



# Siebel Tuning and Monitoring - An Organizational Approach

by

Eyal Markovich (eyal@totalapm.com)  
Total APM Corporation

## **Abstract**

This document discusses an organizational approach for monitoring and tuning Siebel applications. It presents Total APM's best practices for monitoring and tuning Siebel eBusiness applications using the Precise i<sup>3</sup> monitoring solution.

November 2008

## Acknowledgements

I would like to acknowledge the contribution of Girma Watumo a Senior Principal Consultant at Oracle Corporation. Girma and I worked on several projects and I truly value his friendship. Besides being a friend, Girma is a highly skilled Siebel consultant with in-depth technical knowledge of the Siebel Application. He provided me with important insights on some key Siebel Applications items related to performance.

## About Total APM Corporation

Total APM is a leader in application performance management, developing customized methodologies and processes for implementing practical performance management solutions for our customers. Our expertise is in tuning enterprise applications, and, in particular, Siebel applications. Utilizing Precise i<sup>3</sup> to implement, monitor and tune applications, we provide mentoring and training services that help you improve and optimize performance management. Total APM, a partner of Precise Software Solutions, is a reseller of the popular Precise i<sup>3</sup> suite. We consider ourselves to be a genuine VAR (Value Added Reseller), as we provide our customers with a total package: licensing, services and the support required to make your i<sup>3</sup> investment a successful one.

Total APM also manages and operates the leading Precise i<sup>3</sup> user group at [www.AskEyal.com](http://www.AskEyal.com)

## About the Author

Eyal Markovich is an industry leader in performance tuning of large enterprise systems, and the Siebel Application, in particular. With more than 15 years experience working for Precise/Veritas/Symantec, Eyal is one of the industry's foremost experts in Precise i<sup>3</sup>. His past experience in developing Precise i<sup>3</sup> and working with Precise/Symantec customers gives him a unique expertise not only in i<sup>3</sup>, but also in application technology (such as, Siebel, Oracle, MSSQL, .Net, and Java) as well as in customers' applications and requirements.

Copyright © 2008 Total APM Corporation. 1580 Sherman #801 Evanston, IL 60201 U.S.A.  
All rights reserved.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

## Contents

Introduction .....	4
Siebel application performance .....	5
Performance as part of the implementation process .....	5
What level of performance can you expect from a Siebel application .....	6
A Siebel upgrade as a factor contributing to performance .....	7
Why is a load-generating tool not enough? .....	7
Using an APM expert .....	8
About the Performance Test Environment (PTE) .....	8
About the Performance team .....	9
About the i <sup>3</sup> toolset .....	11
The process .....	11
Pre-Production—building a performance test plan .....	12
Pre-Production—analyzing performance test execution .....	12
Additional ways to identify performance problems .....	13
Analyzing a performance defect .....	14
Testing possible solutions .....	14
Appendix A—Precise i <sup>3</sup> toolset .....	16
Low overhead 24/7 monitoring .....	16
Provides historical information .....	16
Identifies user activity vs. LDAP .....	17
Allows you to focus on a single user's activity .....	18
Displays top SQLs and lets you map them to a Siebel event .....	19
Provides a resource utilization breakdown of Siebel activities .....	20
Captures bind variables .....	21
Displays real execution plans .....	22
Displays top objects in plan .....	22
Reports on changes .....	23
Meets the needs of both DBAs and Siebel teams .....	23
Helps identify users that require additional training .....	23

## Introduction

Oracle's Siebel Customer Relationship Management (CRM)™ application offers market-leading CRM solutions based on the latest Web technologies. It is a complex enterprise application consisting of multiple architectural tiers (N-tiers). Siebel performance monitoring and tuning have become a critical aspect of any Siebel deployment with the need to satisfy the demands of partners, suppliers, and customers.

Siebel application performance is defined as the Siebel application's ability to function, and generally measured in response time or throughput. Some of the factors affecting a Siebel application's performance include inadequate hardware, poor network infrastructure, heavy customization, excessive network round trips, extensive data volume, poor tuning of database objects, and improper database configuration. The best way to tackle Siebel application's performance issues is to anticipate, plan, and consequently avoid the problems. However, this is not always possible. For example, it is often difficult to anticipate and account for all the performance issues that can arise during a major Siebel version upgrade, with an accompanying database platform upgrade. Furthermore, while pre-deployment testing is the most cost-effective and risk-averse approach toward ensuring that Siebel eBusiness enterprise applications operate at acceptable levels of performance, many performance problems arise only in production. Consequently, regardless of the time you spend testing, you will still need to have a good solution for monitoring and tuning Siebel production.

This document will present best practices for monitoring and tuning Siebel eBusiness applications. The document is intended for organizations that want to ensure their Siebel deployment meets the performance demands of users and customers.

Precise i<sup>3</sup> (also known as Symantec/i<sup>3</sup>) is an application performance monitoring and management tool that has been utilized by Total APM as an integral part in the process of monitoring, troubleshooting, and solving Siebel performance problems.

There are several good Oracle Siebel product documents (especially Siebel Book Shelf 7.8/8.x), Tech Notes, Alerts, Support Articles, and FAQs that address Siebel Performance and testing. This document does not repeat what is already discussed in these documents. Also, please refer to other papers for a discussion of additional topics, such as, hardware selection, configuration, and testing.

Examples of specific Oracle Siebel documentation that can be used include the Book Shelf > Performance Tuning Guide (7.8 or 8.x), and Technical Note 582: Oracle CBO and Siebel Business Applications.

While this document focuses on the methodology and process of monitoring Siebel application, the white paper "*Siebel Oracle Database Monitoring and Tuning – Detailed techniques*" discusses some key problems and issues that contribute to the poor performance experienced by many Siebel implementations. You can use this paper to gain a better understanding of what

can cause slow performance in your implementation. It examines the performance issues encountered, and the remedies used during several Siebel upgrades, from Siebel 6 to 7, in several Siebel implementations. The paper can be downloaded at [www.totalapm.com/whitepapers.html](http://www.totalapm.com/whitepapers.html).

## Siebel application performance

Improving the performance of your Siebel application begins when you recognize the importance of performance management as an integral part of any successful implementation. Identifying a performance problem, finding its root cause and uncovering possible solutions is not a trivial task. You should not expect to allocate two weeks to performance improvements just before going-live with a project, and expect to achieve good results.

Whether performing a large scale upgrade, developing a new module, or investigating production performance, the need for performance management and the way to go about it is similar. In this section, we'll discuss Total APM's best practice recommendations for managing performance in the Siebel application.

Total APM's solution framework is built on three basic principles:

- Recognize the importance of managing performance *throughout* the implementation processes.
- Have the right tool-set in place.
- Build the methodology and processes required to address performance problems.

## Performance as part of the implementation process

One important message of this document, is that performance monitoring must be part of your implementation process. By placing proper emphasis on performance, starting with the implementation process, you will come to understand and appreciate how a good performance monitoring solution can contribute to the performance and success of your application.

## What level of performance can you expect from a Siebel application

Out of the box, a Siebel application is optimized for performance. However, the Siebel application provides extensive capabilities that enable you to build custom solutions to satisfy even the most complex business requirements. As a result, it is easy to run into performance issues as you tailor a solution to meet your customer's needs. In the Siebel development world, a distinction is made between custom configuration and customization, the later requiring more extensive changes. Nevertheless, since the performance impact of each Siebel change does not necessarily depend on its magnitude, in this paper we have chosen to use the term customization to refer to both custom configuration and heavy customization of the Siebel solution.

Note that, while a proper architectural design is required for every Siebel implementation, and each of the Siebel tiers need to be studied and sized appropriately, this paper does not address these important issues, which require application server tuning (such as, how many Web servers to use), and object manager tuning. In addition, this paper does not address questions such as, how many CPUs, speed or memory is required for EAI servers, or questions regarding what should be the maxTask setting. While these issues are critical for successful implementation, and improper settings will result in poor performance and response time, it has been our experience that sizing the Siebel application is an issue that most customers identify prior to going live. Most customers identify improper sizing at early stages of the implementation. The decrease in hardware costs, especially memory, also contributes to this. This document therefore assumes that the Siebel system has been properly sized.

Still, most Siebel customers experience performance problems. In some cases, our customers experienced a View or a PDQ that performed poorly for all of their users. In other cases, a View performed well for most of their users, but resulted in a timeout for others. Moreover, we have seen cases where Query (View) performance was acceptable for several weeks and then suddenly worsened; or cases where performance in production was completely different than that experienced during development, QA, training, etc.

It is therefore necessary for customers to integrate performance management into their implementation strategy. This includes carefully analyzing every customization or configuration proposal for its impact on performance, and carrying out proper performance testing, monitoring, and performance tuning. By using the methodology and the i<sup>3</sup> toolset presented in this paper, you can immediately identify several tuning candidates and devise a simple solution. We call these "low-hanging-fruit". By having the proper tool set and methodology in place, you can be assured that when performance decreases, you will have all the necessary data you require to identify the root cause of the problem.

## A Siebel upgrade as a factor contributing to performance

The best practice process suggested below is recommended for all Siebel customers, regardless of the specific implementation stage they are in. Specifically, a Siebel application's performance is often worsened after upgrading to a major Siebel and database version release. This is especially true for implementations requiring extensive customization and application interfaces. As a result, it is very important that customers make performance planning, testing, and tuning part of their Siebel upgrade efforts.

The following two examples highlight the complexity of the upgrade with regards to performance:

Consider an upgrade to a new version of Oracle. In most cases a Siebel upgrade from any Siebel version prior to Siebel 7.5 to version 7.8 or 8.x requires an accompanying upgrade to Oracle 10g and changes to the optimization method from rule-based to cost-based. Even though out-of-the-box queries have been optimized, queries generated by customizations must be re-examined. Furthermore, out-of-the-box queries may cause performance problems when statistics are not collected or when data is skewed in a way that causes Oracle CBO to pick the wrong plans.

Siebel Application 7.x includes architectural changes that can cause unexpected performance problems. For example, during an upgrade from a pre Siebel 7.x version to 7.x or later, the ownership of activities changes from individual to team ownership. S\_ACT\_EMP is introduced by Siebel as an intersection table to store an M:M relationship between employees and activities—adding complexity to activity queries. Siebel customers that spent time tuning Activity queries in Siebel 6.x (as for example, by adding customized indexes) may be surprised to find that the new Activity queries are slow after the upgrade. The previous customized indexes may not be in use anymore. Such a performance problem, which is typically identified toward the end of the upgrade project, may lead to a delay in going live or to poor performance in production. It is, therefore, critical to be capable of resolving these cases as soon as possible.

## Why is a load-generating tool not enough?

Typically customers use a load generating tool for testing and creating load on the system (Mercury LoadRunner is the most commonly used tool). While load-generating tools are very valuable for performance testing, they should not be used as the only performance monitoring tool. Load generating tools are typically set to observe averages, and the 95 percentile response-time of transactions. These measures may be misleading in some cases. For example, an Activities transaction that was executed 10,000 times may report a 1 second average, while the first execution took 5 minutes to complete. In this case, the average is satisfactory, as it would be below what most clients specify as an acceptable average. However, for the Siebel end-user who has to wait 5 minutes every morning when navigating to this view, this poses a performance problem. As we will discuss in detail below, using i<sup>3</sup> in conjunction with a load generating-tool, allows you to identify performance problems, as well as find their root cause, and propose a solution.

In addition, it is not realistic to assume that you can script **all** users' navigations. You must, therefore, assume that you will face additional performance problems in production. Consequently, a performance management solution is required.

### Using an APM expert

The following table, taken from the Siebel book, "Planning a Successful Siebel Implementation", suggests a list of roles for a successful project team.

Project Team Member Title
Project Manager
Business Analyst
Application Architect
Technical Architect
Subject Matter Expert
User

Total APM recommends adding to this list an additional role: "Application Performance Expert". When selecting your APM expert, make sure that s/he is familiar with your performance toolset (Total APM recommends using i<sup>3</sup>) and is an expert in Siebel performance tuning. If you hire a consultant as an APM expert, you should also verify that the consultant has the experience to train your internal team regarding how to use the performance toolset and to understand and carry out the performance process.

Typical questions that should be asked when choosing an APM expert include:

- How long has the expert been tuning Siebel applications?
- How long has the expert been working with i<sup>3</sup>?
- Does the consultant have the experience required to train your IT team to use i<sup>3</sup>?

### About the Performance Test Environment (PTE)

The use of a dedicated Performance Test Environment (PTE) is part of Total APM's methodology and process. All attempts must be made to make PTE a replica of what the production environment would be like at go-live (hardware capacity, data volume, infrastructure, authentication, user, etc.). If the budget for this new system is a concern, we recommend having less Web/application/doc servers but creating an identical replica of the database as that of the production database. Many of the database performance problems will not be identified unless the same quantity of data is stored in the database.

When planning the Siebel system, the PTE environment should be taken into consideration and planned as early as possible. The PTE should be used mainly

for load/performance testing and should remain active after going live. You should also plan on adding performance testing to your acceptance criteria, which should be carried out in the PTE.

For example, one of our customers developed a new view with two new applets, based on business requirements (this was not a part of the upgrade process). The modification was tested for functionality in QA, and was promoted to production, without a performance test. This led to poor performance for several new queries, forcing the customer to take Siebel down the next morning and rollback the change. This could have been avoided had a performance test been included as part of the acceptance criteria.

## About the Performance team

In order for the suggested best practice process (described below) to succeed, it is important to add a virtual performance team to your methodology.

This team should be set up and in place throughout all stages of the implementation process. In some cases (as for example, in the case of a large Siebel upgrade project) this would require full time resources, while in other cases only part time resources assigned or called on a need-to-know basis, are required. In any case, you should form the team and implement the process as described below.

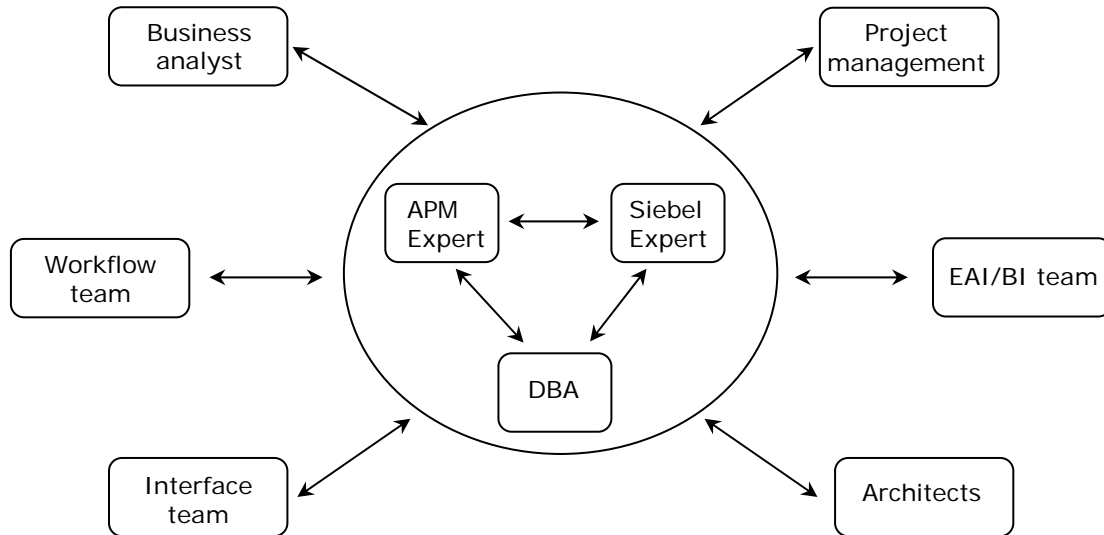
Assuming a large Siebel upgrade project, upgrading from Siebel 6.x to 7.8, the optimal performance team should include the follow members:

**APM Expert**—A full time performance expert (i<sup>3</sup> and tuning expert) should be available throughout performance testing, go-live and the first few days of the post go-live performance management efforts. For some of our customers, Total APM also trained an internal staff regarding the use of i<sup>3</sup> and provided remote APM services, on a per-need basis, for the investigation of production incidents.

**Siebel Expert**—A dedicated experienced Siebel architect with extensive knowledge in Siebel configuration, WF Processes/Policies, Siebel Server architecture, Siebel infrastructure, and Siebel Application Integration, should be involved. The Siebel architect would, from time to time, coordinate with other Siebel experts, Siebel Expert Services, and Siebel support organization resources on some specific issues.

**A DBA**—The DBA's time can be shared between the performance team and other duties. However, as the post upgrade go-live date approaches the performance team should have access to 90% of the DBA's time. During post upgrade tuning, the team should have access to 100% of the DBA's time.

The above team's skill-composition is necessary regardless of the size of the project. However, for smaller upgrade efforts (or post implementation), there is no need for the individuals to be exclusively dedicated to performance. Obviously, the above team players are simply key members dedicated to the narrower area of performance tuning. Performance efforts cannot succeed without the support of the entire project team including business analysts, project managers, technical architects, etc.



During a recent project, where the customer implemented the above structure, and following a few success stories, all members of the Siebel project recognized the value that the team offered towards the success of the project. What commonly ensued was that the performance team would receive frequent requests from other teams to review performance during the development phase, before problems were even reported. For example, we were not surprised to receive a request from the WF team to look into the performance of a new WF policy that was executed during the night, in PTE, even before they released it for performance testing (this was part of a unit testing).

## About the i<sup>3</sup> toolset

In order to monitor and tune the Siebel application, it is important to collect and analyze the necessary metrics and data. With the right information, you can identify performance problems (hopefully proactively before the end users themselves start to notice the problem and complain) and get to the root cause of any performance problem.

Total APM has been using Precise i<sup>3</sup> to monitor and tune Siebel applications. From our vast experience with Precise i<sup>3</sup>, we strongly believe in the i<sup>3</sup> technology, and it is the only tool we use to monitor Siebel. Precise i<sup>3</sup> is used by many Fortune 500 companies to monitor all types of applications, not just Siebel applications.

The following is a partial list of some of the impressive features that make i<sup>3</sup> the perfect solution for Siebel monitoring. Precise i<sup>3</sup> contains many more useful and important features, but we will only touch on the main features that can assist you in Siebel database monitoring in this paper.

- [Low overhead 24/7 monitoring](#)
- [Provides historical information](#)
- [Identifies user activity vs. LDAP](#)
- [Allows you to focus on a single user's activity](#)
- [Displays top SQLs and lets you map them to a Siebel event](#)
- [Provides a resource utilization breakdown of Siebel activities](#)
- [Captures bind variables](#)
- [Displays real execution plans](#)
- [Displays top objects in plan](#)
- [Reports on changes](#)
- [Meets the needs of both DBAs and Siebel teams](#)
- [Helps identify users that require additional training](#)

See Appendix A for a detailed description of each feature.

## The process

The following section describes Total APM's suggested process methodology (hereafter, the process) for integrating performance management into Siebel application release cycle. While the process discusses the early stages of a new implementation or upgrade, post production-implementation customers can still benefit from this process by performing the production-related steps. Note: the rest of the document assumes that a PTE environment was set up, the performance team was established, and the customer owns the i<sup>3</sup> toolset.

### Pre-Production—building a performance test plan

Performance test plans should be developed along with other QA test plans. These plans should be developed with functionality, scalability, and performance in mind.

Performance load tests should be performed in four phases using a load generator tool (such as Mercury Load Runner) in PTE:

- Phase I—load tests, including simple navigations, in views.
- Phase II—load tests including targeted queries that were prioritized by architects and key business users.
- Phase III—load tests that include testing interface response times. Testing each interface in isolation is recommended.
- Phase IV—load tests (including end-to-end fully integrated tests).

The above structure allows you to easily identify scalability and performance problems. Once a scalability or performance problem is recognized, it will be easier to identify the source and root cause of a problem.

### Pre-Production—analyzing performance test execution

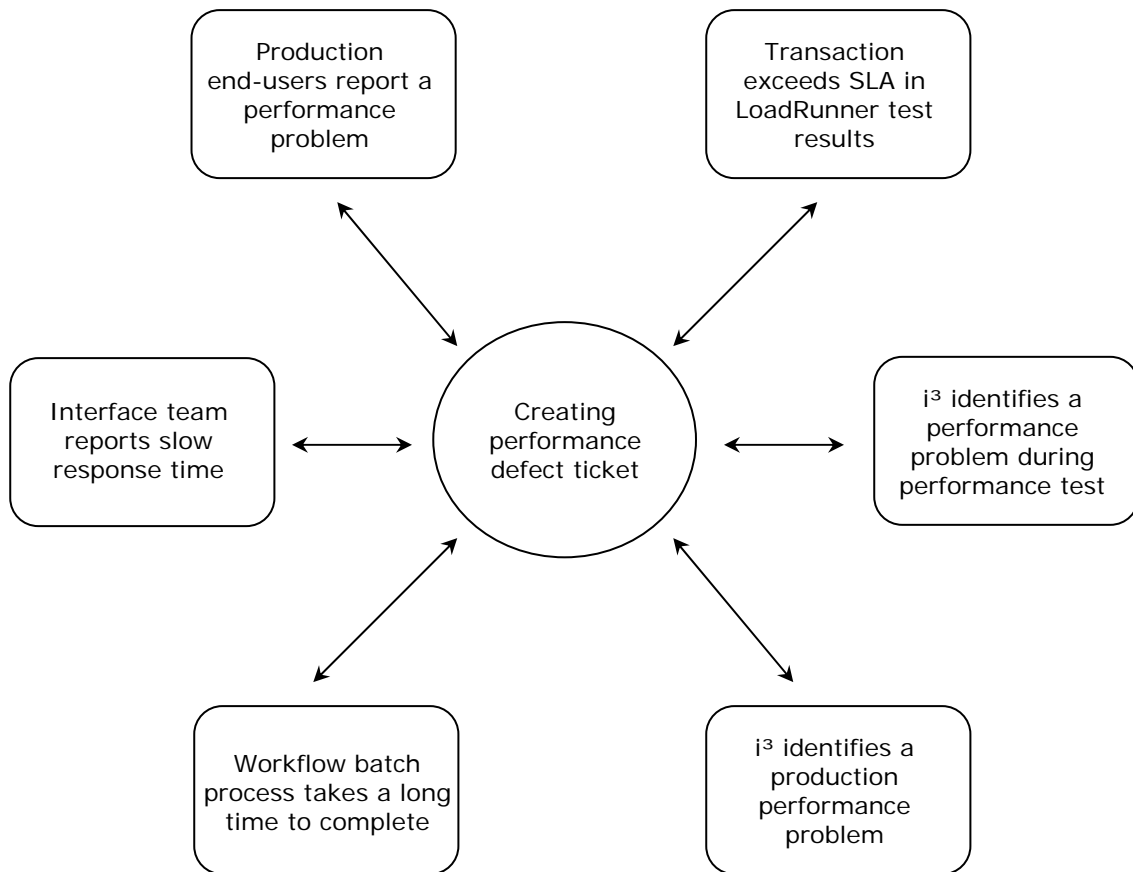
Performance testing is a common practice in most organizations. Typically, performance is measured on an average, or 95 percentile of response time of transactions. These measures, while important, must be used in conjunction with other measures. The reason being that while data taken on an average, and in 95 percentiles, identifies views with consistently slow response time, it may fail to recognize views that perform slowly during the first execution but very fast in subsequent visits. This is a typical database behavior where data is not initially in the buffer.

The i<sup>3</sup> toolset can be very valuable in such cases. The i<sup>3</sup> expert can create a performance analysis report after each execution of a test. The report could then be used to identify performance bottlenecks (database, network or servers, Siebel, Web, Document, etc). The i<sup>3</sup> report should be viewed as a supplement to the reports created by other tools (for example, Mercury LoadRunner, and BMC). It allows the customer to obtain a complete picture of the scalability and performance of the Siebel application. If a potential performance problem is identified, a performance defect should be opened and assigned to the performance team.

### Additional ways to identify performance problems

In a production and pre-production environment, performance problems can be reported by several sources. Once a performance problem is reported, the goal is to verify, as quickly as possible, that a problem indeed exists, identify the root cause of the problem, and finally find a solution. This matches the i<sup>3</sup> methodology exactly, and it is therefore best to let the performance team investigate the problem using i<sup>3</sup>. In other words, once a problem is identified, a performance defect ticket should be opened and assigned to the performance team for further investigation.

The following diagram shows an example of the list of events that can lead to performance defects. As you can see, the events correspond to several stages in the implementation process.



## Analyzing a performance defect

The first task in analyzing a performance problem involves mapping the issue to a particular Siebel transaction, Siebel event, or sub-event. This also includes identifying the bottleneck where most of the time is spent. In our experience, in 90% of the time, slow performance can be traced to the database response time or SQL issues. For comparison purposes, it is recommended to baseline the performance problem before implementing any solution. We typically use two sources to obtain this information—the i<sup>3</sup> report describing the end user response time, and the SQL Plus autotrace output.

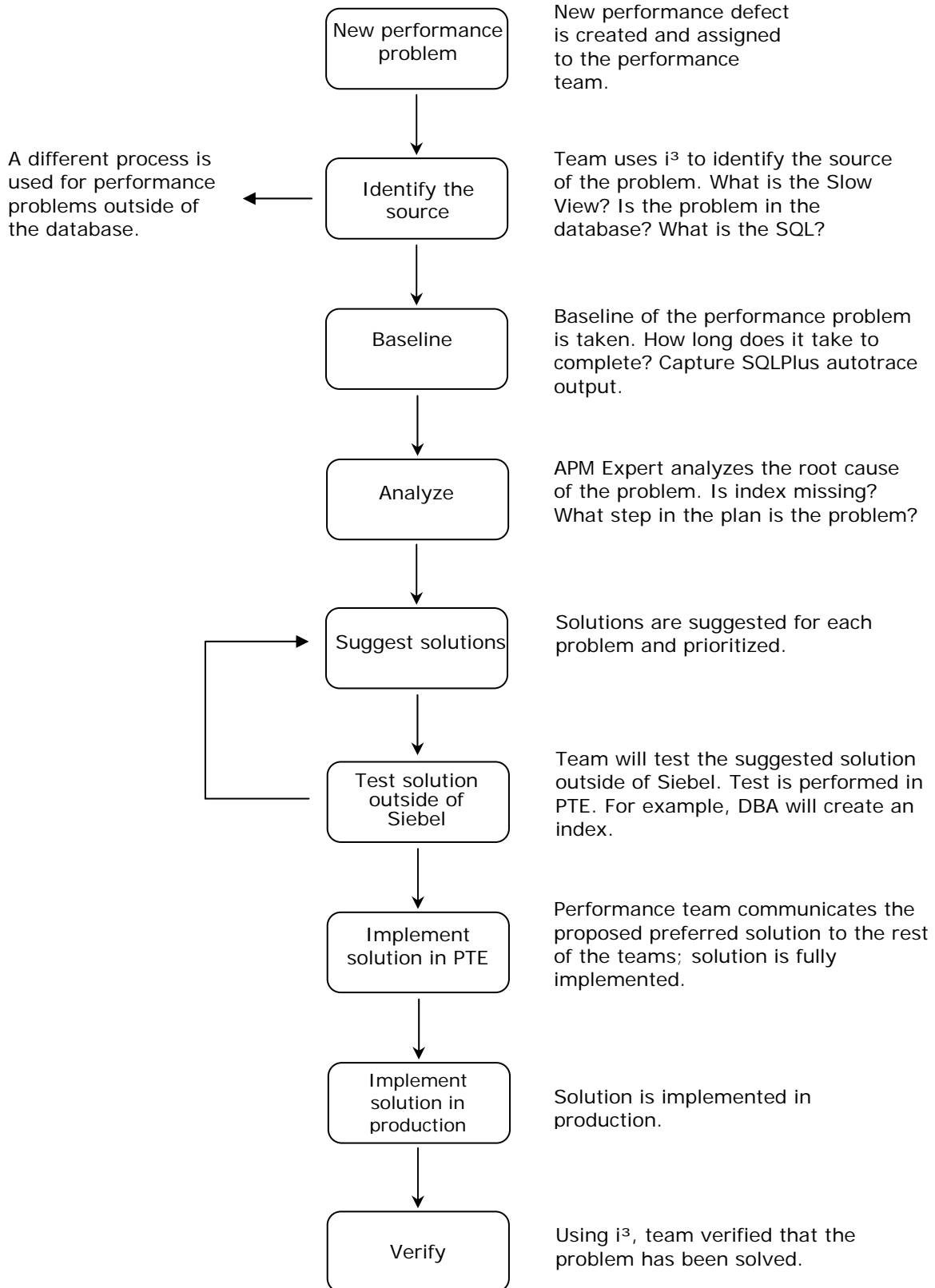
If the performance problem is identified as a SQL database problem, the APM expert should investigate the performance of the query and check if the query has been optimized. This task should be performed by the APM expert, outside of Siebel, by analyzing the query using i<sup>3</sup> or by executing the query directly from SQL Plus (additional tips on how to set up Siebel hints and capture the bind variable for SQL Plus execution are presented later on in this paper). After the source of the problem has been identified, the team should brainstorm for possible solutions. Obviously, one should first verify that the suggested solution is supported by Siebel. This is where i<sup>3</sup>, database, and Siebel skills come together.

## Testing possible solutions

Once the team comes up with a possible solution, the solution should be tested before being implemented. The testing should be done in the PTE environment where the performance problem can be recreated. This process is time consuming and may require the testing of several different alternatives, before an optimal solution can be found.

Below is an example of a typical workflow of events for a slow Siebel navigation event:

- Problem was analyzed by APM expert and identified as a Database performance problem.
- Baseline was taken in PTE by executing the problematic View in Siebel and running the query from SQL Plus.
- A potential Index was identified. The Performance team DBA creates an index.
- An APM expert checks if the index was used and if it was helpful in SQL Plus.
- The APM Expert identified that the CBO did not pick the new index or the performance gain of the new index was not high. A different index that will ensure better results is identified.
- The DBA drops the previous index and creates the new index.
- An APM expert verifies that the problem has been resolved.
- The Siebel expert performs the required modification in Siebel Tools and the change is promoted to production.



## Appendix A—Precise i<sup>3</sup> toolset

### Low overhead 24/7 monitoring

Precise i<sup>3</sup> was designed as a monitoring solution targeted for a production environment. As such, the overhead of the i<sup>3</sup> monitors is very low (less than 2%) and you can keep it running and monitoring at all times. This ensures that if a performance problem occurs, you will have the required information to analyze the problem and get to the root cause of the issue. Relying on a monitoring solution that needs to be turned on whenever a problem occurs is risky, since you will be missing vital information, and you can never be sure when the performance problem occurred. In addition, some problems affect all users while others only affect a single user. For example, one user can constantly complain about the performance of the Activity view while this view works fine, and within its SLA, for thousands of other users. It is, thus, imperative to have a tool that collects information all the time on all activities in the system.

### Provides historical information

Another advantage of i<sup>3</sup> is the ability to look up, observe, and analyze historical information. i<sup>3</sup> allows you to answer questions like:

- Which users used the My Activity view yesterday at 2 PM?
- Why was system performance slow last week?
- Has this query always performed so badly? If not, what has changed?

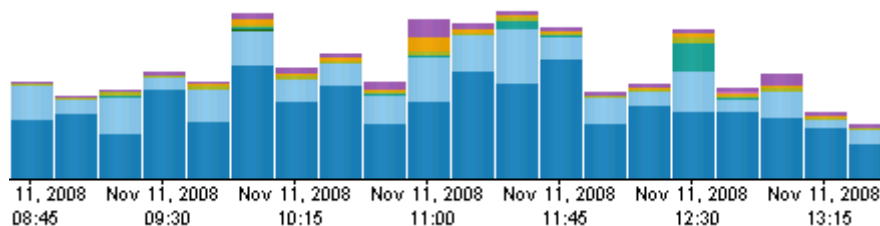


Figure 1 Historical information displayed by i<sup>3</sup>

i<sup>3</sup> also allows you to examine performance over time, identify trends and bottlenecks, as well as zoom in on a problematic time range for further investigation. To analyze a performance problem that happened several hours or days ago you do not need to recreate the problem. You would only need to use i<sup>3</sup> to examine the activities that occurred during the timeframe that the problem occurred.

## Identifies user activity vs. LDAP

One of the main challenges (especially when Siebel uses LDAP or Active Directory authentication, as most implementations do) when monitoring a Siebel application is to correlate end-user activities in the database. From the Oracle database point of view, all Siebel database connections share the same session identifiers. They are all connected to the database using the same user, program, etc. For example, almost all database connections will use the siebmtshmw or siebmtshmw.exe as a program name. So, if a specific user complains about slow response time, how can you identify if the activity from this user is slow because of the database being accessed? How will you identify the Views that this user is using or the queries executed by this user?

i<sup>3</sup> collects this information and allows you to access it easily.



Figure 2: Top Siebel views sorted by Oracle resources

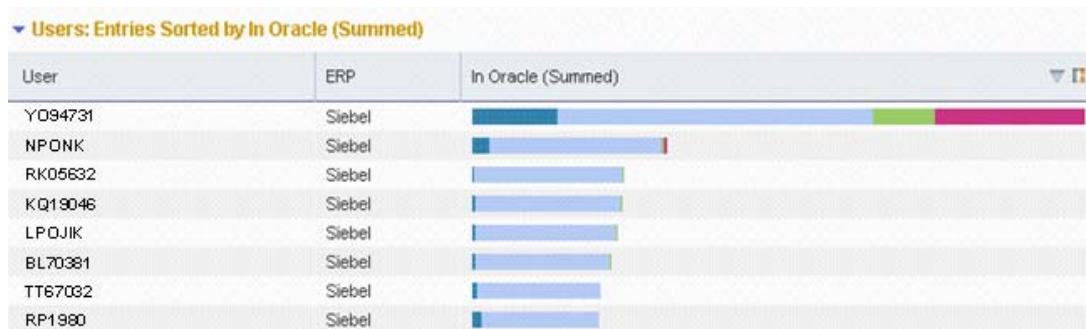


Figure 3: Top Siebel users sorted by Oracle resources

## Allows you to focus on a single user's activity

With a large Siebel implementation, with thousands of users, one faces a real challenge when it comes to monitoring a single user's activity. How can you easily identify what Views and SQL a specific Siebel user executed yesterday between 9:30–10AM?

i3 allows you to digest the performance information in many ways. You can easily filter out activity or zoom in on a single user (view, Siebel server, Siebel application, etc.). For example, with a few clicks of the mouse, you can see the queries that a single user (a Siebel user and not a database user) executed from a specific view, at a given time.



Figure 4: Focusing on a single user (in this case, YO94731), and listing Siebel Views executed by this user

## Displays top SQLs and lets you map them to a Siebel event

Similar to other tools that report on database performance, i<sup>3</sup> can display the list of queries that consume most of the resources. Most Siebel customers have a hard time, however, relating SQL to a user's specific navigation or query event in the browser interface, when trying to tune Siebel performance issues in test and production environments. In typical development and unit test phases of Siebel implementation, developers use Siebel- or Oracle-provided SQL diagnostic tools, such as the "/s" switch (found in the shortcuts for the Siebel Developer Web Client's start path), and Siebel log files. These tools are not well suited for subsequent test and production phases. Furthermore, most performance issues are not discovered during the development and unit test environments since these environments are not set up to run performance tests and the data volume in these environments generally does not compare to production. As a result, the need for a robust third party diagnostic tool becomes apparent. Precise i<sup>3</sup> fulfills this need and provides additional performance management functionalities, such as, monitoring and notifications. Precise i<sup>3</sup> provides the bridge between the database world and the Siebel world. Using i<sup>3</sup>, you can easily identify the list of Siebel users that executed each query, the view that the statement was executed from, in addition to the Siebel business object that executed each query.

▼ Statements: Entries Sorted by In Oracle (Summed)

	Statement	In Oracle (Summed)	%	Executions	In Oracle (Avg)
TUNE	60442.19964.07288.42124		7.77%	7659	00:00:05.519
TUNE	05401.02664.23760.47953		7.10%	1	10:44:00.020
TUNE	50125.04928.15071.33292		5.10%	126	00:03:40.120
TUNE	21997.31466.48770.55057		2.53%	2451	00:00:05.611
TUNE	37231.38965.02901.12125		2.43%	375402	00:00:00.035
TUNE	34792.37843.40691.47820		2.40%	170071	00:00:00.077
TUNE	22612.46074.38666.12210		1.89%	3	00:57:12.727
TUNE	41548.26133.24957.35619		1.83%	14	00:11:53.176
TUNE	25914.34692.11580.13426		1.17%	13	00:08:11.067
TUNE	14931.01206.16180.24260		1.05%	112	00:00:51.176
TUNE	47805.63468.50597.61845		1.02%	258	00:00:21.588
TUNE	06690.09259.31259.62588		1.02%	42316	00:00:00.131
TUNE	42399.57617.49650.63745		0.83%	3	00:25:04.573
TUNE	13053.27319.06659.29880		0.80%	197	00:00:22.151
TUNE	53106.13481.47937.05060		0.76%	1	01:09:37.820
TUNE	10480.44020.24534.37619		0.71%	176	00:00:22.159
TUNE	45410.29777.35549.27773		0.65%	6025	00:00:00.590

Figure 5: List of top queries

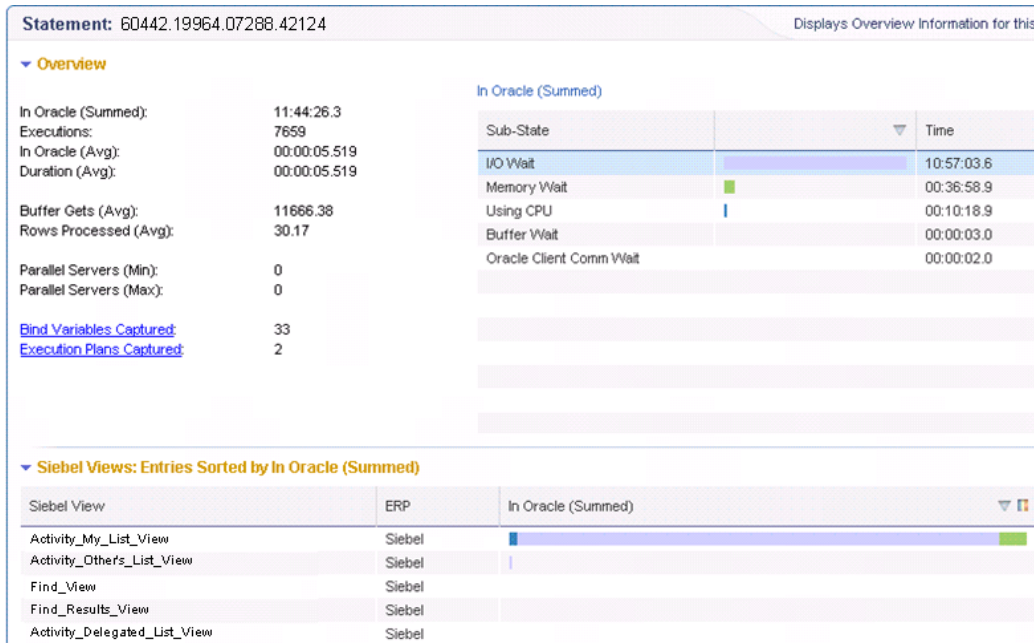


Figure 6: List of Views executing the top statement in the list

### Provides a resource utilization breakdown of Siebel activities

Knowing that a query's database time is high is not enough information to help you in your analysis. It is also important to understand the database performance profile of the application, in general, and for each statement, in particular. Precise i<sup>3</sup> shows the breakdown of activities for each wait group. Understanding the wait type allows you to explore different solutions for each problem. For example, a faster I/O subsystem will not improve high CPU utilization. High CPU utilization can be improved by improving the plans of the top queries.

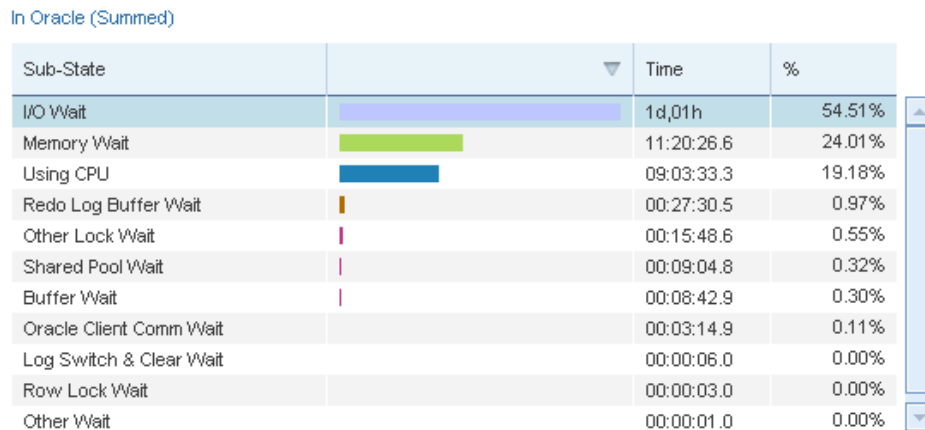


Figure 7: Resource utilization breakdown

## Captures bind variables

Siebel uses statements containing bind variables throughout the application. Bind variables enhance performance, since Oracle does not need to hard-parse the query every time it is executed. However, it also has some limitations. It is difficult for the optimizer to select one optimal execution plan (see section on [Bind variable vs. literals—a large result set vs. a small result set](#)), and it is difficult for you to monitor the query execution. What if a query runs 99 times fast and 1 time extremely slow? You need to be able to identify if this is due to the bind variable values. Precise i<sup>3</sup> collects the bind variables for the top queries, as shown in the following figure.

Bind variables metadata

Bind Name	Bind Value	Column Name
:1	2-713PR4	T2.EMP_ID
:2	11/06/2008 00:00:00	
:3	Not Started	T2.ACT_EVT_STAT_CD
:4	In Progress	T2.ACT_EVT_STAT_CD
:5	On Hold	T2.ACT_EVT_STAT_CD
:6	Acknowledged	T2.ACT_EVT_STAT_CD
:7	Diary Expired	T2.ACT_EVT_STAT_CD

Figure 8: A Bind set collected for a single query with 7 bind variables

In addition to the ability to see the values, this feature can be of great assistance when you need to investigate the performance of a query in Oracle (using i<sup>3</sup>, Toad, EM, etc). A set of bind variables is necessary to execute the query and analyze its performance.

## Displays real execution plans

Another advantage of the i<sup>3</sup> suite is the ability to report the real execution plan that Oracle uses. We have found several cases where a query that performed well in PTE performed badly in production. Moreover, a query that was running for sub-seconds in production suddenly ran for entire minutes and basically brought the entire system down. In almost all of these cases, the Oracle CBO decided to change the execution plan. It is critical to verify that the plan was changed as soon as possible since it provides you with the necessary information to understand why the query is slow. In addition, it is necessary to view the real plan that the CBO is using and not the estimated plan that development tools show (like TOAD). Since Siebel introduces several session hints, you want to make sure that the plan you are viewing is what is really being used.



Figure 9: Two plans for the same query. The second plan has a 20% higher cost

## Displays top objects in plan

Another challenge that Siebel developers face is the complexity of the execution plan of typical Siebel queries. It is common to see execution plans that span more than 150 steps. If you need to investigate a plan, you need to know which steps in the plan are the problematic, or bottleneck steps. You cannot trust the CBO to identify bottlenecks, since in many cases you are investigating the plan because the optimizer picked a bad plan. Instead, you should concentrate on understanding which objects were the hottest objects for each query. Knowing this will allow you to focus on these objects and find the problem faster.

		Type	Object	%
		Heavy Sequential I/O on Table	S_ACT_EMP	81%
		More Than One Real Plan Was Detected		
		Bind Values Were Collected		

Figure 10: i<sup>3</sup> identified that 81% of the waits are on S\_ACT\_EMP

## Reports on changes

At times you will run into a performance problem on a transaction that used to perform fine. In these cases, it is always helpful to investigate what has changed. Precise i<sup>3</sup> lets you track and report database changes over time. Changes can include adding or dropping indexes, changing init.ora parameters, or even changes to object statistics.

## Meets the needs of both DBAs and Siebel teams

Precise i<sup>3</sup> for Siebel is a very popular tool amongst the entire Siebel team. Tailored to monitor Siebel, the Siebel team uses Precise i<sup>3</sup> to gain visibility into the performance of Siebel end users, Siebel servers, etc. Precise i<sup>3</sup> is also used by DBAs since it provides them with complete visibility into database performance. The fact that one tool can be used to measure and report an entire application's performance helps reduce the blame-storming that often takes place when investigating performance problems. Very often you'll hear statements such as, "This is a network issue not a database issue", or "The storage is fine the problem lies in the database or Siebel code", on sites that don't work with the i<sup>3</sup> product suite. The i<sup>3</sup> product suite puts an end to blame-storming since one tool is all you need to identify where the bottlenecks and problems lie.

## Helps identify users that require additional training

In many cases, some SQL queries execute for a long time in the database and consume high resources. Further investigation typically identifies that there is little to be done on the database to improve the performance of these queries. The problem is not in the way the database is executing the SQL, but lies in the way the SQL is being executed in the first place. The database simply executes what it was asked to do, but sometimes takes 15 hours to complete.

In our experience, many of these cases were merely a result of lack of end-user training. This can include users that tend to execute open queries or don't know how to narrow down the number of records efficiently. Alternatively, it can include users that use two asterisks, too many times, in like searches, or users that sort on ALL views with millions of rows in the result table. Precise i<sup>3</sup> can help find these invalid queries, and display the list of Siebel end-users that executed them.