

# Precise for Sybase usage concepts

This section includes the following topics:

- [About application performance tuning](#)
- [About viewing Current Activity](#)
- [About SQL statement tuning](#)
- [About instance statistics tuning](#)

## About application performance tuning

Precise for Sybase allows you to proactively monitor, analyze, and tune Sybase instances. By capturing, measuring, and correlating performance metrics from all critical system components, Precise for Sybase provides a complete view of your application's performance. It detects business performance problems and helps you find suitable and comprehensive solutions thereby ensuring that your business applications always perform at peak efficiency.

Detailed information on the resource consumption of an application is extensive and includes data on CPU Consumption, I/O Waits, Internal Database Waits, and so on. Still, with Precise for Sybase you can easily identify the top resource consumers at different levels and at different times.

For example, you may want to start at the Instance level and then drill down to different Login names and the SQL statements they executed. You can view the status of current sessions, or focus your analysis on a particular time period, and examine their consumption patterns during the last minute, or further back in the past, such as yesterday, last week, or last year. You can also examine changes in resource consumption patterns over time in time slices of 15 minutes, hours, four hours, days, weeks, or months.

## About viewing Current Activity

The Current tab displays near real-time information on all sessions that were connected to your Sybase instance during the last minute. Use the Current tab to solve bottlenecks, such as those caused by locked sessions or runaway processes. The Current tab displays instance activity information up to the last time slice of activity.

You can select a Sybase instance and examine its resource consumption during the last minute, such as CPU Usage, I/O Wait, and Lock Wait.

The application resource consumption is displayed on the instance level or as a summary of the session level of resource consumption.

If you identified a blocking situation that is currently in progress (for example, there is a session waiting for a lock), select the Locks view to view the current sessions that are blocking the other sessions. The information is displayed in a tree format, which helps you easily identify which are the locked sessions, who they are being locked by, and what is the lock level. You can drill down to the sessions involved in the lock to obtain more information about what is causing the lock. For example, you can drill down to a locking session to determine which SQL and which program are currently being executed.

## Analyzing instance performance

While the Current tab displays near real-time data, it is recommended to begin the actual tuning process is the Activity tab.

The Activity tab allows you to analyze performance behavior over time or when the problem is reported (since in many cases the performance problem may no longer be evident from the data displayed in the Current tab). Using the Activity tab, you can analyze what happened at a specific point in time, identify problematic time periods, and drill down to analyze a bottleneck that caused a performance problem. You can begin by selecting an Instance and viewing the In Sybase overtime graph to identify problematic time periods.

In addition, long-term analysis is also required for understanding performance patterns and predicting future resource consumption. You can use the Activity tab to detect resource consumption patterns and suspicious deviations in past behavior.

The Activity tab can also be referred to when planning your system's capacity, performing period-to-period comparisons, and determining how best to optimize storage. Use the Activity tab to examine resource consumption of your Sybase Instance, over time, as well as the resource consumption of your databases, programs, logins, and so on.

You can also identify dominant resource consumption by selecting the Overview view.

The Overview view provides a breakdown of the In Sybase states, such as, CPU Usage, I/O Waits, Lock Waits, and Remote Waits. Examining this data allows you to pinpoint the type of resource causing the bottleneck. If for example you identify that Sybase is suffering from an I/O wait problem, upgrading your server to a much more powerful machine will probably fail to provide you with the expected results.



A high CPU rate exceeding 90% may indicate that indexes are not being used.

## Identifying resource consumers

Precise for Sybase lets you drill down to additional levels to determine which entities are responsible for a particular performance problem. In the Association area you can observe different entities, such as databases, login names, users, programs, and client machines, to pinpoint the application component that consumes the most database resources.

For each entity (as, for example, the program entity), you can access the following information:

- Entities that are related to the entity (such as, the logins or machines that executed the program).
- Resources consumed by the entity. This enables you to determine if the entity is CPU or I/O bound, or waiting for a lock.

- Resource consumption patterns, which allows you to observe and identify trends in resource consumption. You can drill down even further to other entities associated with the selected entity in the Association area.

For example, once you have identified a problem in a specific program you can drill down to the batches or statements the program issued, to determine the source of the problem.

## Associating with offending SQL statements

Having identified a major resource-consuming entity, you can further drill down to the statements and batches executed by that entity. These may include long-running SQL statements or SQL statements that use few resources but are executed frequently.

For each statement or batch, you can access the following information in Precise for Sybase:

- Entities that executed the statement or batch.
- Resources consumed by the statement or batch, which enables you to determine if the statement or batch is CPU or I/O bound, or waiting for a lock, and so on.
- Resource consumption patterns that allow you to observe if the statement or batch execution is affected by excessive use of the same resource, by another statement or batch.

## Analyzing previous blocking situations

One of the main reasons that statements possess long execution times is that they are waiting for a resource that is locked by another session.

Using Precise for Sybase, you can easily identify the following:

- Statements waiting for a locked object
- Locked objects (that is, contended objects)
- Sessions causing locks

In Precise for Sybase, a statement waiting for the lock on a resource to be freed is in a Lock Wait. The session holding the lock is called a Blocker session.

## About SQL statement tuning

Most problems occur in SQL statements or batches. Precise for Sybase lets you to easily identify the most resource-consuming SQL statement or batch and view a detailed explanation of the execution plan of the SQL statement or batch. It can also help you resolve the problem by recommending which index would provide a better execution plan and reduce resource consumption.

## Identifying the most resource consuming statements or batches

Use the Activity tab to identify the most resource-consuming SQL statements or batches over a selected time period. By focusing in on a specific time period you can apply your efforts to tuning a problematic time frame, such as a batch window or end-of-the-month processing.

In the Association area of the Activity tab, you can associate to a specific database entity and then from there to its associated SQL statements.

When selecting an SQL statement, you can view a breakdown of its resource consumption over time, its text, and all locking contentions that it participated in (time spent waiting for a resource).

## Collapsing constants into bind variables

When an application uses constant values and literals instead of bind variables (parameter markers), many different SQL statements actually constitute the same SQL statement with different values.

When each of these similar SQL statements is treated as a different statement, it is hard to summarize their resource consumption to identify the best candidate for SQL tuning.

Precise for Sybase provides a unique algorithm that ignores constant values and groups all similar statements into collapsed statements. This lets you identify the most resource-consuming collapsed statement.

The performance data for the collapsed statement is a summary of all performance data for all statements in the group. Most of the time, all the statements that were grouped together into one collapsed statement have the same execution plan. However, different values in the WHERE clause may lead the optimizer to choose a different execution plan.

Precise for Sybase enables you to select a Collapsed statement and associate to all the different execution plans of the SQL statements that were grouped together.

## About viewing an execution plan

To view the execution plan of the statement and understand how the optimizer chooses to perform the statement, launch to the SQL tab. Each branch in the execution plan represents one operation. Each operation is displayed using the following format:

```
[<execution order>] <operationtype> (<options>) <accessed object>[<cost in percentage>]
```

When examining the execution plan, always start by exploring it from right to left and top to bottom (according to the execution order in a branch on the tree). The first operation to be performed has execution order 1, followed by 2, and so on.

## Understanding an execution plan

By observing the access plan of a selected statement or batch, you can analyze the access path chosen by the Sybase Optimizer. Click the Statistics tab to compare the costs of the execution plan operators and quickly locate the most resource-consuming operation in the execution plan.

Clicking the icon on the left-most column in the Statistics tab will highlight the corresponding row in the execution plan tree.

The information displayed in the Operations tab helps you understand the nature of the execution plan, its major operations, and the major access types to objects (index scan, index seek, table scan, and so on).

## Analyzing the statement execution plan over time

Precise for Sybase allows you to track Access Plan changes and compare them with the resource consumption of the statement to understand how the changes in the Access Plan affected its performance and execution plan.

## Examining object space usage over time

Another way to tune a database object is to examine its space usage. The Collect Space Utilization process collects table and index space information daily (by default).

Space usage information allows you to examine space allocated versus space used by a table or an index. This enables you to easily keep track of object growth over time and decide how to allocate memory and connect objects to database devices in the specified database.

## About instance statistics tuning

The Precise for Sybase Collector Instance Statistics process collects a wide range of performance counters from the operating system, some which belong to Sybase and some to the operating system (OS) itself. These performance counters are sampled every minute and displayed in the Current and Statistics tab. When loaded into the PMDB, the counters are aggregated and kept at a slice level. The counters are then aggregated to daily, weekly, and monthly levels.

## Observing instance statistics over time

The Statistics tab provides many performance counters grouped into several predefined overtime graphs that enable you to locate various aspects of performance problems, such as CPU, paging, I/O, and network. The statistics tab can be used to monitor your system's current state as well as previous states. It is also possible to monitor a specific counter associated with a particular performance group by selecting the required counter.