



**N•CYCLES**

software solutions

# **WebDBA White Paper**

Proactive Monitoring to Prevent Common Database  
Failure Points

March 2001

65 Germantown Court  
Suite 205  
Cordova, TN 38125  
Phone 901-756-2705  
Fax 901-758-1597

1616 West Gate Circle  
Nashville, TN 37027  
Phone 615-403-3168  
Fax 630-214-4909

[www.ncycles.com](http://www.ncycles.com)

## Table of Contents

Table of Contents .....	1
WebDBA Overview .....	2
Architecture .....	2
Database Monitoring Functions.....	2
Overview .....	2
Tablespace Monitoring .....	2
Problem Segment Monitoring .....	3
Table Segment Monitoring .....	3
Index Segment Monitoring .....	3
Rollback Segment Monitoring .....	4
Database Lock Monitoring .....	4
Database Session Monitoring .....	4
Database File Monitoring .....	4
Database User Monitoring .....	5
Database SQL Monitoring.....	5
Future Enhancements .....	5
Data Auditing and Routing Functions .....	5
Overview .....	5
Auditing .....	5
Routing .....	6

---

## **WebDBA Overview**

The WebDBA application was designed as both a monitoring and a maintenance tool for managing several Oracle databases simultaneously, with a key focus on automation. This tool provides constant, proactive monitoring of many databases with little human intervention. WebDBA allows companies to run an Oracle environment with a “hands-off” mentality. However, when issues do arise, WebDBA delivers with its robust Alerting and Notification functionality.

## **Architecture**

WebDBA was designed as a web-based application in order to make accessibility and portability much better. Unlike most administration tools where the software must be installed on the machine doing the administration, WebDBA allows the DBA to access the application from any web browser with connectivity back to the server running WebDBA. This can span from intranet to internet access.

Based on the three tier model, WebDBA runs as a Java Servlet on any web server supporting Java Servlets. Connectivity to the database is provided via JDBC drivers. Future enhancements to the system will include a Java application running as a daemon on the server which will gather statistical information on a timely basis, along with data agents which will run on each database server, gathering information not available from the database itself. The culmination of these provides the majority of functionality required for database monitoring.

## **Database Monitoring Functions**

### **Overview**

WebDBA monitoring functions allow a quick method of determining the state of critical portions of the Oracle database. By consistently monitoring the activity and state of objects in the database, data integrity and up time can be maximized. WebDBA allows a database administrator to quickly examine tablespaces and storage, database objects whose storage segments have any problems. It allows the administrator to locate and when necessary eliminate database session which are holding locks on tables. Functionality is also provided for managing database user accounts, from creating, to modifying tablespace association.

### **Tablespace Monitoring**

One of the most important aspects a DBA needs to keep track of in an Oracle database is the amount of data in the Oracle tablespaces. A tablespace is an area of disk consisting of one or more disk files whose size is allocated prior to bringing the tablespace online (or in the addition of adding a datafile, that new file is utilized by the existing tablespace). Inside the tablespaces, Oracle objects, such as tables, indexes, rollback segments, etc are stored. Due to the fixed size of the tablespaces, they can fill up as data is added to the database. When the amount of data in the tablespace exceeds (or attempts to exceed)

the maximum storage capacity of the tablespace, an error is generated and NO DATA gets inserted. This scenario can be detrimental to any database, especially production databases. To avoid this overflow, it is necessary to monitor on a regular basis the amount of data in each of the tablespaces which make up an Oracle database.

WebDBA allows the monitoring of these tablespaces by providing the user a quick view of all tablespaces in the database, along with their total size, bytes in use, bytes free, and a percentage used. To make the monitoring even easier, all tablespaces that fall within 80-89 percent of capacity are highlighted in green and any that are 90 percent or more of capacity are highlighted in red and flash. This highlighting helps the database administrator quickly locate potential failure points and address them.

In addition, there is graphing capability which allows the DBA to get a visual representation of their database.

## Problem Segment Monitoring

Along the same lines as tablespace monitoring, segment monitoring is equally crucial. Just as tablespaces are pre-allocated, segments are allocated with an initial size, which defines the first amount of space to use when the segment is created, a next size, which defines the size of the second extent to be allocated, a maximum number of extents to allocate to and a percent increase which defines extents following the next extent. When inserting data into the segments, should the max extent value be reached, NO DATA will get inserted. Again, this scenario is detrimental to the database, keeping data from being stored.

A quick screen in WebDBA lets the database administrator see any segment which is within 20 extents of it's max extents. Should a segment show up in this listing, the DBA should determine whether to increase the max extents for the segment, or coalesce the object and reallocate extents. In addition, the problem segment screen displays segment information: owner, name, type, bytes, extents, max extents, extents available, next extent and percent increase. This information is equally crucial in determining the next step for the DBA.

Future enhancements in this section will provide the ability to make modification to the storage parameters of this object.

## Table Segment Monitoring

The table segment monitoring section allows the DBA to view only the segments of type table. In the same way the problem segment monitor displays information about the segments, table segments are shown with any segment within 20 extents of it's max extent (i.e. it's available segments < 20) is highlighted to stand out.

Future enhancements could include the ability to modify storage parameters for a table.

## Index Segment Monitoring

Index segment monitoring again follows the same format as problem segments and table segments, only display segments of type index. Any index segment coming within the 20 segments of its max is highlighted.

Future enhancements could include the ability to modify storage parameters for an index.

## Rollback Segment Monitoring

Unlike other segment types, rollback segments may often have most or all of its extents allocated. The rollback segment monitoring section shows only segments of type rollback. Again segments exceeding the 20 extent threshold from max extents are highlighted.

Future enhancements will include the ability to modify the storage parameters of rollback segments, bring rollbacks on or off line, and create new rollback segments.

## Database Lock Monitoring

With any shared database, the possibility of locks on objects in the database is normal procedure. However, on occasion, an application will obtain a lock on a row and for whatever reason, never release it. When this happens it is extremely beneficial to be able to identify the offending user and application. WebDBA allows for this with the database lock monitoring screen. This screen displays all locked objects in the currently selected database. Of the information show, a DBA can usually determine the user and application causing the lock. Should it be necessary to terminate the session holding the lock, this can be done from this screen by selecting the kill session link.

Future enhancements could include quick linking to the users information and the ability to terminate the session holding the lock.

## Database Session Monitoring

The session monitoring function of WebDBA allows a DBA to view all current sessions in the database. Information provided on each session include the session id, status, schema, operating system user, process, machine logged on from, program which has the session open, and the login time of the session. This information can be valuable for determining number concurrent users on the database, and which applications are utilizing the most connections.

Future enhancements include quick linking to the user associated with this session as well as the ability to terminate the session.

## Database File Monitoring

While database files are not very dynamic, it is helpful to have a method to examine file associations to tablespaces and physical locations of those datafiles. In order to accommodate this, WebDBA displays tablespaces with their associated data files along with size, status, and autoextend information along with them.

Future enhancements could include the ability to examine physical disk space on the device where the tablespaces data files are held.

## Database User Monitoring

The user monitoring function gives a listing of all users defined in the Oracle database. It provides userid, name, account status, expiration date, default tablespace, temporary tablespace, creation date, and profile information for each user. In addition, users can be added and modified from this screen.

Future enhancements could include the ability to view and modify all objects owned by the user.

## Database SQL Monitoring

One of the most crucial aspects of database application programming is tuning the SQL statements to achieve maximum performance. While it would be extremely difficult, if not impossible, for a DBA to go through the code of every application running against the database and tune it, WebDBA makes it easier by providing a list of all SQL statements in the Oracle SQL area which have been executed, along with the user executing them and the number of times they have been executed. With this list, the DBA can find statements which are executed most often and ensure that they are properly tuned.

Future enhancements could include the computation of cost (for cost based optimization) of each of the statements in the SQL area, along with possible scenarios for SQL tuning.

## Future Enhancements

While the current version of WebDBA supports the functionality discussed above, it will be enhanced extensively by providing supporting maintenance functions for all the monitoring functions. This would include the ability to manipulate most of the objects which are monitored in the above functions. Also, statistical information will be gathered over time to help administrators understand the running growth of the database.

## **Data Auditing and Routing Functions**

### Overview

The WebDBA application has an auditing and routing feature which allows the database administrator to create and populate auditing tables on any table of any schema which the administrator has access to. WebDBA can create and remove auditing procedures on the fly, as well as temporarily disabling auditing. Further functionality allows the administrator to determine which elements of a table constitute a change, and upon that criteria being met, execute a plug-in module to the WebDBA tool to create any message type desired, then have that message type routed via FTP, E-Mail or IBM MQSeries to a defined destination.

### Auditing

While auditing generally falls in the schema of the application developer, WebDBA allows the database administrator to create and perform table level auditing on any Oracle database set up in the system. The administrator simply selects one of the database's and schema's which he/she wishes to perform auditing on. From this point a table is selected to audit, then WebDBA installs an auditing table and the trigger mechanism

which populates that audit table. Every field in the original table is duplicated in the audit table along with the name of the user making the modification, a timestamp of the modification, and a unique sequence number for tracking.

## Routing

Should the administrator decide that they wish to notify a user when a field changes in the database, an audit can be set up on the table holding that field. Then WebDBA allows them to define that field as a routed key and determines what message type to send, across what transport, and to where. Once the route has been set up, any time that field changes in the database, a message will be sent to the designated destination.