Achieving a Superior Ownership Experience for Oracle Business Intelligence Enterprise Edition

An Oracle Whitepaper
August 2008

ORACLE

INTRODUCTION

Business Intelligence (BI) is no longer the domain of a few power analysts or a point application used by a small target audience of middle and upper management. BI has evolved into a mission critical application that is being used by many thousands of users across the broad expanse of the enterprise. Oracle Business Intelligence Enterprise Edition is a comprehensive and integrated suite of Analytic Tools designed to bring greater business visibility and insight to the broadest audiences of users, allowing any user in an organization to have Web-based self-service access to up-to-the-moment, relevant, and actionable intelligence. It features a highly unified and scalable architecture centered on a sophisticated Oracle BI Server that provides semantic integration of data, spanning operational and analytical data sources. Oracle also offers an entire suite of pre-built analytic applications (Oracle BI Applications) built on the OBI-EE platform. These pre-built analytic applications are used to optimize performance for sales, service, marketing, contact center, finance, supplier/supply chain, HR/workforce, and executive management. To help you maximize the value of these mission critical applications, and to deliver a Superior Ownership Experience, Oracle provides a set of tools that facilitate top-down application management and cover the entire application lifecycle.

TOP-DOWN APPLICATION MANAGEMENT

A key requirement for managing Oracle BI EE and Oracle BI Applications is the ability to manage the entire application stack, which includes BI-specific components such as Oracle BI Server, Oracle BI Presentation Server, Oracle BI Scheduler, Oracle BI Cluster Controller and Oracle BI DAC Server, as well as infrastructure components such as databases and operating systems. All these components must work optimally together in order to deliver availability and performance required of your Oracle BI application. Therefore, it is important that all these components be managed together.

Traditional system management tools tend to focus on a silo approach of management – handling each component individually, and then trying to piece together information about the health of the application environment from the bottom up. Oracle Enterprise Manager goes beyond this bottom up approach by providing a top-down perspective as well. The top-down approach differs from the traditional bottom-up approach in that it provides not only a system-oriented view of the application's health, but an end user perspective as well. This makes it easier to assess the business impact of system events, and manage the Oracle BI application environment according to business requirements.

The top-down management approach starts with the Business Intelligence Management Pack, which extends Oracle Enterprise Manager to manage Oracle BI EE and Oracle BI Applications. Through the service level management (SLM) capabilities of the pack, you can model the availability and performance requirements for your Oracle BI application, and then monitor your application environment according to these requirements. This approach helps you focus your resources on issues that are truly important – those that actually impact your business.

The Business Intelligence Management Pack is complemented by other Oracle products such as Oracle BI Client Tools (including Oracle BI Administration Tool and Oracle BI Scheduler Job Manager), Oracle BI Data-Warehouse Administration Console (DAC), Oracle BI Catalog Manager, Oracle Application Testing Suite, Oracle Database Management Packs, Oracle Middleware Management Packs, Oracle Provisioning Pack and System Monitoring Plug-in's for third party technologies to provide management coverage for your entire system environment, and support for each phase of the application lifecycle.

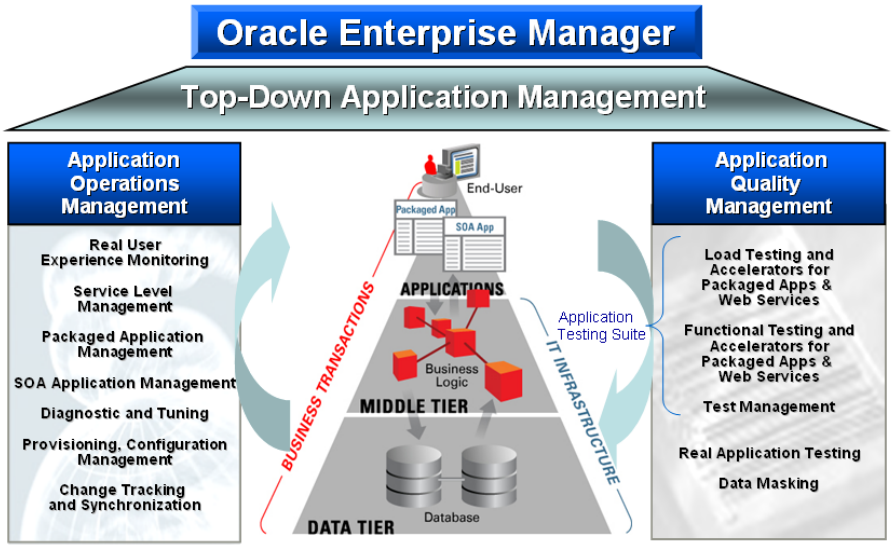

Figure 1 – Oracle Business Intelligence Management Solution

COMPLETE APPLICATION LIFECYCLE COVERAGE

The deployment of Oracle BI application goes through three distinct phases – Implement, Manage, and Optimize. In the very first implementation cycle, you need to plan your business model, design and customize your Oracle BI repository or use a pre-built DAC metadata repository and make customization changes to best fit your business model. In subsequent cycles, you may take an already deployed Oracle BI repository and make further changes to the Physical, Business Model and Mapping and Presentation layers, design, execute and monitor ETL

processes, or upgrade to a new release of Oracle Business Intelligence Enterprise Edition. Throughout the implementation process, you need to constantly test the application to make sure that it performs properly. You would also be constantly migrating configuration changes from development to test to staging environments. Ultimately, when you are ready to go live with your Oracle BI application, you would deploy your tested configuration from staging to production environment.

As you enter production, the focus shifts to management. You need to monitor the performance and availability of the application from both end user and system component perspectives. If a problem is detected, you need to triage the problem quickly and engage the right expert to locate the problem root cause. In addition, you need to monitor operational changes that are made to the environment on an on-going basis to ensure that these changes do not introduce problems into your environment.

Lastly, you need to "fine tune" your environment in order to achieve further optimization. The starting point of this process is a set of service level and capacity utilization reports that provide insight on the performance, availability and resource utilization of your application. You may use the information provided by these reports to decide whether to apply software patches from Oracle, tune the database, or make other functional adjustments to the Oracle BI application in order to improve application end user experience.

In the following pages, we'll describe how you may use various Oracle technologies to accomplish tasks in each of the three application lifecycle phases.

IMPLEMENT

Challenge 1 – Ensuring Functional Conformance to Business Requirements

Running functional tests to ensure an application's implementation conforms to business functional requirements is critical to the successful adoption of the application. However, relying on redundant manual testing is an inefficient use of quality assurance (QA) resources, especially for regression tests that need to be run over and over whenever changes are made to the application. Most automated testing solutions on the market carry a steep learning curve and require that your testers become programmers in order to test particularly for complex applications like Oracle BI EE, which provide extensive advanced functionality. Worse still is the fact that even after dealing with this learning curve, most automated testing solutions are either designed for functional testing or for load testing with no link between the two. This forces your testers to use multiple tools, learn different scripting languages, and maintain separate test scripts to test the same application. Oracle Application Testing Suite (ATS) eliminates the problems associated with traditional test automation tools, by providing a comprehensive solution for testing Web-based technologies. This allows your testers to quickly and efficiently test

both Web applications and Web Services in a single integrated test platform for automated functional and regression testing, load testing and test process management. Its unique integrated scripting platform enables automated test scripts to be used for both functional testing and load testing, eliminating the need to learn multiple test tools or create custom test harnesses.

Oracle Functional Testing for Web Applications, part of ATS, is an easy-to-use automated functional and regression testing tool. It provides a simple and intuitive visual scripting interface that enables testers to automate complex Web transactions. It also includes a wizard-driven scripting interface for creating automated Web Services test scripts utilizing WSDL files, which are automatically parsed to generate web service requests. Your testers can then create multi-step Web or web service transactions, parameterize data inputs, leverage default and custom test cases for validation and execute these test scripts without programming.

Oracle Test Manager for Web Applications, another component of ATS, manages the entire test process, including test requirements, manual and automated test cases, and defects identified during testing. Oracle Test Manager for Web Applications maximizes the return on investment for your testing tools by providing a centralized repository for storing all your test assets, which is accessible through a simple and intuitive Web-based interface and can be completely customized to fit your test process.

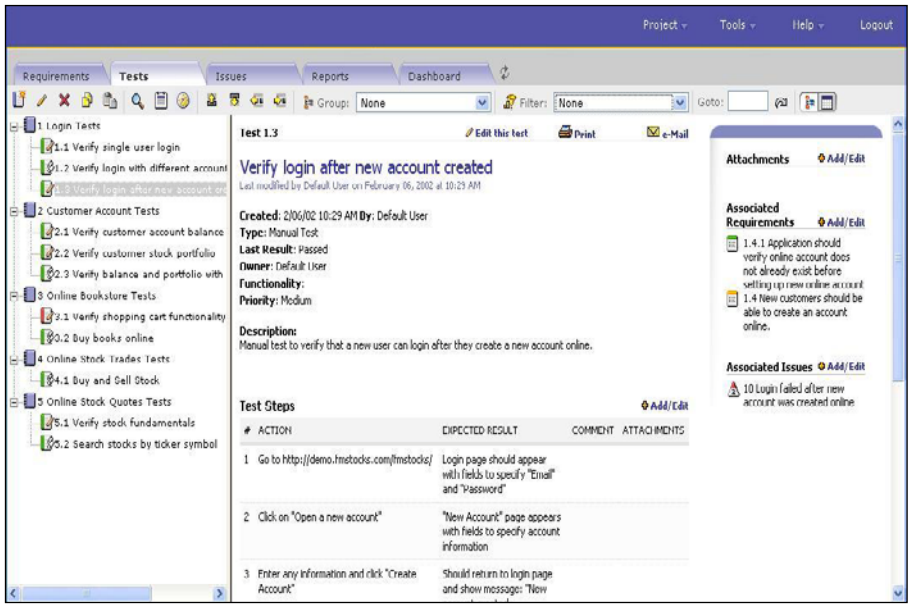

Figure 2 – Oracle Functional Testing for Web Applications

Challenge 2 – Projecting Capacity Utilization and Achieving Scalability

Load testing is a critical activity to ensure that the application will scale and consume resources efficiently when it is used by a large number of users. Manual testing is not an alternative for load testing since you can't realistically simulate production loads or have testers provide objective performance results. However, automated load testing can be a challenge given the difficulty of automating complex transactions and accurately simulating live user loads. Furthermore, as load tests are run, your testers need to be able to work with your application developers to quickly analyze performance problems that are identified during the tests in order to satisfy application performance requirements and optimize capacity utilization.

Oracle Load Testing for Web Applications, the third component of ATS, provides load, scalability, and stress testing of applications. The same scripts that your testers create with Oracle Functional Testing for Web Applications for functional testing can be reused for load testing in Oracle Load Testing. With Oracle Load Testing for Web Applications, your testers can configure one or more scripts to run with hundreds or thousands of concurrent users simulating the load that their Web applications would experience in production to assess performance. Oracle Load Testing for Web Applications not only stresses your application to simulate the impact of end-user workloads, but also enables rigorous validation that protocol-based legacy client/server testing tools cannot provide.

Figure 3 – Oracle Load Testing for Web Applications

As load tests are run, you can also use Business Intelligence Management Pack to observe the application's behavior under various load profiles. You may capture statistics for all critical Oracle BI components including the Oracle BI Server and

Oracle BI Presentation Server and monitor performance and utilization metrics of the underlying server machines. All these captured metrics would be stored in Oracle Enterprise Manager's repository, and can be used to establish performance baselines that provide context for production monitoring.

In addition, your testers and application developers may use Business Intelligence Management Pack's Dashboard Reports tool to troubleshoot failed queries and view detailed statistics on dashboard usage generated from the load tests. You may view the breakdown of response time based on database time, compile time and overall time for all generated queries and view detailed information on failed queries including error code, error message, and failed SQL statements.

Challenge 3 – Orchestrating Controlled Deployment of the Tested Application

After functional and load tests confirm the functional compliance and performance characteristics of the application, you are ready to have your administrators deploy the application into production. It is very important to make sure that the application that is deployed into production maintains the exact configuration that was tested in functional and load tests. Otherwise, the application might not behave as it is expected.

Oracle BI Administration Tool provides a unified framework to package the functional contents and inter-layer mappings of your Oracle BI repositories and deploy them into a production environment. The Administration Tool supports multi-user development allowing developers to check out projects, make changes and then merge the changes into the master repository. Once the master repository has been successfully tested, it can be deployed into production by simply copying the repository's RPD file into the production environment.

To determine if business models within the master repository are consistent, you can use the Check Consistency command to check for compilation errors. You can also use the Compare Repositories option to compare the contents of your current master repository to a backup repository to view the differences in all objects in the Physical, Business Model and Mapping, and Presentation layers.

Once the Oracle BI repository has been successfully migrated into production, you can use the Oracle BI Catalog Manager to migrate the Presentation Catalog. The Presentation Catalog stores the content that users create in Answers and Dashboards. This content, which includes items such as folders, links, and objects (that is, requests, filters, prompts, dashboards, and so on), is stored in a directory structure of individual files. With Catalog Manager, you can migrate the entire Presentation Catalog or selectively migrate folders, dashboards and reports into the production environment. You can also archive the Presentation Catalog and maintain the permissions and timestamp associated with the archive.

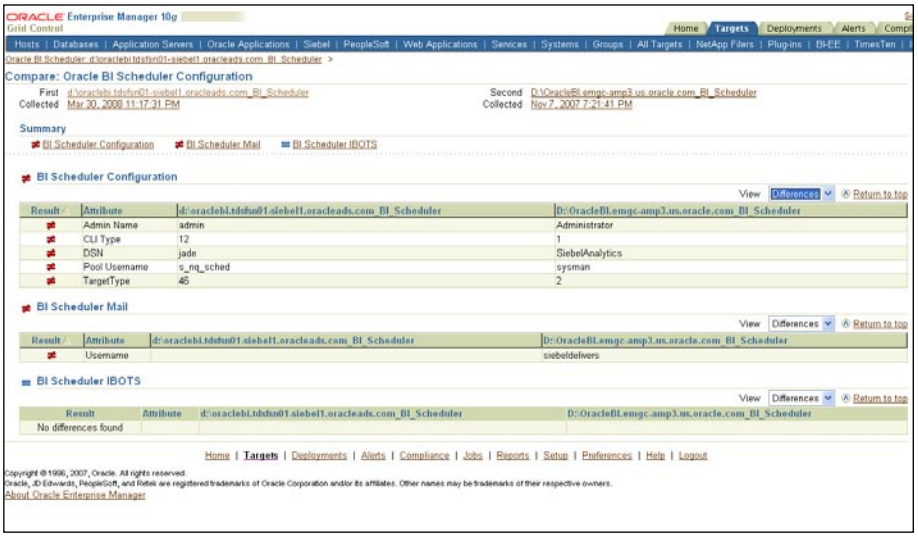

Figure 4 – Configuration Comparison

To ensure that the configurations of all critical Oracle BI components in your production environment are consistent with your staging or test environments, your administrators may also use Business Intelligence Management Pack's Configuration Comparison tool to compare the configuration in the production environment against the test or stage environments.

MANAGE

Challenge 4 – Aligning IT Priorities with Business Demands

A common dilemma in organizations is balancing business needs with IT spending. IT management constantly needs to satisfy business owners while keeping a lid on spending and increasing IT efficiency. Key questions that need to be answered include:

- What are the IT dependencies of a business process? How can we determine if business problems are caused by IT issues?
- When changes are made to the application environment, what is the potential impact on the business?
- How do we prioritize IT activities according to business needs?
- What is the impact of IT on business?

Some key performance indicators (KPI) needed to answer the question may be traditional IT system-based indicators while others may need to be derived from the business applications.

Business Intelligence Management Pack's service level management capabilities help you define service level objectives (SLO) based on business requirements, model the end-to-end service down to the system components it depends on,

monitor performance against these goals, and report on service level agreement (SLA) compliance to key stakeholders.

Service Level Objectives can be specified not only in terms of the system-level metrics for the components supporting the service, but also in terms of end user experience metrics. Business Intelligence Management Pack is unique in allowing all these classes of metrics to be used in measuring service levels. The basis for the service level management capability is a modeling facility that allows you to define a business service to be composed of component services and supporting infrastructure.

With Business Intelligence Management Pack, a service modeled with all the Oracle BI components is provided out-of-box – allowing you to view information on the availability of the service based on the underlying Oracle BI components that host the service or based on service tests that most closely match the critical functionality of your business process. Aggregated information on the status of the service and underlying components are summarized on the Oracle BI Service home page allowing you to obtain an overall perspective on the environment and monitor service level agreements (SLAs) in real-time. Additionally, the Business Intelligence Management Pack allows you to create customized reports that can be used to communicate SLA compliance to the line-of-business (LOB).

Challenge 5 – Proactive Monitoring of the Complete Application Environment

In order to deliver the application service level required by your business, your administrators need to monitor your entire application environment proactively. This requires them to monitor all the components that make up your Oracle BI environment, including the Oracle BI Server, Oracle BI Presentation Server, Oracle BI Scheduler, Oracle BI DAC Server, Oracle BI Cluster Controller, server machines, network and storage devices. The key metrics that your administrators need to monitor include component up/down status, load, resource utilization, performance, exceptions such as errors/warnings etc… Many administrators prefer to monitor the application environment in a "lights out" manner – alerting the administrators only when a problem occurs and allowing them to concentrate on their other duties when the application is functioning normally.

Business Intelligence Management Pack provides an integrated solution for proactively monitoring one or multiple Oracle BI application environments from a single console. Using the pack, your administrators may monitor the health of all critical Oracle BI components, including but not limited to the Oracle BI Server, Oracle Presentation Server and Oracle BI Scheduler. Thresholds may be defined against server and component statistics such as CPU utilization, the current number of sessions or active logins, and up/down status of servers and components.

Log files that are associated with the various Oracle BI components (e.g. NQServer.log) can be monitored by specifying error codes, or by defining regular expressions that match the log messages. In addition to relying on system performance metrics and error logs, you may use Business Intelligence Management Pack's Service Tests to record synthetic web transactions that include a combination of one or more navigation paths within the application to be used as the criteria for determining the service's availability. For example, the Interactive Dashboards requires that a user successfully log on to the Oracle BI Dashboards and access one of the available dashboards for the service to be considered available. Enterprise Manager uses these logical tasks or 'transactions' to define the availability of the Oracle BI application. These synthetic web transactions are recorded, and the stored transaction or 'service test' can be launched at a user-defined interval from strategic locations across the user-base.

While monitoring the various statistics, you may rely on Business Intelligence Management Pack's built-in event management capabilities. Notification methods could be defined to send email, trigger SNMP traps to forward alerts to third party management tools, or to kick off custom scripts. Notification may be defined according to a schedule, so that different administrators who are on duty at different times would get the alerts during their shifts.

To reduce the possibility of false alarms, Business Intelligence Management Pack uses several tactics to throttle the rising number of alerts. First, you may define an alert to go off only if a certain condition persists for a certain number of sampling intervals. This approach prevents a singular rogue event such as a spike from triggering un-necessary alert. Second, you may define a "Notification Rule" to stop sending alerts after a certain number of attempts so that you are not alerted repeatedly if a condition persists and you already know about it. Furthermore, you may define threshold alerts against metric snapshots so that the alerts are based on deviation from observed behavior of the components.

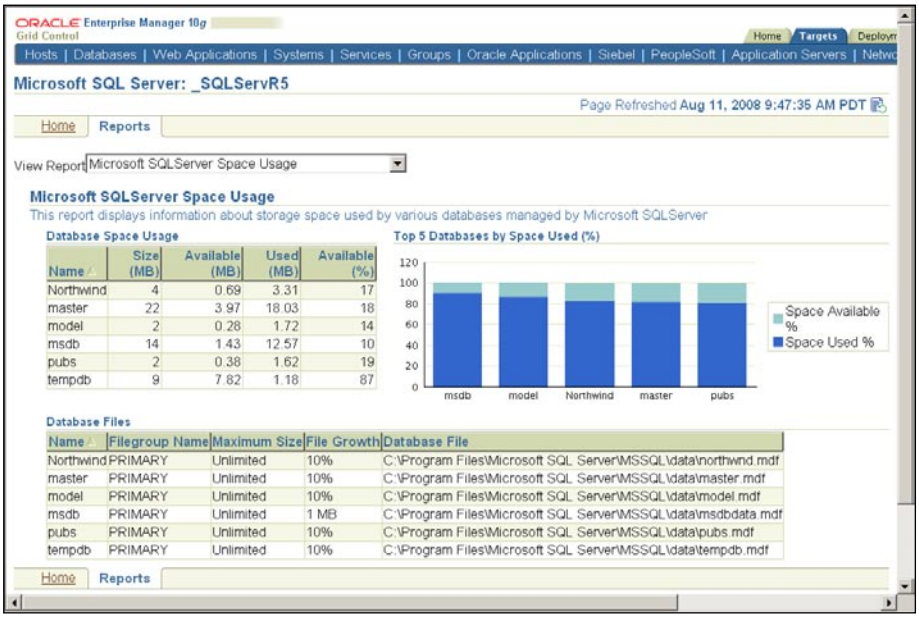

Figure 5 – Microsoft SQL Server Monitoring

Besides managing the Oracle BI components, Oracle Enterprise Manager provides a range of management packs and system monitoring plug-in's to cover the infrastructure components that support the Oracle BI application. You may mix and match these additional packs and plug-in's to complement the core application monitoring provided by the Business Intelligence Management Pack.

If you use Oracle Database, use Oracle Database Diagnostic Pack for deep monitoring of database's functions such as tablespace, buffer pool, memory, CPU and I/O. If you use Microsoft SQL Server or IBM DB2, use System Monitoring Plug-in for Non-Oracle Database to perform similar type of monitoring.

Lastly, to monitor infrastructure technologies such as F5 Big-IP Load Balancer, EMC Storage Arrays and NetApp Filers, Oracle offers System Monitoring Plug-in for Network Devices and System Monitoring Plug-in for Storage Devices. Management data collected through these plug-in's as well as from database and middleware packs can be combined with system and end user experience data collected from the Business Intelligence Management Pack on the same Oracle Enterprise Manager instance to give Oracle BI administrators a holistic, top-down and end-to-end view of the entire Oracle BI environment and the extended infrastructure.

Challenge 6 – Monitoring End User Experience

No matter how well tuned the application is during testing, production performance problems may still occur because of unforeseen usage or interdependencies with other components of the IT infrastructure. Studies indicate that most application performance issues are still reported first by

application end users before IT administrators find out about them. Unfortunately, this delay means that business operations have been impacted.

Your administrators need to proactively identify the end user issues before the end user community is impacted by a performance problem. The first step in guaranteeing end-user satisfaction is to learn about the end-user performance experience. Some of the questions that your IT staff needs to answer related to the end-user performance experience are:

- Are end-users satisfied with the application performance?
- Are end users able to complete key monitoring and analysis tasks timely and successfully?
- Is the application performance problem impacting all the users or limited to a geographical region?

There are two ways to monitor your end users' experience. The first method is by using Business Intelligence Management Pack's synthetic service test. These tests are designed to simulate key end user activities such as logging in and navigating through critical pages of Oracle BI Dashboards. The tests are run via "beacons" from locations within your network to actively measure the performance and availability of your Oracle BI application from an end user perspective. Because these tests are played back automatically via beacons and do not rely on actual end users being present, they can be used for accurate performance trending analysis and for proactive monitoring.

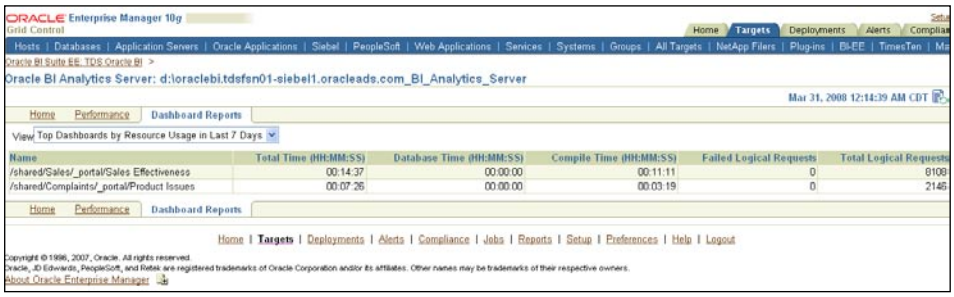

Figure 6 – Dashboard Reports

In addition to synthetic service tests, the Business Intelligence Management Pack allows you to view detailed information about dashboard usage through "Dashboard Reports" that summarize key query statistics. Dashboard Reports enable you to maximize the value of your application by delivering insight into real end user experiences. They provide key statistics for usage trending analysis – allowing you to view detailed information about the top dashboards by resource usage as well as top users by resource usage. In addition to usage trending, these reports can help you diagnose performance problems by showing you a complete listing of failed dashboards along with the error code/message and SQL statements associated with the failed queries and a breakdown of dashboard response time based on database time, compile time and overall time for end-user queries. With

such detailed analysis capabilities you can reduce support costs by lowering call center volumes, accelerate problem resolution of poorly performing dashboards, and help businesses adapt to changing needs by providing insight into business trends and user preferences. Dashboard Reports integrate performance analysis and usage analysis into a single offering, enabling business and IT stakeholders to develop a shared understanding into their application user experience.

Complementing Business Intelligence Management Pack's Dashboard Reports is Oracle BI's customizable Usage Tracking dashboard. Like the Dashboard Reports, the Usage Tracking tool provides you with detailed statistics on queries per day, user or subject area – allowing to you perform detailed usage trending analysis. Additionally, Usage Tracking provides a detailed view of the longest running queries along with the logical SQL statements associated with them and a breakdown of the queries' response time based on database time, compile time and overall time for query execution – allowing you to diagnose performance problems and proactively monitor the availability of dashboards. You can monitor the real-user experience, set Key Performance Indicators (KPIs), and use Oracle BI Delivers to trigger alert notifications for incidents that violate them. Reports generated by Usage Tracking can be analyzed by line of business (LOB) owners who review and optimize the business performance, IT managers who are responsible for availability and performance of an application and IT operators who run day-to-day operations such as monitoring and diagnosing application performance.

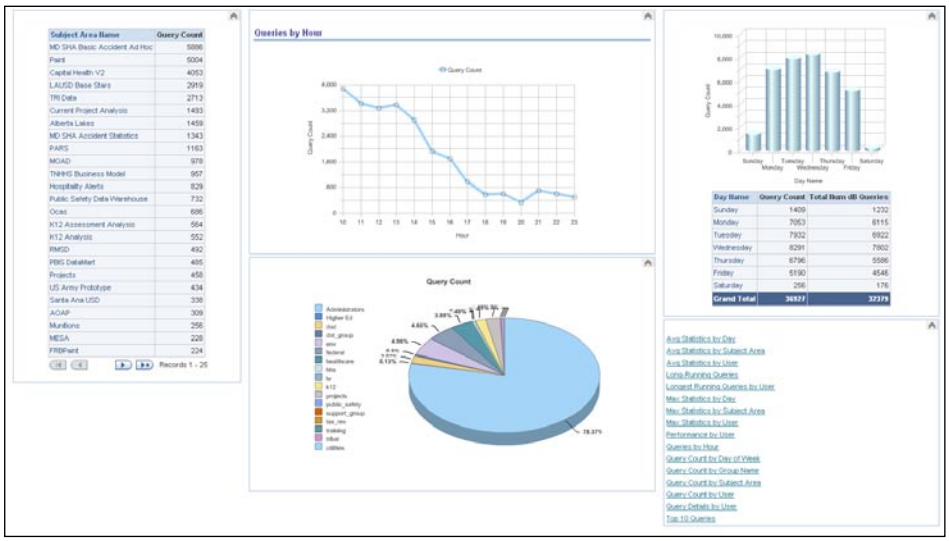

Figure 7 – Oracle BI Usage Tracking Dashboard

Challenge 7 – Diagnosing Production Problems Quickly

When problems are detected, you need to fix them quickly in order to minimize impacts to your end users. Diagnosing problems can be a very tedious task often involving guesswork because of difficulties in accessing pertinent diagnostic information and because of the large number of components in an application environment. Performing diagnostics can be a resource intensive task, often requiring several people involved in managing the application environment – including application administrators, database administrators, OS administrators and network administrators. If every problem needs the attention of all the administrators, then the task of diagnosing problems will be very expensive and time consuming to perform.

Figure 8 – Failed Dashboards (Enterprise Manager's Dashboard Reports) and Longest Running Queries (Oracle BI Usage Tracking)

Business Intelligence Management Pack simplifies diagnostics by presenting relevant diagnostic information and providing tools to analyze information from the different parts of the application environment. The pack simplifies initial problem triage so that the task can be done quickly and with fewer people. It also provides deep diagnostic capabilities to identify problems that are caused by a specific Oracle BI component.

The starting point of a diagnostic effort is examining the Oracle BI Service. Oracle BI Service home page shows the availability of the service based on the underlying Oracle BI components that host the service or based on service tests that most closely match the critical functionality of your business process. Aggregated information on the status of the components and service tests and critical alerts are summarized on this page allowing you to obtain an overall perspective on the environment before you proceed to deeper investigation.

From the service home page, you may view the availability history of the service and the current service levels. Then, begin the triage process by examining service test data to see whether the problem is network location specific. If it is network specific, you may then engage the network administrator to resolve the problem. If not, you may want to bring up metric history information of the various servers and components to see if the problem is due to over utilization or lack of resource.

Business Intelligence Management Pack automatically saves all the metrics that are collected from your application and its environment, so you can go back to a point in time to examine the state of the system when the problem occurred.

For problems that are more intermittent or are tied to specific queries or users, use Business Intelligence Management Pack's Dashboard Reports as well as Oracle BI's Usage Tracking to obtain detailed information on real end-user performance. You may retrieve detailed statistics on queries per day, user or subject area, identify the failed dashboards or longest running queries and view the error code/message and logical SQL statements associated with them, and drill down to a particular query to view a breakdown of response time based on database time, compile time and overall time for query execution.

For problems that may be system configuration related, use Business Intelligence Management Pack's configuration analysis tool to locate the cause. You may query against Oracle Enterprise Manager's configuration management database (CMDB) to find out whether any Oracle BI component parameter has changed. You may also compare configuration settings across different Oracle BI components and between servers to find out why there are discrepancies in behavior amongst different environments.

If you use Oracle Database, you may use Oracle Database Diagnostic Pack to carry out deep database level diagnostics. The pack includes a self-diagnostic engine built right into Oracle Database kernel, called Automatic Database Diagnostic Monitoring (ADDM). ADDM periodically examines the state of the database, automatically identifies potential database performance bottlenecks, and recommends corrective actions. Oracle Database Diagnostic Pack presents ADDM's findings and recommendations in a convenient and intuitive fashion, and guides administrators step-by-step to quickly resolve performance problems by implementing ADDM's recommendations. ADDM starts its analysis by focusing on the activities that the database is spending most time on and then drills down through a sophisticated problem classification tree to determine the root cause of problems. The problem classification tree used by ADDM encapsulates decades of performance tuning experience of Oracle's own performance experts and it has been specifically designed to accurately diagnose the most frequently seen problems, such as CPU and I/O bottlenecks, poor connection management, undersized memory, resource intensive SQL statements, lock contention, etc… Each ADDM finding has an associated impact and benefit measure to enable prioritized handling of the most critical issues. To better understand the impact of the findings over time, each finding has a descriptive name that facilitates search, a link to number of previous occurrences of the finding in the last 24 hours, and affected instances.

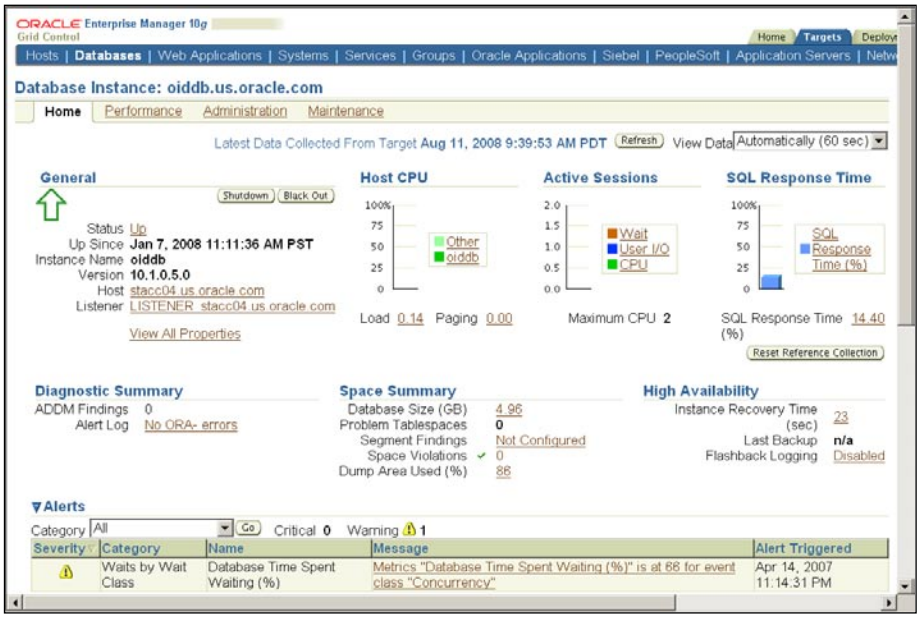

Figure 9 – Oracle Database Diagnostics

OPTIMIZE

Challenge 8 – Making Fact-Based Optimization Decisions

Optimizing an application is a time consuming task often surrounded by myths and legends, few of them based on facts. Like diagnostics, application optimization is very hard to do unless you have access to the right information. Business Intelligence Management Pack, along with Oracle Database Tuning Pack, provides the information that you need to make fact-based optimization decisions.

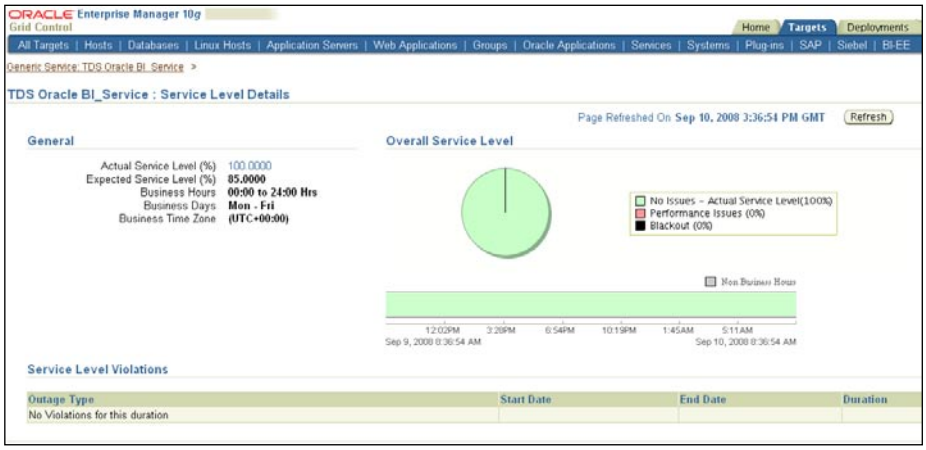

Figure 10 – Service Level Report

The starting point of the optimization process is Business Intelligence Management Pack's service level management reports. Based on service level indicators collected from the application over a period, these reports indicate whether the Oracle BI application provided the performance and availability needed to support critical business operations. These reports are further complemented by capacity utilization reports of the underlying components, and by Dashboard Reports that show the usage patterns of the application.

With this information, you may then decide whether you need to invest in further optimization, which may include tasks such as adjusting the functional configuration of your application, applying patches from Oracle, tuning Oracle BI Server and other Oracle BI components, or tuning the database.

To optimize Oracle BI components, you need to consider several statistics collected during run-time. These statistics are gathered by the Business Intelligence Management Pack and are stored in Oracle Enterprise Manager's repository. You may retrieve them in reports that show the graph of these metrics over time to understand how the application behaves or compare the metrics across different servers to see if your servers are load balanced properly. Using this information, you may work with your application developers to modify your application's functional configurations if they prove to be too resource intensive, or use Oracle BI Administration Tool to make further changes to the Physical, Business Model and Mapping and Presentation layers. You may obtain detailed information about Oracle BI Server's cache and database query performance through performance charts that can help you identify problems and optimization opportunities. Using this information, you may decide to tune the database queries or tweak the cache configuration to best fit your application's load profile. You can also monitor the performance of the Oracle BI DAC Server and view a summary of completed runs with information about the duration, total steps, completed steps, stopped/failed steps and running steps for all completed runs. Based on the performance information retrieved for the Oracle BI DAC Server, you may decide to tweak the queued ETL tasks or fix the failed ones to optimize performance.

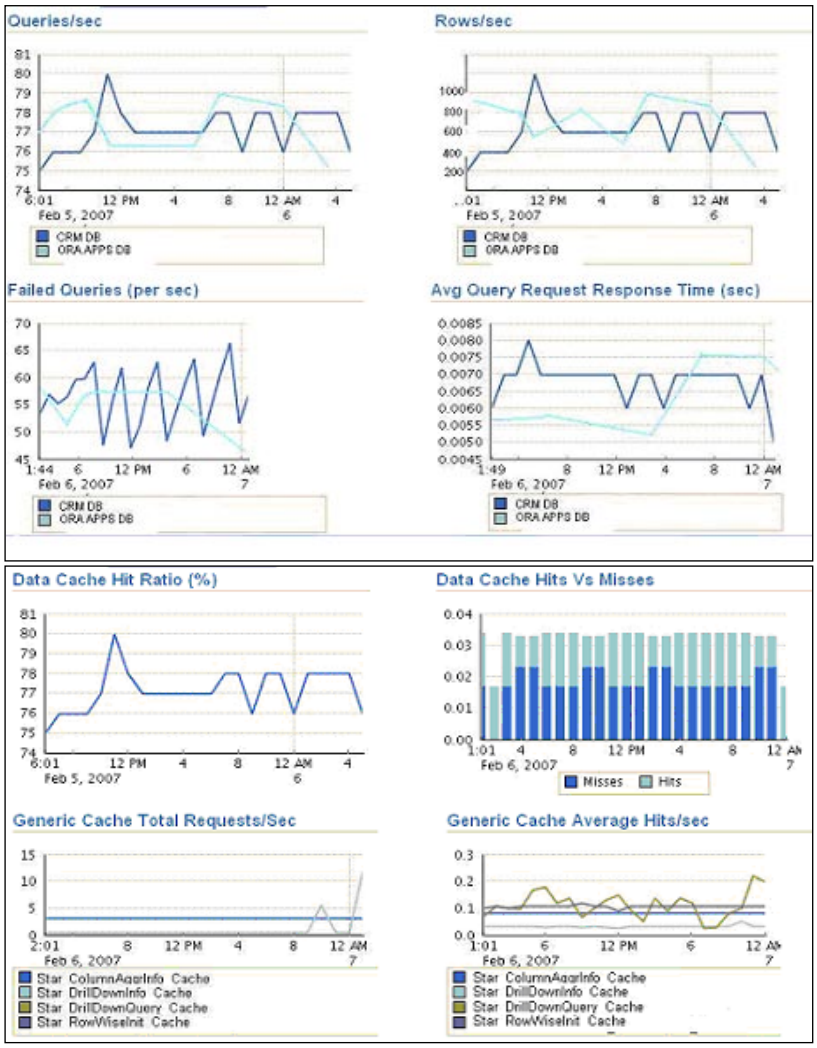

Figure 11 – Oracle BI Server Performance Charts in Enterprise Manager

For tuning the database, use Oracle Database Tuning Pack if Oracle database is one of the data sources in your application environment. Since Oracle BI's queries are based on SQL statements, you may tune the long running queries and make significant changes to database performance using indices, database system component tuning, and SQL profiling to tune the execution plans.

Manual SQL tuning is a complex process that presents many challenges. It requires expertise in several areas, is very time consuming, and requires an intimate knowledge of the schema structures and the data usage model of the application. All these factors make manual SQL tuning a challenging and resource intensive task that is ultimately very expensive for businesses.

SQL Tuning Advisor is Oracle's answer to all the pitfalls and challenges of manual SQL tuning. It automates the SQL tuning process by comprehensively exploring all the possible ways of tuning an SQL statement. The analysis and tuning is

performed by the database engine's significantly enhanced query optimizer. Four types of analysis are performed by the SQL Tuning Advisor:

- Statistics Analysis: The query optimizer needs up-to-date object statistics to generate good execution plans. In this analysis objects with stale or missing statistics are identified and appropriate recommendations are made to remedy the problem.
- SQL Profiling: This feature, introduced in Oracle Database 10g, revolutionizes the approach to SQL tuning. Traditional SQL tuning involves manual manipulation of application code using optimizer hints. SQL Profiling eliminates the need for this manual process and tunes the SQL statements without requiring any change to the application code. This ability to tune SQL without changing the application code also helps solve the problem of tuning packaged applications. Packaged application users now no longer need to log a bug with the application vendor and wait for several weeks or months to obtain a code fix for tuning the statement. With SQL profiling the tuning process is automatic and immediate.
- Access Path Analysis: Indexes can tremendously enhance performance of an SQL statement by reducing the need for full table scans. Effective indexing is, therefore, a common tuning technique. In this analysis, new indexes that can significantly enhance query performance are identified and recommended.
- SQL Structure Analysis: Problems with the structure of SQL statements can lead to poor performance. These could be syntactic, semantic, or design problems with the statement. In this analysis relevant suggestions are made to restructure selected SQL statements for improved performance.

The output of this analysis is in the form of recommendations, along with a rationale for each recommendation and its expected performance benefit. The recommendation relates to collection of statistics on objects, creation of new indexes, restructuring of the SQL statement, or creation of a SQL Profile. A user can choose to accept the recommendation to complete the tuning of the SQL statements.

Oracle Database Tuning Pack 11g also provides the ability to reorganize objects. Managing the space usage of your tablespaces efficiently by removing wasted space is not only a good space management practice but it also enhances performance by reducing unnecessary disk I/Os. Reorganization is used for:

- Rebuilding indexes and tables that are fragmented
- Relocating objects to another tablespace
- Recreating objects with optimal storage attributes

Oracle Database Tuning Pack 11g provides a wizard that can perform reorganization at schema and tablespace levels, and gives the option for both online and offline reorganization. The wizard also provides an impact analysis report as well as a review script that contains the exact operations that will be performed. This helps users to precisely understand the implications of the operation before implementing it.

Challenge 9 – Masking Production Data to Test Optimization Changes

In carrying out optimization tasks, it is often beneficial to use real production data in order to assess optimization impacts accurately. However, using real data may raise information security and privacy concerns. Safeguarding production data and preventing leaks of confidential or sensitive information to non-production users has become a corporate imperative for all organizations – thanks to the slew of global regulations governing data privacy. The Sarbanes Oxley Act of 2002 in the US or the Financial Instruments Exchange Law (FIEL) of Japan (also called J-SOX) provides enhanced standards on internal controls for corporate information. The Health Insurance Portability and Accountability Act (HIPAA) of 1996 in the US or the European Union's Data Protection Directive are a part of the global laws governing the privacy of personal data related to individuals. Even credit card payment processors have adopted Payment Card Industry (PCI) standards regarding the use and sharing of credit card information.

If you use Oracle Database, you may use Oracle Database Masking Pack to overcome this problem. Data Masking Pack helps you obfuscate sensitive data selectively, preserving the realism of test data set while protecting sensitive information at the same time. Data masking rules are highly configurable, and you may control the algorithm for masking the data in order to preserve relevant data semantics that are useful to creating realistic test scenarios.

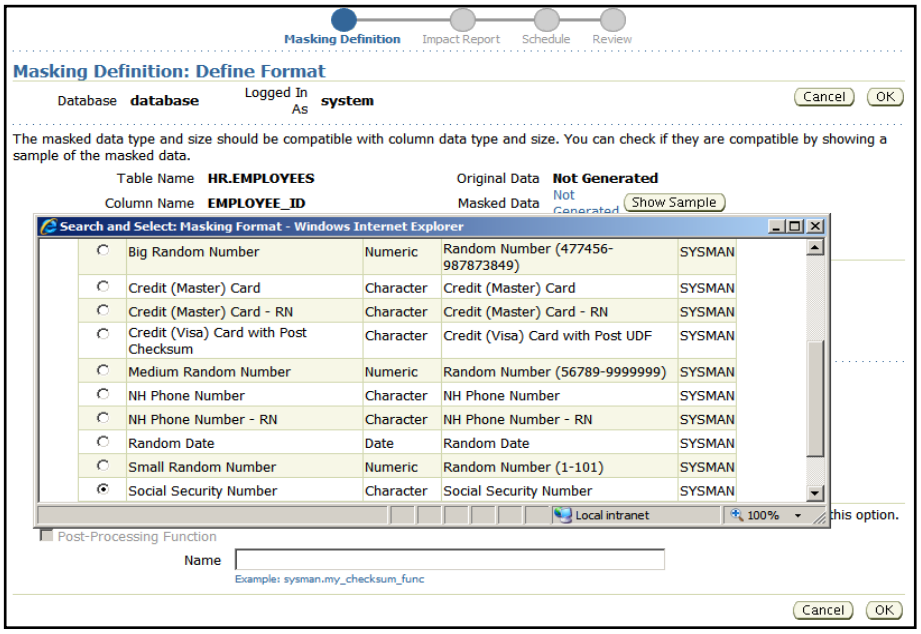

Figure 12 – Data Masking Definition

Data Masking Pack uses an irreversible process to replace sensitive data with realistic-looking but scrubbed data based on masking rules and ensures that the original data cannot be retrieved, recovered nor restored. The Data Masking Pack helps maintain the integrity of the application while masking data.

The Data Masking Pack provides out-of-the-box mask primitives for various types of data, such as random numbers, random digits, random dates, and constants. Organizations can also use other built-in masking routines, such as shuffling, which shuffle the value in a column across different rows. This is useful when the range of values in a column is often not known and the shuffling of values in the same table provides a sufficient degree of privacy protection. For organizations requiring that the masked value be realistic but not based on the original data, the Data Masking Pack can replace the original data, such as names and addresses, with data containing fictitious names and addresses derived from external data sources.

Organization with specialized masking requirements can also add user-defined mask formats to the collection of the mask formats. These user-defined formats, defined using PL/SQL, provide an unlimited degree of flexibility in generating mask format appropriate to the organizations' business or industry segment. Information security administrators can then create complex and composite masks based on combinations of various masking formats - both standard and user-defined. For example, a mask for common credit card numbers can be defined as unique sixteen digit numbers that begin with 4 or 5, which are then verified for check-sums to be compliant with PCI standards.

Data Masking Pack uses a highly efficient and robust mechanism to create masked data. The Data Masking perform bulk operations to rapidly replace the table containing sensitive data with an identical table containing masked data while retaining the original database constraints, referential integrity and associated access structures, such as INDEXes, PARTITIONs, and access permissions, such as GRANTs. Unlike masking processes that are traditionally slow because they perform table updates, the Data Masking Pack takes advantage of the built-in optimizations in the database to disable database logging and run in parallel to quickly create a masked replacement for the original table. The original table containing sensitive data is dropped from the database completely and is no longer accessible.

Challenge 10 – Managing Configuration Changes and Achieving Compliance

As optimization changes are made against server and component parameters, it is important to be able to keep track of the changes for diagnostic and compliance purpose. Traditionally, people have relied on manual methods of maintaining change history, often keeping the information in spreadsheets. The manual approach is very tedious, and is often inaccurate. Sometimes, changes are made temporarily for testing purposes, but end up becoming permanent as the person who made the change forgets to back it out, and this causes what is known as configuration drifts that can impact application performance and availability over time.

Business Intelligence Management Pack's configuration management capabilities automate configuration management activities. The BI Management Pack provides a view of configuration items and their dependencies within and across each other. You can manage configuration drift through scheduled comparison with "gold configuration" baselines. Administrators can track, analyze and report on configurations while capturing and storing configuration data that is used for the automation of the entire configuration management process.

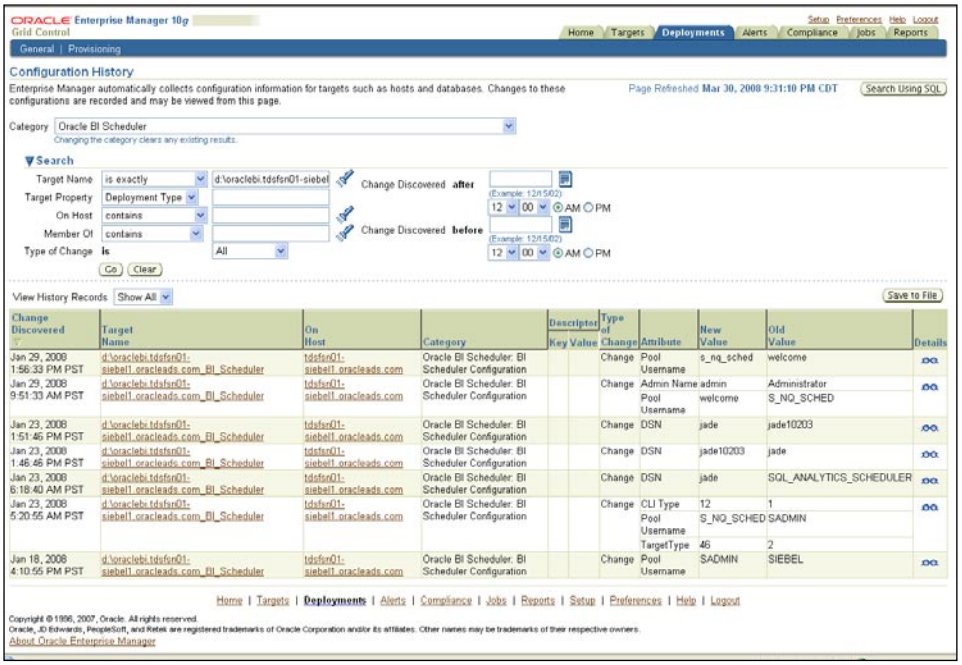

Figure 13 – Configuration Change History

Business Intelligence Management Pack's configuration management support is the foundation to all Service Support processes, enabling effective incident management and service-level management. It enables faster mean-time-to-repair through root cause analysis by isolating and correlating problems to the exact infrastructure or Oracle BI component that is causing failure and by auditing change history for all targets and parameters. It reduces the risks involved in rolling out changes to production environments by identifying the impact of changes on deployed applications and users.

Here are some of the key features of configuration management capabilities:

- Automated discovery of Oracle BI components such as Oracle BI Server, Oracle BI Presentation Server, Oracle BI Scheduler, Oracle BI Cluster Controller, Oracle BI DAC Server, and their association with the underlying host and operating system
- View and analyze Oracle BI component configurations
- Out-of-box and customizable configuration searches
- Compare configurations
- Historical change tracking
- Configuration reports

SUMMARY

Through Oracle Business Intelligence Management Pack and other Oracle management and testing products, you can start centralizing the management of your Oracle Business Intelligence Enterprise Edition application on Oracle Enterprise Manager. These products complement bundled application tools, such as Oracle BI Client Tools (including Oracle BI Administration Tool and Oracle BI Scheduler Job Manager), Oracle BI Catalog Manager and Oracle BI Data-Warehouse Administration Console (DAC), which provide tactical administrative functions. The management packs leverage Oracle Enterprise Manager's top-down application management capabilities to facilitate proactive management and ITIL best practices implementation that cover the complete application lifecycle. You can use Oracle Enterprise Manager as the unified console to manage your entire application infrastructure, including all your application instances, the SOA-based fabric that you use to connect your applications, both Oracle and non-Oracle databases and middleware, as well as your servers, storage and network devices, all of which impact your application's performance and availability. Through these tools, you can achieve a Superior Ownership Experience in manageability and quality for your Oracle BI application, and deliver the application service level required to meet your business needs.

Achieving a Superior Ownership Experience for Oracle BI EE
August 2008
Authors: Amjad Afanah
Contributing Authors: Chung Wu, Bob Ertl, Joe Fernandes

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com