

## Knowledge Database Support Tool

<b>Title</b>	<b>Performance Measurement in Siebel 7 Version and up – An Overview</b>		
<b>Description/ Purpose</b>	Provide a Consultant, Administrator, or Business Unit Analyst with a basic understanding of Performance Measurement in Siebel 7 or 8 environment. This paper is designed as a functional overview for use by customers and consultants when approaching performance issues in a Siebel 7 or 8 environment. (Focus: Siebel 7)		
<b>Key Words</b>	<b>Application Server, Response Time</b>		
<b>Methodology Phase(s)</b>	<b>Design and Implementation</b>		
<b>Methodology Resource(s)</b>	Consulting, Administrators, Business Unit Analysts		
<b>Product Area(s)</b>	All		
<b>Product Versions</b>	All versions of 7 and 8	<b>RBDMS Version</b>	All
<b>Related Files</b>		<b>3<sup>rd</sup> Party Software</b>	<b>None</b>
<b>Visibility</b>	Consulting, partners, customers, prospects		

## Background

With the migration of Siebel Products from a client/server architecture to a web based architecture, system administrators are faced with a new and more dynamic performance and tuning metric than existed with prior releases of Siebel. Siebel 7 uses a thin client to connect via a web browser to a remote server containing the various applications and data stores. This new functionality of the ultra-thin client has given rise to a number of questions within the customer community. Questions have arisen as to how to monitor performance in this type of environment to insure high levels of response time to the customer while maintaining high customer satisfaction.

This particular paper seeks to extend the documentation found in the Siebel Bookshelf by discussing some suggested performance and tuning metrics along with a suggested approach to monitoring the system for performance degradation. This functionality is especially helpful and is most desired by those customers who are using the Siebel 7 product in a high volume, global marketplace wherein the business unit requires high response times for all segments of their business.

## Contact Info

Author of this whitepaper: Denis Sanders dsanders@tsgonestop.com

## Siebel Documentation

### *Bookshelf*

#### Siebel Server Administration Guide

## Basic Functionality

After the installation of Siebel 7 and the migration to production of the product, the role of the System Administrator shifts from installation to maintenance of the system. A high level of user acceptance can be quickly negated by a system that does not provide data in a timely manner back to the business unit. This degradation of response time hits particularly hard in a customer service environment such as a call center. Customers will grow impatient while the call center agent waits for the return of a browser screen. This failure of rapid response time generates a high level of dissatisfaction within the business community and within the customer community. As a result, the System Administrator is called upon to increase system response time to the business unit. We will now begin to explore some performance metrics to assist the Siebel Administrator in solving slow performance problems.

In the best of all possible worlds, performance measurements are constantly taken on systems to note any degradation in response time. Administrators can monitor the performance curve over time and, for example, note that the Siebel response time from Europe is slowest on Friday. This type of measurement gives the Administrator time to formulate the appropriate tuning metric and gradually make the changes to the system until the optimum performance metric is reached. However, in the real world performance degradation is not noted until it has occurred. This causes a much more complex issue for the Siebel Administrator as there is no baseline to determine what event or events may have led to the performance problem. At this point, the

System Administrator must resist the temptation to attack the problem without taking appropriate steps to define the issue.

The system administrator is now faced with creating sizing or performance models. In a normal situation, the computer system has a particular Siebel 7 configuration running, a network interfacing Siebel with a back office system, and an application workload running on the various components of the system. All of these require certain resources. The Siebel Administrator, when presented with a performance problem, measures the performance, analyzes the results of this measurement, and begins to make configuration changes. Another part of this equation beyond workload and configuration are those random fluctuations in system use that may occur only at certain times (end of year, quarter, etc.) that cause a spike in resource demand. This spike can be averaged over time into workload metrics and then used as a valid part of the calculations to be discussed in the next few pages.

## Understanding Workload Measurement

In actuality, the workload parameter in a performance equation is the most difficult to handle. In some cases, the administrator will have total control over the Siebel environment. For example, the Siebel environment could (but probably not) be running in an isolated environment with all of the network and CPU resources dedicated solely to the Siebel Application. However, in reality, the workload component of Siebel is more than likely only part of the load imposed on a computing environment. To this end, we must learn to look beyond Siebel and look at the total computing setup at a company.

Aside from being hard to measure, workload is the most difficult point to cite as the reason for system degradation. One must take extreme care to drill down into the workload component of the performance metric to insure that workload is not merely a symptom of a much more systemic problem. Care must be taken to examine all components of the application environment (network, database, etc) prior to settling on the workload as the sole tuning point.

Dealing with workload, the fundamental types of workload measurement are throughput, response time, queue length, and resource utilization. One can say with certainty that almost all of the measurements to be made will fit into one of these categories. A Best Practice at this point is to always insure that you are collecting at least one measurement for each of these points when reviewing the performance of a Siebel process. Care must be taken to not focus on one measurement at the expense of another measurement, which could contain the true measurement of the failure point.

It is important to note that it is the steady state average throughput, response time, queue length, and utilization that are being measured. One must always use the average over time in any algorithms related to performance. Using a snapshot number taken at random most probably give a completely skewed results and mislead the tuning effort.

## Fundamental Measurement Types

There are several fundamental types of measurement that we touched on above. In this section we will define the types for use later in calculations.

- Throughput is the amount of work that is completed in a given amount of time. One example is found in the transactions per second measurement quoted for databases. It is important to never confuse throughput with bandwidth. Bandwidth is the best possible speed ignoring any overhead or latency. Maximum throughput is the actual amount of work that can be done at 100 percent utilization. An example of this difference is

illustrated in networks. The network vendor provides a bandwidth number, which must be reduced by the protocol in use, any overhead in hardware, and various packet containers used by the transmission.

- Response time is the measure of time that the user has to wait for some work to complete. For example, response time is the time elapsed from when a customer service agent enters a customer name into the system and the customer data is returned back to the agent from the database. Response time is best stated as the amount of time elapsed for a CPU to complete a process and return results to the calling program.
- Think time is a measurement of how long the user must wait before starting another operation. Think time can be illustrated in the amount of time that elapses between when a document request is made to Proposal Generator and the actual draft of the document is displayed on the agent's workstation. High think time reduces throughput and system utilization.
- Service time is the time taken to process the request once the request has reached the front of the queue. At low utilization levels, the queue is empty, so the response time is the same as service time. Service time is calculated:
  - $\text{System Utilization/System Throughput} = \text{Service Time}$
- Queue length is the number of requests that are waiting for service. A long queue increases the response time but does not affect the service time. It is important to note that a long queue can also imitate a hung system resulting in the needless shutdown of Server components beyond those used by the offending process. Queue length is calculated:
  - $\text{Throughput} \times \text{Response Time} = \text{Queue Length}$
- Utilization is a measure of how much of the computer's resources were used to perform the work. It is the system busy time as opposed to the system total time.

These definitions are important in the Siebel 7 environment, as the product has many more touchpoints that must be monitored for performance. These terms will enable staff to understand the measurement metrics necessary to pin down problematic response areas and to encourage the use of monitoring tools in the system.

## Measuring Workload on a Live System

Now that we have a basic understanding of terms used, we can discuss measuring of workload on a live Siebel 7 system

When monitoring a live system, care must first be taken that whatever monitoring tools used will consume some type of overhead. CPU resident tools will cause a slight skew to the CPU performance numbers simply because you have added one more process to the system. However, the tool selected will provide overhead numbers that can be factored into any manual calculation of work resources used.

To begin, we must find a way to quantify the throughput on a live Siebel 7 system. Response times are very hard to obtain on a live system but can be inferred by taking random timings at various workstations and from the overall system average utilization. In order to find utilization, it is best to review the appropriate Siebel and system logs for CPU or I/O usage. Typical execution times for a task are also important factors to be included into the equation at this time. Essentially, we are quantifying the amount of work done over a particular period of time as throughput.

Once throughput has been captured, we need to evaluate the customer-mandated throughput to determine the ratio of actual throughput to expected throughput. With this in mind, we must now build a model of the system as it has changed over time including application additions, user additions, database changes, configuration changes, and network configuration. Once these variables have been captured, we can then build a model of the system in a test environment and begin testing various solutions to the performance issue.

It is important to have a test system available upon which to test our solutions to the response time issue. Many small problems have become disasters when a change is made to a live environment without a full understanding of the ramifications of the change upon the entire system.

We will now discuss examples of how this type of measurement can function in a Siebel 7 environment.

## Suggested Performance Measurement Tasks

- Define the business purpose of the system. How is the Siebel 7 Application being used? High volume transaction processing? Sales? Order processing?
- Who is using the system and where are the users located? Remote? Local?
- What type of connection is being used? Frame? VPN? Copper? Fibre? Insure that the customer has run a full set of diagnostics on the network. Make sure you capture the bandwidth and overhead numbers to correctly calculate the work completed over time.
- Define who is stating there is a performance problem. Is the performance problem restricted to one part of the network or is the issue system wide? Narrow down the location as closely as possible.
- What is the system configuration? The configuration of the basic Siebel system is important as changes to configuration can have a widespread impact. Were the systems provided with at least the minimum RAM, CPU speed, and disk size as specified by Siebel?
- What is the Siebel configuration? Have any reviews of the configuration been completed by Expert Services? The placement of servers and how each dedicated client has been configured must be compared with best practices as specified by Siebel Systems.
- What application software is in use? This is a key question in that other memory or CPU intensive software can cause a great impact on system response time in general. Of particular note here, it is to insure that either Expert Services or a database vendor consultant check out the database to insure that the database is configured for optimum performance.
- What are the busy processes on the system doing? Basic tools in the quest for answers to performance issues are the current processes that are shown in the System Monitor. One can select the various items to chart (Disk cache, memory, etc.) and chart the current impact to the system. Once one has collected the charted data, one can further analyze the problems at hand.
- In line with the above point, it is important to note the CPU and disk utilization levels. How busy is the CPU and what is the proportion of user and CPU system time? In the case of processor intensive tasks, one should take care to note how much of the CPU is being used by the process at the time of generation. In line with this point, one should note how busy the disk is to insure that the forms being generated are not becoming I/O bound and thus realizing a slowdown in response time.
- Once one has checked out the disk system, one must then study paging on the system. Again we can use the system monitor to provide a gross cut at the data. We can then

evaluate the data and determine if we are receiving page faults or paging has become excessive. Heavy paging can be indicative of a memory shortage.

- One should also determine what has changed recently on the system. At times nothing will have changed in the Siebel System but performance will begin to degrade rapidly. Here is where one must determine if, for example, a high traffic node has been added to the network that is consuming resources and blocking the smooth functionality of the Siebel System. From a client machine perspective, what was added on the client machine to cause a slowdown? Memory intensive processes such as CAD/CAM or games can cause an unforeseen slowdown in the process.

An example of the above can be summarized in the findings of the recent issue with Siebel 7 running in a global company.

## Example of Performance Measurement

In this example, a customer was experiencing performance degradation in two areas: overall connectivity from a remote site and slow document generation time.

At first glance, it would appear that these issues were probably caused by a slow network connection. However, it was decided among the account team that the issue would be approached following the path suggested above.

In approaching the network problem, the first issue was to isolate the point of failure. By asking various users, it was determined that the only area having performance degradation and random disconnects was the Northern Europe Division. At this point, the team involved the customer's networking team and tasked them with running diagnostics on the frame relay in question. By running level three diagnostics on the frame relay, it was found that certain configuration parameters did not match those of the other frame relays working in areas that did not have performance issues. The team then determined that the issue was not with the Siebel product but rather was within the configuration of the network component.

The point of the above example is that the Siebel implementation team adopted a logical enterprise wide view of the issue. This led to the problem being solved quickly and efficiently by following a logical debugging model.

Once the network connection problem was solved, the issue with quotes was attacked. Following the metric suggested in this paper, the team began the performance enhancement by asking questions of the users. During this question period, it was found that performance was satisfactory in all phases of the implementation with the exception of the quotes module. During the generation of the document, the system would, at times, emulate a locked system. The System Administrator would then cycle the Siebel Server requiring all users to log out and back in to the system. This cycle resulted in a high level of customer dissatisfaction and a slow down of processing of work.

The network was again checked to validate any issues. No issues within the network were found. The hardware configuration in the production environment was compared to the model suggested in the Expert Services Sizing Review and no discrepancies were found. It was found through examination of the throughput that sufficient network capability was in place to process the document. Further calculations noted that there would be a minimal performance impact upon the system even with a full complement of current users connected to the system.

Continuing with the debugging effort, the team examined the components involved in the production of the quote. It was noted that Siebel 7 used Microsoft Word on the Server to process the document templates needed to prepare the written document for the customer. The templates had been generated by the team to accommodate the specific needs of the customer. As the team moved through the problem, it was decided to monitor the CPU and memory loads to note any issues. No anomalies were noted that would indicate high CPU or memory usage. All of the calculations for performance showed no issues.

The next step was to measure the response time between the Siebel Server component and Microsoft Word. At this point it was found that the connection between Microsoft Word and the Siebel Server did not release during the generation of certain documents. Trace files were then gathered and an examination of CPU time and response time revealed that Microsoft Word had entered an event wait state. That is, Word was waiting to display an error message on the UI. However, in the thin client model, the UI is not available to Word thereby causing Word to enter a wait state.

In order to test out the above theory, the offending template was installed on a dedicated client and tested. The results proved that the template had been corrupted and Word was attempting to pass an error message back to the calling UI. The template was regenerated and the issue was solved.

In summary, this example was presented to show the application of the performance measurement methodology and metrics to an actual situation.

## **Conclusion**

This paper has presented a brief, high-level discussion of performance metrics that can be used in measuring performance within a Siebel installation. An example of issue resolution was presented with a methodology that was followed to trace and resolve a performance related problem. Performance and tuning is a complex issue and Expert Services as well as the Siebel TAM should be engaged to proactively assist in resolution of these issues. Additional reading can be found on the Siebel Bookshelf.