding GemTalk holidays.

Training and Consulting

GemTalk Professional Services provide consulting to help you succeed with GemStone products. Training for GemTalk products is available at your location, and training courses are offered periodically at our offices in Beaverton, Oregon. Contact GemTalk Professional Services for more details or to obtain consulting services.

Table of Contents

Chapter

1 VSD Overview

VSD (Visual Statistics Display) is a graphical tool that allows you to view and interpret the statistical information that can be generated by a running GemStone server or by GemBuilder for Smalltalk. GemStone utilities write this statistical data to a formatted file, and the VSD utility reads these files and displays the data graphically.

1.1 Statmonitor and VSD

Every commercial aircraft in service contains a flight data recorder, commonly called a black box. Black boxes are used to help determine what went wrong if the aircraft ever crashes. Without it, crash investigators would have very little to go on and the cause of many crashes would never be known.

In a production GemStone application, the *statmonitor* program serves the role of the black box. It runs in the background, logging the values of all cache statistics to a disk file. Should a problem with the repository occur, you can review these statistics to determine the cause, allowing you to identify problems and tune GemStone applications for better performance.

Data alone, however, is not enough; you need tools to display this data to allow analysis. VSD allows you to view and interpret the statistical information gathered by GemStone tools such as statmonitor. VSD reads the statmonitor files and displays the values in a graph. This graphical user interface gives you a meaningful way to see how your GemStone system changes over time, based on periodic samples of the system's state.

Many questions about repository performance are impossible to answer after the fact unless statistics data files are available. For example:

- Why did `markForCollection` take longer than normal?
- Why did the backup run slower than before?
- Why was there more repository growth than normal?
- Which Gem session was consuming the most CPU or I/O during a time of poor performance?
- Which Gem had the most number of commit failures?

If you're trying to answer questions such as these, GemStone statistics are your best source of information.

While VSD is a standalone executable, it requires statmonitor or GBS Stat Monitor to generate data:

- *statmonitor* is a stand-alone GemStone server executable with a command-line interface. This is provided as a utility with your GemStone server product and version, and is required to be the same product and version as the shared page cache it will be attached to. It does not log into GemStone nor use a GemStone session. Statmonitor writes data files to disk, in a platform- and version- independent format, in a text format.
- *GBS Stat Monitor* is a GBS/VW process that can be enabled and disabled within a VW image. Like statmonitor, this writes data files to disk, in a platform- and version-independent format, as a binary .gfs file
- *VSD* provides a graphical user interface for viewing the files that statmonitor or GBS produces. The VSD executable is provided for a variety of platforms.

  Any version of VSD running on any platform can load statistical data files generated from any other platform and version of the GemStone server or GBS, although older versions of VSD may not be able to read statmonitor files produced by more recent GemStone server or GBS versions.

Statmonitor in the GemStone server

The statmonitor executable is provided as part of the GemStone/S 64 bit or 32-bit GemStone/S Server, and is described in the *System Administration Guide* for that version of the server.

Statmonitor requires a shared page cache, either the Stone's shared page cache, if the stone is running on that machine, or a remote shared page cache; or, in some versions of GemStone, the -X option can be used to collect system statistics.

For example, to start statmonitor, at the command line, enter a command similar to:
`statmonitor -f` *outputFile* `-i` *sampleInterval stoneOrCacheName*

Statmonitor provides a large number of command line options to control monitoring, including the number and range of processes to monitor and how long to run. These options are provided in the *System Administration Guide* for your platform, or by executing `statmonitor -h` at the command line.

Statmonitor can be terminated by CTRL-C, or when a limit is reached as specified on the command line, and will shut down when the repository is shut down.

GemStone recommends running statmonitor at all times for production systems. If the GemStone system encounters an error, it may be able to report the nature of the error, but diagnosis of what led to that error often requires understanding previous activity and the actions of other processes, which can only be understood from statmonitor data. And some important questions, such as why you ran out of memory or why the repository is out of disk space, can only be analyzed using data from before the problem occurred.

Statmonitor statistics data files

Statmonitor writes text files to disk, either as plain text or in gzip compressed form.

These files contain formatted information, platform- and version- independent, that is read and displayed in VSD. VSD does not directly connect to processes in the GemStone server, but it can startup the statmonitor utility via the command line interface, and read the output files that are generated by statmonitor.

The statmonitor data file format changes very rarely. The latest version of VSD can always read statmonitor files generated by older as well as recent versions of the GemStone server.

These files normally have the extension .out or .out.gz, although you may specify any file name.

File Size

For each sample, statmonitor writes a line for each process being monitored to a data file. This data file continues to grow as long as statmonitor is running.

With small intervals and a large number of processes, over time this file may become very large. A longer sampling interval will keep file sizes more reasonable, but may not be fine enough granularity to catch certain types of problems.

Writing the file in compressed format will save space.

Statmonitor includes options to regularly terminate the statmonitor file and begin a new one, either by time or by the number of samples collected. For example, you may wish to start a new file every 24 hours, and delete old files after a few days.

When you are diagnosing specific behaviors, a 1-second sample is the most convenient, although sub-second intervals may be needed. Monitoring running systems will usually require a longer interval to keep the file sizes manageable.

GBS Stat Monitor in GBS

GBS/VW can be configured to collect statistics describing the performance of its internal operations. These statistics are written to a file named GBSStatArchive-<dateAndTime>.gfs in the current VisualWorks working directory.

GBS main statistic types are for GbsSession and GbsSessionManager. GBS can also be configured to collect per-class cache inventories, in which each process line represents a Class that has instances in the classMap.

to start collecting both main and cache inventory statistics, in GBS execute:

```
GBSM statsEnabled: true
```

to start collecting only the main statistics, in GBS execute:

```
GBSM mainStatsEnabled: true
```

to start collecting only the cache inventory statistics, in GBS execute:

```
GBSM statMonitorRunning: true
```

To stop collecting statistics, use the same method with a false argument.

For more information, refer to the *GBS User's Guide* chapter on profiling.

GBS statistics data files

Statistical data files written by GemBuilder for Smalltalk (GBS) are written in a binary form, with the extension .gfs, for GemTalk Flexible Statistics file.

1.2 Installing VSD

There is no need to run VSD on the same machine or on the same platform as the one on which the GemStone server or GBS is running. VSD on any machine and platform can read statistics generated on any other machine and platform. If you will be running VSD on a machine other than the one in which the GemStone server software is installed, you can download the appropriate platform of VSD and install it on that machine.

On UNIX platforms, VSD requires X-Windows.

Distribution

VSD is provided as part of the server distribution for all versions of GemStone/S and GemStone/S 64 Bit on all platforms except Windows. On Windows, only the GemStone/S 64 Bit v3.2 and later client distributions include VSD.

VSD is also available for download on the GemTalk website, at http://gemtalksystems.com/vsd/

Installation on all platforms

Installation is simple. The following instructions use / as a path separator, although the process is the same on Windows.

1. Download the VSD distribution, and unzip it into your VSD installation directory, *vsdInstallDir*.

   This will create two subdirectories in *vsdInstallDir*; *vsdInstallDir*/bin, which contains the VSD executable files, and *vsdInstallDir*/lib, which contains several subdirectory trees containing the VSD library files.

2. It is convenient, but not required, to add *vsdInstallDir*/bin to the machine search path.

   If the bin directory is not added to the machine search path, you may execute vsd directly from *vsdInstallDir*. VSD will search for the library files under the /lib subdirectory of the executable's parent directory, so you do not need to add this directory to the path.

Chapter

2 VSD's Graphical Interface

VSD provides a graphical way to view the data that is recorded by statmonitor. When you open VSD on this data, you will see a very large number of statistics that were recorded for potentially a very large number of GemStone processes. Not all of these are going to be useful; your task is to locate the interesting and useful information.

2.1 Starting VSD

You can start VSD from your windowing system or from the command line.

- Double click on `vsd` in a graphical file browser
  or `vsd.exe` in Windows Explorer.
- Execute `vsd` at the command line.
  You can either provide the full path to the vsd executable, navigate to the directory containing vsd, or, if the directory containing the vsd executable is on your machine search path, just enter 'vsd'.

You may start VSD and automatically open a specific data file by:

- execute vsd, specifying the statistics file name on the command line.

  For example:

  `prompt> vsd` *statmonDataFileName*

  vsdinstalldir`/bin/vsd` *statmonDataFileName*

- On windows, you drag a statmonitor data file and drop it on vsd.exe in Windows Explorer

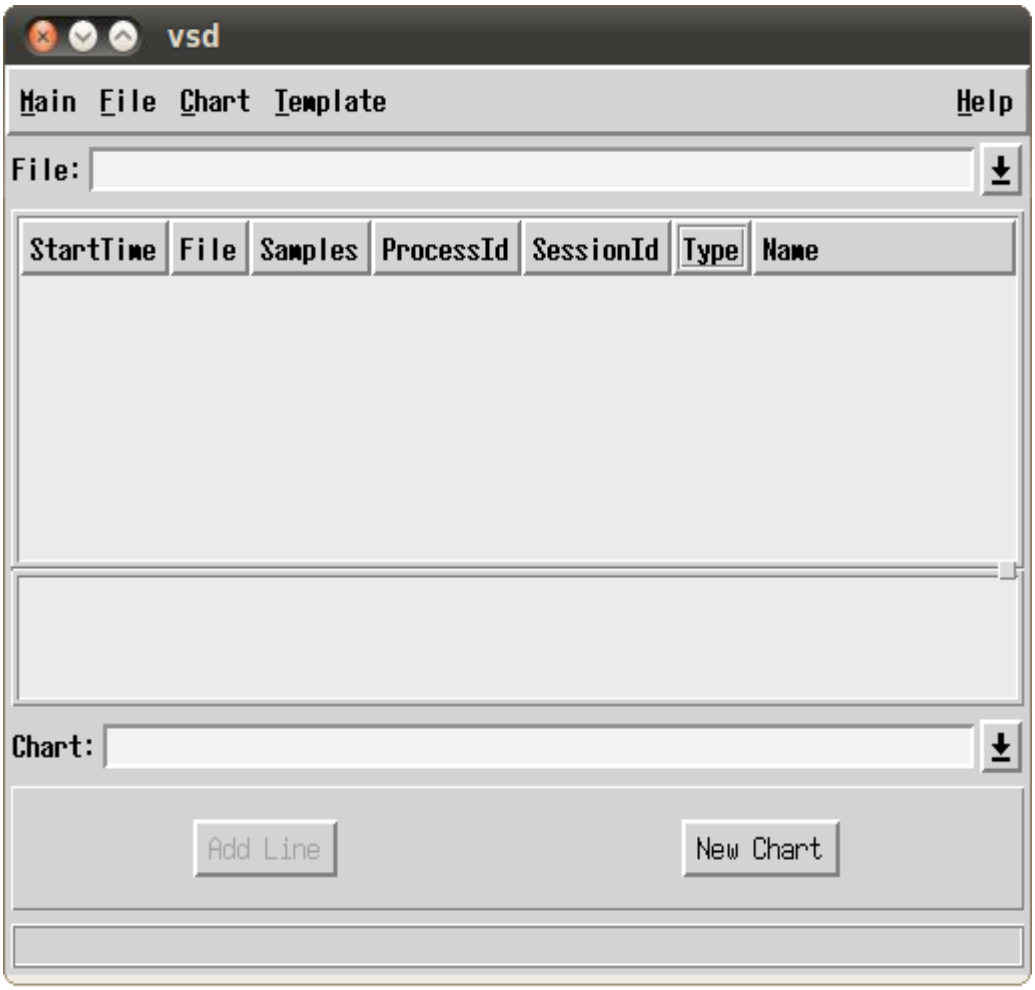
The main VSD window will appear; if this is your first time running VSD, an additional window will appear providing access to the version history and help.

2.2 Main VSD Window

When VSD is started, it displays the Main window. This window has a single instance and remains open while you are running VSD.

The main window displays the list of processes and other information about the statmonitor file; here, you will locate and select processes and statistics for which you want to view charts.

This window also contains many configuration options that control how your charts are displayed. These options are stored, so next time you open VSD you will get the same options. For more details, see "VSD Configuration Files" on page 29.



Loading data for viewing

If you opened VSD with a particular data file or files as an argument, the main window will display the processes listed from that file. Otherwise, you need to load a data file, or start monitoring a running server.

Loading an existing statmonitor file

To load a statmonitor file into VSD, use the menu item **Main > Load Data File...**

This brings up a file dialog that allows you to browse for an existing statmonitor output file.

If you know the name of the file you would like to load, you can also type the full path in the File entry box.

Updating the view if the data file is updated

You may load a statmonitor data file that is still being updated by the live system. As new data is added to this file, you have the option to have vsd update its view.

If you select **File > Auto Update**, VSD automatically updates your display, and any associated charts, any time the data file changes. Otherwise, if auto update is disabled, you can choose **File > Update** periodically to update the displays when you choose.

Loading multiple data files

You can load multiple statmonitor files using the menu item **Main > Load Data File...**, or by specifying multiple files on the command line. Each file is loaded, though you can view the list of processes for only one file at a time. To switch between them, click the down-arrow next to the File entry box, then select a file from the list.

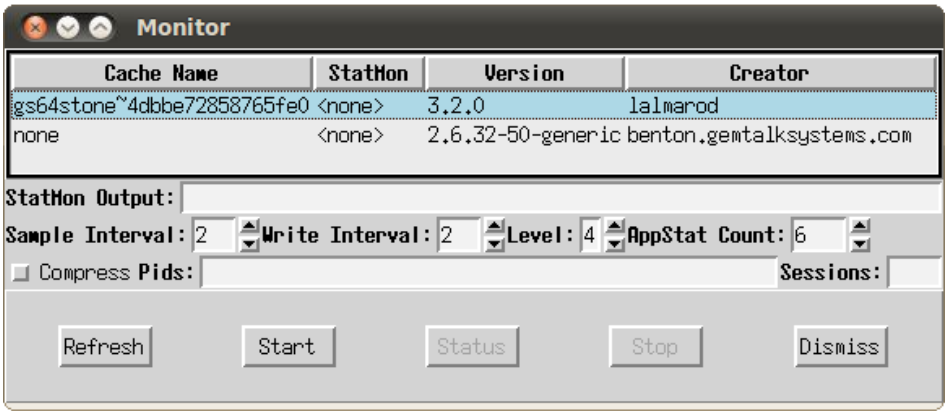
If your multiple data files are logically a single data set; (for example, you start statmonitor on the same repository, shut it down, and start it again), then you can append the second data file rather than load it. This is done using the **File > Append Data File** menu item, or by using the -a command line argument.

Monitoring statistics on a running GemStone server

You can use VSD to start statmonitor and automatically read directly from the data file statmon is currently creating. This options is not available on Windows.

To monitor a live system:

Step 1. Bring up the Monitor window using the menu item **Main > Monitor**



The Monitor window will display a list of anything that statmonitor can monitor on this host; all shared caches on this machine, and the system itself.

Step 2. Select the cache you wish to monitor, and the monitoring options.

These options are a subset of those offered by the server statmonitor utility. Options are:

- The **file name** for the statmonitor output.

  By default, this is empty, and statmonitor will create a file with the default name in the current directory. You must have write permission for the directory in which this file is located. For example, if you started vsd while in the $GEMSTONE/bin directory, you must specify a path and filename for another directory, in which you do have write permission.
- The sample interval (in seconds)—that is, how frequently to read the cache.
- The write interval—the maximum number of seconds to wait before flushing the samples to the output file. This determines the rate your monitoring view is updated.
- The statistics level—the amount of non-GemStone, OS-level information to collect. Larger numbers result in more and differing sets of statistics being sampled. Available levels vary by platform.
- The number of application statistics (shared counters) to gather.
- Whether to compress the output file.
- You can select to monitor only processes with specific Process Ids or Session Ids. Normally, all processes are monitored.

Step 3. Click the **Start** button to start monitoring.

This will start statmonitor, and create a statmonitor data file in the current working directory. If statmonitor startup fails, the Monitor status window will appear which shows the statmonitor command line.

The Monitor windows will remain open. You can use the **Dismiss** button to close the window; this does not affect monitoring.

VSD loads the data file as soon as it has some samples in it (unless you have explicitly turned off Auto Update).

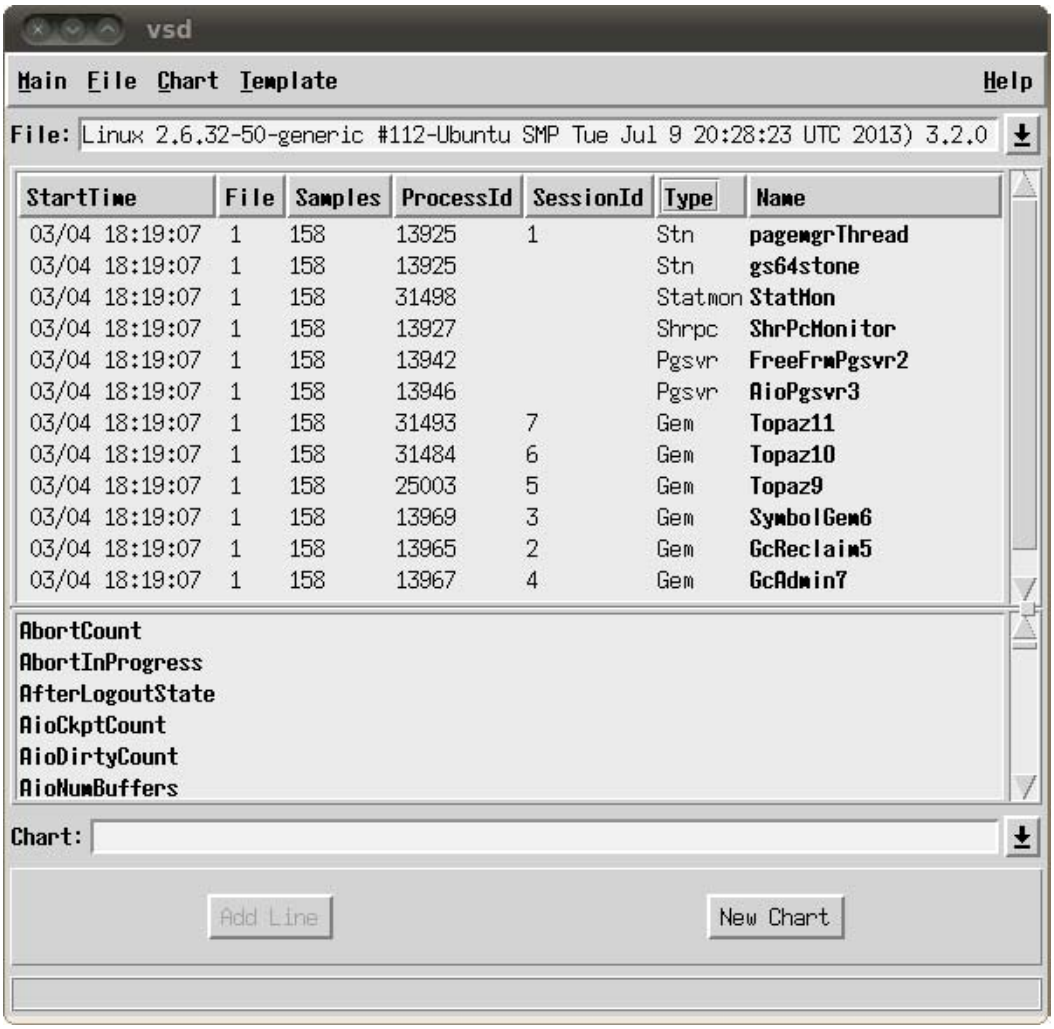
The statmonitor data file that is created by monitoring will grow as long as you are monitoring. If you have a large system and collect samples frequently, this file may become very large.

Step 4. Click the **Stop** button to stop monitoring.

If you have dismissed the Monitor window, you can re-open it using the menu item **Main > Monitor**. Select the cache you have been monitoring to enable the **Stop** button.

Note that the statistics data that is created will remain, and can be viewed later. It must be explicitly deleted if you no longer need it.

After you've loaded the data file or started monitoring, the VSD window looks something like this:



Process List

The upper list pane contains a list of all the entities—processes, thread-based sessions within processes, and other entities such as the disks—for which data is recorded in the statmonitor data. Depending on the options used to collect the data, this may include all GemStone processes or a small subset, and may include a number of non-GemStone entities including the operating system, disks, etc. While these are not all strictly "processes", in the following discussion this term is used to include all entities that appear in the process list.

For each process, session, or other entity, the display includes:

- The **StartTime** - the time that process first recorded a sample, adjusted to local time on the machine executing VSD.
- The **File** id for the file in which it was recorded. This is of interest if you have more than one file loaded; the file id corresponds to the number in the file select drop-down.
- The total number of **Samples** for that process recorded in the data file.

  If you are monitoring, processes that are currently running and for which samples are being taken are shown in green, and these number will change as new data samples are taken.
- the OS **ProcessID** for the process. Multi-threaded processes in which the threads are GemStone sessions will have individual entries for each thread, and list the same process Id for each.
- the GemStone **SessionId** for that processes. This only applies for GemStone sessions.
- the process **Type**.

  GemStone processes fall into a number of types, which govern which statistics apply. The common process types are:

  Stn — Stone

  Shrpc — shared page cache monitor

  Pgsvr — AIO or free frame page server, or a page server for a remote session

  Gem — Gem; includes system Gems, RPC Gems, and linked sessions such as linked Topaz

  Statmon — statmonitor

  Additional types will appear if you are also monitoring OS information, including platform-specific statistics.
- the process **Name**.

  This is assigned by statmonitor or GBS's Stat Monitor, and displayed in VSD. It may be assigned for Gem processes within the GemStone server; see "Process names in GemStone" on page 25.

Sorting

By selecting a column header in the main windows, you can sort all the processes by that column's attribute.

For example, if you want to find the Stone, you can sort by Type. Or, if you know the session ID of the Gem, you can sort by SessionId and scroll down under you find that Gem.

Selecting Process

When you select a process, the list of statistics relevant for that process is displayed in the statistics pane. Multiple selection is allowed.

Selecting Sets of Processes

The pop-up menu option **Select** allows you to select all processes matching a particular criteria. The options are:

All —select all processes

By type —select all processes of the same type as the currently selected process. Does nothing if no process is selected.

By statistic—select all processes for which the currently selected statistics applies. Does nothing if no statistic is selected.

Clear —deselect all

Statistics List

The lower pane displays a list of statistics.

Each process type has a specific set of statistics that are relevant for that process. Many of these are unique for that process type; for example, CommitCount applies to Gem processes, while TotalCommits is meaningful for the Stone. Others, such as UserTime, apply to all processes.

When no process is selected, all statistics in the system are shown. When a particular process is selected, only statistics appropriate for that process type are listed. When multiple processes are selected, only statistics that are common to all the selected process types will be listed.

You may scroll down the list to locate the statistic you want. Typing in the first character will select the first statistic with a name beginning with that letter.

Notice that some statistics are listed in bold font, while others are in regular font. Statistics with the font in bold have non-zero values recorded.

Controlling visibility of statistics

Hiding statistics with only values of zero

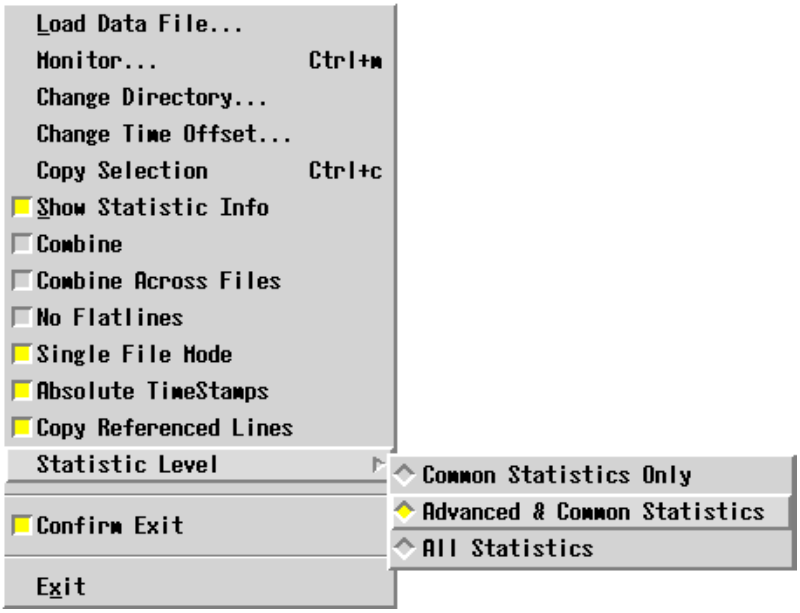
You may hide statistics for which the value of that statistic is 0 for the entire sample set. These are the statistics that are listed in non-bold font.

To hide statistics where there are no non-zero values, check **No Flatlines** in the **Main** menu or in the right mouse button popup menu.

Statistics Level

Each statistic has a characteristic called a *level* reflecting the amount of background knowledge about GemStone processes needed to use it with understanding. You can set up VSD to list only those statistics that are at, or below, a certain level of complexity.

To establish the levels of statistics that you want to display in VSD, choose **Statistic Level** from the VSD window's **Main** menu:



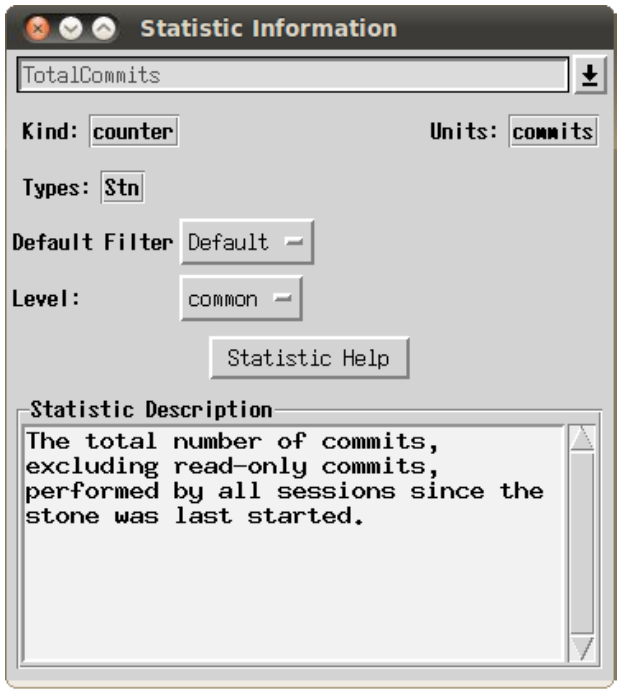
The level of statistics you select will be recorded in your `.vsdrc` file, and will be used automatically the next time you start VSD. See "VSD Configuration Files" on page 29 for details on VSD configuration files.

Meaning of statistics

While many statistics have intuitive names, you will often need to look up details. Chapter 4, "Cache Statistics Reference", starting on page 35, has a complete list of statistics.

You should not expect to understand every statistic recorded for your system. Some of these statistics require an intimate knowledge of GemStone server internals, some are experimental, some may only apply under specific server conditions. GemStone documentation, Techtips, and analysis from GemStone Technical Support will include specific statistics that are particularly helpful.

The information in the current version of VSD, including more detail, is displayed in the Statistics Information Window. To open the Statistic Information, use the menu item **Main > Show Statistics Info**.



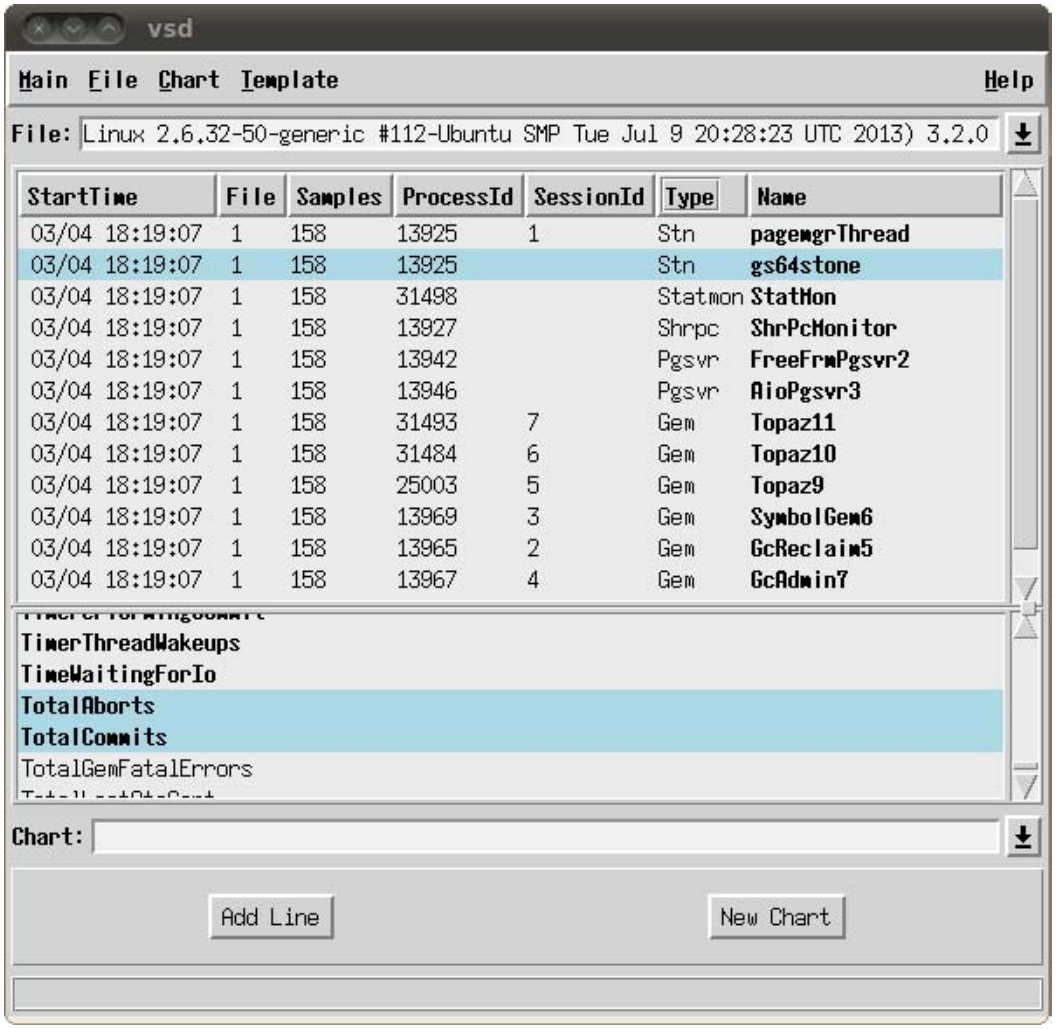
This singleton window can be left open; it will display the details for any selected statistics as it is selected. You can also lookup statistics using the drop down.

Selecting and viewing statistics

After you have selected one or more processes and one or more statistics, you can display these results in a chart by clicking on the **New Chart** button. You can create a named chart by entering a name in the **Chart** name entry field.

If you have a chart open, you can add the selected statistics to an existing chart. Select the chart by name in the drop-down with the **Chart** name entry field, then click on the **Add Line** button. The Chart name is set to the most recently selected chart, so if you do not explicitly select a chart, the statistics will be displayed in the most recently selected chart.

The following shows the main window with one process selected, the Stone, and two statistics selected, TotalAborts and TotalCommits.

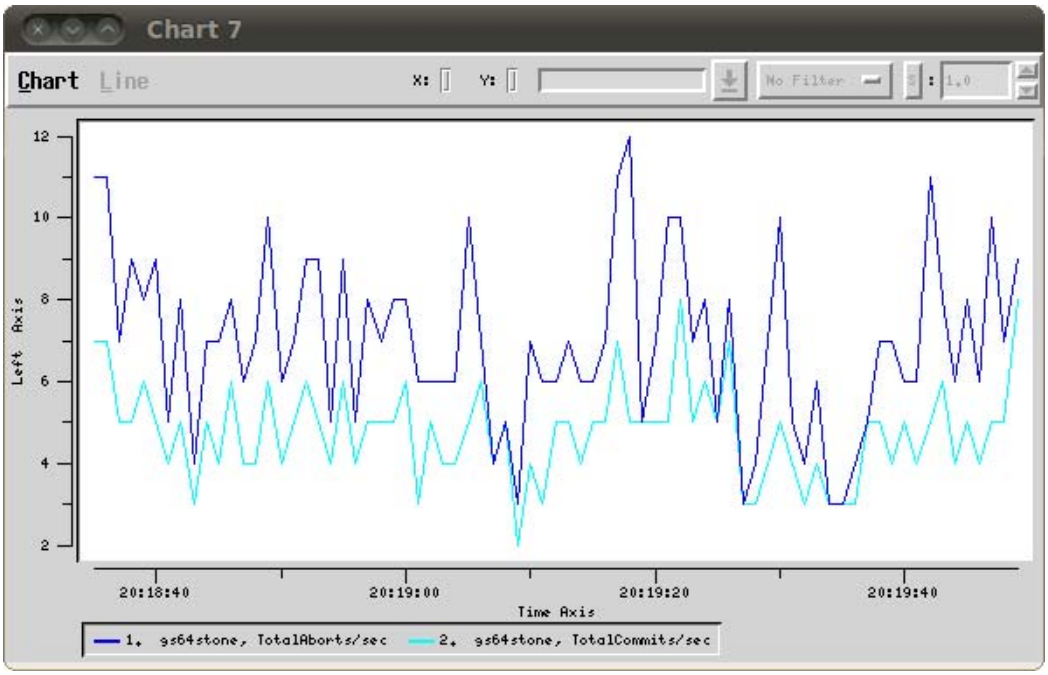


2.3 VSD Statistics Chart

VSD statistics charts allow you to view and analyze the data for the statistics you have selected. You may have many charts open.

Chart Window

When you first open a chart, all the values for all statistics are displayed with its default settings. Each line has a unique color and/or pattern.



At the bottom is the Legend - the list of process statistics that are viewed in this chart. Each is described by Process name and statistics name.

The axis of the graph are labeled; the bottom axis is always time.

The Y-axis is labeled with the number of the units for all the statistics; this may or may not be meaningful, if the statistics used different units of measurement. For example, comparing the counts of the number of commits and the time waiting for I/O, by units, does not directly make sense. However, it can be very useful to see how each varies relative to the other over time.

Chart Options

You can customize and manipulate a VSD chart in many ways to make your analysis easier. A few of the more useful are described here. The VSD application includes help for each menu item.

Zoom

To zoom in on a subsection of the chart, use the menu item **Chart > Zoom In**. This prompts for a rectangle to zoom in to, allowing you to view with more detail the section of the chart that you are interested in. You can zoom in multiple times to narrow your view further.

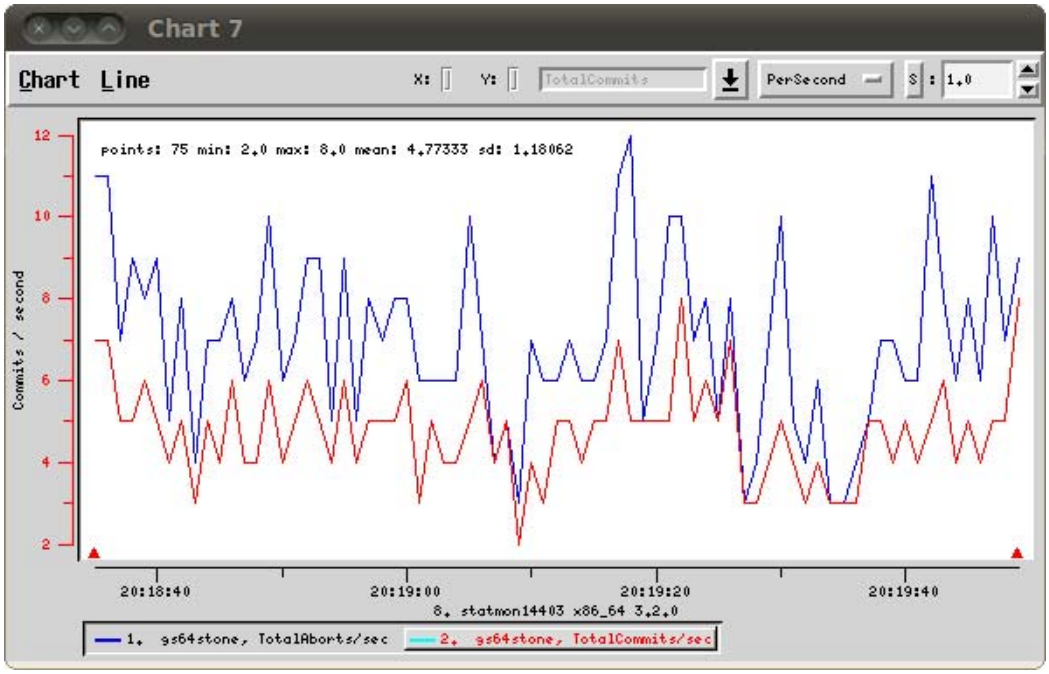
Zoom Out returns you to the previous level of zoom.

Time Format

The times marked on the Y axis, by default are in HH:MM:SS format (with hours in 24-hour format). If your sample spans multiple days, you can set the time format to include the month and day using the **Chart > Time Format** menu item.

Selecting a Line

You can select an individual line from the Legend at the bottom, or by clicking on that line in the graph. Selecting a line colors it red and displays information about that line.



When a line is selected, the following information is available:

- On the graph itself, a brief **summary statistics** for the line. This includes the number of data points for that statistic, the minimum, maximum, and standard deviation are displayed. These units stay the same regardless of scale or transformation.
- When the mouse is over a data point in the graph, the **X** (time) and **Y** (units) of that point are displayed. These units stay the same regardless of scale or transformation.
- the **name** of the statistic.
- the current **filter**. The filter allows you to distinguish absolute values from rates of change. Each statistic has a default filter, which can be changed as needed.
- The scale or other transformation. The scale allows you to compare lines with widely different ranges of values, by scaling one or the other to bring the ranges closer together.

Changing Line Options

Filters

The filter allows you to view either absolute values, rates of change per second or sample. Filter options are:

None — Displays the values for the statistic exactly as they are expressed in the statmonitor text file.

PerSample — Displays the difference between two consecutive samples of the statistic.

PerSecond — Displays the difference between two consecutive samples of the statistic, divided by the number of elapsed seconds between the two samples.

Aggregate — Displays a running total for the per-sample deltas of the statistic.

Scaling

When you have multiple lines with widely different ranges of values, you can increase the visibility of a line relative to the other line by scaling it.

Select the line, and in the upper right, use the arrow indicators to scale up or down the line.

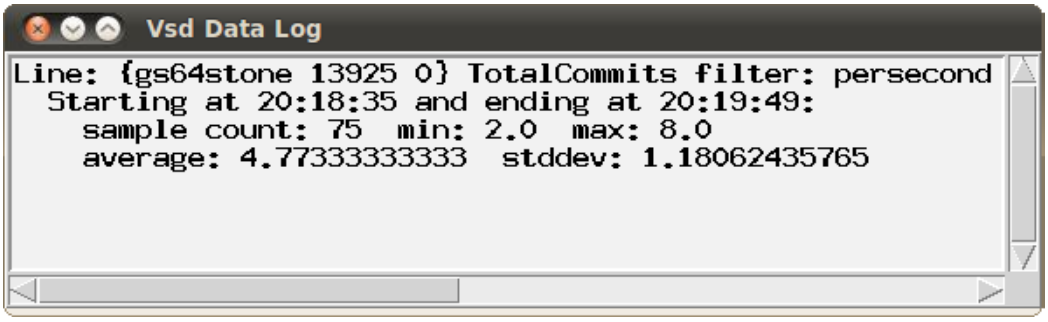
Other Transformations

By default, **S** is selected for the transformation setting, between the filter list and the scaling value entry field. When **S** is selected, the numeric scrolling field is used to scale the selected line.

You may also select other options, rather than **S**. This allows you to transform the line by adding, subtracting, multiplying, or dividing the value in the numeric scrolling field.

Summary of Information on a line

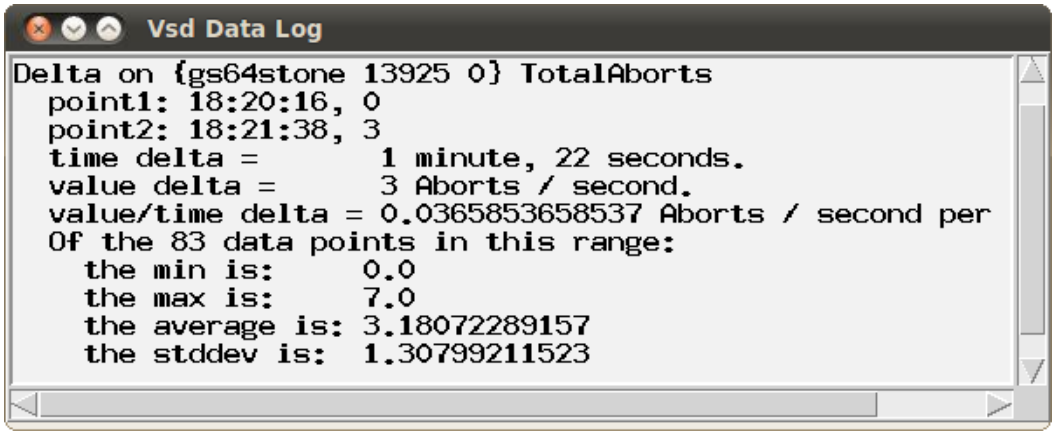
The menu item **Line > Log Info** will open (or append to an already open log window), summary information about a single line. For example:



Summary of Information on a section of line

A very useful tool is collecting statistics on a section of line. For example, you may want to know the commit rate over a single period of intense activity, rather than for the entire period of sampling.

The menu item **Line > Log Delta** will allow you to select two point on the selected line. It then opens (or appends to an already open log window), summary information about that section of line.



Other ways to control the view

There are many ways to control what you are viewing to make analysis of your data easier. Here are a few that are particularly useful.

Combining statistics values for multiple processes

Normally, when opening a chart or adding lines to an existing chart, one line is added for each selected process for each selected statistic. If you have selected a large number of processes — for example, all Gems — this may be cumbersome to see in the resulting chart.

In cases like this, you can combine values for a single statistic for all the selected processes. To combine values, in the Main VSD Window, check the value **Combine** in the **Main** menu or in the right mouse button popup menu.

Graphing against scales on both right and left side Y-axis

By default, all lines in a graph are plotted according to Y-axis units scaled on the left-hand side of the chart. You can instead plot some of the lines according to units scaled on the right-hand side of the chart, which will allow them to use a different scale. This is useful when the lines use different units of measurement from those plotted on the left.

To change the side on which a line is selected, with the line selected in a Chart Window, uncheck the value **Graph on left axis** on the Line menu.

Process names in GemStone

When statmonitor records statistics for Gem processes, it records them with the name "Gem", or for linked topaz logins, "Topaz". This can be modified on the GemStone server after the gem logs in. This can be particularly useful for regular maintenance processes, such as your backup or markForCollection, that are run from scripts.

To do so, as soon possible after login, in the session you want to rename, execute:

```
System cacheName: 'myName'
```

Depending on how soon you can execute this after login and the statmonitor sample rate, you may have one or more samples showing the original default name.

Changing Hours Offset

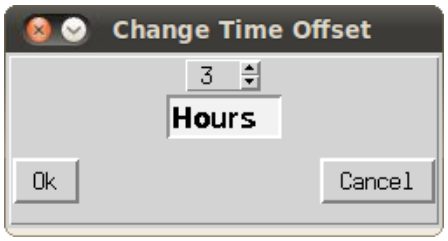
Timestamps are recorded in statmonitor data files using GMT time. When these files are read into GemStone, VSd will translate these times into the current machine's local time.

If you are viewing the data in VSD in the same time zone as the system that generated the statmonitor data, this means that VSD will display dates in the correct time, including adjusting for daylight savings time.

However, if you are viewing statistics in a different time zone, the timestamps will appear to be incorrect.

To correct this, use the main menu option **Change Time Offset...,** and select the number of hours plus or minus to adjust VSD's display.

For example, if you are in U.S. Pacific Time and viewing statistics that were generated by a system running in U.S. Eastern Time, which is three hours ahead, add three hours:



2.4 Using and Creating Templates

VSD templates let you quickly create a chart with a commonly used set of statistics. Templates are helpful if you find yourself performing the same task frequently in VSD — for example, monitoring the same five or six statistics. By creating a template for the statistics that you want to monitor most frequently, you can automate the task of building charts.

VSD is shipped with a set of predefined templates. You can open charts on these templates using **Template > New Template Chart**, and choosing any of the predefined templates. If you create your own custom templates, they will also appear in this list.

You can create a custom template based on a chart you have created in VSD, or by editing the formatted text file that contains the templates. See "VSD Chart Templates" on page 27 for more on the text file and how it is formatted, and editing this file outside of the GUI.

To create a custom template from VSD:

Step 1. set up a chart with the processes, statistics, scaling, and so on, that you want in your template.

Step 2. Chose the **Chart** menu item **Save Template**. This will prompt you to enter a name for the custom template. Hit the enter key when you are done.

You can now use your new template, from the **Template > New Template Chart** menu item.

if you want to remove your template, edit the .vsdtemplates file, and use the menu item **Chart > Reload Template File** to reload the templates file.

Gem and Pageserver Names in Templates

When you create a template that includes processes that are Gems or PageServers, the name of the process, as well as the type, is included in the template. This helps avoid an excessive number of lines appearing when the chart is opened.

The name that is included does not include any serial numbers that are displayed with the name in VSD. So for example, if the chart on which you are creating a template includes a Gem named Topaz19, the template will use Topaz* as the name. When you later use the template to create a chart, lines will be added for all Gems with the name Topaz.

Since you may have many Topaz processes, and cannot determine which one is performing a particular task such as backup or MFC, including Gem and Page server statistics in templates requires more consideration. Providing the Gem with a name, as described under "Process names in GemStone" on page 25, is one solution.

Chapter

3 VSD Template and configuration files

The first time you use VSD, the executable creates default template files and configuration files in your home directory. These files are used on subsequent executions; this allows it to remember the statistic level you have selected, window sizes and locations, and similar details.

You can delete these file at any point, and they will be recreated from the default the next time you use VSD.

3.1 VSD Chart Templates

VSD templates specify a set of processes and statistics that together represent a useful picture of one behavior of the system. VSD is shipped with a set of predefined templates. As described under "Using and Creating Templates" on page 25, you can open charts on these templates, and create your own templates based on charts you have constructed.

To further customize the templates you use, you may want to edit the templates directly. The templates are stored in a text file using specific syntax. This section provides information you need if you wish to edit the template files directly.

VSD Templates File

The default templates are in the file .vsdtemplate, which is written to your home directory the first time you start up VSD.

If you delete the `.vsdconfig` file, then the next time VSD is started, the file will be recreated with default values.

Template Syntax

The template files are written in a formatted form that is understood by the TCL code that supports VSD.

Templates in the .vsdtemplate have the following form:

```
set vsdtemplates(TemplateName) {
        {Type {ProcessName} StatName scale filter axis}
     {Type {ProcessName} StatName scale filter axis}
        ...
    }
```

for example:

```
set vsdtemplates(Garbage) {
        {Stn {*} PagesNeedReclaimSize 1 none y2}
        {Stn {*} ReclaimedPagesCount 1 none y2}
        {Stn {*} EpochGcCount 1 persecond y}
        {Stn {*} ReclaimCount 1 persecond y}
        {Gem+ {*Gc*} CommitCount 1 persecond y}
    }
set vsdtemplates(CacheTooSmall) {
        {Shrpc ShrPcMonitor FreeFrameCount 1 none y2}
        {+ {*} FramesFromFreeList 1 persecond y}
        {+ {*} FramesFromFindFree 1 persecond y}

  }
```

The data for each line in each template includes:

type — One of the following:

```
Stn
Shrpc
Gem
Pgsvr
```

'+' can also be used, which means that all processes that match this line spec will be combined into a single line. + can be used either standalone, meaning all process types, or added to a type name. For details on types, see page 16.

name — Identifies the specific process by process name. You m.ay use * to match multiple process names. For more on process names, see page 16.

stat — The name of a specific existing statistic.

scale — The number of the scale for this particular statistic. It will usually be 1. For more details on scales, see"Scaling" on page 23.

filter — The name of the filter, which will be one of the following:

```
none
persecond
persample
aggregate
```

For details on these filters, see "Filters" on page 23

axis — Either y, the left axis, or y2, the right access. Units for this statistic line are labeled on the specified side y-axis.

Once you have created or edited your template file, save the .vsdtemplates file, and use the menu item **Chart > Reload Template File** to load this file. If you have made syntax errors, this file will not load.

For more about VSD templates and the syntax and options available, see the VSD help under the topic "Template Syntax".

3.2 VSD Configuration Files

VSD provides a number of configuration options, which are set by menu items in vsd. These options are recorded in platform-independent text configuration files in your home directory.

There are two configuration files that together manage the configuration parameters that control what you see on startup.

.vsdrc

This file is used by VSD to record its current configuration. VSD reads the file when it starts and writes the file when it exits.

If you delete the `.vsdrc` file, VSD will start with defaults from .vsdconfig or from the VSD executable. When it subsequently exists, the file will be recreated with the current settings.

You should not modify this file manually. To make changes in the way that VSD is always started, no matter how it was configured at the previous shutdown, edit the .vsdconfig file.

.vsdconfig

You can configure VSD by setting default values in this file. Any value set in `.vsdconfig` will override the same definition in `.vsdrc`.

If you delete the `.vsdconfig` file, then the next time VSD is started, the file will be recreated with default values.

Configuration file format

In most cases, you can configure your default display using by taking advantage of the natural way VSD saves your settings to `.vsdrc` and reads them on restart. By doing this, you get a consistent display without any extra work.

However, if you want to set a configuration that is always used, regardless of what you do during any particular VSD session, you will need to edit the VSD configuration file `.vsdconfig`. The configuration file format is the same in `.vsdconfig` and `.vsdrc`, so you may copy lines between files.

The following sections document many of the relevant configuration parameter names and meanings.

Configuration options are specified in a format that is specific to VSD/TCL, and are not intended for general use by customers.

To include a comment line, prefix the line with #.

Incorrect configuration names and settings encountered in reading the files are ignored; default values will be used in this case.

Configuration options have the format:

```
set vsd (configurationOption) value
```

for example:

```
set vsd(confirmonexit) 1
```

Acceptable *value* arguments depend on the parameter. Many are Integers, but some parameters are strings. Lists are indicated by {}. Boolean statistics are specified as 0 or 1.

Main Window options

The following statistics relate to the main VSD window.

Parameter name	Type	Default	Description
confirmonexit	Boolean (0 or 1)	1	Whether to confirm before exiting VSD, **Main > Confirm Exit**
def:sortbytype	Column label	Type	On file load, the column to use to sort processes.
def:sortdecreasing	Boolean (0 or 1)	1	On file load, the direction of sort for the process list; 0 is decreasing, 1 is increasing.
combineFiles	Boolean (0 or 1)	0	Whether to combine values for all selected processes in all open files when displaying a statistic; **Main > Combine Across Files**
combineInstances	Boolean (0 or 1)	0	Whether to combine values for all selected processes when displaying a statistic; **Main > Combine**
filelist	list of strings	empty list	List of filenames displayed in the Main Window File: dropdown.
noFlatlines	Boolean (0 or 1)	0	Whether to show statistics for a process if all values for that statistic are zero; **Main > No Flatlines**
showCtrInfo	Boolean (0 or 1)	0	Whether to have the Statistics Information window open. **Main > Show Statistics Info**
singleFileMode	Boolean (0 or 1)	1	Whether working with multiple data files is allowed. **Main > Single File Mode**
templatesUseSelection	Boolean (0 or 1)	0	Whether the template applies only to the selected processes.

Chart Window Display Options

The following parameters describe the defaults for including and omitting information from chart windows.

Parameter name	Type	Default	Description
absoluteTSMode	Boolean (0 or 1)	1	Whether to use absolute timestamps for charts. **Main > Absolute Timestamps**
activecolor	string	red	The color of the selected line on a chart. **Chart > Selected Line Color...**
def:linestyle	linear | step | natural | quadratic	linear	How the line is rendered on charts; **Chart > Default Line Style**
def:showcrosshairs	Boolean (0 or 1)	0	Whether to show cursor crosshairs on charts, **Chart > Show CrossHairs**
def:showcurxy	Boolean (0 or 1)	1	Whether to show current X and Y values on a chart; **Chart > Show Current Values**
def:showgridlines	Boolean (0 or 1)	0	Whether to show gridlines on charts, **Chart > Show Grid Lines**
def:showlegend	Boolean (0 or 1)	1	Whether to show the legend (the process and statistic name and line color for the chart lines) on charts; **Chart > Show Legend**
def:showlinestats	Boolean (0 or 1)	1	Whether to show text with points, min, max, mean, and standard deviation on graph; **Chart > Show Line Stats**
def:showminmax	Boolean (0 or 1)	0	Whether to show max and min values on chart menu bar; **Chart > Show Max and Min**
def:xaxis:showtitle	Boolean (0 or 1)	1	Whether to display label on x axis, **Chart > Show Time Axis Title**
def:xformat	date | time | elapsed	time	How to display the time on the x-axis of a chart; **Chart > Time Format**
def:yaxis:showtitle	Boolean (0 or 1)	1	Whether to display y axis label on left side, **Chart > Show Left Axis Title**
def:y2axis:showtitle	Boolean (0 or 1)	1	Whether to display y axis label on right side, **Chart > Show Right Axis Title**

Monitor Window Options

The following parameters apply to the Monitor window, describing options for how to start statmonitor.

Parameter name	Type	Default	Description
wv:statmoninterval	integer	20	Monitor window, default statmonitor sample interval
wv:statmonflush	integer	0	Monitor window, number, default statmonitor flush interval
wv:statmonlevel	integer	1	Monitor window, number, default statmonitor statistics level
wv:statmonappstats	integer	0	Monitor window, number, default statmonitor number of app stats
wv:statmonsessions	list of integers	empty list	Monitor window, default for list of session ids to monitor in statmon
wv:statmonpids	list of integers	empty list	Monitor window, default for list of process ids to monitor in statmon
wv:statmoncompress	Boolean (0 or 1)	0	Monitor window, default for if statmonitor should write compressed file

Window locations and display

The location, size, and proportion of the Main Window, the Chart window, and the Statics info file are recorded.

`def:geometry` sets the size and location of the main window.

`instancePaneSize` sets the height of the Process pane in the main window.

`statisticPaneSize` sets the height of the Statistics pane in the main window.

`geometry:.datawin` sets the size and location of the chart window.

`geometry:.ctrinfo` sets the location of the statistics information window.

Fonts

Window title, menu, and other fonts are based on operating system default fonts, and are not controllable. However, you may set the font for text panes and for charts.

Main window text fonts

`textfont` is a list of font attributes that controls what is used to display text on a chart windows. This configuration is used for most text on a chart: legend, axis labels, summary information, etc. Attributes include the font family, size, weight, slant, underline and overstrike are options. These can be set using **Help > Choose Text Font...**. For example,

```
textfont {-family fixed -size 10 -weight normal -slant roman
                -underline 0 -overstrike 0}
```

Chart window text fonts

`smallfont` is a list of font attributes that controls what is used to display text on a chart windows. This configuration is used for most text on a chart: legend, axis labels, summary information, etc. Attributes include the font family, size, weight, slant, underline and overstrike are options. These can be set using **Chart > Choose Chart Font...**. For example,

```
smallfont {-family fixed -size 12 -weight normal -slant roman
                -underline 0 -overstrike 0}
```

Chapter

4 Cache Statistics Reference

This chapter describes specific statistics that are written by GemStone and GBS tools. The lists of statistics are provided here for convenience when using VSD; VSD can read files with any arbitrary set of statistics, and the particular set of statistics that are present in a data file are specific to the tool that generated them.

The references in this chapter are separated by how they are collected; GemStone server product statistics, statistics available when statmonitor performs System monitoring on available platforms, and statistics generated by GBS.

When you are browsing a statistical data file, you may look up the definition in the Statistics Information window. See the section “Meaning of statistics” on page 18 for more information.

4.1 GemStone Server process Statistics

This section lists statistics that are generated by the GemStone/S 64 Bit server. This includes the following process types:

- Stone
- Gem
- SPC monitor (Shared Page Cache Monitor)
- Pgsvr (AIO, Free Frame, or remote Gem Page Server)
- Statmon
- All (all of the above, excluding Statmonitor)

Some statistics are platform-specific, and have platforms listed. Note that no host statistics are available for the Macintosh.

Note

This statistical reference include statistics that are generated by recent versions of GemStone/S 64 bit and in some older versions,. Any specific statmonitor data file will not include all these statistics.

AbortCount (Gem)

The number of aborts executed by a Gem process (or by an application linked to a Gem) since the Gem was most recently started.

AbortInProgress (Gem)

A boolean that indicates if the session is currently performing an abort operation.

AfterLogoutState (Stone)

Bits from the after-logout state machine in stone.

AioCkptCount (Pgsvr)

The number of dirty pages written from the shared page cache to disk to satisfy a checkpoint.

AioDirtyCount (Pgsvr)

The number of dirty pages written from the shared page cache to disk by Stone's AIO page server during normal operation.

AioNumBuffers (Stone)

The current setting of the STN_MAX_AIO_REQUESTS configuration parameter.

AioNumEmptyBuffers (Stone)

The number of internal AIO buffers that are currently empty. The stone will block when starting a tranlog write unless at least 3 buffers are empty.

AioRateLimit (Pgsvr)

The current I/O rate being used by the page server, expressed in I/O operations per second. The page server will perform no more than this number of I/O operations per second on average.

AioRateMax (Pgsvr)

The current maximum I/O rate being used by the page server, expressed in I/O operations per second. The page server will perform no more than this number of I/O operations per second on average.

AioWriteFailures (Pgsvr)

Total number of write errors detected by this page server, including write errors that succeeded on retry.

AllOtherSleepTime (All, on Solaris only)

The number of milliseconds the process has been sleeping for some reason not tracked by any other stat.

AsyncFlushesInProgress (Pgsvr)

The number of pending extent flush operations.

AsyncWritesInProgress (Pgsvr)

Is zero or one for an aio page server. It is one when the aiopgsvr is doing checkpoint io. Always zero for a stone and a page server that is not an aio page server and in GemStone/S 6.3 and later.

BackupHighWaterPage (Stone)

The lowest page ID for which the GC reclaim gems will temporarily ignore because a full backup is in progress. Pages with IDs less than this value will be reclaimed normally. Page IDs equal to or greater than this value will not be reclaimed. The session performing the full backup will increase this value as the full backup progresses.

BackupRecordPagesWrittenByGem (SPC monitor)
BackupRecordPagesWrittenByStone (SPC monitor)

The number of backup record pages in the cache because of a write done by a Gem or by the Stone, respectively.

BitlistPagesWrittenByGem (SPC monitor)
BitlistPagesWrittenByStone (SPC monitor)

The number of bitlist pages in the cache because of a write done by a Gem or by the Stone, respectively. A bitlist page is a form of a bit array.

BitmapPageReads (All)

The number of bitmap pages read by the process since it was last started. These page reads are actual disk reads and not reads from the shared page cache.

BmCHeapPages (Gem, Stone)

The number of temporary bitmap pages allocated in private heap memory to store bitmap structures.

BmInternalPagesWrittenByGem (SPC monitor)
BmInternalPagesWrittenByStone (SPC monitor)

The number of bitmap internal pages in the cache because of a write done by a Gem or by the Stone, respectively.

BmLeafPagesWrittenByGem (SPC monitor)
BmLeafPagesWrittenByStone (SPC monitor)

The number of bitmap leaf pages in the cache because of a write done by a Gem or by the Stone, respectively.

CacheScanCount (SPC Monitor)

Number of times the shared page cache monitor statistics thread has scanned the entire shared page cache.

CacheSlotIndex (All)

Index of the slot in the shared page cache used by this process.

CacheStartQueueSize (Stone)

The number of sessions waiting for remote cache startup.

CharacterIO (All, on Solaris only)

The number of characters read and written.

CHeapSizeKB (All)

The size (in KB) of the C heap area; that is, the total size of results of UtlMalloc for which UtlFree has not been called. This statistic is computed only in slow or fastdebug executables.

CheckpointCount (Stone)

The number of checkpoints that have been written since the Stone repository monitor was last started. Writing a checkpoint implies that all of the data and meta information needed to recover the data corresponding to the commit record associated with the checkpoint have been written to the disk(s) containing the extent(s) that make up the repository. Thus, the last checkpoint in the transaction log determines how much data in the log must be recovered when there is a system crash.

In full logging mode, the checkpoints are controlled completely by the STN_CHECKPOINT_INTERVAL configuration parameter. In partial logging mode, a checkpoint may be written more often if the size of the transaction exceeds the value set by the configuration parameter STN_TRAN_LOG_LIMIT. If partial logging is in use, a rapidly increasing CheckpointCount indicates that STN_TRAN_LOG_LIMIT may be set too small.

CheckpointState (Stone)

This internal statistic indicates the state of the checkpoint process:

-1 — checkpoints are suspended
0 — checkpoint not active
1 — AIO servers writing pages
2 — pre-fsync processing
3 — synchronous fsync
4 — asynchronous fsync
5 — second fsync
6 — dispose alloc pages shadowed
7 — write log record

These values are provided for information only and are subject to change without notice.

ClassesRead (Gem)

The number of classes copied into VM memory since the start of session.

CleanSlotsInRecoveryCount (SPC monitor)

The current number of slots being recovered which were owned by a process which shutdown cleanly.

CleanSlotsRecoveredCount (SPC monitor)

The total number of slots the shared cache monitor has recovered because a client process shutdown cleanly.

CleanSlotThreadTimeRunningMs (SPC monitor)

Approximate real time in milliseconds that the clean slot recovery thread in the shared page cache monitor has spent doing work.

ClientAbortInProgress (Pgsvr)

A boolean indicating if the page server's client Gem is currently performing an abort operation.

ClientAborts (Pgsvr)

Number of times the client gem of this page server aborted a transaction.

ClientCommits (Pgsvr)

Number of times the client gem of this page server committed a transaction.

ClientLostOtsSent (Pgsvr)

The number of Lost OT root signals sent from the stone to the page server's client.

ClientPageReads (Pgsvr)

The number of pages that have been transmitted by a page server to its client. This statistic is implemented only for cache slots used by a page server.

ClientPageWrites (Pgsvr)

The number of pages that have been transmitted by a client to its page server. This statistic is implemented only for cache slots used by a page server.

ClientPid (Gem, Pgsvr)

The process ID of the client process associated with a Gem or page server process.

ClientSigAbortsSent (Pgsvr)

The number of SigAborts sent from the stone to the page server's client.

ClockHandFrameId (Pgsvr)

The shared page cache frame ID that the page server clock hand currently references.

CodeCacheSizeBytes (Gem)

The total size in bytes of copies of GsNMethods that are in the code generation area and ready for execution, as of the end of mark/sweep.

CodeCacheSizeKBytes (Gem)

Total size in kilobytes of copies of GsMethods that are in the code generation area and ready for execution, as of the end of mark/sweep.

CodeGenClearSendCachesCount (Gem)

Total number of times the gem has cleared all method send caches in code gen. (2x)

CodeGenFullCount (Gem)

Number of times the code gen space was found to be full.

CodeGenGcCount (Gem)

The number of times the code generation area has been garbage collected.

CommitCount (Gem)

The number of commits executed by a Gem process (or application linked to a Gem) since the Gem was most recently started.

CommitQueueAddedToRunQueueCount (Stone)

The number of times sessions from the commit queue were added to the run queue.

CommitQueueAddedToRunQueueSessionCount (Stone)

The number of sessions from the commit queue which were added to the run queue.

CommitQueueSessionNotReadyCount (Stone)

The number of times the stone was ready to assign the commit token to a session in the commit queue but no session was not ready to receive the token.

CommitQueueSize (Stone)

The number of Gem session processes waiting for the commit token.

CommitQueueThreshold (Stone)

The current setting of the STN_COMMIT_QUEUE_THRESHOLD configuration parameter. This setting determines how large the commit queue must be before the stone will defer commit record disposal. A value of -1 indicates this feature is disabled and commit record disposal will never be deferred.

CommitRecordCount (Stone)

The number of outstanding commit records that are currently being maintained by the system. A number larger than the STN_SIGNAL_ABORT_CR_BACKLOG configuration option indicates that there is a process in a transaction that is preventing the Stone from reclaiming (garbage collecting) the resources associated with those commit records. Large values are usually accompanied by continuing growth in the size of the repository.

CommitRecordDisposalsDeferredCount (Stone)

The number of times the stone deferred commit record disposal because the number of sessions in the commit queue exceeded the STN_COMMIT_QUEUE_THRESHOLD setting.

CommitRecordDisposalState (Stone)

An internal statistic used to analyze the commit record disposal routines.

CommitRecordPagesWrittenByGem (SPC monitor)
CommitRecordPagesWrittenByStone (SPC monitor)

The number of commit record pages in the cache because of a write done by a Gem or by the Stone, respectively.

CommitRecordReleases (Stone)

Number of times stone has released a session's reference to a commit record due to an abnormal event. Routine events that cause commit records to be released such as commit, abort, and logout are not included in this statistic.

CommitRecordsDisposedCount (Stone)

The total number of commit records disposed by the Stone.

CommitRecordsReadAbortCount (Gem)

The number of commit records read while processing an abort.

CommitRecordsReadCommitBeforeCommitQueueCount

The number of commit records read by the session before it requested the commit token during a commit operation.

CommitRecordsReadCommitInCommitQueueCount

Number of commit records read by the session after it has requested but not yet received the commit token.

CommitRecordsReadCommitWithoutTokenCount (Gem)

The number of commit records read while processing a commit and the gem is waiting for the commit token (2x)

CommitRecordsReadCommitWithTokenCount (Gem)

The number of commit records read while processing a commit and the Gem has the commit token.

CommitRetryFailureCount (Gem)

The number of commits that failed after exceeding the retry count.

CommitsSinceLastEpoch (Stone)

Number of commits, excluding reclaim commits by GcGems, since the start of the currently open epoch.

CommitTokenSession (Stone)

The session ID of the gem holding the commit token. If no gem is holding the commit token, the value will be zero.

CompressedLogPagesWrittenByGem (SPC monitor)
CompressedLogPagesWrittenByStone (SPC monitor)

The number of compressed log pages in the cache because of a write done by a Gem or by the Stone, respectively.

CompressionCount (Pgsvr)

Number of times the compression library was called to compress data.

CompressionKbIn (Pgsvr)

Total size of data compressed before compression, expressed in kilobytes.

CompressionKbOut (Pgsvr)

Total size of data compressed after compression, expressed in kilobytes.

CompressionTimeCpu (Pgsvr)

CPU time in milliseconds spent performing data compression.

CompressionTimeReal (Pgsvr)

The real time in milliseconds spent performing data compression.

ContinueTransactionCount (Gem)

Total number of times the session executed the "System continueTransaction" method.

CountBagInteriorPagesWrittenByGem (SPC monitor)
CountBagInteriorPagesWrittenByStone (SPC monitor)

The number of count bag interior pages in the cache because of a write done by a Gem or by the Stone, respectively.

CountBagLeafPagesWrittenByGem (SPC monitor)
CountBagLeafPagesWrittenByStone (SPC monitor)

The number of count bag leaf pages in the cache because of a write done by a Gem or by the Stone, respectively.

CountMapLeafPagesWrittenByGem (SPC monitor)
CountMapLeafPagesWrittenByStone (SPC monitor)

These statistics are obsolete; the page kind is no longer used.

CrashedSlotsInRecoveryCount (SPC monitor)

The current number of slots being recovered which were owned by a crashed process.

CrashedSlotsRecoveredCount (SPC monitor)

The total number of slots the shared cache monitor has recovered because a client process shutdown abnormally.

CrashedSlotThreadTimeRunningMs (SPC monitor)

Approximate real time in milliseconds that the crashed slot recovery thread in the shared page cache monitor has spent doing work.

CrPageLocateForDisposeCount (Stone)

Number of times a commit record page was not found in the stone's commit record cache at commit record disposal time.

DataFaultSleepTime (All, on Solaris only)

The number of milliseconds the process has been faulting in data pages.

DataPageReads (All)

The number of data pages read by the process since it was last started. These page reads are actual disk reads and not reads from the shared page cache.

DataPagesWrittenByGem (SPC monitor)
DataPagesWrittenByStone (SPC monitor)

The number of data pages in the cache because of a write done by a Gem or by the Stone, respectively.

DataRSS (All, on AIX only)

The data resident set size.

DataVmSize (All, on AIX only)

The data virtual memory size.

DeadDataPagesWrittenByGem (SPC monitor)
DeadDataPagesWrittenByStone (SPC monitor)

The number of dead data pages in the cache because of a write done by a Gem or by the Stone, respectively. Dead data pages are intended for disposal.

DeadNotReclaimedKobjs (Stone)

The same as DeadNotReclaimedObjs, but in units of 1000s of objects. This statistic is present for server versions that did not support a sufficiently large value range.

DeadNotReclaimedObjs (Stone)

DeadNotReclaimedObjs is the number of objects that have been determined to be dead (current sessions have indicated they do not have a reference to the objects) but have not yet been reclaimed.

DeadObjsReclaimedCount (Stone)

The total number of dead objects reclaimed since the Stone repository monitor process was last started. For a system in "steady state" for a particular application, look for a uniform discovery rate per garbage collection epoch. Increasing the duration of the epoch should increase this value, but that could also cause larger swings in the amount of free space in the repository.

DecompressionCount (Pgsvr)

Number of times the compression library was called to decompress data.

DecompressionKbIn (Pgsvr)

Total size of data decompressed before decompression, expressed in kilobytes.

DecompressionKbOut (Pgsvr)

Total size of data decompressed after decompression, expressed in kilobytes.

DecompressionTimeCpu (Pgsvr)

CPU time in milliseconds spent performing data decompression.

DecompressionTimeReal (Pgsvr)

The real time in milliseconds spent performing data decompression.

DeferCkptCompleteCount (Stone)

The number of pending commit operations for which the checkpoint will allow itself to be deferred before it completes.

DepMapKeysChanged (Gem)

The total number of objects with DependencyMap changes committed by this session.

DirtyListSize (Gem)

The number of modified committed objects in the temporary object memory dirty list.

DirtyPageSweepCount (Pgsvr)

The number of times the page server has swept the cache for dirty pages or frames to add to the free list.

EmptyBmLeafPagesWrittenByGem (SPC monitor)
EmptyBmLeafPagesWrittenByStone (SPC monitor)

The number of empty bitmap leaf pages in the cache because of a write done by a Gem or Stone, respectively.

Env1sendCacheSerialNum (Gem)

Counter on env 1 send site caches.

EpochForceGc (Stone)

This statistic is 1 when a user has requested that Epoch garbage collection be started manually, and is otherwise zero.

EpochGcCount (Stone)

The number of times that the Epoch garbage collection process was run by the Admin GcGem since the Admin GcGem was started. For a system in steady state, look for uniform periods between runs or a uniform run rate.

EpochLastDuration (Stone)

The duration of the last completed epoch garbage collection operation in seconds.

EpochNewObjs (Stone)
EpochNewKobjs (Stone)

The number of new objects that were created during the last epoch. In some versions, in units of 1024.

EpochPossibleDeadObjs (Stone)
EpochPossibleDeadKobjs (Stone)

The number of possible dead objects found by the last Epoch garbage collection. In some versions, in units of 1024.

EpochScannedObjs (Stone)
EpochScannedKobjs (Stone)

The number of objects scanned by the last Epoch garbage collection. In some versions, in units of 1024.

ExportedSetSize (Gem)

The number of objects in the ExportSet. The ExportSet is a collection of objects for which the Gem process has handed out an Oop to GemBuilder or Topaz. Objects in the ExportSet are prevented from being automatically garbage collected in the Gem's temporary object memory. The ExportSet is used to guarantee referential integrity for objects only referenced by an application, that is, objects that have no references to them within the Gem.

The application program is responsible for timely removal of objects from the ExportSet. The contents of the ExportSet can be examined using hidden set methods defined in class System. In general, the smaller the size of the ExportSet, the better the performance is likely to be. There are a couple of reasons for this relationship. The ExportSet is one of the root sets used for garbage collection. The larger the ExportSet, the more likely it is that objects that would otherwise be considered garbage are being retained. One threshold for performance is when the size of the export set exceeds 2K objects. When its size is smaller than 2K objects, the export set is stored as a single disk page. When its size is larger than 2K, the export set occupies more than one page and is likely to cause additional I/O.

ExportedSetSizePinnedInMemory (Gem)

The number of objects in the ExportSet that cannot drop out of temporary object memory nor be garbage collected by in-memory collection of temporary objects. Will include any objects represented in ExportedSetSize that are not committed. Can be less than ExportedSetSize if ((System gemConfigurationAt:GemDropCommittedExportedObjs) == true).

ExtentFlushCount (All)

The cumulative number of file flush operations performed on any extent by the process. Note that extents residing on raw partitions do not require flushing. On UNIX systems, file flushes are performed by calling the fsync() function. During a checkpoint, each extent is flushed once, except for the primary extent which is flushed twice. Most extent flushes are performed by the AIO page servers.

ExtentGrowTimeMs (Stone)

Total amount of real time in milliseconds the stone has spent increasing the size of the repository by growing the extent(s).

ExtentGrowTimePerGrow (Stone)

Average amount of real time in microseconds it takes the stone to perform a single grow operation on an extent.

ExtentGrowTotal (Stone)

Number of times the stone has increased the size of the repository by growing the extent(s).

FailedCommitCount (Gem)

The number of attempts to commit that failed due to concurrency conflicts.

ForcedDisconnects (Stone)

Number of session logouts for which stone forced the disconnect of OOB socket.

FragmentBmPagesWrittenByGem (SPC monitor)
FragmentBmPagesWrittenByStone (SPC monitor)

The number of bitmap fragment pages in the cache because of a write done by a Gem or by the Stone, respectively.

FrameCount (SPC monitor)

The size of the shared cache in frames.

FramesAddedToFreeList (All)

The number of frames added to the free list since the session (for Gems), shared page cache, or Stone started.

FramesFromFindFree (All)

The number of times the process acquired a page frame in the shared page cache by scanning the cache entries. The process tries to find free frames this way instead of taking them from the free list when the number free is below the value set by the GEM_FREE_FRAME_LIMIT configuration option. While scanning for free frames under those conditions is desirable from a system perspective, it represents additional overhead for the particular session.

FramesFromFreeList (All)

The number of times the process acquired a page frame in the shared page cache from the list of free frames.

FreeFrameCacheNumFrames (All)

The number of frames present in the free frame cache for the process.

FreeFrameCacheSize (All)

The number of frames that the process will add or remove from the shared free frame list in a single operation. A value of zero indicates that free frames are not cached and will be added or removed from the shared free frame list one at a time.

FreeFrameCount (SPC monitor)

The number of unused page frames in the shared page cache. This statistic gives some indication of the utilization of the cache, but it is not tunable. This statistic is valid (non-zero) only for the shared page cache monitor process slot.

FreeFrameLimit (All)

When the number of free frames in the shared page cache is less than the FreeFrameLimit, the Gem scans the cache for a free frame rather than use one from the free frame list, so that the Stone process can use the remaining free frames.

FreeOops (Stone)
FreeOopsK (Stone)

The number of free OOPs in the free list that have not been allocated to a Gem and committed. In some versions, in units of 1024.

FreePages (Stone)

The size of the free page pool for the repository. Free space in the repository is calculated at 16 KB (for GemStone/S 64 Bit) for each page in the free pool.

FreePceCacheEntries (All)

Number of free PCEs (Page Cache Entries) currently in the free PCE cache of the process. The capacity of the free PCE cache is equal to the capacity of the free frame cache.

FreePceCount (SPC monitor)

Current size of the list of free PCEs (Page Cache Entries).

GarbageCollectionState (Gem)

The state of a garbage collection task in the Admin GcGem or a user performing a garbage collection operation. Values are defined as follows:

0 - Garbage collection not active
1 - Initialization
2 - Mark-sweep
3 - Compute dead objects
4 - Remove not dead objects
5 - Finalize weak references
6 - Start record dead
7 - Write possible dead objects to tranlog
8 - Finish record dead
9 - Write not dead objects to tranlog
10 - Send not dead objects to stone

GcHighWaterPage (Stone)

The lowest page ID for which the GC reclaim gems will temporarily ignore because a garbage collection operation is in progress. Pages with IDs less than this value will be reclaimed normally. Page IDs equal to or greater than this value will not be reclaimed. The session performing the garbage collection will increase this value as the operation progresses

GciBytesRcvd (Gem)

Number of bytes received from the RPC GCI client.

GciBytesSent (Gem)

Number of bytes sent to the RPC GCI client.

GciPhysBytesRcvd (Gem)

Number of bytes received from the RPC GCI client before decompression.

GciPhysBytesSent (Gem)

Number of bytes sent to the RPC GCI client after compression.

GciRpcCommandsServiced (Gem)

The number of GCI RPC commands serviced by this Gem.

GciRpcKeepAlivePacketCount (Gem)

Number of keep-alive packets received from the GCI client on a remote host.

GciRpcLastCommandServiced (Gem)

The numeric identifier of the last command executed by this gem on behalf of its RPC client. Not used by linked gems.

GciRpcTimeInClientRequests (Gem)

Approximate total number of real milliseconds spent executing requests from the GCI client. Not used by linked gems.

GciRpcTimeInLastCommand (Gem)

Approximate number of real milliseconds spent executing the last command from this gem's RPC client. Not used by linked gems.

GcLockKind (Stone)

Indicates the state of the garbage collection lock and why it is being held:

0 - Free
1 - MarkForCollection
2 - FindDisconnectedObjects
3 - Pre-MarkGcCandidates (loading candidate objects)
4 - MarkGcCandidates
5 - Epoch GC
6 - Reclaim All
7 - (not used)
8 - Repository Scan (listInstances, listReferences, etc)

GcVoteState (Stone)

GcVoteState tracks the progress of a number of phases within garbage collection. Possible values are:

0 - Not voting
1 - Processing possible dead symbols (in older versions: voting on possible dead set).
2 - Waiting for all gems to vote (in older versions: vote complete, waiting for WSUS).

3 - Gems have finished voting (in older versions:WSUS in progress).
4 - Write-set-union sweep in progress (in older versions: WSUS complete).
5 - Write-set-union sweep completed.

Note that all of these phases must complete before the PossibleDead objects can be "promoted" to DeadNotReclaimed objects. The Admin GcGem is responsible for performing the possible dead write set union sweep, and must be running for this to occur.

For details about the voting phase of garbage collection, refer to the *System Administration Guide* for your version of GemStone.

GcWsUnionSize (Stone)

The number of objects in the write set union used to finalize possible dead objects. All objects in the WSU must be scanned to determine if any objects reference one or more possible dead objects.

GcWsUnionSweepCount (Stone)

The total number of sweeps of the possible dead write set union that have been done by the Admin GcGem since it started.

GemFreePages (Gem)

The number of free pages that the VM or Gem has acquired but has not yet used.

GemHasCommitToken (Gem)

A boolean that indicates whether the Gem holds the commit token.

GemsInCacheCount (SPC monitor)

The total number of Gems using the shared page cache whose process slot you are viewing. This is useful for distinguishing Gems using a local shared page cache from those using a remote shared page cache.

GemTempObjCacheSizeKb (Gem)

The value of the session's GEM_TEMPOBJ_CACHE_SIZE_KB configuration parameter.

GlobalDirtyPageCount (SPC monitor)

The total number of pages in the shared cache that are dirty but not yet eligible for asynchronous writing to the disk because they have not yet been committed. If this value is very large, then very large transactions may be filling the cache. Otherwise, if the Stone repository monitor is running on this cache, the Stone's private page cache size may be too small. This statistic is available only for the shared page cache monitor's slot (currently Slot 0).

GlobalStat00 — GlobalStat47 (Stone)

User-defined statistics that can be written and read by any Gem on any Gem server; that is, these statistics are stored in the shared page cache of the Stone, rather than that of the machine on which the Gem is running. There are 48 global cache statistic slots available.

GsMsgCount (Stone)

The number of messages processed by the Stone on behalf of Gems. This statistic can help determine how busy the Stone is.

GsMsgKind (Stone)

An integer identifying the kind of message the Stone is currently processing.

GsMsgSessionId (Stone)

The session identifier of the Gem for which the Stone is currently processing a message.

HeapKBytes (All, on Solaris only)

The size of the process's heap in kilobytes.

ImageKBytes (All, on Linux and Solaris only)

The size of the process's image in kilobytes.

IndexProgressCount (Gem)

Used to monitor the status of certain index operations:

0 - Inactive
1 - Identity index audit in progress.
2 - Equality index audit - auditing root terms.
3 - Equality index audit - auditing NSC counts.
4 - Equality index audit - auditing btree counts.
5 - IndexManager>>removeAllIndexes in progress.

The ProgressCount statistic is also used to indicate the progress of some of these operations. When it is used, it starts at the number of path terms to be checked and is decremented each time the audit of a path term has completed.

InputBlocks (All, on AIX and Solaris only)

The number of input blocks.

InvalidPagesWrittenByGem (SPC monitor)
InvalidPagesWrittenByStone (SPC monitor)

The number of invalid (i.e. empty) pages in the cache because of a write done by a Gem or by the Stone, respectively.

IVolCSW (All)

The number of times the process was forced to do a context switch.

KernelFaultSleepTime (All, on Solaris only)

The number of milliseconds the process has been faulting in kernel pages.

LastCommandFromClient (Pgsvr)

The numeric index of the last message sent to the page server by its client.

LastCommitRecordReleaseReasonCode (Stone)

A code indicating the reason the stone released the commit record reference for a session as follows:

1 - Lost OT root.
2 - Logout processing when the logout was not requested.
3 - System terminateAllSessionsReferencingOldestCr method.
4 - System stopZombieSession method.

LastCommitRecordReleaseSessionId (Stone)

The session ID of the last session for which the stone released the commit record reference due to an abnormal event.

LastErrorNumber (Gem)

The number of the last error processed by the session. Fatal errors are not reported in this statistic.

LastMarkSweepReasonCode (Gem)

An internal code which indicates the reason for the last Mark/Sweep operation.

LastScavengeReasonCode (Gem)

An internal code which indicates the reason for the last scavenge operation.

LastSessionFatalError (Stone)

The session ID of the last session that reported a fatal error to the stone.

LastSessionIdStopped (Stone)

The session ID of the last session that was sent a stop session message.

LastSessionIdTerminated (Stone)

The session ID of the last session that was forcibly logged off by the stone.

LastSessionLostOt (Stone)

The session ID of the last session that was sent a SigLostOtRoot signal.

LastSessionSigAbort (Stone)

The session ID of the last session that was sent a SigAbort.

LastSigTermGemPid (Stone)

The process ID of the last local session to which the stone sent a SIGTERM signal to the gem process.

LastSigTermGemSessionId (Stone)

The session ID of the last local session to which the stone sent a SIGTERM signal to the gem process.

LastSigTermPageServerSessionId (Stone)

The session ID of the last remote session to which the stone sent a SIGTERM signal to the page server process.

LastSigTermPgsvrPid (Stone)

The process ID of the last remote session to which the stone sent a SIGTERM signal to the page server process.

LastSleepTimeBetweenScans (SPC monitor)

Number of milliseconds the shared page cache monitor statistics thread slept after the last complete scan of the cache.

LastSleepTimeWithinScan (SPC monitor)

Number of milliseconds the shared page cache monitor statistics thread last slept while in the loop that computes cache statistics. The thread will sleep for a short interval each time it scans 100,000 cache frames.

LastSmcQueueSize (Stone)

The size of the SMC (Shared Memory Communication) queue when it was last processed by the stone.

LdiThreadOperations (Stone)

The number of wakeups from semaphore wait in stone NetLdiCall Thread.

LocalCacheAllocatedPceCount (Gem, Stone)

The number of page cache entries that the process has allocated for collision chains.

LocalCacheFreeFrameCount (Gem, Stone)

The number of frames in the private page cache that are free.

LocalCacheFreePceCount (Gem, Stone)

The number of page cache entries in the free pool.

LocalCacheOverflowCount (Gem, Stone)

The number of times that a page was moved from private cache to the shared cache because the private cache was full.

LocalCachePceCountLimit (Gem, Stone)

The maximum number of page cache entries that the process will allocate before performing a reclaim.

LocalCachePceReclaimCount (Gem, Stone)

The number of page cache entry reclaim operations performed by the process.

LocalCacheStalePcesRemovedCount (Gem, Stone)

The number of stale page cache entries that the process has removed from its private page cache lookup table. Stale PCEs occur when a reference to page in the shared cache becomes invalid because the page is removed from the cache by another process.

LocalCacheValidPcesRemovedCount (Gem, Stone)

The number of valid page cache entries removed by a reclaim operation.

LocalDirtyPageCount (SPC monitor)

The total number of pages in the shared cache that are dirty and eligible for asynchronous writing to the disk. The Stone's AIO page server will write these pages to the disk. This statistic is available only for the shared page cache monitor's slot (currently Slot 0).

LocalPageCacheHits (All)

The number of times a page lookup found the page in the shared page cache. No I/O was required to access the page.

LocalPageCacheMisses (All)

The number of times a page was not found in the shared cache and a read operation was required to get the page.

LockReqQueueSize (Stone)

The number of Gem session processes waiting for a commit to complete so that their lock request can be serviced.

LockWaitQueueSize1 — LockWaitQueueSize10 (Stone)

The number of sessions waiting for the Application object lock with the given number. (System waitForApplicationWriteLock:queue:autoRelease:)

LockWaitSleepTime (All, on Solaris only)

The number of milliseconds the process has been waiting for a user lock.

LogHighQueueAdds (Stone)

Number of additions to the LogHighQueue. See LogHighQueueSize.

LogHighQueueSize (Stone)

The number of sessions waiting for transaction log writing to be not saturated. This queue holds mostly committing sessions.

LoginLogFlushes (Stone)

Number of times the stone's login log thread has flushed writes to the login log file.

LoginLogThreadOperations (Stone)

Number of operations performed by the stone's login log thread

LoginQueueSize (Stone)

The number of sessions waiting for login completion (2x)

LoginRequestsCount (Stone)

The total number of login requests processed since the Stone started.

LogLowQueueAdds (Stone)

Number of additions to the LogLowQueue. See LogLowQueueSize.

LogLowQueueSize (Stone)

The number of lower priority sessions waiting for transaction log writing to be not saturated. This queue holds primarily sessions logging resourceIds such as possibleDead or notDead.

LogRecordPagesWrittenByGem (SPC monitor)
LogRecordPagesWrittenByStone (SPC monitor)

The number of transaction log record pages in the cache because of a write done by a Gem or by the Stone, respectively.

LogRecordsIoCount (Stone)

The number of physical write operations performed on the transaction logs since the Stone repository monitor process was last started. The minimum write to a transaction log is 512 bytes (one log record). The maximum number of bytes written in a single I/O to the transaction log is 64K. The implication for performance tuning is that to achieve the best throughput (in transactions per second) you would like to have as few as possible writes to the transaction logs. The technique for achieving this is to tune the size of the transactions so that each transaction writes one or more completely filled 64K records.

LogRecordsWritten (Stone)

The number of log records that have been written to the transaction logs since the Stone repository monitor process was last started. The size of a log record is 512 bytes.

LogWaitQueueSize (Stone)

The size of the queue that holds sessions waiting for space to become available in a transaction log. This queue should be empty or nearly so unless the space for logging transactions has been exhausted.

LookupCacheSerialNum (Gem)

Counter on per-class method lookup caches.

LosOtDeferLastReason (Stone)

An integer value indicating the most recent reason deferred sending a LostOtRoot signal to a session. The codes are defined as follows:

0 - none - no LostOtRoot signals have been deferred.
1 - the session was executing a garbage collection primitive (markForCollection, etc).

2 - the session held the commit token or was waiting in the commit queue.
3 - the session was blocked on an internal stone queue.
4 - the session was waiting in the run queue or the SMC queue.
5 - the session was performing an abort.
6 - the session was a system gem.

LostOtDeferCount (Stone)

The total number of times Stone has deferred sending a LostOtRoot signal to a session.

LostOtDeferLastSession (Stone)

The session ID of the last session for which the stone deferred the sending of a LostOtRoot signal.

LostOtPagesWrittenByGem (SPC monitor)
LostOtPagesWrittenByStone (SPC monitor)

The number of lost OT pages in the cache because of a write done by a Gem or by the Stone, respectively.

LostOtsReceived (Gem)

The number of Lost OT Root signals received and recognized by this session.

LostOtsSent (Gem)

The number of Lost OT Root signals the Stone has sent to this session, although the session may be in a sleep or I/O wait state and not yet aware of having received the signal. (See LostOtsReceived (Gem), above.)

LwpCurCount (All, on Solaris only)

The current number of light weight processes that exist in the process.

LwpTotalCount (All, on Solaris only)

The total number of light weight processes that have ever contributed to the process's statistics.

MainThreadTimeRunningMs (SPC monitor)

Approximate real time in milliseconds that the main shared page cache monitor thread has spent doing work.

MajFlt (All)

The number of times the process has had a page fault that needed disk access.

MarkSweepCount (Gem)

The number of mark/sweeps executed by the in-memory garbage collector.

MaxImageSize (All, on Linux only)

The maximum (high water) size of the process's image in kilobytes.

MaxRSS (All, on AIX and Linux only)

The high water mark of the processes resident set size.

MaxUserSessions (Stone)

The maximum number of user sessions stone is configured for.

MaxVotingSessions (Stone)

Maximum number of sessions that can concurrently vote on possible dead objects at the end of an MFC (`markForCollection`) or Epoch garbage collection.

MeSpaceAllocatedKBytes (Gem)

(Only in 3.1.0.5)Memory allocated for remSet, in-memory oopMap and map entries, at end of mark/sweep. (31x)

MeSpaceAllocatedBytes (Gem)

The number of bytes allocated for remSet, in-memory oopMap, and map entries.

MeSpaceUsedKBytes (Gem)

Memory in use for remSet, in-memory oopMap and in-use map entries, at end of mark/sweep. (31x)

MeSpaceUsedBytes (Gem)

The number of bytes occupied by remSet, in-memory oopMap, and in-use map entries.

MessageKindToStone (Gem, Pgsvr)

The message type of the most recent message sent to the Stone.

MessagesToStnProcessingCommit (Gem)

The number of messages sent to the Stone while the gem is processing its part of the commit.

MessagesToStnStoneCommit (Gem)

The number of messages sent to the Stone while the Stone is processing its part of the commit.

MessagesToStnWaitingForCommit (Gem)

The number of messages sent to the Stone while waiting for the commit token.

MessagesToStone (Gem, Pgsvr)

The number of messages sent from a session process to the Stone repository monitor.

MethodsRead (Gem)

The number of GsNMethods copied into VM memory since the start of this session.

MilliSecPerIoSample (All)

MilliSecPerIoSample is used as a parameter to implement the configurable I/O limit for GemStone processes. Because a process's I/O rate currently is sampled every five I/O operations, this statistic is computed as 1000/(ioLimit * 5). A value of 1 means that the process has no I/O limit. If the time in milliseconds since the last sample equals or exceeds MilliSecPerIoSample, the process can perform another I/O operation; if not, the process sleeps. MilliSecPerIoSample is particularly useful in limiting I/O rate of a process that is executing a long-running operation. If you want to calculate the current I/O limit, it is given by 1000/(MilliSecPerIoSample * 5).

MinorFaults (All)

The number of times the process has had a page fault that did not need disk access.

MlClearAllCount (Gem)

Number of times that all send-site caches were cleared.

MlFullLookupCount (Gem)

Number of times that fullMethodLookup was called.

MlLuCacheGrowCount (Gem)

Number of times that a per-class lookup cache was grown to larger table size.

MlLuCacheLargeCount (Gem)

Number of times that a per-class lookup cache became a large object.

MlLuCacheResetCount (Gem)

Number of times that a per-class lookup cache was reset.

MlPolyCacheCreateCount (Gem)

Number of times that a polymorphic send-site cache was created.

MlPolyCacheFullCount (Gem)

Number of times that a polymorphic send-site cache overflowed.

MlPolyCacheGrowCount (Gem)

Number of times that a polymorphic send-site cache was grown.

MsgRecv (All, on AIX and Solaris only)

The number of messages received by the process.

MsgSent (All, on AIX and Solaris only)

The number of messages sent by the process.

MtActiveThreads (Gem)

The number of threads actively working on a multi-threaded task in a Gem executable.

MtMaxThreads (Gem)

The maximum number of threads currently configured for a multi-threaded task in a Gem executable. Set by the method that initiated the operation.

MtPercentCpuActiveLimit (Gem)

Can be set by a session to control the maximum number of active threads working on the task. Value must be between 0 and 100. When a non-zero value is set, the controller thread attempts to keep the percentCpuActive stat below this value by limiting the number of threads working on the task.

MtThreadsLimit (Gem)

Can be set by a session to control the maximum number of active threads working on the task. Must be less than or equal to MtMaxThreads.

MultObjPagesWrittenByGem (SPC monitor)
MultObjPagesWrittenByStone (SPC monitor)

The number of multiple object pages in the cache because of a write done by a Gem or by the Stone, respectively.

NetWriteThreadSocketWrites (Stone)

The number of socket write operations attempted in stone NetWrite thread.

NetWriteThreadWakeups (Stone)

The number of wakeups from semaphore wait in stone NetWrite thread.

NewGenSizeKBytes (Gem)

Used memory in the new generation at the end of mark/sweep or scavenge. (31x)

NewGenSizeBytes (Gem)

The number of used bytes in the new generation at the end of mark/sweep.

NewObjsCommitted (Gem)

The number of new objects committed by the most recent transaction committed by this process.

NewObjsCommittedNotLogged (Gem)

Number of newly created objects that were committed but not added to the tranlog because they are reachable from the NotTranloggedGlobals root object.

NewSymbolRequests (Gem)

The number of symbol creation requests by a session to the symbol creation Gem.

NewSymbolsCount (Gem)

The number of new symbols created by this session.

NextSleepTime (Pgsvr)

The number of milliseconds that the page server will sleep during the next sleep period. This value is adjusted to regulate the IO rate of the page server.

NotifyQueueSize (Stone)

The number of Gem session processes using notifiers.

NumCacheWarmers (Stone)

The number of cache warmer gem processes currently operating on the shared cache.

NumEphemerons (Gem)

The number of live ephemerons remaining at the end of the last mark/sweep garbage collection.

NumInLdiQueue (Stone)

The number of uncompleted requests in the stone NetLdiCall thread input list.

NumInLoginLogQueue (Stone)

Number of sessions queued to have entries written to the stone's login log file.

NumInLostOtQueue (Stone)

Number of requests in the stone's Lost OT root queue.

NumInMainInpQueue (Stone)

The number of requests from other threads in the stone main thread input list.

NumInNetReadWorkList (Stone)

The number of tasks in the NetRead thread work list, produced by NetPoll action functions and not yet processed by the NetRead thread.

NumInNetWriteQueue (Stone)

The number of uncompleted requests in the input list to the stone NetWrite thread.

NumLiveSoftRefs (Gem)

The number of instances of SoftReference in temporary object memory found during an attempt to clear SoftReferences. This counter remains at zero until temporary object space grows to at least GEM_SOFTREF_CLEANUP_PERCENT_MEM.

NumNonNilSoftRefs (Gem)

The number of instances of SoftReference in temporary object memory with non-nil, non-special values, that were found during an attempt to clear SoftReferences. This counter

remains at zero until temporary object space grows to at least GEM_SOFTREF_CLEANUP_PERCENT_MEM.

NumProcsSleepingForLock (SPC monitor)

Number of processes on this shared page cache that are currently sleeping while waiting to acquire a spin lock.

NumRefsStubbedMarkSweep (Gem)

The number of references contained in copies of committed objects that were stubbed (converted to a Pom objectId) by in-memory mark/sweep.

NumRefsStubbedScavenge (Gem)

The number of in-memory references that were stubbed (converted to a Pom objectId) by in-memory scavenge.

NumSharedCounters (SPC monitor)

The number of shared counters for this shared cache.

NumSoftRefsCleared (Gem)

The number of times that the in-memory garbage collector has cleared the `value` instance variable in instances of SoftReference.

NumVotingSessions (Stone)

The number of sessions that the stone has requested to vote on possible dead objects.

ObjectMemoryGrowCount (Gem)

On AIX and HPUX only, the number of times object memory was grown or shrunk by allocating a new virtual memory region with mmap(). Always zero on other platforms.

ObjectsRead (Gem)

The number of committed objects copied into VM memory since the start of this session.

ObjectsReadInBytes (Gem)

Total size in bytes of committed objects copied into VM memory since start of session.

ObjectsRefreshed (Gem)

The number of committed objects in VM memory that have been re-read from the page cache after transaction boundaries, since the start of this session.

ObjectTablePageReads (All)

The number of object table pages read by the process since it was last started. These page reads are actual disk reads and not reads from the shared page cache.

ObjsCommitted (Gem)

The number of objects committed by the most recent transaction committed by this process.

ObjsCommittedNotLogged (Gem)

The total number of objects (new and modified objects) reachable from the NotTranloggedGlobals root object which the session has committed.

OldestCrSession (Stone)

The session ID of a session referencing the oldest commit record. Note that more than one session may reference a commit record, which prevents it from being reclaimed. A value of 0 (in recent versions of GemStone/S 64 Bit) or -1 indicates that the oldest commit record is not referenced by any session.

OldestCrSessNotInTrans (Stone)

The session ID of a session that is not in a transaction that is currently referencing the oldest commit record, which may be preventing it from being reclaimed. Note that more than one session may reference a commit record. A value of 0 (in recent versions of GemStone/S 64 Bit) or -1 indicates that the oldest commit record is not referenced by any session.

OldGenFullCount (Gem)

Number of times the old gen space was found to be full.

OldGenPreGcSizeBytes (Gem)
OldGenPreGcSizeKBytes (Gem)

Used memory in the old generation at the start of mark/sweep. (2x, 31x)

OldGenSizeBytes (Gem)
OldGenSizeKBytes (Gem)

The number of used bytes in the old generation at the end of mark/sweep. In some versions, in units of 1024.

OmBytesFlushed (Gem)
OmKBytesFlushed (Gem)

The total number of temporary object memory bytes flushed to Pom during commit attempts for this session. In some versions, in units of 1024.

OopNumberHighWaterMark (Stone)
OopNumberKHighWaterMark (Stone)

The highest number of object identifiers allocated by the system. In some versions, in units of 1024.

OopsReturnedByGemsCount (Stone)

The total number of free oops returned to the stone by any gem.

OtherPageReads (All)

The number of pages read by the process that were not object table, data, or bitmap pages since the process was started. These page reads are actual disk reads and not reads from the shared page cache.

OtInternalPagesWrittenByGem (SPC monitor)
OtInternalPagesWrittenByStone (SPC monitor)

The number of object table internal pages in the cache because of a write done by a Gem or by the Stone, respectively.

OtLeafPagesWrittenByGem (SPC monitor)
OtLeafPagesWrittenByStone (SPC monitor)

The number of object table leaf pages in the cache because of a write done by a Gem or by the Stone, respectively.

OutputBlocks (All, on AIX and Solaris only)

The number of output blocks.

PageDisposesDeferred (Stone)

The number of times a page disposal had to be deferred. This deferral can be caused by an asynchronous operation (checkpoint) being in progress on the page or by the page being attached or locked.

PageIoCount (All)

The number of page I/O (page read or page write) calls done by the process since it was last started. Each I/O call may read or write more than one page.

PageIoTimeOverallAvg (All)

The average duration of a page I/O (page read or page write) call in microseconds.

PageIoTime10SampleAvg (All)

The average duration of a page I/O (page read or page write) call in microseconds. The average is computed for the last ten I/O operations, and is updated every ten I/O operations.

PageIoTime100SampleAvg (All)

The average duration of an I/O call (page read or page write) in microseconds. The average is computed for the last 100 I/O operations, and is updated every ten I/O operations.

PageLocateCount (All)

The number of times that the process located a page. The page may have been read from disk or found in the cache.

PageManagerMaxWaitTimeMs (Stone)

Current value of the STN_PAGE_MGR_MAX_WAIT_TIME configuration option.

PageManagerStarvedCount (Stone)

Number of times the page manager session waited for service from the stone longer than STN_PAGE_MGR_MAX_WAIT_TIME milliseconds.

PageMgrCompressionEnabled (Stone)

A boolean value: true if the page manager session is compressing the list of pages that it sends to remote cache page servers, false otherwise.

PageMgrPageRemovalRetryCount (Stone)

Number of times the page manager session has retried removing one or more pages from the shared page caches because the first attempt to remove the pages failed.

PageMgrPagesNotRemovedFromCachesCount (Stone)

The total number of pages that the Page Manager was unable to remove from one or more shared page caches.

PageMgrPagesPendingRemovalRetryCount (Stone)

The current number of pages that could not be removed from shared page caches by the page manager on the first attempt and are waiting to be retried.

PageMgrPagesReceivedFromStoneCount (Stone)

The total number of pages that the Page Manager session received from the Stone to remove from shared page caches.

PageMgrPagesRemovedFromCachesCount (Stone)

The total number of pages that the Page Manager has successfully removed from all shared page caches.

PageMgrPolls (Stone)

The number of times the pagemanager thread has polled.

PageMgrPrintTimeoutThreshold (Stone)

The time threshold in seconds used by the page manager to decide if a message should be printed to its log file indicating that a remote cache page server was slow to respond.

PageMgrRemoteCachePgsvrTimeout (Stone)

Number of seconds the page manager session will wait to receive a response from a remote cache page server.

PageMgrRemoveFromCachesCount (Stone)

The total number of times that the Page Manager has attempted to remove pages from shared page caches.

PageMgrRemoveFromCachesPageCount (Stone)

The total number of pages that the Page Manager has attempted to remove from shared page caches. This statistic includes pages processed by page removal retry operations, which occur whenever a page cannot be removed from a shared page cache on the first attempt.

PageMgrRemoveMaxPages (Stone)

The maximum number of pages the Stone will return to the page manager session in a single batch.

PageMgrRemoveMinPages (Stone)

The minimum batch size of pages that the page manager will process. The page manager will request pages to process from the Stone only if the Stone cache statistic PagesWaitingForRemovalInStoneCount exceeds this value.

PageMgrRemovePagesFromCachesPollCount (Stone)

The number of times that the Page Manager called poll() or select() to determine which cache page servers have completed removing pages from their shared caches. This statistic represents the value during the most recent page disposal operation and is not cumulative. It varies between zero (when there are no remote shared caches on the system) and the number of remote shared page caches.

PageMgrSleepMs (Stone)

The number of milliseconds the pagemanager thread has slept.

PageMgrSleepState (Stone)

The number of times pagemanager thread has slept.

PageMgrThreadWakeups (Stone)

The number of times pagemanager thread has woken up.

PageMgrTimeWaitingForCachePgsvrs (Stone)

The total amount of real time in milliseconds that the page manager has spent waiting to receive data from remote cache page servers.

PageReads (All)

The number of pages read by the process since it was last started. These page reads are actual disk reads and not reads from the shared page cache. Note that a single operation may read more than one page.

PageReadsProcessingCommit (Gem)

The number of pages read while the Gem is processing its part of the commit.

PageReadsStoneCommit (Gem)

The number of pages read while the Stone is processing its part of the commit.

PageReadsWaitingForCommit (Gem)

The number of pages read while waiting for the commit token.

PageServersInCacheCount (SPC monitor)

The total number of page servers attached to the shared cache.

PagesNeedReclaimSize (Stone)

The amount of reclaim work that is pending, that is, the backlog waiting for the Reclaim GcGem to reclaim.

PagesNeedRemovingThreshold (Stone)

Threshold for page manager to process the backlog described by PagesWaitingForRemovalInStoneCount. (2x)

PagesNotFoundInCacheCount (SPC monitor)

The number of pages needing to be removed from the cache that were not found in the cache. This statistic is updated by the cache page server on remote caches, or by the page manager on the Stone's shared page cache.

PagesNotRemovedFromCacheCount (Pgsvr)

The number of pages the page server was unable to remove from the cache. Requests to remove pages come from the stone. This statistic is updated by the cache page server on remote caches, or by the page manager on the Stone's shared page cache.

PagesRemovedDirtyFromCacheCount (SPC monitor)

The number of pages successfully removed from the cache by the cache page server or the Page Manager at the Stone's request. This statistic is updated by the cache page server on remote caches, or by the page manager on the Stone's shared page cache.

PagesRemovedFromCacheCount (SPC monitor)

The total number of pages successfully removed from the cache by the cache page server or the Page Manager at the stone's request.

PagesReturnedByGemsCount (Stone)

The total number of free pages returned to the Stone by any Gem.

PagesWaitingForRemovalDeferred (Stone)

Number of deferred persistent pages waiting to be processed by the Page Manager gem.

PagesWaitingForRemovalInStoneCount (Stone)

The number of pages in the Stone that are waiting to be removed from the shared page cache by the Page Manager.

PagesWaitingForRemovalPersist (Stone)

Number of persistent pages waiting to be processed by the Page Manager gem.

PagesWaitingForRemovalTemp (Stone)

Number of temporary pages waiting to be processed by the Page Manager gem.

PageTablesMemoryKB (All, on Linux only)

The amount of memory dedicated to low-level page tables.

PageWaitQueueSize (Stone)

The size of the queue that holds sessions waiting to be allocated free pages. This queue should be empty or nearly so unless the repository is below its free space threshold.

PageWrites (All)

The number of pages written by the process since it was last started. These page writes are actual disk writes and not just writes into the shared page cache. Unless a large data load is in process, the number should be low for all processes except the Stone's AIO page server process.

PcesAddedToFreeList (All)

Number of PCEs (Page Cache Entries) this process added to the free PCE list.

PcesRemovedFromFreeList (All)

Number of PCEs (Page Cache Entries) this process removed from the free PCE list.

PercentCpuUsed (All, on Linux and Solaris only)

The percentage of recent CPU time used by the process.

PercentMemoryUsed (All, on AIX and Solaris only)

The percentage of real memory used by the process.

PermGenFullCount (Gem)

Number of times the perm gen space was found to be full.

PermGenObjsChanged (Gem)

Number of objects in perm gen that were changed by another process at the last transaction boundary.

PermGenObjsChangedTotal (Gem)

Total number of objects in perm gen that were changed by another process during the life of the session.

PermGenSizeBytes (Gem)
PermGenSizeKBytes (Gem)

The number of used bytes in the perm generation at the end of mark/sweep. Perm generation holds copies of Classes. In some versions, in units of 1024.

PersistentPagesCommittedCount (Gem)

The total number of pages made persistent (committed) by the session. This statistic is updated during commit processing.

PersistentPagesDisposed (Stone)

The number of persistent pages (pages already checkpointed) that have been disposed of while in the Stone's private cache.

PgsvrCheckpointState (Pgsvr)

The state of checkpoint processing within an AIO pgsvr:

0=not active
1=writing dirty pages
2=finished write dirty
3=in fsync

PgsvrWaitQueueSize (Stone)

The number of remote sessions in the Page Server Wait Queue. This queue is used to finalize sessions which are logging out.

PinnedDataPagesCount (SPC monitor)

The number of pinned data pages found in the cache.

PinnedOtPagesCount (SPC monitor)

The number of pinned object table pages found in the cache.

PinnedPagesCount (All)

The number of pages the process has pinned (locked) in the shared cache. Pages may be pinned by more than one process at the same time.

PinnedPrivatePagesCount (All)

The number of pages that the process has pinned (locked) in its private page cache.

PinnedSharedCount (SPC monitor)

The number of pages pinned by more than one process.

PinnedTotalCount (SPC monitor)

The total number of pinned pages found in the cache.

PomGenScavCount (Gem)

The number of times scavenge has thrown away the oldest pom generation space.

PomGenSizeBytes (Gem)
PomGenSizeKBytes (Gem)

The number of used bytes in the pom generation at the end of mark/sweep. Pom generation holds clean copies of committed objects. In some versions, in units of 1024.

PossibleDeadObjs (Stone)
PossibleDeadKobjs (Stone)

The number of objects previously marked as dereferenced in the repository, but for which sessions currently in a transaction might have created a reference in their object space. An object is not declared ("promoted to") dead until each active session verifies the absence of such references during its next commit or abort. In some versions, in units of 1024.

PossibleDeadSymbols (Stone)

The number of symbols found to be not referenced in a markForCollection.

PostCheckpointPages (Pgsvr)

The count of pages written out by the page server during post-checkpoint processing.

PreemptedBitmapPages (Pgsvr)

The number of bitmap pages removed from the shared page cache by this page server.

PreemptedCommitRecordPages (Pgsvr)

The number of commit record pages removed from the shared page cache by this page server.

PreemptedDataPages (Pgsvr)

The number of data pages removed from the shared page cache by this page server.

PreemptedObjectTablePages (Pgsvr)

The number of object table pages removed from the shared page cache by this page server.

PreemptedOtherPages (Pgsvr)

The number of pages removed from the shared cache by this page server that were not data, object table, commit record, or bitmap pages.

PrimitiveNumber (Gem)

The primitive number currently being executed by the session, or 0 if the session is not in a primitive. The session only sets this value for long-running primitives. A non-zero value also indicates the session is immune from termination due to the STN_GEM_TIMEOUT mechanism.

Note: the PrimitiveNumber stat is set to 9999 if the session has executed System class >> disableStoneGemTimeout

ProcessesWaitingForQueueLocks (SPC monitor)

Number of processes attached to the shared cache that are spinning while attempting to acquire a queue lock.

ProcessId (All)

The operating system processId for the process associated with this shared page cache process slot.

ProcessName (All)

This statistic identifies the process kind (Gem, Stone, page server, or shared page cache monitor).

ProgressCount (Gem)
ProgressKobjs (Gem)

You can use this statistic to monitor the progress of certain Repository methods that may run for extended periods.

- During objectAudit, ProgressCount is set to the number of live data pages at end of the object table scan, then decremented as data pages are scanned.
- During _findPagesContainingOops:, ProgressCount is set to the total number of pages in the repository and decremented as pages are processed.
- During markForCollection, ProgressCount is incremented as live objects are marked during the marking phase, reset to zero, and incremented as possible dead objects are computed.
- During epoch garbage collection, ProgressCount is incremented as live objects within epoch are marked, then reset to zero.
- During fullBackupTo: and restoreFromBackup:, ProgressCount tracks the number of objects written to or restored from the backup file(s).
- During objectAudit, ProgressCount is incremented as the object table is scanned, then decremented as data pages are scanned.
- While reading and writing an FDC file of OOPs, ProgressCount is incremented as OOPs are read or written.

RcConflictCount (Gem)

The number of commits that resulted in an RC conflict.

RcRetryQueueSize (Stone)

The number of sessions waiting for the RcRetry object lock. (System waitForRcWriteLock:)

ReadLocksSize (Gem, Stone)

The number of objects that are read locked.

RebuildScavPagesForCommitCount (Gem)

The total number of times the gem rebuilt its list of scavengeable pages while processing a commit.

RecentActiveProcessCount (SPC monitor)

The number of active processes attached to the shared page cache. This statistic is computed every half second and has decays quickly.

ReclaimCount (Stone)

The number of reclaims performed by a Reclaim GcGem process since the Stone repository monitor was last started.

ReclaimedPagesCount (Stone)

The number of pages reclaimed by a Reclaim GcGem process since the Stone repository monitor process was last started. The count indicates the number of pages that have been or will soon be placed back into the repository's pool of free pages.

ReclaimedSymbols (Stone)

The number of symbols that have been reclaimed since the stone was started.

RecoverFreeFrameWaitTime (Stone)

The time in seconds spent by the main recovery thread waiting for free frames in the shared page cache. Recovery performance may be improved by increasing the size of the shared page cache.

RecoverNumBufs (Stone)

The total number of logEntryBuffers allocated in the heap.

RecoverNumBufsForSessions (Stone)

The number of logEntryBuffers used to hold records for sessions not yet committed.

RecoverNumBufsInFreeList (Stone)

The number of logEntryBuffers currently in the freeList.

RecoverNumBufsInWorkQueue (Stone)

The number of logEntryBuffers currently in the workQueue.

RecoverReadThreadWaitTime (Stone)

The time in seconds that the reader thread spent waiting for the main recovery thread to catch up processing the log buffers.

RecoverReclaimOopsWaitTime (Stone)

The time in seconds spent by the main recovery thread waiting for oops to be reclaimed. Recovery performance may be improved by adding more reclaim gems.

RecoverTimeLag (Stone)

The difference in commit time (restoreLogCommitTime - originalCommitTime), gives a relative idea of how hot a hot standby is.

RecoverTranlogBlockId (Stone)

The block ID of the transaction log currently being replayed during system recovery or restore.

RecoverTranlogFileId (Stone)

The file ID of the transaction log currently being replayed during system recovery or restore.

RejectedProcsCount (SPC monitor)

The total number of processes which attempted to connect to the shared page cache but were rejected because the maximum number of processes had already attached.

RemoteCachesNeedServiceCount (Stone)

The number of remote caches that require service from the Page Manager. Caches need service when they are starting or shutting down.

RemoteSharedPageCacheCount (Stone)

The total number of remote shared page caches attached to the system.

RemoteSharedPageCacheMax (Stone)

Current setting of the STN_MAX_REMOTE_CACHES configuration parameter

RepBkupRestPagesWrittenByGem (SPC monitor)
RepBkupRestPagesWrittenByStone (SPC monitor)

These statistics are obsolete; the page kind is no longer used.

ReposSizeMB (Stone)

The size of the stone's repository in MB.

ResponsesSentEarly (Stone)

Number of times the stone was able to send a response to the client gem or page server without using the semaphore.

ResponsesSentNormal (Stone)

Number of times the stone sent a response to the client gem or page server using the semaphore.

Root40PagesWrittenByGem (SPC monitor)
Root40PagesWrittenByStone (SPC monitor)

The number of root 4.0 pages in the cache because of a write done by a Gem or by the Stone, respectively.

RootPagesWrittenByGem (SPC monitor)
RootPagesWrittenByStone (SPC monitor)

The number of root pages in the cache because of a write done by a Gem or by the Stone, respectively.

RSSData (All, on Linux only)

On Linux, the data resident set size. On other platforms, the combined data and stack resident set size.

RSSKBytes (All, on Linux and Solaris only)

The size of the process's resident set size in kilobytes.

RSSLib (All, on Linux only)

The library resident set size. Always zero in Linux 2.6 and later.

RSSStack (All, on Linux only)

The stack resident set size.

RSSText (All, on Linux only)

The text resident set size.

RunQueueSize (Stone)

The number of Gem session processes waiting for service from the Stone repository monitor.

ScavengeCount (Gem)

The number of scavenges executed by the in-memory garbage collector.

ScavengeOverflows (Gem)

Number of times an in-memory GC scavenge operation could not be completed because old space was full. The scavenge will be promoted to a mark/sweep GC.

ScavsPromToMkSwCount (Gem)

Number of times a scavenge operation was promoted to a Mark/Sweep operation.

SessionId (All)

The GemStone sessionId associated with this client.

SessionPerformingBackup (Stone)

The session ID of the session that is performing a full backup; 0 if a full backup is not in progress.

SessionStat00 — SessionStat47 (Gem)

These are computed by user code to define statistics associated with a session. There are 48 session cache statistic slots available.

SessionWithGcLock (Stone)

The sessionId of the session holding the GcLock. A value of 1 indicates that the lock is held by Stone recovery or restore; 0 means that the lock is not issued.

ShadowedPagesCount (Gem)

The number of data pages added to the reclaim list due to commits by this Gem. This statistic is only updated during a commit.

SharedMemorySize (All, on AIX only)

The integral size of shared memory.

SigAbortsReceived (Gem)

The number of times the Stone has signaled this session to abort, that it has received and recognized.

SigAbortsSent (Gem)

The number of times the Stone has signaled this session to abort, although the session may be in a sleep or I/O wait state and not yet aware of having received the signal. (See SigAbortsReceived (Gem), above.)

SignalsReceived (All, on AIX and Solaris only)

The total number of operating system signals this process has received.

SleepDuringDisposeTempPageCount (Gem)

Total number of times the Gem slept while waiting to dispose of a temporary page. Each tick of the counter represents 5 milliseconds of sleep. If you encounter excessive counts, please report them to GemStone Technical Support.

SlotsCrashedCount (SPC monitor)

The total number of slots for which the shared cache monitor has attempted recovery because a client process shutdown abnormally.

SlotsFreeCount (SPC monitor)

The number of free process slots currently available in the cache.

SlotsTotalCount (SPC monitor)

The maximum number of processes that can concurrently attach to the shared page cache.

SpinLockCount (SPC monitor)

The current setting of the SHR_SPIN_LOCK_COUNT parameter. It determines how many times a process will attempt to acquire a spin lock before sleeping on a semaphore.

SpinLockFreeFrameSleepCount (SPC monitor)

The number of times the process was forced to sleep on a semaphore while attempting to acquire the free frame list spin lock.

SpinLockFreePceSleepCount (SPC monitor)

The number of times the process was forced to sleep on a semaphore while attempting to acquire the free page cache entry spin lock.

SpinLockHashTableSleepCount (SPC monitor)

The number of times the process was forced to sleep on a semaphore while attempting to acquire a hash table spin lock.

SpinLockNewSymSleepCount (SPC monitor)

The number of times a process was forced to sleep on a semaphore waiting for the NewSymbols spinLock.

SpinLockOtherSleepCount (SPC monitor)

The number of times the process was forced to sleep on a semaphore while attempting to acquire a shared counter spin lock.

SpinLockPageFrameSleepCount (SPC monitor)

The number of times the process was forced to sleep on a semaphore while attempting to acquire a page frame spin lock.

SpinLockSmcQSleepCount (SPC Monitor)

Number of times a session was forced to sleep on a semaphore waiting for the Smc spinLock. The SMC queue allows sessions to communicate with the Stone via shared memory. (2x)

StackKBytes (All, on Solaris only)

The size of the process's stack in kilobytes.

StatsThreadTimeRunningMs (SPC monitor)

Approximate real time in milliseconds that the statistics thread in the shared page cache monitor has spent doing work.

StnAioCompleted1Suspend (Stone)

Number of AIO requests completed after one call to aio_suspend(). (2x)

StnAioCompletedNoSleep (Stone)

Number of AIO requests that were completed without sleeping. The AIO thread in stone will sleep for a short time if aio_suspend() returns prematurely several times. (2x)

StnAioCompletedNoSuspend (Stone)

Number of AIO requested completed without calling aio_suspend(). (2x)

StnAioCompletionFailures (Stone)

The number of aio_write() operations for which either aio_error() or aio_return() reported that the AIO failed.

StnAioFsyncCount (Stone)

Number of times the fsync() call was made by the AIO wait thread in stone.

StnAioFsyncFailures (Stone)

The number of fsync() operations in AioWait thread which failed.

StnAioLastSuspendCount (Stone)

Number of times that aio_suspend() was called to complete the last AIO request. (2x)

StnAioMainTimeInAioWrite (Stone)

Total real time in milliseconds that the main thread spent executing the aio_write() call. aio_write() is used to initiate asynchronous writes to the tranlog and should not block.

StnAioNumWriteThreads (Stone)

Value of the STN_NUM_AIO_WRITE_THREADS Stone configuration parameter, which determines how many threads are dedicating to writing to the tranlog.

StnAioSuspendPrematureReturn (Stone)

Total number of times the aio_suspend() call indicated an AIO completed, but the aio_error() call returned EINPROGRESS, indicating the AIO is not complete. (2x)

StnAioSuspendTimeoutCount (Stone)

Total number of times the aio_suspend() timer expired because no completed AIO requests were found. (2x)

StnAioSyncWritesAfterCancel (Stone)

Total number of times the AIO wait thread performed a blocking write because an async IO was cancelled. (2x)

StnAioTimeInFsyncMs (Stone)

Number of real milliseconds the AIO wait thread in stone spent blocked by the fsync() call when flushing data to the tranlog.

StnAioTotalSleepCount (Stone)

Total number of times the AIO wait thread in stone went to sleep while waiting for an AIO to complete. (2x)

StnAioWaitLastTime (Stone)

Time in microseconds that it took the AIO thread to complete the last AIO request. (2x)

StnAioWaitsForWork (Stone)

The number of times the Stone's AIO wait thread waited on semaphore for more work.

StnAioWaitTotalTime (Stone)

Approximate total real time in milliseconds that the AIO wait thread spent waiting for asynchronous I/O requests to complete.

StnAioWriteEAGAIN (Stone)

Total number of times the aio_write() call returned a status of EAGAIN to the main thread in stone. (2x)

StnAioWriteFailures (Stone)

The number of aio_write() calls by the Stone main thread which failed with other than EAGAIN. Stat should normally be zero.

StnAioWriteQueueHighWaterSize (Stone)

High water mark of the StnAioWriteQueueSize statistic.

StnAioWriteQueueSize (Stone)

Size of the tranlog write request queue.

StnAioWritesQueuedCount (Stone)

Total number of tranlog write requests that have been queued.

StnAioWriteThreadsIdle (Stone)

Number of tranlog write threads in the Stone process that are currently idle.

StnCrBacklogThreshold (Stone)

Current setting of the STN_CR_BACKLOG_THRESHOLD Stone configuration parameter.

StnGetLocksCount (Stone)

The total number of times the Stone retrieved the lock set and passed it to a remote gem.

StnLoopCount (Stone)

The total number of times the Stone has executed its service loop. If this number remains unchanged for a significant period (for example, ten seconds or so), the Stone has hung.

StnLoopHibernateCount (Stone)

The total number of times the Stone went to sleep waiting for a network event to occur. This state occurs when StnLoopState is set to 17.

StnLoopNoWorkThreshold (Stone)

Current setting of the STN_LOOP_NO_WORK_THRESHOLD stone configuration parameter.

StnLoopsNoWork (Stone)

Number of times the stone has executed its main service loop and found no work to perform. This counter is reset to zero each time the stone performs work.

StnLoopsSinceSleep (Stone)

Number of times the stone has executed its main service loop since sleeping. This counter is reset to zero each time the stone sleeps while waiting for work.

StnLoopState (Stone)

An integer that identifies where, in the Stone control loop, the Stone process is currently executing. For a meaningful statistic, set your sample rate to faster than a second. For state definitions, consult GemStone technical support.

StnMainWaitsForFreeAio (Stone)

The number of times the stone main thread waited for free AIO buffers to become available.

StnMessagesNeedWakeUpCount (Stone)

Number of requests sent to the stone when the stone was sleeping and was awoken.

StnMessagesNoWakeUpCount (Stone)

Number of requests sent to the stone when the stone was already awake.

StnRemoteCachePgsvrTimeout (Stone)

Current value of the STN_REMOTE_CACHE_PGSVR_TIMEOUT stone configuration parameter. (2x)

StnSmcSpinLockCount (Stone)

Current value of the STN_SMC_SPIN_LOCK_COUNT stone configuration parameter.

StnTranQToRunQThreshold (Stone)

Current setting of the STN_TRAN_Q_TO_RUN_Q_THRESHOLD stone configuration parameter.

StoneCommitState (Stone)

This internal statistic is used to determine the state of the Stone's commit processing.

StoppedTime (All, on Solaris only)

The number of milliseconds the process has been stopped.

SwapCount (All, on AIX and Solaris only)

The number of times the process swapped out of memory to disk.

SymbolCreationQueueSize (Stone)

The number of sessions waiting for a Symbol creation request to be processed.

SystemCalls (All, on Solaris only)

The total number of system calls.

SysTime (All)

The number of milliseconds the process has been using the CPU to execute system calls.

TargetFreeFrameCount (SPC monitor)

The minimum number of unused page frames the free frame page server(s) will attempt to keep in the cache. The free frame count can still fall below this value if the cache contains mostly dirty pages, which free frame page servers cannot preempt.

TargetPercentDirty (Pgsvr)

The percent of dirty pages that AIO page servers try to maintain in the shared cache. If the dirty pages are below this target then the I/O rate will be limited on the next scan. If the dirty pages are above this target then the I/O rate will be set to AioRateMax.

TempObjSpacePercentUsed (Gem)

The approximate percentage of the total reserved temporary object memory for this session which is in use. Sessions are likely to encounter an out of memory error if this value approaches or exceeds 100%. This statistic is only updated at the end of a mark/sweep operation.

TempPagesDisposed (Stone)

The number of temporary pages (pages allocated since the last checkpoint) that have been disposed.

TextFaultSleepTime (All, on Solaris only)

The number of milliseconds the process has been faulting in text pages.

TextRSS (All, on AIX only)

The text resident set size.

ThreadCount (All, on AIX and Linux only)

Number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.

TimeInCommitRecordDisposal (Stone)

The total amount of real time in milliseconds that the Stone has spent disposing commit records. Commit records disposed during repository startup are not included in this statistic.

TimeInFramesFromFindFree (All)

The cumulative number of milliseconds that the Gem or Stone has spent scanning the shared page cache for a free frame since the session (for Gems) or Stone started.

TimeInGetPagesForPageMgr (Stone)

Real time in milliseconds spent by the stone processing requests from the page manager session for lists of pages to remove from shared page caches.

TimeInMarkSweep (Gem)

The real time in milliseconds spent in in-memory garbage collector mark/sweeps.

TimeInPgsvrNetReads (Stone, Gem)

The cumulative amount of real time in milliseconds that a session or Stone has spent reading network data from a page server.

TimeInPgsvrNetWrites (Stone, Gem)

The cumulative amount of real time in milliseconds that a session or Stone has spent writing network data to a page server.

TimeInProcessPagesFromPageMgr (Stone)

Real time in milliseconds spent by the stone processing the list of pages from the page manager session which were removed from shared page caches.

TimeInScavenges (Gem)

The real time in milliseconds spent in in-memory garbage collector scavenges.

TimeInStnGetLocks (Stone)

The total time spent by the Stone retrieving the lock set and passing it to a remote gem.

TimeInStonePageDisposal (Stone)

The elapsed real time in milliseconds the Stone spent performing page disposal operations.

TimeInUpdateUnionsCommit (Gem)

The total real time the gem spent updating its unions while waiting for the commit token.

TimeInUserActions (Gem)

Approximate total number of real milliseconds spent executing user action code.

TimeInWaitsForOtherReaders

The real number of milliseconds the process spent waiting for the read of another process to complete.

TimeLastEpochGc (Stone)

Time the last epoch garbage collection operation was completed by the Admin GcGem, expressed in seconds since January 1, 1970, 00:00:00 UTC.Time to end of the currently open epoch is computed from here.

TimePerformingCommit (Stone)

The number of milliseconds of real time the Stone has spent performing commit processing, not including time taken by asynchronous writes to the transaction log.

TimePerformingReadIo (Pgsvr)

The total amount of real time in milliseconds the page server has spent reading pages from disk.

TimePerformingReadRequests (Pgsvr)

The total amount of real time in milliseconds the page server has spent performing read requests for its client. This statistic includes time reading pages from disk and time searching the shared page cache for pages.

TimeProcessingCommit (Gem)

The cumulative amount of time in milliseconds that the Gem session process has spent doing the processing for commits while it has the commit token.

TimerThreadWakeups (Stone)

The number of wakeups from sleep in stone Timer thread.

TimeSleepingMs (Pgsvr)

Total real time the process has spent sleeping, in milliseconds.

TimeStoneCommit (Gem)

The cumulative amount of time in milliseconds that the Gem session process has waited for the Stone repository monitor to complete commits by this session.

TimeWaitingForCommit (Gem)

The cumulative amount of time in milliseconds that the Gem session process has spent waiting for its turn to commit, that is, the time waiting for the commit token and the Stone's processing time for serialization.

TimeWaitingForIo (All)

The total real time in milliseconds that the process has spent waiting for I/O calls that read or write pages to complete. Only page I/O is included in this statistic. Other types of I/O (such as transaction log writes by the Stone process) are not included.

TimeWaitingForStone (Gem, Pgsvr)

The total time the session spent waiting for a response from the Stone.

TimeWaitingForSymbols (Gem)

Cumulative elapsed time in milliseconds waiting for symbol creation requests to be processed.

TimeWritingStats (Statmon)

The elapsed time in milliseconds that the statmonitor spent collecting and writing the statistics.

TotalAborts (Stone)

The number of abort operations, including read-only commits, performed system-wide since the Stone was started.

TotalBitmapPages (SPC monitor)

Total number of bitmap pages present in the shared page cache.

TotalBmPageReads (SPC Monitor)

Total number of bitmap pages read by all processes currently attached to the shared page cache.

TotalCommits (Stone)

The total number of commits (excluding read-only commits) performed by all processes since the Stone repository monitor was last started.

TotalCrPages (SPC monitor)

Total number of commit record pages present in the shared page cache.

TotalDataPageReads (SPC Monitor)

Total number of data pages read by all processes currently attached to the shared page cache.

TotalDataPages (SPC monitor)

Total number of data pages present in the shared page cache.

TotalFramesAddedToFreeList (SPC Monitor)
TotalKFramesAddedToFreeList (SPC Monitor)

Total number of frames added to the free list by all processes currently attached to the shared page cache. In some versions, in units of 1024.

TotalFramesFromFindFree (SPC Monitor)

Total number of frames found by scanning the cache for all processes currently attached to the shared page cache.

TotalFramesFromFreeList (SPC Monitor)
TotalKFramesFromFreeList (SPC Monitor)

Total number of frames taken from the free list by all processes currently attached to the shared page cache. In some versions, in units of 1024.

TotalFramesFromFreeList (SPC monitor)

Total number of frames taken from the free list by all processes currently attached to the shared page cache.

TotalFramesInFreeFrameCaches (SPC monitor)

Total number of free frames present in the free frame caches of all processes currently attached to the shared page cache.

TotalGemFatalErrors (Stone)

The total number of fatal errors reported to the stone by all gems.

TotalLocalPageCacheHits (SPC Monitor)
TotalLocalPageCacheKHits (SPC Monitor)

Total number of local page cache hits for all processes currently attached to the shared page cache. In some versions, in units of 1024.

TotalLocalPageCacheMisses (SPC Monitor)
TotalLocalPageCacheKMisses (SPC Monitor)

Total number of local page cache misses for all processes currently attached to the shared page cache. In some versions, in units of 1024.

TotalLostOtsSent (Stone)

The total number of SigLostOtRoot signals sent by the stone.

TotalMiscPageReads (SPC Monitor)

Total number of miscellaneous pages read by all processes currently attached to the shared page cache.

TotalNewObjsCommitted (Stone)

The total number of new objects committed by all Gems.

TotalOtherPages (SPC monitor)

Total number of pages present in the shared page cache which are of a page kind other than object table, data, bitmap or commit record.

TotalOtPageReads (SPC Monitor)

Total number of object table pages read by all processes currently attached to the shared page cache.

TotalOtPages (SPC monitor)

Total number of object table pages present in the shared page cache.

TotalPageReads (SPC Monitor)

Total number of pages read by all processes currently attached to the shared page cache.

TotalPageWrites (SPC Monitor)

Total number of pages written by all processes currently attached to the shared page cache.

TotalPcesAddedToFreeList (SPC Monitor)
TotalKPcesAddedToFreeList (SPC Monitor)

Total number of PCEs (Page Cache Entries) added to the free list by all processes currently attached to the shared page cache. In some versions, in units of 1024.

TotalPcesInFreePceCaches (SPC Monitor)

Total number of free PCEs (Page Cache Entries) present in the free PCE caches of all processes currently attached to the shared page cache.

TotalPcesRemovedFromFreeList (SPC Monitor)
TotalKPcesRemovedFromFreeList (SPC Monitor)

Total number of PCEs (Page Cache Entries) removed from the free list by all processes currently attached to the shared page cache. In some versions, in units of 1024.

TotalProcsInCacheCount (SPC monitor)

he total number of processes currently connected to the cache, including crashed processes that have not yet had their cache slot recovered.

TotalSessionsCount (Stone)

The total number of sessions currently logged in to the system.

TotalSessionsStopped (Stone)

The total number of stop session requests initiated by a gem or by the stone.

TotalSessionsTerminated (Stone)

The total number of sessions forcibly logged off by stone.

TotalSigAbortsSent (Stone)

Total number of SigAbort signals the stone has sent to sessions since the stone has been running.

TotalSigTermsSentToGems (Stone)

Total number of times the stone has sent a SIGTERM signal to a gem process.

TotalSigTermsSentToPageServers (Stone)

Total number of times the stone has sent a SIGTERM signal to a page server process.

TotalWaitsForOtherReader (SPC Monitor)

Total number of PageRead operations avoided by all processes currently attached to the shared page cache.

TotEphemeronsFired (Gem)

Total number of ephemerons whose key has been garbage collected. Only updated at the end of a mark/sweep GC.

TrackedSetSize (Gem)

The number of objects in the Tracked Objects Set, as defined by the GCI.

TranlogFileId (Stone)

The file id of the transaction log to which the most recent tranlog entry was written.

TranlogKBytesWritten (Gem)

Total number of KB written to the tranlog by stone on behalf of this session.

TranlogRecordId (Stone)

The record id of the most recent transaction log entry to be written.

TranlogRecordKind (Gem)

The kind of tranlog record last written to the tranlog by stone on behalf of this session.

TranlogRecordsWritten (Gem)

Total number of physical tranlog records written to the tranlog by stone by this session.

TranlogsFull (Stone)

A flag indicating if all configured tranlog partitions or directories are full; 1 means full.

TransactionLevel (Gem)

Indicates if the session is in a transaction. Possible values are:

1 indicates the session is in a transaction.
0 indicates that the session is not in a transaction, but allows a consistent view for reads. Session is subject to sigAbort or lostOT
-1 indicates that the session is in a transactionless state

TrapTime (All, on Solaris only)

The number of milliseconds the process has been in system traps.

TteCrPageFreeCount (Stone)

Total number of free commit record page entries in the stone's internal commit record cache.

TteCrPageFreePoolSize (Stone)

Total number of commit record page entries allocated in the stone's internal commit record cache.

TteFreeCount (Stone)

Total number of free TTEs (transaction table entries) in stone.

TteFreePoolSize (Stone)

Total number of TTEs (transaction table entries) allocated by stone.

UnsharedDataSize (All, on AIX only)

The integral operating system unshared data size.

UnsharedStackSize (All, on AIX only)

The integral unshared stack size.

UpdateUnionsCommitCount (Gem)

The total number of times the gem updated its unions while waiting for the commit token. This count will be at least one for every commit.

UserTime (All)

The number of milliseconds the process has been using the CPU to execute user code.

VolCSW (All)

The number of voluntary context switches done by the process. Note that this counter is always 0 on HP-UX.

VoteNotDead (Gem)

The number of objects that the Gem process removed from the possible dead set the last time that it voted on the possible dead.

VoteNotDeadObjs (Stone)
VoteNotDeadKobjs (Stone)

The total number of objects that have been voted 'not-dead' by all Gem processes thus far in the voting cycle. In some versions, in units of 1024.

VoteOnDeadCount (Gem)

Number of times the session has voted on dead objects.

WaitCpuTime (All, on Solaris only)

The number of milliseconds the process has been waiting for a CPU due to latency.

WaitingForSessionToVote (Stone)

The sessionId of a session which the system is waiting for to complete the voting on possible dead objects. A zero value indicates that it is not waiting.

WaitsForOtherReader (All)

The number of PageRead operations avoided by waiting for read already in progress by another process.

WorkingSetClearedCount (Gem)

Number of times the working set was cleared at a transaction boundary because a large number of objects in the working set were changed by another session.

WorkingSetObjsChanged (Gem)

Number of objects in the working set that were changed by another process at the last transaction boundary.

WorkingSetObjsChangedTotal (Gem)

Total number of objects in the working set that were changed by another process during the life of the session.

WorkingSetSize (Gem)

The number of objects in memory that have an object Id assigned to them. Approximately the number of committed objects that have been faulted in plus the number that have been created.

WriteLocksSize (Gem)

The number of objects that are write locked.

4.2 System Statistics on Linux

This section lists statistics that are generated when statmonitor on Linux is started using options that specify to include system statistics.

Note

The set of system statitics that can be collected varies by server product and version. Not all statistics may be generated by statmonitor in any particlar version of GemStone.

ActiveAnonMemoryKB

The amount of non-file backed memory that has been used more recently.

ActiveFileMemoryKB

The amount of memory used for buffering files that has been used recently.

ActiveMemoryKB

The amount of memory that has been used more recently and usually not reclaimed unless absolutely necessary.

AllocatedSwapKB

The number of kilobytes of swap space have actually been written to. Swap space must be reserved before it can be allocated.

AnonHugePagesKB

The amount of non-file back memory backed by huge memory pages.

AnonymousMemoryKB

The amount of non-file backed memory mapped into userspace page tables.

BounceMemoryKB

The amount of memory used for bounce buffers for block devices.

CachedMemoryKB

The amount of memory used as cache memory.

CachedSwapKB

The amount of swap used as cache memory.

CommitLimitKB

The total amount of memory currently available to be allocated on the system.

CommittedAsKB

The amount of memory presently allocated on the system, including memory allocated by processes that has not yet been used.

CPUs

The number of online cpus on the local machine.

DirtyMemoryKB

The number of kilobytes of memory in the dirty list.

FileBufferSizeKB

The amount of memory used in file buffers.

FreeMemoryKB

The number of kilobytes of memory in the free list.

HardwareCorrupted

A boolean indicating if the system has detected a memory failure.

HugePagesFreeKB

The amount of memory in the huge pages pool that has not yet been allocated.

HugePageSizeKB

The size of a huge memory page in kilobytes.

HugePagesRsvdKB

The amount of memory in the huge pages pool for which a commitment to allocate from the pool has been made, but no allocation has yet been made.

HugePagesSurpKB

The amount of memory in the huge pages pool above the value in /proc/sys/vm/nr_hugepages.

HugePagesTotalKB

The total amount of memory in the huge pages pool.

InactiveAnonMemoryKB

The amount of non-file backed memory that has not been used recently.

InactiveFileMemoryKB

The amount of memory used for buffering files that has not been used recently.

InactiveMemoryKB

The amount of memory which has been less recently used. It is more eligible to be reclaimed for other purposes.

InputKBytes

Amount of data received by the network adaptor in kilobytes.

InputPackets

The total number of packets read from the network interface.

InputPacketsDiscarded

The total number of input packets discarded.

KernelDataMemoryKB

The amount of memory used by the kernel for caching data structures.

KernelDataReclaimableMemoryKB

The amount of memory used by the kernel for caching data structures that may be reclaimed.

KernelDataUnreclaimableMemoryKB

The amount of memory used by the kernel for caching data structures that cannot be reclaimed.

KernelStackMemoryKB

The amount of memory used by the kernel stack.

LoadAverage1

The average number of threads ready to run over the last minute.

LoadAverage5

The average number of threads ready to run over the last five minutes.

LoadAverage15

The average number of threads ready to run over the last fifteen minutes.

LockedMemoryKB

The amont of memory that has been locked using mlock(2) or similar calls. Locked memory cannot be swapped.

MappedMemoryKB

The amount of memory which has been mapped to files.

MulticastInputPackets

The total number of multicast packets read from the network interface.

NfsUnstableMemoryKB

The amount of memory used by NFS pages sent to the server, but not yet committed to stable storage.

OutputKBytes

Amount of data sent by the network adaptor in kilobytes.

OutputPackets

The total number of packets written to the network interface.

PacketsOutboundDiscarded

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

PacketsOutboundErrors

The number of outbound packets that could not be transmitted because of errors.

PacketsReceivedErrors

The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

PercentCpuActive

The percentage of the total available time that has been used to execute user or system code.

PercentCpuIdle

The percentage of the total available time that has been spent sleeping.

PercentCpuIOWait

The percentage of the total available time that has been spent waiting for disk I/O to complete.

PercentCpuNice

The percentage of the total available time that has been used to execute user code in low priority mode.

PercentCpuOther

The percentage of the total available time that has been used to execute tasks which do not belong to any other category.

PercentCpuSystem

The percentage of the total available time that has been used to execute system (i.e. kernel) code.

PercentCpuUser

The percentage of the total available time that has been used to execute user code.

PhysicalMemoryKB

The amount of physical memory on the machine in kilobytes.

Processes

The number of processes in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. Each process represents the running of a program.

QueueLength

The number of IO operations currently in progress.

ReadKBytes

The total number of kilobytes read from the operating system I/O device.

ReadOperations

The total number of read operations done on the operating system I/O device.

ServiceMs

The average number of milliseconds it took to complete an IO.

SharedMemoryKB

The amount of memory enabled for sharing between multiple processes via shmat(2) and mmap(2) with the MAP_SHARED attribute set.

UnallocatedSwapKB

The number of kilobytes of swap space that have not been allocated.

UnevictableMemoryKB

The amount of memory that cannot be swapped.

Utilization

The average percentage of time the disk was busy (not idle).

WaitMs

The average number of milliseconds an I/O spent in the IO queue.

WritebackMemoryKB

The amount of memory which is actively being written back to disk.

WritebackTmpMemoryKB

Amount of memory used by FUSE (Filesystem in Userspace) filesystems.

WriteKBytes

The total number of kilobytes written to the operating system I/O device.

WriteOperations

The total number of write operations done on the operating system I/O device.

4.3 System Statistics on Solaris

This section lists statistics that are generated when statmonitor on Solaris on SPARC or Solaris on x86 is started using options that specify to include system statistics.

Note

The set of system statitics that can be collected varies by server product and version. Not all statistics may be generated by statmonitor in any particlar version of GemStone.

AckedBytes

The total number of bytes acknowledged by received TCP ack segments.

AcksForUnsentData

The total number of acknowledgment TCP segments received for unsent data.

AcksReceived

The total number of acknowledgment TCP segments received.

AcksSent

The total number of acknowledgment TCP segments sent.

ActiveQueueLength

The average number of operations being performed on a system I/O device.

Operations go into the active queue when they are taken from the wait queue because the system bus and device are ready. They leave it when the device completes the operation.

ActiveQueueResponseTime

The average milliseconds and operation stays in the queue. Its calculated by dividing the queue length by total number of operations taken out of the queue.

Operations go into the active queue when they are taken from the wait queue because the system bus and device are ready. They leave it when the device completes the operation.

ActiveQueueServiceTime

The average active queue service time in milliseconds. Its calculated by dividing the time the queue was busy by total number of operations taken out of the queue.

Operations go into the active queue when they are taken from the wait queue because the system bus and device are ready. They leave it when the device completes the operation.

ActiveQueueUtilization

A percentage that represents how much of the queue is being used. Its calculated by dividing the time the queue was busy by the total time available.

Operations go into the active queue when they are taken from the wait queue because the system bus and device are ready. They leave it when the device completes the operation.

AllocatedSwap

The number of megabytes of swap space have actually been written to. Swap space must be reserved before it can be allocated.

AnonymousPagesFreed

The total number pages that contain heap, stack, or other changeable data that have been removed from memory and added to the free list.

AnonymousPagesPagedIn

The total number pages that contain heap, stack, or other changeable data that have been allocated in memory and possibly copied from disk.

AnonymousPagesPagedOut

The total number pages that contain heap, stack, or other changeable data that have been removed from memory and copied to disk.

BroadcastInputPackets

The total number of broadcast packets read from the network interface.

BroadcastOutputPackets

The total number of broadcast packets written to the network interface.

Collisions

The total number of packet collisions that have happened on the network interface.

ConnectionFailures

The total number of times TCP connections failed to be established. At the lowest level this means that they have made a direct transition to the CLOSED state from the SYN-SENT state or the SYN-RCVD state, or a direct transition to the LISTEN state from the SYN-RCVD state.

ConnectionsActive

The total number of times a client socket has explicitly connected to a listening server socket.At the lowest level this means that a socket has made a direct transition to the SYN-SENT state from the CLOSED state.

ConnectionsEstablished

Current number of established TCP socket connections on the machine. At the lowest leve this means that a socket's current state is either ESTABLISHED or CLOSE-WAIT.

ConnectionsPassive

The total number of times a listening server socket has accepted a connection from a client. At the lowest level this means that a socket has made a direct transition to the SYN-RCVD state from the LISTEN state.

ConnectionsReset

The total number of times established TCP connections have been closed.At the lowest level this means a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.

ControlSegmentsSent

The total number of control (syn, fin, rst) TCP segments sent.

CopyOnWriteFaults

The total number of times a private copy of a shared page needed to be made due to a write to the shared page.

CPUs

The number of online cpus on the local machine.

CrossCalls

The total number of inter-cpu cross-calls.

DelayedAcksSent

The total number of delayed acknowledgment TCP segments sent.

DeviceNotReady

The total number of errors that have occured due to the device not being ready.

DuplicateAcks

The total number of duplicate acknowledgment TCP segments received.

ExecPagesFreed

The total number readonly pages that contain code or data that have been removed from memory and returned to the free list.

ExecPagesPagedIn

The total number readonly pages that contain code or data that have been copied from disk to memory.

ExecPagesPagedOut

The total number readonly pages that contain code or data that have been removed from memory and will need to be paged in when used again.

FailedMutexEnters

The total number of times a thread entering a mutex had to wait for the mutex to be unlocked.

FailedReaderLocks

The total number of times readers failed to obtain a readers/writer locks on their first try. When this happens the reader has to wait for the current writer to release the lock.

FailedWriterLocks

The total number of times writers failed to obtain a readers/writer locks on their first try. When this happens the writer has to wait for all the current readers or the single writer to release the lock.

FileSystemPagesFreed

The total number of pages, that contained the contents of a file due to the file being read from a file system, that have been removed from memory and put on the free list.

FileSystemPagesPagedIn

The total number of pages that contain the contents of a file due to the file being read from a file system.

FileSystemPagesPagedOut

The total number of pages, that contained the contents of a file due to the file being read from a file system, that have been removed from memory and copied to disk.

FreeMemory

The number of megabytes of memory in the free list.

HalfOpenDrops

The total number of half open connections dropped. Non-zero values usually indicate a SYN flood attack.

HalfOpenQueueFull

The total number of connection refused due to the help open listen queue (q0) being full.

HardErrors

The total number of hard errors that have occured on the I/O device.

HATMinorFaults

The total number of hat faults. You only get these on systems with software memory management units.

IdleThread

The total number of times the idle thread has been scheduled to run.

IllegalRequest

The total number of illegal request errors that have occured on the I/O device.

InputBytes

The total number of bytes read from the network interface.

InputErrors

The total number of errors associated with read operations on the network interface.

InputPackets

The total number of packets read from the network interface.

InputPacketsDiscarded

The total number of input packets discarded.

Interrupts

The total number of interrupts that have occurred on the computer.

InterruptsAsThreads

The total number of interrupts as threads. This does not include clock interrupts.

InterruptsBlocked

The total number of interrupts blocked/preempted/released.

KeepAliveDrops

The connections dropped due to the failure of a keep alive probe.

KeepAliveProbes

The total number of times a probe needed to be sent out due to a keep alive timer expiring.

KeepAliveTimeouts

The total number of keep alive timeouts.

ListenQueueFull

The total number of connections refused due to a listen queue being full.

LoadAverage1

The average number of threads ready to run over the last minute.

LoadAverage5

The average number of threads ready to run over the last five minutes.

LoadAverage15

The average number of threads ready to run over the last fifteen minutes.

LockedPages

The current number of physical memory pages on the machine that are locked.

LogicalBlockReads

The total number of logical block I/O read operations.

LogicalBlockWrites

The total number of logical block I/O write operations.

MajorPageFaults

The total number of times a page fault required disk io to get the page. A page fault occurs when a process refers to a virtual memory page that has been paged out.

MediaErrors

The total number of media errors that have occured on the I/O device.

MessageCount

The total number of msgrcv() and msgsnd() system calls.

MulticastInputPackets

The total number of multicast packets read from the network interface.

MulticastOutputPackets

The total number of multicast packets written to the network interface.

NoDevice

The total number of errors that have occured due to a device not being available.

OutputBytes

The total number of bytes written to the network interface.

OutputErrors

The total number of errors associated with write operations on the network interface.

OutputPackets

The total number of packets written to the network interface.

OutputPacketsDiscarded

The total number of output packets discarded.

PageDemonCycles

The total number of revolutions of the page daemon's scan

PageFreeReclaims

The total number of page reclaims done from the free list. These reclaims are much cheaper than those that need to go to disk.

PageIns

The total number of times pages have been brought into memory from disk by the operating system's memory manager.

PageOuts

The total number of times pages have been flushed from memory to disk by the operating system's memory manager.

PageReclaims

The total number of page reclaims; both from the free list and from disk. Page reclaims are caused by a reference to a page that has been stolen from a process by the page daemon.

PagerRuns

The total number of times the pager daemon has been scheduled to run.

PagesFreedAutomatically

The total number of pages that have been added to the free list be either the pager daemon or automatically.

PagesPagedIn

The total number of pages that have been brought into memory from disk by the operating system's memory manager.

PagesPagedOut

The total number of pages that have been flushed from memory to disk by the operating system's memory manager.

PagesScanned

The total number pages examined by the pageout daemon. When the amount of free memory gets below a certain size, the daemon start to look for inactive memory pages to steal from processes. A high scan rate is a good indication of needing more memory.

PagesSwappedIn

The total number of swapped out pages that have been brought back into memory from disk.

PagesSwappedOut

The total number of pages that have been moved from memory to disk due to a swap out operation.

The amount of virtual memory which is active.

PathnameLookups

The total number of pathname lookups.

PercentCpuActive

The percentage of the total available time that has been used to execute user or system code.

PercentCpuIdle

The percentage of the total available time that has been spent sleeping.

PercentCpuIOWait

The percentage of the total available time that has been spent waiting for disk I/O to complete.

PercentCpuSwapWait

The percentage of the total available time that has been spent waiting for paging and swapping to complete.

PercentCpuSystem

The percentage of the total available time that has been used to execute system (i.e. kernel) code.

PercentCpuUser

The percentage of the total available time that has been used to execute user code.

PercentCpuWaiting

The percentage of the total available time that has been spent waiting for I/O, paging, or swapping.

PhysicalAsyncBlockWrites

The total number of physical asynchronous block I/O write operations.

PhysicalBlockReads

The total number of physical block I/O read operations.

PhysicalBlockWrites

The total number of physical block I/O write operations, both synchronous and asynchronous.

PhysicalMemory

The amount of physical memory on the machine in megabytes.

PredictiveFailureAnalysis

A metric that can be used to predict the next failure on the I/O device.

Processes

The number of processes in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. Each process represents the running of a program.

ProcsInIOWait

The number of processes waiting for block I/O at this instant in time.

ProtectionFaults

The total number of times memory has been accessed in a way that was not allowed. This results in a segementation violation and in most cases a core dump.

RawIOReads

The total number of raw I/O read operations.

RawIOWrites

The total number of raw I/O read operations.

ReadKBytes

The total number of kilobytes read from the operating system I/O device.

ReadOperations

The total number of read operations done on the operating system I/O device.

ReceivedDuplicateBytes

The total number of TCP data bytes received in duplicate segments. Incoming data may be duplicated when an acknowledgment is lost or delayed and the other end retransmits a segment that actually arrived correctly the first time. This situation can be a sign that the remote systems are retransmitting too quickly and needs tuning or a patch.

ReceivedInorderBytes

The total number of TCP data bytes received in the correct order.

ReceivedOutOfOrderBytes

The total number of TCP data bytes received in the wrong order. If this value is high compared to (ReceivedInorderBytes > ReceivedInorderBytes) then it could be a sign of routing problems.

ReceivedPartialDuplicateBytes

The total number of TCP data bytes received in partially duplicated segments. Incoming data may be duplicated when an acknowledgment is lost or delayed and the other end retransmits a segment that actually arrived correctly the first time. This situation can be a sign that the remote systems are retransmitting too quickly and needs tuning or a patch.

Recoverable

The total number of errors that have occured on the I/O device that are recoverable.

ReservedSwap

The number of megabytes of swap space reserved for allocation by a particular process.

RetransmittedTcpBytes

The total number of bytes resent in TCP data segments. If this value is more than 30% of (SentTcpBytes > SentTcpBytes) you may have some bad network hardware, a congested route that is dropping packets, or and operating system that needs a patch.

RetransmitTimeoutDrops

The total number of connections dropped due to a retransmit timeout.

RetransmitTimeouts

The total number of TCP retransmit timeouts.

SchedulerRunCount

The total number of times the system scheduler has put a thread in its run queue.

SchedulerSwapCount

The total number of times the system scheduler has swapped out an idle process.

SchedulerWaitCount

The total number of times the system scheduler has removed a thread from the run queue because it was waiting for a resource.

Segments

The total number of TCP segments that have been sent or received using the TCP protocol.

SegmentsReceived

The total number of TCP segments that have been received, including those received in error. This count includes segments received on currently established connections.

SegmentsRetransmitted

The total number of retransmitted TCP segments, that is, segments transmitted containing one or more previously transmitted bytes. If this value is more than 30% of (SegmentsSent > SegmentsSent) you may have some bad network hardware, a congested route that is dropping packets, or an operating system that needs a patch.

SegmentsSent

The total number of TCP segments that are sent, including those on current connections, but excluding those containing only retransmitted bytes.

SemaphoreOps

The total number of semaphore operations.

SentTcpBytes

The total number of bytes sent in TCP data segments.

SoftErrors

The total number of soft errors that have occured on the I/O device.

SoftwareLockFaults

The total number of faults caused by software locks held on memory pages.

SwapIns

The total number of times a process that had been swapped out to disk is brought back into memory.

SwapOuts

The total number of times an idle process has been swapped out from memory to disk. If this ever happens its a sign that free memory stayed low long enough to trigger swapping.

SystemCalls

The total number of system calls.

SystemExecs

The total number of exec() system calls.

SystemForks

The total number of fork() system calls.

SystemMinorFaults

The total number of minor page faults in kernel code. Minor page faults do not require disk access.

SystemReads

The total number of read() and readv() system calls.

SystemVForks

The total number of vfork() system calls.

SystemWrites

The total number of write() and writev() system calls.

ThreadCreates

The total number of times a thread has been created.

ThreadMigrates

The total number of times a thread (lwp) has migrated from one CPU to another.

TotalCSW

The total number of context switches from one thread to another on the computer. Thread switches can occur either inside of a single process or across processes. A thread switch may be caused either by one thread asking another for information, or by a thread being preempted by another, higher priority thread becoming ready to run.

TotalIVolCSW

The total number of times a thread was forced to give up the cpu even though it was still ready to run.

TotalPages

The current number of physical memory pages on the machine.

TransportErrors

The total number of transport errors that have occured on the I/O device.

Traps

The total number of traps that have occurred on the computer.

UnallocatedSwap

The number of megabytes of swap space that have not been allocated.

UnreservedSwap

The number of megabytes of swap space that are free. If this value goes to zero new processes can no longer be created.

UnswappableMemory

The amount of allocated memory, in megabytes, that can not be swapped out.

UserMinorFaults

The total number of minor page faults in non-kernel code. Minor page faults do not require disk access.

WaitQueueLength

The average number of operations in the driver's wait queue for a system I/O device. Operations go into the wait queue when they are performed on the I/O device and leave it when both the system bus and device are ready to activate the operation.

WaitQueueResponseTime

The average milliseconds and operation stays in the queue. Its calculated by dividing the queue length by total number of operations taken out of the queue. Operations go into the wait queue when they are performed on the I/O device and leave it when both the system bus and device are ready to activate the operation.

WaitQueueServiceTime

The average wait queue service time in milliseconds. Its calculated by dividing the time the queue was busy by total number of operations taken out of the queue. Operations go into the wait queue when they are performed on the I/O device and leave it when both the system bus and device are ready to activate the operation.

WaitQueueUtilization

A percentage that represents how much of the queue is being used. Its calculated by dividing the time the queue was busy by the total time available. Operations go into the wait queue when they are performed on the I/O device and leave it when both the system bus and device are ready to activate the operation.

WriteKBytes

The total number of kilobytes written to the operating system I/O device.

WriteOperations

The total number of write operations done on the operating system I/O device.

ZeroFilledPages

The total number of pages have been block-cleared to contain all zeros.

4.4 System Statistics on AIX

This section lists statistics that are generated when statmonitor on AIX is started using options that specify to include system statistics.

Note

The set of system statitics that can be collected varies by server product and version. Not all statistics may be generated by statmonitor in any particlar version of GemStone.

BadPages

The number of bad pages.

BytesReceived

The number of bytes received on the network interface, including framing characters.

BytesSent

The number of bytes sent on the network interface, including framing characters.

Collisions

The total number of packet collisions that have happened on the network interface.

ConnectionsEstablished

Current number of established TCP socket connections on the machine. At the lowest leve this means that a socket's current state is either ESTABLISHED or CLOSE-WAIT.

CPUs

The number of online cpus on the local machine.

DiskCount

The total number of disks attached to the system.

DiskReads

The total number of read operations on the disk. If this counter is always 0 then 'diskperf -y' needs to be run.

DiskSizeFree

The total free space of all disks in megabytes.

DiskSizeTotal

The total size of all disks in megabytes.

DiskWrites

The total number of write operations on the disk. If this counter is always 0 then 'diskperf -y' needs to be run.

DiskXferRate

The total number of reads via adapter.

DiskXfers

The total number of transfers to/from disk.

FileBufferCacheSize

The amount of real memory used to buffer files.

FreeMemory

The number of megabytes of memory in the free list.

InputErrors

The total number of errors associated with read operations on the network interface.

InputKBytes

Amount of data received by the network adaptor in kilobytes.

InputPackets

The total number of packets read from the network interface.

InterruptDev

The number of device interrupts.

InterruptSoft

The number of software interrupts.

KBytesRead

The number of kilobytes read by an operating system IO device.

KBytesWritten

The number of kilobytes written by an operating system IO device.

LoadAverage1

The average number of threads ready to run over the last minute.

LoadAverage5

The average number of threads ready to run over the last five minutes.

LoadAverage15

The average number of threads ready to run over the last fifteen minutes.

LogicalBlockReads

The total number of logical block I/O read operations.

LogicalBlockWrites

The total number of logical block I/O write operations.

MessageCount

The total number of msgrcv() and msgsnd() system calls.

NetworkInterfaceCount

The number of network interfaces connected to the system.

OutputErrors

The total number of errors associated with write operations on the network interface.

OutputKBytes

Amount of data sent by the network adaptor in kilobytes.

OutputPackets

The total number of packets written to the network interface.

PacketsOutboundErrors

The number of outbound packets that could not be transmitted because of errors.

PacketsReceived

The number of packets received on the network interface.

PacketsReceivedErrors

The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

PacketsSent

The number of packets sent on the network interface.

PageCycles

The number of page replacement cycles that have occurred.

PageFaults

The total number of Page Faults by the threads executing in this process. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This will not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.

PageIns

The total number of times pages have been brought into memory from disk by the operating system's memory manager.

PageOuts

The total number of times pages have been flushed from memory to disk by the operating system's memory manager.

PageScans

The number of page scans that have occurred.

PageSpaceFree

The amount of paging (swap) space which is free.

PageSpaceReserved

The amount of paging (swap) space which is reserved.

PageSpaceTotal

The total size of the paging (swap) space.

PagesPagedIn

The total number of pages that have been brought into memory from disk by the operating system's memory manager.

PagesPagedOut

The total number of pages that have been flushed from memory to disk by the operating system's memory manager.

PageSteals

The number of page steals that have occurred.

PagesVirtualActive

The amount of virtual memory which is active.

PercentCpuActive

The percentage of the total available time that has been used to execute user or system code.

PercentCpuIdle

The percentage of the total available time that has been spent sleeping.

PercentCpuIOWait

The percentage of the total available time that has been spent waiting for disk I/O to complete.

PercentCpuSystem

The percentage of the total available time that has been used to execute system (i.e. kernel) code.

PercentCpuUser

The percentage of the total available time that has been used to execute user code.

PhysicalMemory

The amount of physical memory on the machine in megabytes.

Processes

The number of processes in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. Each process represents the running of a program.

ProcessorSpeed

The clock speed of the processors in megahertz.

QueueLength

The number of IO operations currently in progress.

RawIOReads

The total number of raw I/O read operations.

RawIOWrites

The total number of raw I/O read operations.

ReadKBytes

The total number of kilobytes read from the operating system I/O device.

RealMemoryInUse

The amount of real memory currently in use.

RealMemoryNonSys

The amount of memory used by non-system (user) segments.

RealMemoryPinned

The amount of real memory that cannot be paged out.

RealMemoryProc

The amount of memory used by process segments.

RealMemorySys

The amount of memory used by system segments.

SchedulerRunCount

The total number of times the system scheduler has put a thread in its run queue.

SchedulerSwapCount

The total number of times the system scheduler has swapped out an idle process.

SemaphoreOps

The total number of semaphore operations.

SharedMemoryKB

The amount of memory enabled for sharing between multiple processes via shmat(2) and mmap(2) with the MAP_SHARED attribute set.

SystemCalls

The total number of system calls.

SystemExecs

The total number of exec() system calls.

SystemForks

The total number of fork() system calls.

SystemReads

The total number of read() and readv() system calls.

SystemWrites

The total number of write() and writev() system calls.

TotalCSW

The total number of context switches from one thread to another on the computer. Thread switches can occur either inside of a single process or across processes. A thread switch may be caused either by one thread asking another for information, or by a thread being preempted by another, higher priority thread becoming ready to run.

TransferCount

A count of the read and write operations done by an operating system I/O device.

Traps

The total number of traps that have occurred on the computer.

WriteKBytes

The total number of kilobytes written to the operating system I/O device.

4.5 GBS Statistics

This section lists statistics that are generated when GBS is configured to write statistics.

Note

The set of GBS statistics varies by version. Not all statistics may be generated in any particular version of GBS.

bytesSentByStoreTraversal (GbsSession)

The number of bytes sent cumulatively by store traversal calls

bytesTraversed (GbsSession)

The cumulative number of bytes returned by traversal calls

cacheStatSampleTime (GbsSessionManager)

The amount of time spent sampling cache statistics

changedObjNotifications (GbsSession)

The number of changed object notifications received

clientMapSize (GbsSessionManager)

The number of entries in the client map (stObjectCache)

freeOopsFetched (GbsSession)

The number of free oops fetched

gbsBytesCached (Cache)

The number of bytes occupied by instances of the class in the stObjectCache.

gciCallProtectInvocations (GbsSessionManager)

The number of accesses to the GciCallProtect semaphore (a GbxCInterface shared variable)

gciCallsToGem (GbsSession)

The number of GCI calls made that communicate with the gem

gciCallsToGemTime (GbsSession)

The amount of time spent in GCI calls that communicate with the gem

gciErrors (GbsSession)

The number of errors reported by GciErr()

lostOtRoots (GbsSession)

The number of lostOTRoot signals received

mainStatSampleTime (GbsSessionManager)

The amount of time spent sampling manager and session main statistics

nbEndResultNoProgress (GbsSession)

The number of times GciNbEnd() was called when the result wasn't ready and no progress was made

nbEndResultReady (GbsSession)

The number of times GciNbEnd() was called and a result was ready

nbEndResultProgressed (GbsSession)

The number of times GciNbEnd() was called and progress was made

numSessions (GbsSessionManager)

The number of logged in GbsSessions

numInstances (Cache)

The number of instances of the class in the stObjectCache.

objectsStoredByTraversal (GbsSession)

The total number of objects stored by store traversal calls

objectsTraversed (GbsSession)

The total number of objects (with or without a value buffer) received by traversal call

serverMapSize (GbsSession)

The number of entries in this session's server map (gsObjectCache)

sessionListProtectInvocations (GbsSessionManager)

The number of accesses to the sessionListProtect semaphore

sessionProtectInvocations (GbsSession)

The number of times the sessionProtect semaphore has been invoked

sessionSignals (GbsSession)

The number of gem-to-gem signals received

sigAborts (GbsSession)

The number of signaled aborts received

storeTraversals (GbsSession)

The number of store traversal calls made

traversalUnpackingTime (GbsSession)

The total number of milliseconds spent unpacking traversal buffers

traverseCalls (GbsSession)

The number of traversal calls (all types, including more traversal)

traverseCallTime (GbsSession)

The amount of time spent in traversal calls