

Examining statistical information

This section includes the following topics:

- [About the Statistics workspace](#)
- [How the Statistics workspace is structured](#)
- [About the entities you can examine in the Statistics workspace](#)
- [How to identify performance problems](#)

About the Statistics workspace

The Statistics workspace displays statistical information on all .NET instances. The workspace can be used to monitor your system's current state as well as historical statistical information.

The Statistics workspace provides hundreds of raw performance counters (all Performance Monitor counters that are relevant for .NET). For each counter, Precise for Microsoft .NET can display the summed, average, minimum, or maximum, depending on the context. You can view the current status either in intervals of one time slice or historical information over a period of time at a higher summary level.

The Statistics workspace enables you to provide answers to the following types of questions: What was the number of successful and failed requests in the last 6 hours? or Does .NET spend a lot of time in the garbage collector?

Use the Statistics workspace to periodically examine the health of your system. Alternatively, you can use the Statistics workspace to fully analyze a performance problem. For example, you can use the overtime graph for a specific counter to get a wide view and examine the counter behavior in the suspicious instance and time slice.

How the Statistics workspace is structured

The Statistics workspace displays information on a selected entity and its associated entities. For example, it is possible to associate with all counters that are related to a specific instance, by selecting the Counters entity from the Association controls.

When you open the Statistics workspace from another workspace, you will be focused on an individual instance. The workspace is always launched at the instance level, even if the currently selected entity is not an instance.

The selected entity is always reflected in the workspace heading, which serves as a point of orientation. The highest-level entity you can view information for in the Statistics workspace is the Tier. You can view information on an instance by selecting it from the Instance list.

About the Main area in the Statistics workspace

The Main area shows comprehensive information on the selected entity. You can choose from several views to examine the entity from different angles.

About the Association area in the Statistics workspace

The Association area provides corresponding information on the entities associated with the selected entity (displayed in the Main area). You can view information on one type of entity at a time, such as instances only, by selecting an item from the Association controls.

From the Association area, you can also drill down to another entity by clicking a row in the table. A drill-down affects the whole workspace. When you drill down to another entity, the workspace heading displays the new selection; the Main area displays information on the newly selected entity, and the Association area displays the entities associated with the selected entity.

If you want detailed information on a specific server counter, in the Association area, click the row of the counter that you want to view detailed information for. The workspace heading indicates the newly selected entity, and the Main area displays an overtime graph for the counter you drilled down to. There is no Association area data for a counter.

See "How most tabs are structured" on page 9.

About the entities you can examine in the Statistics workspace

The Statistics workspace displays information on different entities. This section provides an overview of all entities, their meaning, and their views.

The following entities are available:

- Instances
- Counters



For an explanation of the counters displayed in the Statistics workspace, move your pointer over the counter name in the table to view a ToolTip, or see the Microsoft performance monitor tool.

About the Counter entity in the Statistics workspace

The Counter entity displays information on the specific counter over time. The following views are available in the Main area:

- Overview

You cannot associate a counter with other entities. The Association area is blank.

Getting an overview of a counter in the Statistics workspace

The overview displays a bar graph that illustrates the counter behavior over time. A counter can collect different types of information, such as the summed number of bytes received or send, or the maximum or average number of pooled connections. Accordingly, if the selected counter collects average information, the graph displays the average overtime values; if it collects maximum information, it displays the maximum overtime values, and so on.

Viewing information on counters in the Association area

The table below describes the table that is displayed in the Association area when you view information on counters.

Table 1 Counter information in the Association area

Column	Description
Category	Displays the perform category of the counter.
Counter Name	Displays the unique counter name.
Counter Value	Displays the sampled counter value.

How to identify performance problems

To determine whether your .NET application is performing optimally it is necessary to analyze performance measurements over time. The Statistics workspace provides information about many Microsoft performance counters related to .NET and ASP.NET instances, as well as some server related counters.

They are grouped into several predefined views and categories that enable you to locate performance problems such as CPU, GC, and memory usage.

Below is a list of common performance problem areas. Each section will refer to important counters to watch using the Statistics workspace when analyzing your instance performance.

How to examine ASP.NET-related statistics

You can examine your ASP.NET instance counters over time to confirm the ASP.NET normal operation over time. For example, you can find out if your ASP.NET instance experiences many failed requests. If so, it indicates a problem with your application or server that needs further investigation.

To view ASP.NET related counters, do the following

1. In the Time Frame list, choose the period of time you want to analyze.
2. In the Instance list, choose the ASP.NET instance you want to analyze.

Counters can be examined by drilling down on a specific counter in the counters table area.

Table 2 Main counters

Counter	Description
Request Counters	<p>Requests Total. Shows the number of requests served by your ASP.NET instance (IIS). These includes all pages processed by the ASP.NET ISAPI filter - not just .aspx, but also .asmx, .axd, .ashx and others.</p> <p>Requests Failed. Shows the number of requests that failed. A high number might indicate an exception condition is thrown repeatedly in your application.</p> <p>Other Requests counters. Requests not authorized, Requests authorized, Requests succeeded, Requests Timed Out., Requests Current, Requests Rejected, Requests Queued, Requests Not Found.</p>
Session Counters	Session Abandoned, Sessions Timed out, Sessions Total.
Worker Process Counters	Worker Process restarts and Worker Process Running counters can be used to detect situations where due to a failure your ASP.NET application restarts frequently.
ASP.NET Error Counters	<p>ASP.NET errors counters can be used to detect errors thrown by your web application and not handled by your code. Explanation about these counters can be found in the SmarTune workspace Advice tabs regarding these counters:</p> <ul style="list-style-type: none"> • Errors Total • Errors Unhandled during Execution • # of Exceptions thrown

Cache Usage Counters	<p>These counters can be used to detect bad usage of cache by your application. There are several ways to use caching in your ASP.NET application. Explanation about these counters can be found in the SmarTune workspace Advice tabs regarding these counters:</p> <ul style="list-style-type: none"> • Output Cache Hits / Misses / Entries • Cache API Hits / Misses / Entries • Cache Total Hits / Misses / Entries
----------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

How to examine Garbage Collection related statistics

You can examine your .NET instance Garbage Collection related counters over time to confirm abnormal behaviors. To view .NET Garbage Collection related counters, do the following

1. In the Time Frame list, choose the period of time you want to analyze.
2. In the Instance list, choose the .NET instance you want to analyze.
3. On the Main area View controls, click **Garbage Collection**.

The % Time in GC (Avg) counter shows the time your application spends doing garbage collection. A high value indicates you are experiencing problems with your applications memory management scheme. This could occur for many reasons; the most common are:

- Allocating too many objects
- Low memory in the machine running the application

The Allocated Bytes/sec displays the rate of bytes per second allocated on the GC Heap. This counter is updated at the end of every GC. Other important GC counters, for .NET instance displayed in graphs are:

- Number of GC Collections
- CLR Generations

How to examine CPU & Memory related statistics

You can examine your .NET instance memory and CPU related counters over time to confirm abnormal behaviors. To view .NET memory and CPU related counters, do the following

1. In the Time Frame list, choose the period of time you want to analyze.
2. In the Instance list, choose the .NET instance you want to analyze.
3. On the Main area View controls, click **CPU & Memory**.

The CPU Consumption (Avg) counter shows average CPU time consumed by your .NET application. Use this graph to find peaks of high CPU usage. High CPU usage can be caused by many things, including bad programming, too much time in GC, etc.

The Used Memory (Avg) counter shows average memory consumed by your .NET application. Use this graph to detect patterns of increasing memory usage, possibly caused by a memory leak in your code.