

# nocoug

JOURNAL

Official Publication of the Northern California Oracle Users Group

VOL. 17, No. 3 · AUGUST 2003

\$15



## NoCOUG Gets Two Awards!

NoCOUG was recognized by the International Oracle Users Group during the IOUG Live! 2003 Conference in April/May. Our acting IOUG Representative at the conference, Colette Lamm, took home two awards for NoCOUG

at the GAP Networking Luncheon. Many user groups were nominated, and Colette accepted awards of excellence for our website and journal. Many thanks to Vadim Barilko, who has been our webmaster for the past year and to Lisa Loper, our Journal editor for the past two years! ▲

## NoCOUG Board Position Available – Marketing Director

**Join our exciting NoCOUG team today!**

The NoCOUG Board is seeking a Marketing Director who will be responsible for establishing and executing strategic marketing plans to promote NoCOUG and extend our reach to new members. The person in this position will work closely with the Membership Coordinator and NoCOUG board members to identify marketing channel opportunities that communicate the benefits of NoCOUG to Oracle professionals throughout the greater Bay Area. This is a volunteer position.

To express your interest or for any questions, please contact [board@nocoug.org](mailto:board@nocoug.org). We are also seeking an Assistant Journal Editor. (See page 2 for details.) ▲



### TABLE OF CONTENTS

NoCoug Gets Two Awards.....	1
NoCoug Board Position Available.....	1
Seeking an Assistant Journal Editor .....	2
NoCOUG Board .....	2
Publication and Submission Format.....	2
Advertising Rates.....	2
President's Message .....	3
Board Member Spotlight.....	5
Oracle Temporary Tables, Temporary Segments and Temporary Tablespaces.....	6
Other Users Groups.....	10
Sponsorship Appreciation.....	11
Treasurer's Report.....	11
Killing the Oracle DBMS Job .....	13
Third Party Grants.....	16
Too Clever for Words: Oracle9i Log Miner...17	
The Database Gets Better, but the Metrics Look Worse .....	22
NoCOUG Summer Conference Schedule.....	26

### –ADVERTISERS–

LECCOTECH.....	4
Mission Critical 24/7 .....	4
BMC Software .....	12
Database Specialists .....	14
Embarcadero Technologies.....	14
Quest Software.....	25



Lisa Loper

# Seeking an Assistant Journal Editor

NoCOUG exists for the education and representation of the users of Oracle Corporation's database and tools software—

those who contribute to the NoCOUG Journal help achieve that vision.

If you would like to get involved, now is a great time. I am seeking an Assistant Journal Editor. There are also a number of short and long-term assignments that you can take on, depending on your time and interests.

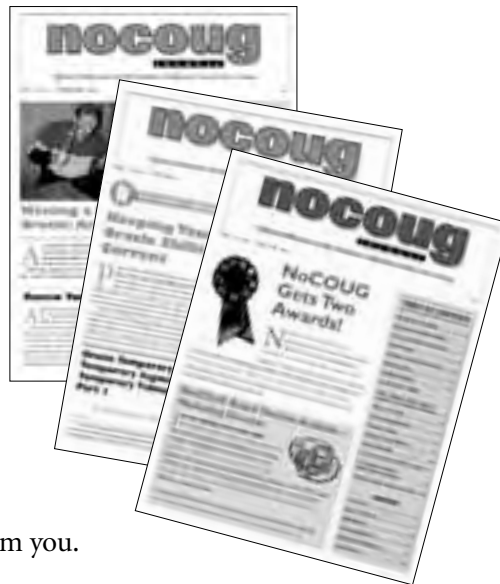
They include:

- Soliciting and reviewing articles
- Coordinating production of the *Journal*
- Writing brief tech tips
- Writing in-depth articles

Just get in touch with me at [journal@nocoug.org](mailto:journal@nocoug.org) or call (415) 344-0500 ext. 42.

I look forward to hearing from you.

—Lisa Loper  
*Journal Editor*



## Publication and Submission Format

The *NoCOUG Journal* is published four times a year by the Northern California Oracle Users Group approximately two weeks prior to the quarterly regional meetings. Please send your questions, feedback, and submissions to: Lisa Loper, NoCOUG Journal Editor, at [journal@nocoug.org](mailto:journal@nocoug.org).

The submission deadline for the upcoming November issue is October 1, 2003. Article submissions should be made in electronic format via email if possible. Word documents are preferred.

*NoCOUG does not warrant the NoCOUG Journal to be error-free.*

Copyright ©2003 by the Northern California Oracle Users Group. Permission to reproduce articles from this publication, in whole or in part, is given to other computer user groups for nonprofit use, with appropriate credit to the original author and the *Northern California Oracle Users Group Journal*. All other reproduction is strictly prohibited without written permission of the editor. Two copies of each reprint should be sent to the editor.

## NOCOUG BOARD

### President

Roger Schrag, Database Specialists, Inc.  
[rschrag@dbspecialists.com](mailto:rschrag@dbspecialists.com)

### Vice President

Darrin Swan, LECCOTECH  
[darrin@leccotech.com](mailto:darrin@leccotech.com)

### Treasurer/Secretary

Judy Lyman, Contra Costa County Public Works  
[gooma@california.com](mailto:gooma@california.com)

### Membership and Past President

Joel Rosingana, Independent Consultant  
[joelros@pacbell.net](mailto:joelros@pacbell.net)

### Webmaster

Eric Hutchinson, Independent Consultant  
[erichutchinson@comcast.net](mailto:erichutchinson@comcast.net)

### Journal Editor

Lisa Loper, Database Specialists, Inc.  
[lloper@dbspecialists.com](mailto:lloper@dbspecialists.com)

### Vendor Relations

Mike DeVito, BankServ  
[mike@devito.com](mailto:mike@devito.com)

### IOUG-A Representative

Hamid Minoui, UPS  
[hamid.minoui@ups-scs.com](mailto:hamid.minoui@ups-scs.com)

### Members, At Large

Jen Hong, Cisco Systems  
[hong\\_jen@yahoo.com](mailto:hong_jen@yahoo.com)

Vilin Roufchaie, Cingular Wireless  
[vilin.roufchaie@cingular.com](mailto:vilin.roufchaie@cingular.com)

Eric Buskirk, Mission Critical 24/7  
[ebuskirk@mc247.com](mailto:ebuskirk@mc247.com)

### Conference Agenda Coordinator

Colette Lamm, Independent Consultant  
[colette\\_lamm@yahoo.com](mailto:colette_lamm@yahoo.com)

### Marketing Director

Vacant

## ADVERTISING RATES

### Contact: Nora Rosingana

325 Camaritas Way  
Danville, CA 94526  
Ph: (925) 820-1589

*The NoCOUG Journal is published quarterly.*

Size	Per Issue	Per Year
Quarter Page	\$100	\$320
Half Page	\$200	\$640
Full Page	\$400	\$1,280

*Personnel recruitment ads are not accepted.*



# The Well-Rounded Oracle Professional



Roger Schrag

In this column last quarter I shared some of my thoughts about how Oracle DBAs and developers can keep their technical skills current. With technology being the fast moving target that it is, keeping up-to-date is not an easy feat. But there is more to being a successful Oracle professional than technical know-how. This time around I'd like to discuss these "soft skills" that are so very important.

## Exceptional DBAs and developers

Key non-technical traits shared by the truly exceptional Oracle DBAs and developers out there include: strong communication skills, attention to detail, a proactive attitude, and the ability to see the big picture. It isn't hard to see why these characteristics are so important—you could be the best PL/SQL programmer in the world, but if the application you build does not do what the user needs it to do, then the project was not successful. Hence, the soft skills can be just as important as the technical skills.

## Honest Communication

Abundant, straightforward, respectful communication with the other members of the team and with users and customers is key to the success of any project and the smooth operation of any organization. Telling others the truth—which may not necessarily be what they want to hear—in a tactful, respectful way will benefit everyone and avoid unpleasant surprises down the road. If a project is running behind schedule, people will usually appreciate learning of the delay as soon as possible. They won't be thrilled about the delay, but they will be happier to learn of the delay early on, rather than on the date the system was supposed to be rolled out.

## Attention to Detail

Being thorough and paying attention to details benefits everyone because people's expectations are more likely to be

met the first time around. Application bugs, oversights and omissions, and database disasters all can stem from a lack of attention to detail. Users appreciate when developers review all of the specifications carefully one last time before deploying a new form in production. Similarly, everyone will be happier if the DBA reviews the procedure carefully one last time before beginning the database upgrade, especially if that review uncovers a detail that had been missed before.

## A Proactive Approach

Taking a proactive approach often saves time and misery for everyone. The earlier in the process that a bug or mistake is discovered, the easier it is to fix. And the best design flaw of all is the one that never happened. Think about how you can prevent problems from occurring in the first place. For example, DBAs shouldn't wait for a tablespace to fill and space allocation errors to occur before deciding it is time to add more disk storage to the database server. Similarly, users will be much happier if the developers point out an apparent omission from the specification, instead of just coding the application according to the incomplete specification.

## Learning Opportunities

Bugs are sometimes euphemistically called "enhancement opportunities." Likewise, every experience is a learning opportunity. If a code change is put into production and a batch job that used to complete in ten minutes now takes ten hours, getting to the root of the problem, understanding it, and then fixing it will likely enable you to avoid falling into the same trap next time. Simply throwing more hardware or more indexes at a problem most likely won't lead to as effective a solution, and the chances of the same

*(continued on page 4)*

## Mark Your Calendars!

Our Summer Conference will take place on Thursday, August 21, 2003 at ChevronTexaco in San Ramon. You won't want to miss it!

## President's Message

(continued from page 4)

problem or a similar one coming back to bite you again are quite high.

### Being a Team Player

Being a good teammate is important to the success of any project. Teammates in this case could be fellow DBAs or developers, business analysts, managers, users, or customers. Everyone shares a common goal, namely the success of the project. If the project fails then everyone will suffer, not just those persons responsible for the failure. The most successful developers and DBAs don't lose sight of this fact when they see a problem brewing elsewhere on a project. They step in and assist their teammates, even if it would be so much easier to say, "Hey that's not my problem!"

### More Than Pure Technical Talent

There are many knowledgeable Oracle DBAs and developers out there, but the truly exceptional ones have more going for them than pure technical talent. The best folks out there have a well-rounded skillset including good communication skills, attention to detail, a proactive approach, the desire and ability to learn from every experience, and good teamwork. Keeping your technical skills up-to-date might seem like a full-time job, but don't forget to focus on the soft skills too!

### Work on Your Soft Skills

A great way to work on your soft skills, believe it or not, is to contribute material to NoCOUG. By writing an article for the *NoCOUG Journal* or giving a presentation at a NoCOUG conference you'll be giving your communication skills a workout. You'll also have the opportunity to pay close attention to the details, and you'll definitely be able to learn from the experience. I have heard many authors say that the best way to master something new is to write a paper about it. If you are interested in learning more about contributing an article or giving a presentation at NoCOUG, feel free to drop me an email at [rschrag@dbspecialists.com](mailto:rschrag@dbspecialists.com) or chat with any member of the NoCOUG board.

### Speaking of NoCOUG Conferences

You won't want to miss the Summer Conference coming up Thursday, August 21 at Chevron-Texaco in San Ramon. There will be 12 technical sessions to choose from, plus a keynote, vendor exhibits, networking opportunities, and more. Jonathan Lewis, the Oracle performance expert in the UK who often answers tough questions on Metalink's forums, will be giving the keynote and also leading one of the technical sessions. Check out the latest conference details and let us know you are coming at [www.nocoug.org](http://www.nocoug.org). I hope to see you there! ▲

**LECCO**  
**SQL Expert**  
SQL with a Higher IQ

LECCO SQL Expert automates the task of performance tuning and optimization of database applications by providing developers and DBAs with expert knowledge of SQL. Through its proprietary Feedback Searching engine, it provides optimum SQL statements—guaranteed to provide the correct results. No guesswork or hopeful suggestions, statements are actually rewritten.

- Oracle
- Sybase
- MS SQL Server

Join the ranks of the experts and download a FREE evaluation copy today!

Where the experts go for performance.

**LECCOTECH**  
www.leccotech.com

That's right...  
**no more sleepless nights!**

- Monitoring and Support
- Customized Solutions
- Free monitoring up to 24/7\*

\* The Fine Print: With minimum professional services commitment. See website for details.

**MISSION CRITICAL 24/7**

ORACLE PartnerNetwork CERTIFIED PARTNER

510.352.7300 • 866.352.7300  
info@mc247.com • www.mc247.com



# Meet Eric Buskirk

As one of the newest members of the NoCOUG board of directors, Eric Buskirk has brought a lot of enthusiasm and new ideas to the board. He joined the board earlier this year as a Director, at large. That means Eric has been doing a variety of board activities, pitching in wherever he can, and we've all been grateful that he's shared his knowledge and time. Eric also stays very busy as the founder and president of Mission Critical 24/7 ([www.mc247.com](http://www.mc247.com)), a technology consulting firm based in San Leandro. Look for Eric and introduce yourself at one of our upcoming conferences. Until then, here's a little bit more about him.

### *How do you make time to participate on the NoCOUG board while managing the responsibilities of your company?*

I wanted to join an organization that gives something back to society. Originally, I joined the local chamber of commerce, but I found that the members were self-employed or worked for organizations that were too small to recognize the importance of a database or enterprise system. I've found that NoCOUG members share similar interests; thus I've committed extra time to this organization.

### *What do you think makes being on the board worthwhile?*

I'm glad to be part of an organization where I can provide value. It's rewarding to see NoCOUG grow and evolve throughout time.

### *Back to your company, Mission Critical 24/7: What does your company do, and what is your primary role?*

Mission Critical 24/7 provides remote and on-site monitoring and database administration primarily to supplement a full-time DBA, developer or project manager. I was motivated to start this company while spending a

winter working on an SAP/Oracle project in Minneapolis. They had one DBA who was responsible for managing several dozen mission critical databases. This resulted in an extremely overworked employee and a very nervous company. Having just completed the Oracle Certified Professional (OCP) exams, this turned out to be a good match of my technical skills to a neglected market niche.

### *What, specifically, prompted you to get involved with Oracle technology?*

Throughout my career as a programmer, data architect or project manager I had usually worked with Oracle databases; however, I was always at the mercy of the DBA. Several years ago I realized that I hadn't taken any post-graduate courses or training, so I decided that completing the OCP would be a reasonable goal and it would reduce my reliance on DBAs.

### *So, when you're not working or volunteering with NoCOUG, is there any time left to do things you enjoy?*

Yes, I try to allocate time for a variety of activities such as hiking, rollerblading and backpacking. My favorite activity, however, is traveling. I've spent a significant amount of time in Europe, Australia, New Zealand, and east Africa, in addition to much of the United States. I'm always curious about new places, cultures, and people.

### *How long have you lived in the Bay Area? What do you consider to be your hometown?*

Mesa (a suburb of Phoenix), Arizona is where I grew up. I lived in Tucson and Flagstaff for undergraduate and graduate schools, respectively, and I lived in New York City prior to coming to the Bay Area. I've lived here for a total of four years and I consider this my home.



Eric Buskirk

### *What's one thing about you that would surprise most people?*

I plead the Fifth.

### *What book are you currently reading?*

I've been reading *A Man and His Machine: Thomas Watson, Sr. and the Making of IBM*. Other favorites are works by Ron Chernow including *House of Morgan* and *Titan*, biographies about J.P. Morgan and John D. Rockefeller, respectively. Michael Crichton is my favorite fiction writer. As a middle child I found *Born to Rebel: Birth Order, Family Dynamics, and Creative Lives*, by Frank J. Sulloway, to be very interesting. Oh yeah, my true passion is reading books by Oracle Press.

### *What is your favorite electronic toy?*

Ironically, I'm not very interested in electronic gadgets. I don't have a Palm Pilot or PDA, but I'm fascinated with how technologies such as computers, databases and the Internet can make businesses more efficient—though this should be expected of a business owner.

### *And, finally, back to something Oracle-related. What is your favorite Oracle resource website?*

I've frequently used [Tahiti.oracle.com](http://Tahiti.oracle.com) to search Oracle documentation. Also, [Database Specialists \(www.dbspecialists.com\)](http://DatabaseSpecialists.com) is a great resource for white papers and scripts. ▲

# Oracle Temporary Tables, Temporary Segments & Temporary Tablespaces

## Part II

By Hamid Minoui, UPS-Supply Chain Solutions



*Hamid Minoui*

### Abstract

Part I of this paper addressed the characteristics of different types of Oracle's temporary segments, and the circumstances under which they are created and used. The main focus of the second part of the paper is the temporary tablespace, the tablespace in which most temporary segments are allocated. The advantages of temporary tablespaces that are "locally managed" over those that are "dictionary managed" will also be covered.

### Overview

Oracle uses two types of temporary segments: sort segments, and other temporary segments such as those associated with global temporary tables. A temporary tablespace is the space in which sort segments and other temporary segments are created and managed. A temporary tablespace is a tablespace, which is defined to be the default temporary tablespace for a database user. Once a tablespace is designated as a user's temporary tablespace, it will be used 'internally by Oracle' as the temporary work area to fulfill the users temporary object requirements, such as disk sorts.

Until Oracle 7.3 there was no tablespace appropriate for the special management of temporary segments. As of that release a tablespace can be defined, with a special CONTENT type of TEMPORARY, for better management of temporary segment extents. With the introduction of locally managed tablespaces in Oracle 8i, a more efficient way to manage temporary segments became possible. With locally managed temporary tablespaces, Oracle users now have a more appropriate type of tablespace for handling disk sorts and other operations requiring temporary segments.

The comparison between different types of tablespaces assigned as temporary to database users, and the way the Oracle instance manages space using them is the subject of this article. Its purpose is to demonstrate which tablespace category is more appropriate to be designated for this role.

### Tablespace Extent Management

Before discussing the various categories of temporary tablespaces, the differences between a locally managed and dictionary managed tablespaces are worth mentioning. As of Oracle 8i, two methods of space management within a tablespace became available for Oracle to keep track of where free space exists in the tablespace.

The SMON (system monitor) process, which is responsible for managing space, maintains records where free space exists and frequently coalesces free space in tablespaces with nonzero pctincrease. For a dictionary managed tablespace, Oracle server processes and records the information about free space in the data dictionary tables (such as UET\$, FET\$) whenever new extents are allocated or freed. In addition, for the changes performed to the dictionary tables, Oracle generates undo and redo information.

For each locally-managed tablespace, Oracle maintains a bitmap in all data file headers for the tablespace. The bitmap represents a block or a group of blocks in the data file, and each bit in the map represents whether that block is free or not. When extents are allocated or freed, Oracle changes the bit value to indicate the status of the block. These changes do not generate rollback information because they do not update tables in the data dictionary (except for tablespace quota information). The locally managed tablespace automatically tracks adjacent free space, eliminating the need for SMON to coalesce free extents. This method of tablespace management significantly reduces the overhead associated with dictionary based extent management. It also reduces recursive space management in the database. Recursive space management occurs in dictionary-managed tablespaces, if consuming or releasing space in an extent results in another operation that consumes or releases space in a rollback segment or data dictionary table. Segments in a locally-managed tablespace are not governed by storage parameters such as INITIAL, NEXT, PCTINCREASE, MINEXTENTS and MAXEXTENTS. Instead, the size of the extent, for all segments in the tablespace, is either the value specified by the UNIFORM clause or system-managed, if AUTOALLOCATE is used at the time the tablespace is created.

If UNIFORM is specified, each bit in the bitmap represents one extent of size UNIFORM SIZE. Therefore, each bit maps a group of n blocks, where n is (UNIFORM SIZE/DB\_BLOCK\_SIZE). As an example, the number of blocks in a group to which each bit of a locally-managed table, with UNIFORM size of 64K in a database has DB\_BLOCK\_SIZE of 4k, maps is 16 blocks. (64k/4k).

Under Oracle 8i, all tablespaces except for the SYSTEM tablespace can be created as locally-managed. Oracle 9i allowed for the SYSTEM tablespace to be locally managed only with the AUTOALLOCATE option.

Beginning with Oracle 9i, all tablespaces are locally managed by default. In addition, if the parameter COMPATIBLE is set to 9.0.0 all subsequent tablespaces to be created will have the default extent management of LOCAL, otherwise the default behavior is that they be created as dictionary managed.

Fortunately, Oracle provides a procedure to migrate dictionary-managed tablespaces to locally-managed. The requirement has the tablespace in READ-WRITE mode before using the procedure. The name of the procedure is DBMS\_SPACE\_ADMIN.TABLESPACE\_MIGRATE\_TO\_LOCAL.

### Temporary and Permanent Tablespaces

Now let's compare the temporary segments created in PERMANENT tablespaces to those that are of type TEMPORARY. The type of the tablespace is recoded in CONTENTS column in the DBA\_TABLESPACE view as PERMANENT or TEMPORARY.

When creating a tablespace using the syntax shown below, a permanent tablespace is created unless the TEMPORARY clause is specified.

```
CREATE TABLESPACE ...[PERMANENT | TEMPORARY]
```

In Oracle versions prior to Oracle 9i, a tablespace of either type can be assigned as a user's temporary tablespace. When a PERMANENT tablespace is assigned for this purpose, all operations requiring temporary segments would cause the creation of such segments and the allocation of their extents by the requesting user process, to be de-allocated and dropped at the end of each operation.

A tablespace of type TEMPORARY allows better managements of temporary segments that are stored in the tablespace. Creating temporary tablespace of type TEMPORARY using the above syntax was made available with Oracle 7.3.

With a tablespace of type TEMPORARY, instead of allocating temporary segments on the fly, and dropping them after the transaction, one single temporary segment is created and shared among other users. The single temporary segment is created, after an instance startup, in reaction to the first operation that requires one, such as disk sorts, and is not dropped until the instance restarts. All subsequent database users requiring disk sorts will share this single temporary segment. Other than the overall space availability in the tablespace, there is no limit to the number of extents that can be acquired by the sort segment.

A temporary tablespace created in this way can only use datafiles and can only be dictionary managed.

### Locally-Managed Temporary Tablespaces

Oracle 8i offered an alternative way of creating a TEMPORARY tablespace. This new syntax, that only allows the creation of locally managed tablespaces, begins with the statement:

```
CREATE TEMPORARY TABLESPACE ... TEMPFILE ...
```

As mentioned above, locally-managed tablespaces are significantly more efficient than dictionary-managed tablespaces. Disk sorts are notoriously poor performers anyway. Your temporary tablespace should be locally-managed because Oracle needs fast access to free space information when performing a disk sort, there is no sense in making your data dictionary yet another bottleneck in the process.

A locally managed-temporary tablespace created in this way can use only tempfiles.

### Temporary datafiles versus ordinary datafiles

As stated above, a locally-managed tablespace of type TEMPORARY uses temporary datafiles or tempfiles; any other types of tablespaces, regardless of the extent management method, use Oracle's ordinary datafile.

Attempting to create a TEMPORARY tablespace with the DATAFILE clause will fail with an ORA-25144 (invalid option for CREATE TABLESPACE with TEMPORARY contents) forcing you to use a tempfile.

Due to the fact that it is locally-managed, the single temporary segment residing in a tempfile improves the concurrence of multiple sort operations and multiple transactions using the same temporary segment. This highly reduces the overhead associated with space management.

Also, tempfiles do not generate redo for temporary tablespace data or modify data outside the temporary tablespace and a tablespace using tempfiles, can not be made read-only. They can be used to add convenience with starting standby databases in read-only mode. A tempfile, due to the fact that they are locally managed, can be re-creating significantly faster than dictionary-managed ones. Because of this, they should not be included in the database backup. The reason you might have wanted to backup a temporary tablespace previously would have been simply to save recovery time, as recreating a large dictionary-managed tablespace can take much longer than recovering it from tape. Since this is not the case with locally-managed tablespaces, it does not make much sense to spend time, space, and system resources to backup many megabytes of empty spaces. The backup strategy should include a 'backup control file to trace' command, which produces an ALTER statement that would add the TEMPFILE to the TEMPORARY table space in the trace file for recovery purposes.

```
ALTER TABLESPACE TEMP ADD TEMPFILE  
'/u2/oradata/MYDB/temp01.dbf' REUSE;
```

Simply use it to recover the temporary tablespace in case of failure. Fortunately, there is no need to change the existing backup scripts that query the V\$DATAFILE or

DBA\_DATA\_FILES in order to exclude the files associated with the temporary tablespace. The tempfiles are not visible views; instead, the views V\$TEMPFILE and DBA\_TEMP\_FILES hold entries for the temporary files.

Keep in mind that during the recovery, Oracle treats tempfiles and datafiles the same way. When there is no backup for tempfiles, you can simply offline the file during, and drop the tablespace associated with the tempfile while the database is open, and then re-create the tablespace using the above statement.

A temporary tablespace does not have to have any tempfile associated with it. You can remove all tempfiles from the tablespace with an ALTER DATABASE TEMPFILE ...DROP command. However, note that if all files of a TEMPORARY tablespace are dropped, any operation requiring the creation of a temporary segment in that TEMPORARY tablespace (such as a query requiring disk sort) will fail with ORA-25153 (Temporary Tablespace is Empty).

### Default Temporary Tablespace

As of Oracle 9i, you can specify a database wide default temporary tablespace when creating a database. Prior to Oracle 9i, the SYSTEM tablespace was the default temporary tablespace and although it was recommended to assign each newly created user a temporary tablespace to be used for this purpose, there was no simple way to enforce a non-SYSTEM tablespace if it was left out from the CREATE USER command. The database default temporary tablespace cannot be permanent or taken offline. However, you can still assign a dictionary-managed SYSTEM tablespace as the default temporary tablespace, but you are discouraged to do that. If a default temporary tablespace is not defined, the SYSTEM tablespace, if dictionary-managed, will still be the database default temporary tablespace as before, but now a message will be written in the ALERTLOG warning that a default temporary tablespace is recommended and necessary in future releases. When the SYSTEM tablespace is locally-managed, a database default temporary segment must be defined.

The DATABASE\_PROPERTIES view shows the name of the database default tablespace in the PROPERTY column for a PROPERTY\_NAME value of 'DEFAULT\_TEMP\_TABLESPACE.'

### Management of Temporary Segments and Extents

As stated previously, the manner in which sort segments are managed differ in tablespace of type TEMPORARY compared to those of type PERMANENT. By designating a tablespace of type TEMPORARY exclusively for sorts, sort segments are managed more efficiently. This designation effectively eliminates serialization of space management operations involved in the allocation and de-allocation of temporary segments, and provides an overall performance improvement over tablespaces with PERMANENT content.

As stated above, during normal operations, user processes that create temporary segments in PERMANENT tablespaces also de-allocate the segment after the completion of the operation. This is also true for operations like CREATE INDEX and DROP, since they cause the creation of temporary segments in PERMANENT tablespaces. Creating sort

segments for a large number of sort requests and dropping them after the completion of each request by many active user processes in the database, can be a tedious and inefficient task.

Besides, a transaction may need more than one temporary segment, and there might also be other processes requiring the allocation of new sort segments at the same time. Assigning many database users to the same default temporary tablespace would make matters worse.

All of these can result in shortage of space and fragmentation (due to extensive allocations and de-allocations of temporary segments) in the tablespace, and with a large number of allocated extents and many active database users, can negatively impact the overall database performance.

In the temporary tablespace of type TEMPORARY, a single temporary segment is created with the first disk sort operation after an instance startup. All subsequent transactions, which need to sort on disk, will share the same sort segment. They do so by allocating new extents, since they cannot share the same extent. The extents in the temporary segment are not de-allocated as long as the instance is up and running; instead, they are marked as free extents to be reused as required once the sort operation using them is completed.

The multiple sort operations and multiple transactions using the same temporary segment reduces the overhead associated with space management during normal database operations, making sorts more efficient in TEMPORARY tablespaces than in their PERMANENT counterparts.

Since temporary segments are not dropped for as long as the instance is up, you will notice when monitoring the size of the available free space in the tablespace, that once disk sorts are finished the amount of available space does not increase, even when there are no users connected to the database. You may also see a large number of extents of type TEMPORARY in the temporary tablespace when querying DBA\_SEGMENTS.

The temporary segment grows to a certain steady state and the information about the segment is stored in the Sort Extent Pool (SEP), which is a component of the shared pool in the System Global Area (SGA). The Sort Extent Pool is protected by a latch (Sort Extent Pool Latch), which synchronizes the access to the SEP. All statements requiring sort in the temporary tablespace first checks the SEP for free extents. Multiple processes requiring access to the SEP would have to wait until the latch becomes available. However, the overall performance gain is the result of not having to allocate and de-allocate extents for each upcoming sort operation.

### SMON and Temporary Segments

SMON is responsible for both the cleanup of temporary segments at database startup and periodical coalescing of free extents the tablespaces with nonzero PCTINCREASE.

Also, if a user session associated with temporary segments dies, the process monitor (PMON) process responsible for cleaning up after terminated user sessions, will call upon SMON to remove the leftover temporary segment. In case of the failure of a user statement (due to some errors), SMON would have the added responsibility of cleaning up (multiple



extents of) the temporary segment extents that were left over from the failed user sessions.

You may notice that SMON is consuming large amounts of CPU time when the database is being opened and also at other times when the instance is running. This is because SMON de-allocates the extent(s) from the temporary segment before dropping the segment. This is done at system startup for dictionary-managed TEMPORARY tablespaces and also upon a failure as mentioned above.

The smaller the size of the default NEXT extent storage parameter the longer it may take to complete the initial cleanup. While the instance is running SMON wakes itself up every five minutes to coalesce free extents in any tablespace with nonzero default PCTINCREASE.

While coalescing, SMON acquires the Space Transaction (ST) enqueue in exclusive mode; and other processes waiting for the enqueue will be blocked. This is usually observed by multiple of ORA-01575 (timeout waiting for space management resource) errors being generated. SMON consumes a high percentage of CPU while coalescing, causing a possible increase in the run queue as other processes wait to get to the CPU in a CPU-bound system.

To identify whether SMON is cleaning up temporary extents, a DBA can query the DBA\_EXTENTS view several times to verify that the number of segments of type TEMPORARY is dropping with the following statements:

```
SELECT COUNT(*) FROM DBA_EXTENTS
WHERE .SEGMENT_TYPE='TEMPORARY' ;
```

To identify that SMON is coalescing free extents the DBA\_FREE\_SPACE view can be checked as below assuming TEMP is the name of the TEMPORARY tablespace.

```
SELECT COUNT (*) FROM DBA_FREE_SPACE
WHERE .TABLESPACE_NAME='TEMP' ;
```

In both cases, if the returned value of the query is dropping, it is most likely that SMON is either cleaning up temporary extents or coalescing temporary segment free space.

### Space Allocation Rules for Temporary Segments

One of the ways to avoid the situations specified above and other problems is to use certain default storage allocation rules when creating a temporary tablespace. By the definition of disk sorts, the data written to disk will equal SORT\_AREA\_SIZE, and the size of your extent should be at least that large. It is recommended that the value of SORT\_AREA\_SIZE fall in the range from 256 KB to 4 MB. The extent size should be large enough to accommodate multi-block writes for both SORT and HASH-JOIN operations. If the extent size is less than the HASH\_MULTIBLOCK\_IO\_COUNT parameter, then an ORA-3232 (unable to allocate an extent of string blocks from tablespace string) may be raised on hash joins when using TEMPORARY table. Therefore, when specifying the default storage for the temporary tablespace, set the INITIAL extents size as a multiple of SORT\_AREA\_SIZE making sure it is at least the size of the HASH\_MULTIBLOCK\_IO\_COUNT parameter plus the value of DB\_BLOCK\_SIZE.

INITIAL=n\*SORT\_AREA\_SIZE + DB\_BLOCK\_SIZE

The extra space for DB\_BLOCK\_SIZE allows for header block storage as well as multiple sorts data to be stored in each sort segment extent. Also, because sort extents should acquire extents of the same size, set INITIAL to the value of NEXT and always set PCTINCREASE to 0.

Even though INITIAL and PCTINCREASE are included as part of the DEFAULT STORAGE definition of the tablespace, the Oracle 8i Tuning Manual indicates that for temporary segments, the server ignores the setting of INITIAL and PCTINCREASE and only uses the value of NEXT to avoid fragmentation.

For Oracle 8 and later, chose a multiplier that sets the INITIAL (and NEXT) to at least the size of DB\_BLOCK\_SIZE \* DB\_FILE\_MULTIBLOCK\_READ\_COUNT.

As of Oracle 9i, multiple block sizes are supported, so choose the block size that matches the block size of the temporary tablespace for DB\_BLOCK\_SIZE in the above formula.

Note that MAXEXTENTS is not used in TEMPORARY tablespaces; the value of this storage parameter is automatically set to unlimited. Also, as of Oracle 8, you can make use of the tablespace MINIMUM EXTENT size, which ensures that every extent size (used or free) in the tablespace is at least as large as, (or a multiple of) this value.

At the same time, do not go overboard by setting a very large value for NEXT, as the unit of space allocation is a single extent. Doing so places a limit on the concurrent sort operations in a single tablespace. For example, in a tablespace of 200 MB, setting INITIAL and NEXT to 100MB limits the number of concurrent sort operations to two, assuming the sorts need one extent each.

Keep in mind that as a general rule, extents of temporary segments should be smaller than the extents of permanent segments due to the higher demand for temporary space where parallel processes or other concurrent operations must share the temporary segment.

### Restrictions

Some restrictions apply to temporary tablespaces. For a tablespace that has a content of type TEMPORARY, it is not possible to create any permanent segments (such as permanent tables, indexes, clusters, and rollback segments) in a tablespace of type TEMPORARY.

Under Oracle 8i, any type of tablespace can be assigned as the default temporary tablespace of a user. However, if a PERMANENT locally managed tablespace is assigned to a user as temporary tablespace, any operation performed by that user (such as sort) requiring temporary segment will produce an ORA-03212 (temporary segment cannot be created in locally-managed tablespace) error. Oracle 9i is even more restrictive in the sense that it does not allow any locally-managed tablespace of type PERMANENT to be defined as temporary tablespace for a user. If you try to define a locally-managed PERMANENT as temporary tablespace for a user, Oracle 9i issues an ORA-12911 (permanent tablespace cannot be temporary tablespace).

An attempt to assign a tablespace of type TEMPORARY to be a user's default tablespace will produce an ORA-12910

(cannot specify temporary tablespace as default tablespace). Also, a locally-managed SYSTEM tablespace cannot be used as the database default temporary segment.

### Obtaining Information About Temporary Segments

The following views can be used to get more information about temporary segments:

**DBA\_SEGMENTS:** To obtain name, tablespace location, owner of both types of temporary segments.

Note that you will only see information on temporary segments in PERMANENT tablespaces while allocated (e.g. sort in progress), but you see information on the single temporary segment in TEMPORARY tablespaces for the life of the instance.

**V\$SORT\_SEGMENT:** For information on current sort segments such as number of (used or free) extents allocated to the segments, size of one extent (in Oracle blocks), the size and name of the tablespaces (of TEMPORARY type) on which the segments reside, the number of active users using the segments, and the sort segments high-water marks.

**V\$SORT\_USAGE:** Available since Oracle 8i, contains session information and the amount of space being used in the temporary segments in the tablespaces of either type (TEMPORARY/PERMANENT) by database users. One such type of information is the current total number of extents and blocks being used. This view is often joined with V\$SESSION to give information about users doing sorts. This view shows only information during the sort.

### Performance Tuning Tips

If there are a lot of contentions for the Sort Extent Pool latch, as reported by v\$latch dynamic view, you should consider increasing the default storage parameter for NEXT extent of the temporary tablespace. If the wait is the result of too many concurrent sorts, increasing the SORT\_AREA\_SIZE could reduce the number of disk sorts by allowing more sorts to take place in memory.

If the ADDED\_EXTENTS and FREED\_EXTENTS columns of the V\$SORT\_SEGMENT view reveals extensive allocation/de-allocation activities, consider resizing the temporary tablespace involved in the sort operation to a larger size. If the MAX\_SORT\_SIZE column of this view indicates the occurrence of large sorts, you probably should assign a different temporary tablespace to the user performing these operations.

Since the SEP is allocated from the SGA, it may affect the other SGA components. The size of SEP is roughly proportional to the number of total extents of all the sort segments in the instance. Each extent takes about 20 bytes out of the shared pool for its SEP structure. Increasing the extent size will reduce the size of SEP, but may increase the amount of temporary storage usage (the last extent by each sort being only partially filled).

### Conclusion

I hope I was able to demonstrate the performance advantages that locally-managed temporary tablespaces offer over dictionary-managed ones, especially for high activities such as disk sorts. Locally managed, temporary tablespaces are recommended as delivering the best performance for global temporary tables and sort operations as they consume the least amount of resources and provide the greatest throughput. I believe, dictionary-managed tablespaces will soon be considered as legacy and its usage may even be deprecated or de-supported some day. Already, under Oracle 9i, locally-managed tablespaces are created by default. There are probably some reasons to still continue to use them for a while. Perhaps many home-grown monitoring scripts are written under the assumption that tablespaces are dictionary-managed. But, there is no reason to continue to use dictionary-managed temporary tablespaces if you are using Oracle 8i or higher. The benefits that locally managed temporary segments offer are very difficult to circumvent. ▲

## Other User Groups

### Local

NorCalOAUG – Northern California Oracle Applications Users Group

- **Contact:** Michael Capelle (650) 562-1167
- **Email:** capelle@tru-course.com
- **Website:** [www.norcaloaug.org](http://www.norcaloaug.org)

### Sacramento

SacOUG – The Sacramento Oracle User Group

- **Contact:** Ravi Verma (916) 705-3261
- **Email:** ravi.verma@telcommand.com
- **Website:** [www.sacoug.org](http://www.sacoug.org)

### International

IOUG – International Oracle Users Group

- **Website:** [www.ioug.org](http://www.ioug.org)

### U.S. Domestic

OAUG – Oracle Applications Users Group

- **Website:** [www.oaug.org](http://www.oaug.org)

ODTUG – Oracle Development Tools User Group

- **Website:** [www.odtug.com](http://www.odtug.com)

Canvassing calls by employment recruiters to local chapter contacts is strongly discouraged.



## SPONSORSHIP APPRECIATION

# Many Thanks to Our Sponsors

NoCOUG would like to acknowledge and thank our generous sponsors for their contributions. Without this sponsorship, it would not be possible to present regular events while offering low-cost membership dues. If your company is able to offer sponsorship at any level, please contact NoCOUG President Roger Schrag at [rschrag@dbspecialists.com](mailto:rschrag@dbspecialists.com). ▲

*Long-term event sponsorship:*

LOCKHEED MARTIN

CHEVRON TEXACO

ORACLE CORP.

PG&E

## Thank you! Year 2003 Gold Level Support Vendors:

- Database Specialists, Inc.
- Embarcadero Technologies
- Golden Gate
- LECCOTECH
- Mission Critical 24/7
- Quest Software, Inc.

*For information about our Gold Level Vendor Program, contact the NoCOUG vendor coordinator via email at: [vendor\\_coordinator@nocoug.org](mailto:vendor_coordinator@nocoug.org)*



## TREASURER'S REPORT

Judy Lyman, Treasurer

<b>Beginning Balance</b>		
April 1, 2003		\$ 60,483.83
<b>Revenue</b>		
Membership Dues	5,689.90	
Meeting Fees	430.00	
Vendor Receipts	1,500.00	
Advertising	960.00	
Interest	33.93	
Tax Refunds	2,022.00	
<b>Total Revenue</b>		<b>\$ 10,635.83</b>
<b>Expenses</b>		
Regional Meeting	3,222.72	
Journal	5,410.85	
Membership	-	
Administration	1,284.43	
Website	75.00	
Board Meeting	599.53	
Miscellaneous	525.00	
IRS	260.00	
FTB Tax	800.00	
<b>Total Expenses</b>		<b>\$ 12,177.53</b>
<b>Ending Balance</b>		
June 30, 2003		<b>\$ 58,942.13</b>



SPACE EXPERT™ GIVES YOU A SMARTER WAY TO  
*visualize* YOUR DATABASE'S SPACE PROBLEMS.

It's not always easy to see where pesky space problems are lurking, ready to degrade the performance of the database that your business depends on. And it's not always easy to keep your database operating at peak levels without lots of highly trained DBAs.

Until now, that is.

Introducing Space Expert™ for Oracle from BMC Software. It intelligently and automatically visualizes, isolates, analyzes and corrects space-related problems. Enabling even a novice DBA to perform the work of many. Which means the IT department's SLAs are good as gold. IT managers can concentrate on managing. Your database's performance is enhanced. And your company is more competitive than ever.

See for yourself. **Register for a 30-day trial copy of Space Expert for Oracle today at [www.bmc.com/distdata/spaceexpert](http://www.bmc.com/distdata/spaceexpert).** Then you'll discover just how easy we make it for you to help your database, and your business, really perform. Because it takes intelligence, not hocus pocus.



# Killing the Oracle DBMS\_JOB

By James F. Koopmann, Co-Founder & Chief Architect, dbDoctor, Inc.



James F. Koopmann

*Take control of Oracle's queue with a step-by-step approach to getting rid of those pesky DBMS\_JOBS.*

Let's face it, Oracle's job-scheduling facility is a wonderful tool for scheduling Oracle-related jobs without having to maintain a cron job on Unix or an AT job in

Windows. It is also very robust and reliable. It is the reliability and robustness that gives many of us our problems.

If you have any form of job running on your system, you will at one time or another come across the issue of a run away job that just doesn't seem to want to end. Or, maybe you will try to shutdown the database only to find out that it is waiting to complete a job. I would like to offer some help in the management of those job queues when they just don't seem to want to end or go away.

A while back, I needed to find information on how to clear the job queue for jobs running with no apparent end in sight. Some were hung, while others just were taking a bad access path to data. I needed to bring down these jobs, do a bit of tuning and then restart the jobs. Well, to my amazement, there just wasn't very much information out on the Web that gave good insight into this process. Basically, the method suggested was to first break the job and then issue an ALTER SYTEM KILL SESSION command. This method does not always work and unfortunately never on my system for the jobs I had. I then called Oracle support and basically got the same answer as what I discovered on the web. They did give me one added piece of information. If the ALTER SYSTEM KILL SESSION didn't work, I was supposed to bounce my database in order to bring down the job queue processes. First of all, this wasn't an option and when I did get the opportunity to bounce the database box, many of the jobs seemed to come right back as strong as ever.

Before writing this article, I did another quick search on the topic of killing dbms\_jobs. To my amazement there still wasn't much good information out there. This is why I want to share my method so that you too won't be stuck up against the wall with this problem and nowhere to turn.

Let's first go through a few different methods of viewing the information about job queues.

## Viewing Scheduled dbms\_jobs

When looking at which jobs have been scheduled, there is really only one view that you need to go to. The dba\_jobs view contains all the information you need to see to determine what has been scheduled, when the jobs have run last, and if any are currently running. Use the following simple script to take a look. Bear with me on the sub-select, I will build on this query as we go on in the presentation.

```
scheduled_dbms_jobs.sql
set linesize 250
col log_user          for a10
col job               for 9999999          head 'Job'
col broken            for a1                head 'B'
col failures          for 99                head "fail"
col last_date         for a18               head 'Last|Date'
col this_date         for a18               head 'This|Date'
col next_date         for a18               head 'Next|Date'
col interval          for 9999.000         head 'Run|Interval'
col what              for a60
```

```
select j.log_user,
j.job,
j.broken,
j.failures,
j.last_date||':'||j.last_sec last_date,
j.this_date||':'||j.this_sec this_date,
j.next_date||':'||j.next_sec next_date,
j.next_date - j.last_date interval,
j.what
from (select dj.LOG_USER, dj.JOB, dj.BROKEN, dj.FAILURES,
dj.LAST_DATE, dj.LAST_SEC, dj.THIS_DATE, dj.THIS_SEC,
dj.NEXT_DATE, dj.NEXT_SEC, dj.INTERVAL, dj.WHAT
from dba_jobs dj) j;
```

## What Jobs are Actually Running

A simple join to the dba\_jobs\_running view will give us a good handle on the scheduled jobs that are actually running at this time. This is done by a simple join through the job number. The new column returned here of interest is the sid, which is the identifier of the process that is currently executing the job.

```
running_jobs.sql
set linesize 250
col sidfor 9999 head 'Session|ID'
col log_user for a10
col jobfor 9999999head 'Job'
col broken for a1 head 'B'
col failures for 99 head "fail"
col last_datefor a18head 'Last|Date'
col this_datefor a18head 'This|Date'
col next_datefor a18head 'Next|Date'
col interval for 9999.000 head 'Run|Interval'
col what for a60
```

```
select j.sid,
j.log_user,
j.job,
j.broken,
j.failures,
j.last_date||':'||j.last_sec last_date,
j.this_date||':'||j.this_sec this_date,
j.next_date||':'||j.next_sec next_date,
```

```

j.next_date - j.last_date interval,
j.what
from (select djr.SID,
dj.LOG_USER, dj.JOB, dj.BROKEN, dj.FAILURES,
dj.LAST_DATE, dj.LAST_SEC, dj.THIS_DATE, dj.THIS_SEC,
dj.NEXT_DATE, dj.NEXT_SEC, dj.INTERVAL, dj.WHAT
from dba_jobs dj, dba_jobs_running djr
where dj.job = djr.job ) j;

```

## What Sessions are Running the Jobs

Now that we have determined which jobs are currently running, we need to find which Oracle session and operating system process is accessing them. This is done through first joining v\$process to v\$session by way of paddr and addr (the address of the process that owns the sessions) and then joining the results back to the jobs running through the sid value. The new columns returned in our query are spid, which is the operating system process identifier and serial# which is the session serial number.

```

session_jobs.sql
set linesize 250
col sid          for 9999          head 'Session|ID'
col spid         for              head 'O/S|Process|ID'
col serial#      for 99999999     head 'Session|Serial#'
col log_user     for a10          head
col job         for 99999999     head 'Job'
col broken      for a1           head 'B'
col failures     for 99          head "fail"
col last_date   for a18         head 'Last|Date'
col this_date   for a18         head 'This|Date'
col next_date   for a18         head 'Next|Date'
col interval    for 9999.000     head 'Run|Interval'
col what        for a60          head
select j.sid,
s.spid,
s.serial#,
      j.log_user,
      j.job,
      j.broken,

```

```

j.failures,
j.last_date||':'||j.last_sec last_date,
j.this_date||':'||j.this_sec this_date,
j.next_date||':'||j.next_sec next_date,
j.next_date - j.last_date interval,
j.what
from (select djr.SID,
dj.LOG_USER, dj.JOB, dj.BROKEN, dj.FAILURES,
dj.LAST_DATE, dj.LAST_SEC, dj.THIS_DATE, dj.THIS_SEC,
dj.NEXT_DATE, dj.NEXT_SEC, dj.INTERVAL, dj.WHAT
from dba_jobs dj, dba_jobs_running djr
where dj.job = djr.job ) j,
(select p.spid, s.sid, s.serial#
from v$process p, v$session s
where p.addr= s.paddr ) s
where j.sid = s.sid;

```

Now that we have a good handle on how we can look at the jobs and the key columns involved, lets go through the steps needed to bring down a job. The following is a 5 to 11 step process that should solve all your problems.

## Bringing Down a DBMS\_JOB

### 1. Find the Job You Want to Bring Down

In order to do anything, you first need to find the job that is giving you a headache. Go ahead and run the running\_jobs.sql. This will give you the prime information, job, sid, serial#, and spid, for the following actions in bringing down the job.

### 2. Mark the DBMS\_JOB as Broken

Use the following command for the job that you have to deal with:

```
SQL> EXEC DBMS_JOB.BROKEN(job#,TRUE);
```



## Need Problem Solvers?

Our team represents some of the most knowledgeable and experienced in the industry. We are authors, speakers, and Oracle Certified Professionals, each with long careers as Oracle experts and problem solvers. We provide remote DBA services and onsite Oracle database consulting.

We partner with you to:

- Increase database uptime and reliability
- Minimize risk of downtime or data loss
- Optimize performance
- Reduce overall cost of operating your database systems

**Call Us Today!**  
(415) 344-0500 • (888) 648-0500  
[www.dbspecialists.com](http://www.dbspecialists.com)

ORACLE | CERTIFIED SOLUTION PARTNER

  
**Database Specialists**



### What kind of a person thinks building and managing enterprise databases is fun?

Hey, it could be you. We might be surprised at how much it can be said that, that actually work really, really well. (Elasticity) for maximum speed and through comprehensive systems, storage performance and security management challenges with innovative, open-innovating users. And Embarcadero's growing list of solutions into things that solve existing fast, secure, junior and specialty for a whole range of development, management and integration demands for all kinds of IT environments. Completely heterogeneous for mixed bag. Needs to support. Small to huge. Over the edge. In fact, 94 of the Fortune 100 are now using Embarcadero's software. Talk to us at 415-344-0500. Or why not download a free, full featured evaluation copy of one of our products and take a look for yourself. [www.embarcadero.com](http://www.embarcadero.com)

EMBARCADERO  
TECHNOLOGIES

This command marks the job so that if we get it to stop, it won't start again. Let's make one thing perfectly clear: after executing this command the job is still running.

As a side note, if you are trying to shut down a database with jobs that run throughout the day, they may hinder your attempts to bring down the database cleanly. This is a wonderful command to make sure no jobs are executing during the shut-down process. Just be aware that you will need to mark the jobs as unbroken when the database comes back up – more on that later.

### 3. Kill the Oracle Session

Since the job is still running and it isn't going to end soon, you will need to kill the Oracle session that is executing the job. Use the following command to kill the job.

```
ALTER SYSTEM KILL SESSION 'sid,serial#';
```

### 4. Kill the O/S Process

More often than not the previous step will still leave the job attached to the database and it will still be running. When this happens you will need to go to the operating system level and get rid of the process that has spawned the running job. In order to do this you must login to the database box and issue the following command, depending on the type of operating system you have.

For Windows, at the DOS Prompt: orakill sid spid  
For UNIX at the command line: kill -9 spid

The orakill is an Oracle command, while kill is a Unix command.

### 5. Check to See If Job Is Still Running

Re-run the session\_jobs.sql script to see if you have gotten rid of the job. If you have there is no reason to go further. Usually Steps 1 through 4 will be sufficient to get rid of a job, but when the job is running wild you will have to continue with Steps 6 through 11, which describe a process for bouncing the job queue process.

### 6. Determine the Current Number of Job Queue Processes

```
SQL> col value for a10
```

```
SQL> select name,value from v$parameter where name = 'job_queue_processes';
```

### 7. Alter the Job Queue to Zero

```
SQL> ALTER SYSTEM SET job_queue_processes = 0;
```

This will bring down the entire job queue processes.

### 8. Validate that No Processes are Using the Job Queue

Re-run the session\_jobs.sql script to see if any jobs are still running. Since we have given a hard stop to the job queue and issued the kill commands, you can now wait until no jobs are running. After



## TECH TIPS

# The Oracle Underground

You'll find the Oracle Underground on the Web at [www.oraFAQ.com/faq.htm](http://www.oraFAQ.com/faq.htm). This site puts a wealth of information at the fingertips of Oracle professionals all over the world. According to the website, this popular place for Oracle users receives "more than 200,000 unique pairs of eyes that visit The Oracle FAQ every month." On the site, you'll find a variety of resources to explore, including:

- Script and Code Exchange
- Books on many topics related to Oracle
- Forums
- Job directories
- USENET Archives

all the jobs quit running, you can now do whatever maintenance or tuning you need to do before proceeding.

### 9. Mark the DBMS\_JOB as Not Broken

You can now reset the broken job to not broken so they can run again. Just issue the command.

```
SQL>EXEC DBMS_JOB.BROKEN(job#,FALSE);
```

### 10. Alter the Job Queue to Original Value

Now set the job queue to its original value so that the jobs can run again.

```
ALTER SYSTEM SET job_queue_processes = original_value;
```

### 11. Validate That DBMS\_JOB Is Running

To make sure everything is back to normal, re-run the above scripts to validate that jobs are scheduled, not broken, and are executing with the next and last dates columns changing.

Oracle has given us a great tool for scheduling activities within the database. As with many things inside the database, things don't always go as planned, nor are we given adequate tools to fix some of the problems we encounter. With the eleven steps outlined here, you will have increased your arsenal to handle those run-away jobs that have given the best of us a few tense moments. ▲

*James F. Koopmann is the Co-Founder and Chief Architect of dbDoctor Inc. ([www.dbdoctor.net](http://www.dbdoctor.net)) He can be reached at [jkoopmann@dbdoctor.net](mailto:jkoopmann@dbdoctor.net)*

# Third Party Grants

By Ian Jones, Senior Staff Consultant, Database Specialists, Inc.

## In Oracle 9iR2 you can finally do “third party grants”

A third party grant is where a DBA logs in as user A and grants a privilege on user B's table to user C. Oracle 8i and earlier do not support third party grants, so DBAs are supposed to log in as User B to grant a privilege on a table owned by User B.

Let's talk about some tricks a DBA can use in Oracle 8i and 9iR2 to accomplish a third party grant. There are a few techniques to choose from. But first, let's briefly mention the enhancement added in Oracle 9i Release 2: There is a new system privilege called “grant any object privilege” that is granted to the DBA role and directly to users SYS, MDSYS, WKSYS and CTXSYS. This new privilege enables third party grants in Oracle 9iR2. E.g.,:

```
CONNECT B/password
CREATE TABLE B.t1 (c1 VARCHAR2(10));
CONNECT A/password
GRANT SELECT ON B.t1 TO C;
```

On a system prior to Oracle 9iR2 the GRANT statement gives the error “ORA-01031: insufficient privileges.” For databases prior to Oracle 9iR2 we will walk through three workarounds to issuing third party grants.

## Option 1: Use DBMS\_SYS\_SQL.parse\_as\_user to issue grants

This option requires execute privilege on the undocumented DBMS\_SYS\_SQL package which by default is very restricted. However, if we are willing to allow access to this package it can be used to perform third party grants. E.g.,:

```
CONNECT / AS SYSDBA
DECLARE
  l_grant_sql VARCHAR2(250);
  l_cursor_id INTEGER;
  l_user_id INTEGER;
BEGIN
  l_grant_sql :=
  'GRANT SELECT ON B.t1 TO C';
  SELECT user_id INTO l_user_id
  FROM dba_users WHERE username = 'B';
  l_cursor_id :=
  SYS.dbms_sys_sql.open_cursor ();
  SYS.dbms_sys_sql.parse_as_user
  (l_cursor_id, l_grant_sql,
  dbms_sql.native, l_user_id);
  SYS.dbms_sys_sql.close_cursor
  (l_cursor_id);
END;
```

## Option 2: Temporarily become user B to issue grants on B's tables

This option works around the restriction by temporarily becoming the schema that owns the objects, i.e., schema B, and issuing the grants from that schema. This is achieved by:

- Storing away the existing encrypted password for schema B from dba\_users
- Changing the password of schema B to an arbitrary known password
- Connecting to schema B and quickly changing the password back to its original encrypted value
- Issuing the grants

Two important things to watch out for: (1) ensure that you correctly perform step a; (2) minimize the duration of steps b and c (users trying to connect to schema B using the original password will fail between steps b and c). E.g.,:

```
CONNECT A/password
SELECT username, password FROM dba_users
WHERE username = 'B';
USERNAME          PASSWORD
-----
B                  EB11FDB815CB0D37
ALTER USER B IDENTIFIED BY [new_password];
CONNECT B/[new_password]
ALTER USER B IDENTIFIED BY VALUES
'EB11FDB815CB0D37';
GRANT SELECT ON B.t1 TO C;
```

## Option 3: Create and run a stored procedure to issue grants on B's tables

This option avoids the need for privilege on DBMS\_SYS\_SQL or changing schema B's password, but is only available if schema A has the “create any procedure” system privilege. Depending upon the situation this may be the safest option of the three presented here. E.g.,:

```
CONNECT A/password
CREATE PROCEDURE B.temp_grant AS
BEGIN
  EXECUTE IMMEDIATE
  'GRANT SELECT ON B.t1 TO C';
END;
/
EXECUTE B.temp_grant
DROP PROCEDURE B.temp_grant;
```

*About the author:*

**Ian Jones** is a Senior Staff Consultant at Database Specialists, Inc. ([www.dbspecialists.com](http://www.dbspecialists.com)) He can be reached at [ijones@dbspecialists.com](mailto:ijones@dbspecialists.com).



# Too Clever for Words: Oracle 9i Log Miner

By Tim Gorman, SageLogix, Inc.

## Introduction

This paper is intended to explain the configuration and usage of the Log Miner feature. This feature was first introduced in the Oracle8i v8.1.5 version of the RDBMS product and was updated considerably in Oracle9i v9.2.0.

The initial functionality of Log Miner was simply diagnostic, for the purpose of reading redo log files and interpreting the contents. From its initial release, Log Miner was capable of reading individual redo records and reconstructing the data into SQL statements. Used this way, Log Miner can be utilized to:

- Debug the actions of one or more sessions
- Provide a high-level audit-trail for one or more sessions
- Recover deleted or modified data from prior to the commit of the transaction

To start a Log Miner “session,” the user would specify a set of specific redo log files and read from them via the SQL interface. Thus, redo log files from the current database or other databases (i.e., Oracle8 v8.0 or higher) could be read and interpreted from any database. For the purpose of translating data dictionary IDs into symbolic names (i.e., OBJECT\_ID into OBJECT\_OWNER, OBJECT\_TYPE, and OBJECT\_NAME or USER\_ID into USER\_NAME), an extract of information from the originating database’s data dictionary was necessary.

Later on, as Oracle9i was introduced, Log Miner became a component of other products used for the replication of data, namely:

- Oracle Data Guard (Logical Standby Database)
- Oracle Streams

In order to support these new features, the basic capability of reconstructing SQL statements from the binary information in the redo records is utilized, but Log Miner itself was enhanced to better support a continuous stream of information flowing through it. Instead of interpreting a set of redo log files explicitly specified by the user, Log Miner must be capable of continuously reading redo log files as they are generated. Moreover, the older method of passing data dictionary information, via generated flat-files, is completely impractical for this mode of operation. Instead, the very same stream of redo log files now also communicates this data dictionary information continuously, simply and elegantly.

## How to Use Log Miner in Oracle8i for Diagnosis

There are four steps to using Log Miner in Oracle8i:

1. Periodically capturing data dictionary information into flat-file extracts
2. Creating a list of redo log files
3. Starting a Log Miner session against that list of redo log files
4. Querying data from the files using the SQL API

## Capturing Data Dictionary Information Extracts

If you were to use a standard operating-system DUMP utility, you would see precious little recognizable text in redo log files. Redo log files contain both data as well as metadata about the data structures holding the data, but the metadata (a.k.a. data dictionary data) is represented only in terms of numeric IDs, not human-recognizable symbolic strings. For example, a schema named SCOTT might have a USERNAME = ‘SCOTT’ and USERID = 1327. It is the USERID value that is stored in the redo log files. Similarly, a table named EMP owned by the schema named SCOTT might have OBJECT\_TYPE = ‘TABLE’ and OBJECT\_NAME = ‘EMP’ and OBJECT\_ID = 20887. It is the OBJECT\_ID value that is stored in the redo log files.

Displaying this numeric data is not very useful for most human beings. Leaving the task of translating these ID values to symbolic names would not be user-friendly, to say the least.

Furthermore, if you were running Log Miner from a database named TEST and reading redo log files generated by another database entirely, you would have to perform that numeric-to-symbolic translation against the data dictionary in the other database. If that other database were unavailable for any reason, you would be out of luck. Again, not very user-friendly.

Last, the data dictionary of any database only contains current metadata about the database. In contrast, redo log files contain data from many points in time. Objects and schemas referenced in the redo log files may no longer exist in the databases that generated the logs, and the data dictionary of that database would no longer have information for translating numeric IDs to symbolic names.

For these reasons, the DBMS\_LOGMNR\_D package supplied with Oracle8i contains a packaged procedure named BUILD to extract snapshots of data dictionary information into flat-files. These extracts can then be used by a subsequent Log Miner session.

Creating these flat file extracts of the data dictionary is the only real prerequisite to using Log Miner. Generally, the decision to use Log Miner for diagnostic purposes is made after an anomaly has occurred and has been logged. In this situation, it is very much like closing the barn door after the cow has already escaped. Especially if the object to be diagnosed has since been dropped. For this reason, I highly recommend considering now whether you anticipate ever running Log Miner against a database. If you feel that there is a possibility that you might, then start running DBMS\_LOGMNR\_D.BUILD on a periodic basis now. Using a job-scheduling program such as DBMS\_JOB, start a daily extract:

```
SQL> variable jobno number
SQL> exec dbms_job.submit(:jobno, -
'begin -
dbms_logmnr_d.build(dictionary_filename=>'lmd_'||to_char(sysdate,'dy'), -
dictionary_locatio =>'/d010/orabkup/PRD'); -
end;', -
```

```

sysdate+(1/1440), -
'sysdate + 1');
SQL> select job, what, next_date, next_sec, broken, failures
2 from user_jobs where job = :jobno;
SQL> exec dbms_job.run(:jobno)

```

The first line of this script declares a SQL\*Plus variable to hold the number of the job after it is created. The second line calls the DBMS\_JOB.SUBMIT procedure to create a job and store the job number in the variable JOBNO. The third through sixth lines (lines 2 through 5 of the DBMS\_JOB.SUBMIT call) builds the SQL statement to be executed by the job. In this case, the SQL statement is really an “anonymous PL/SQL block” containing the call to the DBMS\_LOGMNR\_D.BUILD procedure.

The parameters to the BUILD procedure are DICTIONARY\_FILENAME and DICTIONARY\_LOCATION. The filename consists of the prefix “LMD\_,” the first three characters of the day of the week (derived from the Oracle SQL function to\_char(sysdate, ‘dy’), and the file-extension “.dct.” Thus, if this job is run on a Sunday, then the data dictionary extract will be stored in a file named “lmd\_sun.dct.” On Monday, the filename would be “lmd\_mon.dct.”

Using this file-naming scheme would allow a maximum of seven data dictionary extract files to be stored, one for each day of the week. The BUILD procedure will overwrite any existing files.

The directory in which this file will be created must be accessible to the UTL\_FILE package, as that is the mechanism used to create the flat file. Thus, the initialization parameter UTL\_FILE\_DIR must either contain the directory name “/d010/orabkup/PRD” or the wildcard character “\*”.

### Calling DBMS\_LOGMNR\_D.BUILD from Oracle8 V8.0.X Databases

The SQL script “dbmslmd.sql” is used to create the DBMS\_LOGMNR\_D package. This script resides in the ORACLE\_HOME subdirectory “rdbms/admin.” Normally, the package gets created during database creation as it is called during the standard “catproc.sql” script, when you are running Oracle8i software.

But what if you are running Log Miner against redo log files created by an Oracle8 v8.0.x database? Even though Log Miner is a feature of Oracle8i v8.1.x and above, redo log files created by earlier Oracle8 v8.0.x databases can still be read.

But obviously Oracle8 v8.0.x software does not include the DBMS\_LOGMNR\_D package. In this case, you can steal the file “dbmslmd.sql” from an Oracle8i v8.1.x software distribution and run it against the Oracle8 v8.0.x database.

Only the DBMS\_LOGMNR\_D package can be created on an Oracle8 v8.0.x database, and the presence of the UTL\_FILE package (installed automatically by default) is a prerequisite. The DBMS\_LOGMNR package cannot be created on an Oracle8 v8.0.x database.

### Creating a List of Redo Log Files

In Oracle8i, the procedure ADD\_LOGFILE is called to make a list of redo log files that will be read during the Log Miner session. Redo log files can either be active online redo log files or inactive archived redo log files. Naturally, it is easier and safer to read archived redo log files, since they will not be reused by the database instance and overwritten.

To create a list of ten archived redo log files:

```

SQL> exec dbms_logmnr.add_logfile('/d010/oraarch/PRD/1_1150.arc', dbms_logmnr.new)
SQL> exec dbms_logmnr.add_logfile('/d010/oraarch/PRD/1_1151.arc', dbms_logmnr.addfile)

```

```

SQL> exec dbms_logmnr.add_logfile('/d010/oraarch/PRD/1_1152.arc', dbms_logmnr.addfile)
SQL> exec dbms_logmnr.add_logfile('/d011/oraarch/PRD/1_1153.arc', dbms_logmnr.addfile)
SQL> exec dbms_logmnr.add_logfile('/d011/oraarch/PRD/1_1154.arc', dbms_logmnr.addfile)
SQL> exec dbms_logmnr.add_logfile('/d011/oraarch/PRD/1_1155.arc', dbms_logmnr.addfile)
SQL> exec dbms_logmnr.add_logfile('/d011/oraarch/PRD/1_1156.arc', dbms_logmnr.addfile)
SQL> exec dbms_logmnr.add_logfile('/d012/oraarch/PRD/1_1157.arc', dbms_logmnr.addfile)
SQL> exec dbms_logmnr.add_logfile('/d012/oraarch/PRD/1_1158.arc', dbms_logmnr.addfile)
SQL> exec dbms_logmnr.add_logfile('/d012/oraarch/PRD/1_1159.arc', dbms_logmnr.addfile)

```

The first call to ADD\_LOGFILE, using the option NEW, created a new list. The subsequent nine calls to ADD\_LOGFILE, each using the ADDFILE option, added to the list.

While the DBMS\_LOGMNR\_D package relies on the UTL\_FILE package to perform file I/O, the DBMS\_LOGMNR package does not. It uses I/O mechanisms built-in to the Oracle server processes to read from the redo log files in the list, similar to the way the database recovery mechanism works. So, the directories in which the redo log files reside do not have to be included in the UTL\_FILE\_DIR parameter, as is the case with the data dictionary extract file.

### Starting a Log Miner Session

Once a list of redo log files has been created using the ADD\_LOGFILE procedure, a Log Miner session can be started, using the START\_LOGMNR procedure in the DBMS\_LOGMNR package.

A Log Miner session exists within a database session. Queries performed on the V\$LOGMNR\_CONTENTS view within a Log Miner session act as requests to perform I/O on the redo log files. The Oracle server process supporting the database session performs all of the I/O for the Log Miner session from the redo log files. It then populates the V\$LOGMNR\_CONTENTS view with the requested information, using information read from the data dictionary extract to translate IDs into symbolic names along the way.

In Oracle8i, the START\_LOGMNR procedure has the following parameters:

- **DICTFILENAME**

The parameter DICTFILENAME is used to specify the flat file in which the data dictionary information is stored. This information will be used during the Log Miner session to translate the numeric ID information in the redo log files into symbolic names that people can understand.

- **OPTIONS**

The parameter OPTIONS has only two possible values in Oracle8i. These values are persistent variables defined in the DBMS\_LOGMNR package: SKIP\_CORRUPT and USE\_COLMAP. The SKIP\_CORRUPT flag is used when one (or more) of the redo log files is known to contain corrupted redo records. In this case, Log Miner will bypass the corrupted redo record instead of giving an error message and terminating the current operation. The USE\_COLMAP flag is used to map the names of columns to specific tables that might be referenced in the redo records. The flat file “logmnr.opt” should reside in the same directory as the data dictionary extract; it contains the mapping information for columns. The “logmnr.opt” mapping file contains entries of the format:

```
colmap = SCOTT EMP (1, EMPNO, 2, SAL, 3, JOB, 4, MGR, 5, COMM);
```

which indicates that the table EMP in the schema SCOTT will have the column EMPNO mapped to the columns PH1\_NAME, PH1\_REDO, and PH1\_UNDO in the

V\$LOGMNR\_CONTENTS view. Also, the column SAL will be mapped into the corresponding columns PH2\_NAME, PH2\_REDO, and PH2\_UNDO. In Oracle8i, there are only five sets of these “placeholder” columns in the V\$LOGMNR\_CONTENTS view, so only five columns can be specified. These columns can then be used to query for specific data values in specific column names.

- STARTSCN and ENDCSCN
- STARTTIME and ENDTIME

These parameters can be used to further restrict the range of redo records that can be queried during the Log Miner session. Of course, the list of redo log files represents a restriction, but using these parameters, we can additionally restrict by a range of SCNs (a.k.a. system change numbers) or by time.

Once the Log Miner session has been started, then only the database session hosting it can view redo log file contents using the V\$LOGMNR\_CONTENTS view.

## Querying Data Using the SQL API

There is only one V\$ view comprising the Log Miner API in Oracle8i, named V\$LOGMNR\_CONTENTS:

NameNull?`	Type	Description/comments
SCN	NUMBER	system change number of the redo record
TIMESTAMP	DATE	approximate time of the redo record
THREAD#	NUMBER	redo log thread (i.e. OPS or RAC)
LOG_ID	NUMBER	log sequence number within the redo thread
XIDUSN	NUMBER	rollback segment number in which the txn resided
XIDSLT	NUMBER	slot in the RBS's txn table in which txn resided
XIDSQN	NUMBER	# of times slot in txn table in RBS reused
RBASQN	NUMBER	log sequence number of redo byte address
RBABLK	NUMBER	block within redo log file
RBABYTE	NUMBER	byte offset within 512-byte block within redo log file
UBAFIL	NUMBER	file ID # within rollback segment of undo info
UBABLK	NUMBER	block ID # within rollback segment of undo info
UBAREC	NUMBER	slot # within block within rollback segment of undo
UBASQN	NUMBER	# of times slot within block within RBS reused
ABS_FILE#	NUMBER	absolute file ID # of data
REL_FILE#	NUMBER	relative file ID # of data
DATA_BLK#	NUMBER	block ID # within file of data
DATA_OBJ#	NUMBER	OBJECT_ID within DBA_OBJECTS of data
DATA_OBJD#	NUMBER	DATA_OBJECT_ID within DBA_OBJECTS of data
SEG_OWNER	VARCHAR2(32)	schema name owning data object
SEG_NAME	VARCHAR2(32)	name of data object
SEG_TYPE	NUMBER	type of data object
SEG_TYPE_NAME	VARCHAR2(32)	name of type of data object
TABLE_SPACE	VARCHAR2(32)	name of tablespace in which data object resides
ROW_ID	VARCHAR2(19)	full ROWID value
SESSION#	NUMBER	value of V\$SESSION.SID of txn
SERIAL#	NUMBER	value of V\$SESSION.SERIAL# of txn
USERNAME	VARCHAR2(32)	username executing the txn
SESSION_INFO	VARCHAR2(4000)	additional info
ROLLBACK	NUMBER	0 = txn committed, 1 = txn rolled back
OPERATION	VARCHAR2(32)	type of SQL operation
SQL_REDO	VARCHAR2(4000)	text of SQL statement reconstructed from redo
SQL_UNDO	VARCHAR2(4000)	text of SQL statement reconstructed from undo
RS_ID	VARCHAR2(32)	rollback segment name
SSN	NUMBER	SQL sequence number
CSF	NUMBER	SQL continuation flag (0 = not continued)
INFO	VARCHAR2(32)	additional info
STATUS	NUMBER	txn status
PH1_NAME	VARCHAR2(32)	COLMAP placeholder name #1
PH1_REDO	VARCHAR2(4000)	COLMAP placeholder redo info #1
PH1_UNDO	VARCHAR2(4000)	COLMAP placeholder undo info #1...
PH5_NAME	VARCHAR2(32)	COLMAP placeholder name #5
PH5_REDO	VARCHAR2(4000)	COLMAP placeholder redo info #5
PH5_UNDO	VARCHAR2(4000)	COLMAP placeholder undo info #5

The use of the USE\_COLMAP option, the “logmnr.opt” file, and the “placeholder” columns allows one to query the V\$LOGMNR\_CONTENTS view based on the before and/or after values of specific columns, such as:

```
SELECT *
FROM V$LOGMNR_CONTENTS
WHERE SEG_OWNER = 'SCOTT'
AND SEG_NAME = 'EMP'
AND PH1_NAME = 'ENAME'
AND PH1_REDO = 'SMITH'
AND PH1_UNDO = 'JONES';
```

This query would be searching for any transactions upon the SCOTT.EMP table where the ENAME column was changed from 'JONES' to 'SMITH,' for example.

Queries performed upon this view when a Log Miner session has not been started within the current database session will result in the error message “ORA-00942: table or view does not exist.”

Queries performed on this view during a Log Miner session will initiate read I/O directly to the redo log files in the list. Since there are no indexes upon the redo log files, the performance of very specific queries may be quite slow, as the entire range of redo records and files may need to be scanned.

## Ending a Log Miner Session

---

Ending the database session also ends the Log Miner session. Alternatively, the Log Miner session can be ended by a call to the `END_LOGMNR` procedure in the `DBMS_LOGMNR` package.

## How to Use Log Miner in Oracle9i for Diagnosis

---

Using Log Miner for diagnosis in Oracle9i is similar to Oracle8i, only there are more options.

Specifically, the following options exist:

- Data dictionary extracts can still be captured in flat-files as with Oracle8i, but in Oracle9i they can also be captured into the redo log files themselves, as well.
- Additionally in Oracle9i, if you are reading redo log files generated by the same database that is running the Log Miner session, you can use the online data dictionary itself, instead of an extract to flat file or the redo log file stream. This is just a shortcut for the situation where using the current information in the data dictionary is sufficient for your purposes.
- It is not necessary to specify the file name of every redo log file to be read by the Log Miner session, as in Oracle8i. Instead, it is possible in Oracle9i to specify only the first redo log file and then have the Log Miner session assume that the next file will have the same filename in the same location/directory, just using the next log sequence number.
- An Oracle9i Log Miner session can continuously update its data dictionary information by scanning the incoming redo log file stream for DDL statements, instead of relying on a single static image of data dictionary information, as Oracle8i Log Miner does.

We shall discuss the enhanced capabilities for Oracle9i Log Miner in the following sections.

## Capturing Data Dictionary Information Extracts

---

The `DBMS_LOGMNR_D` package has been greatly enhanced for Oracle9i. Instead of having a single capability, capturing data dictionary extracts to flat-files, the package can now perform additional duties:

- Capture data dictionary extracts to flat-files (as with Oracle8i)
- Capture data dictionary extracts into the redo log file stream
- Change the location and storage of internal tables used for capturing data dictionary extracts into the redo log file stream. By default, these tables are located in the `SYSTEM` tablespace, and they can be moved into another pre-existing tablespace.

There are two procedures belonging to the `DBMS_LOGMNR_D` package:

### Procedure build

---

The parameters expected by this procedure now vary based on the value of the new `OPTIONS` parameter. If the value of `OPTIONS` is `DBMS_LOGMNR_D.STORE_AS_FLAT_FILE` (the default), then the procedure will expect to have values provided for the `DICTIONARY_FILENAME` and `DICTIONARY_LOCATION` parameters, just as with Oracle8i.

However, if the value of the `OPTIONS` parameter is

`DBMS_LOGMNR.STORE_IN_REDO_LOGS`, then no other parameters are needed. In this case, the call to the `BUILD` procedure will populate a series of tables belonging to the `SYS` schema whose names all begin with the prefix "LOGMNR."

Since `INSERT` statements will be utilized to populate these tables, redo will be generated as a result. The redo records from these operations will be written to a redo log file, which in turn will be read by a Log Miner session. The Log Miner session, if configured properly, will recognize the changes to the data values in these tables, and update its in-memory translation map of data dictionary information accordingly.

This is a wonderfully elegant way of solving the problem of continuous data dictionary updates!

Thus, there is no longer a need for the flat-file data-dictionary extracts. Moreover, a Log Miner session can have its data dictionary translation map updated without stopping and restarting the Log Miner session, as was necessary in Oracle8i Log Miner.

## Procedure SET\_TABLESPACE

---

New with Oracle9i, this procedure is intended to move the tables associated with Log Miner out of the all-important `SYSTEM` tablespace.

This routine is expected to be run only once during the life of the database, but it is OK to run it several times if the Log Miner tables need to be moved several times.

## Identifying Redo Log Files

---

The importance of the `ADD_LOGFILE` procedure is less pronounced in Oracle9i than it was in Oracle8i. Whereas it was required to use this procedure to specify the name of every redo log file to be included in the Log Miner session, in Oracle9i this is no longer required.

Instead, if the first redo log file is specified, then the Log Miner can be instructed to make the assumption that subsequent files will reside in the same location with the same filename format, only with an incrementing log sequence number in the name. See the description of the value `DBMS_LOGMNR.CONTINUOUS_MINE` to the parameter `OPTIONS` in the procedure `START_LOGMNR`, below.

## Starting the Log Miner Session

---

The initiation of an Oracle9i Log Miner session is very much like it was in Oracle8i, with some notable additional functionality:

- If the redo log file stream contains data dictionary extract information (see `OPTION => DBMS_LOGMNR_D.STORE_IN_REDO_LOGS`), then the Log Miner session can be started so that it reads that information from the redo log file stream, instead of expecting to find the information in a separate flat file. Set the `OPTIONS` parameter with the value `DBMS_LOGMNR.DICT_FROM_REDO_LOGS`.
- Regardless of whether the data dictionary information comes from a flat file or the redo log file stream, the Log Miner session can be configured to detect any DDL changes and automatically update its in-memory data dictionary translation map. This ensures that the translation map will not become invalid over time. Set the `OPTIONS` parameter with the value `DBMS_LOGMNR.DDL_DICT_TRACKING`.
- The Log Miner session can automatically filter out

uncommitted transactions, so that only committed transactions can be seen. Set the `OPTIONS` parameter with the value `DBMS_LOGMNR.COMMITTED_DATA_ONLY`.

- If redo log files were generated by the same database in which the Log Miner session is running, then the online data dictionary can be used to initialize the in-memory data dictionary translation map. Set the `OPTIONS` parameter with the value `DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG`.
- If only one redo log file was registered in the list for the Log Miner session, then Log Miner can assume the name of the next redo log files based on the same directory location and subsequent log sequence numbers. Set the `OPTIONS` parameter with the value `DBMS_LOGMNR.CONTINUOUS_MINE`.
- For more readable SQL statements to be reconstructed in the `REDO_SQL` and `UNDO_SQL` columns of the `V$LOGMNR_CONTENTS` view, set the parameter `OPTIONS` with the value `DBMS_LOGMNR.PRINT_PRETTY_SQL`.

The `DBMS_LOGMNR` constant values can be added together to combine settings, as each value is a bit-mask. In addition to these options, there are additional diagnostic views available only during the running Log Miner session:

- `V$LOGMNR_DICTIONARY`  
Displays information about the source of the data dictionary translation map, whether it is from a flat file, from the redo log stream, or from the online data dictionary. Can be queried from outside an active Log Miner session.
- `V$LOGMNR_LOGS`  
Displays information about the redo log files used in the current Log Miner session. Can be queried from outside an active Log Miner session.
- `V$LOGMNR_LOGFILE`  
Not documented. More detailed information about each of the redo log files used in the current Log Miner session. Can be queried from outside an active Log Miner session.
- `V$LOGMNR_PARAMETERS`  
Displays information about the `OPTIONS` parameter values specified during the `START_LOGMNR` procedure call, specifically regarding any restrictions set by `START_TIME/END_TIME` or `START_SCN/END_SCN`. Can be queried from outside an active Log Miner session.
- `V$LOGMNR_SESSION`  
Not documented. Information about each Log Miner session, including `SESSION_ID` for joining back to the `V$SESSION` view. Can be queried from outside an active Log Miner session.
- `V$LOGMNR_PROCESS`  
Not documented. Information about the Oracle server process underlying the current Log Miner session. Cannot be queried from outside an active Log Miner session.
- `V$LOGMNR_TRANSACTION`  
Not documented. Information about the transaction listed within the current Log Miner session. Cannot be queried from outside an active Log Miner session.

- `V$LOGMNR_REGION`  
Not documented. Information about internal memory structures used within the current Log Miner session. Cannot be queried from outside an active Log Miner session.
- `V$LOGMNR_CALLBACK`  
Not documented. Information about internal memory structures used within the current Log Miner session. Cannot be queried from outside an active Log Miner session.
- `V$LOGMNR_STATS`  
Not documented. Usage statistics about each Log Miner session, intended to be joined to `V$LOGMNR_SESSION` and/or `V$SESSION`. Can be queried from outside an active Log Miner session.

These additional views may seem unnecessary, but the continuous mining features of Oracle9i Log Miner mean that a Log Miner session might be open for a very long time, and the calls themselves might have been made hours, days, or even months in the past.

Hopefully, some documentation on these views will be made available in the near future.

---

### Ending the Log Miner Session

There is nothing new to Oracle9i Log Miner in this respect; you can end a Log Miner session by ending the database session or by calling the `DBMS_LOGMNR.END_LOGMNR` procedure.

---

### Conclusions

Log Miner has evolved from a limited diagnostic tool into an important data replication component. Along this evolution, it remains a very effective diagnostic tool, its capabilities enhanced by features added to support data replication.

Log Miner can be an important tool for any database administrator, enhancing database availability and data recoverability. Its use as a diagnostic utility has been greatly enhanced as a side effect of its incorporation as a component of new features such as Oracle9i Data Guard Logical Standby Database and Oracle9i Streams, both of which are data replication mechanisms. ▲

---

### About the Author

*Tim Gorman is a principal consultant for SageLogix, Inc., based in Colorado. He has been a "C" programmer on databases on UNIX and VMS since 1983, and has worked with Oracle technology since 1990. He specializes in performance tuning and trouble-shooting, database architecture, and high-availability. Tim is co-author of "Oracle8 Data Warehousing" and "Essential Oracle8i Data Warehousing," published in 1998 and 2000, respectively, by John Wiley & Sons. Tim can be reached at:*

*Corporate Website: [www.SageLogix.com/](http://www.SageLogix.com/)*

*Email: [Tim@SageLogix.Com](mailto:Tim@SageLogix.Com)*

*Personal Website: [www.EvDBT.com/](http://www.EvDBT.com/)*

*This article is reprinted with permission from the IOUG Live! Conference, April 2003.*

# The Database Gets Better, but the Metrics Look Worse

By Jonathan Lewis, JL Computer Consultancy (UK)

## Traditional Metrics

Once upon a time, when Oracle databases were very small, computer memory was very expensive, and discs were very slow, a wise old bird invented the cache hit ratio as a method of measuring one aspect of the performance of an Oracle database system. There were a number of minor variations in the formula, but the basic calculation was simple:

- Cache hit ratio =  $100 \times (1 - (\text{disc-based block accesses} / \text{buffered block accesses}))$

The ideal was to get a figure close to 100 (percent), and the purpose of the calculation was simply to answer the question: do I really need to buy more memory? However, a frequently overlooked detail of the formula was that it assumed that the database had been designed to operate efficiently so that the effects of extra memory would simply allow the same amount of logical work to be done with fewer physical accesses to disk.

Many, many years later this formula, or one of its many descendants, is still touted as an important measure. But over those years it has changed from a purchase metric to an efficiency metric. Instead of saying, "I know that the database design is efficient, let's see if more memory will help," DBAs the world over are more likely to be saying "if only I can increase the cache hit ratio it will make the database more efficient."

Unsurprisingly, though, one of the most common complaints voiced on the Oracle-related Usenet news groups is "I've just upgraded to Oracle version X.X.X.X and my cache hit ratio has just dropped—what's gone wrong?" There are two main answers. First (which we will ignore for the purposes of this paper) is that when you upgrade, you usually need to adjust a few `init.ora` parameters and physical storage features to cater for changes in the optimizer's ability to find exciting new access paths. Secondly, and more importantly for the purposes of this paper, Oracle keeps enhancing its buffer access mechanisms to decrease buffer contention. One side effect of this continual enhancement is that the statistics traditionally used to calculate the cache hit ratio may show dramatic changes in usage.

This is a "problem" that doesn't just apply to the buffer cache hit ratio, of course. In today's high throughput, high concurrency systems—even a database which has been planned and structured intelligently—can suffer from problems of contention for latches (often the cache buffers chains latches and the library cache latches). It is fairly clear that in more recent versions of Oracle (particularly

9.2) various sections of low-level code have been re-engineered to reduce contention at a cost of added CPU. Remember—you can always get a faster CPU, but if you hit a latch you stop. Reducing the probability of hitting a latch by adding more latches, cascading latches, and generally burning more CPU may be a good trade-off to improve concurrency. The side effect of this strategy is to throw doubt on the meaning of the various latch-related "performance ratios."

## A Test Case

The ideal, we are told, is to get the cache hit ratio as close to 100% as possible. So take as an example the following table which shows two sets of (slightly cleaned) statistics—we use the values from `v$sysstat` (`v$sesstat`) for the most commonly used values: "db block gets," "consistent gets" and "physical reads" as the basis for this simple example and their resulting cache hit ratios:

Statistic	Example 1	Example 2
Db block gets	100,000	100,000
Consistent gets	2,000,000	14,000,000
Physical reads	141,000	141,000
Cache hit ratio	93.3%	99%

According to the theory, the first set of figures comes from a database that could do with some improvement, but the second set of figures indicates a pretty efficient database system. However, try running the following two SQL statements on your test database (version 8.1.7.4 or below).

```
create table kill_cpu (n, primary key(n))
organization index
as
select rownum n
from all_objects
where rownum <= 23
;

select count(*) X
from kill_cpu
connect by n > prior n
start with n = 1
;
```

Running under Oracle 8.1.7.4, the select statement will give the result 4,194,304, and record a similar number of consistent gets. The difference between the two sets of statistics tabulated above (hence the resulting cache hit ratios) can be produced by running this rather extreme query just three times. (Expect the query to take between 2 and 5 minutes—depending on platform and CPU speed—as a very rough guideline I assume that most systems will

handle 10,000 buffer accesses per 100MHz of CPU per second).

More significantly though, if you run this query on 9.2.0.2 the number of consistent gets recorded drops to four million to five (yes, five, not five hundred thousand, or even five thousand). In other words, if you migrate your database system from 8.1.7.4 (where I got the statistics for example 2) to 9.2.0.2 (where I got the statistics for example 1) your cache hit ratio could drop dramatically because Oracle has introduced a new (usually) improved method of handling connect by queries.

Of course, I designed an extreme example to increase the visibility of the issue. But if you are running an HR system you may have a significant number of “X is in the line management of Y” predicates in your code; or if you are running a manufacturing system you may have a significant number of “parts explosion, bill of materials” queries, in which case you may see exactly this effect.

Warning—there has been at least one note on Metalink under the Server forum reporting a problem with the Oracle 9 mechanism for connect by queries taking much longer than it used to under Oracle 8. However, the thread never reached a conclusion about what special set of circumstances made this anomaly appear.

## New Technology

It isn't just in the area of connect by queries that Oracle has re-written the code. Over the years, as databases have grown larger and the memory used to support them has increased, Oracle Corp. has recognised that pure I/O costs are not necessarily the most important performance issue, and that the cost of protecting shared memory can be very significant.

This has led to major efforts to produce new strategies for reducing latch contention—in particular on the cache buffers chains latches which are used to protect hash buckets whilst Oracle is searching for buffered data blocks. One of the main strategies for reducing this latch activity is to pin buffers, once they have been located or loaded, if they are likely to be used again by the same process in the very near future. Consequently some block accesses that used to be recorded under the statistic consistent gets will instead be recorded under buffer is pinned count because the blocks are being revisited in the same call, and Oracle has successfully anticipated the need. (In the `kill_cpu` script above, the statistics for Oracle 8.1 will show approximately 2 million visits to pinned buffers - adding another 50% buffer accesses than you would traditionally account for when calculating the buffer cache hit ratio).

Consider the following SQL statement—executed from SQL\*Plus under version 8.1.7.4—which I engineered to perform an index full scan across the primary key index of a 1,000 row table to return every row of the table:

```
select small_vc
from pin_c
where id > 0;
```

If you check the session statistics (`v$sesstat`) before and after running this query, you will find that the number of

consistent gets will be affected as you change the SQL\*Plus arraysize (set arraysize NNN). For example with an arraysize of one you will record around 500 consistent gets but with an arraysize of 500 you will probably see just a handful of consistent gets. Why the difference? The figures below summarize the critical statistics from my test case:

Statistic	Arraysize 1	Arraysize 500
Consistent gets	504	23
Buffer is pinned count	498	980

Note how consistent gets have “migrated” to become buffer is pinned. When we set the arraysize to 500, Oracle gets to an index leaf block, and pins it, then goes tripping back and forth between the table and index without having to hit the cache buffers chains latch for the index block again until either: (a) the leaf block is exhausted and the next leaf has to be visited or (b) the array is full which means the call is complete and data has to be returned to the client.

Pinning has been in use for many years, but the appearance of statistics related to pinning is relatively recent; and these statistics are still not used in any variant of the formula for the cache hit ratio. Since Oracle keeps finding new ways of trading pins for consistent gets, any metric which doesn't cater properly for both statistics is likely to degrade each time you upgrade your version of Oracle (even in the unlikely event that all optimizer options and execution paths remain unchanged across the upgrade). For example, Oracle 9 has introduced table pre-fetching, which is a cost-based feature that allows Oracle to accumulate multiple rows entries from an index and sorting them before accessing any of the targeted table blocks—which could result in increased pinning of table blocks and a matching reduction in consistent gets.

## Dedicated Caches

Of course, we don't even need to look at a detail as esoteric as pinning to discover problems with the traditional approach to measuring the cache hit ratio. Think of the macro-level feature of `buffer_pool_keep` and `buffer_pool_recycle` that appeared in Oracle 8 (replaced by `db_keep_cache_size` and `db_recycle_cache_size` in Oracle 9).

If you look at `v$sysstat` for your underlying values to calculate the cache hit ratio, what misleading figures are you likely to introduce? Think about what you are trying to measure with a cache hit ratio—and what the functions of the keep and recycle pools are.

The keep pool is where you expect to find data every time (after a brief warm-up period), the recycle pool is the thing you use because you never expect to find data cached in memory, so that every visit to an object in the recycle pool is supposed to require a disc read. Your localized cache hit ratio for the recycle pool should presumably be zero, and for the keep pool it should be 100%, or you are not making optimal use of the pools.

The issue, of course, gets more subtle with Oracle 9 where you can define different block sizes for different tablespaces, and then set up a separate block cache for each

of the available block sizes. For example, what would the effects be of creating your undo tablespace at a non-standard size? Should you expect, or require, the undo cache hit ratio (as we might choose to call the hit ratio on this cache) to behave differently from the rest of the database? (The answer is almost certainly yes, but will be application-dependent.)

### Conclusion

Oracle is constantly refining the internal mechanisms of data acquisition. There is often a time-lag, sometimes a very long time-lag, between a new mechanism being implemented, the corresponding statistics being exposed and the working DBA finding out about it, understanding it, and being able to cater for it in their efficiency metrics.

At the end of the day, if you rely on formulas that are based on well-known statistics to “calculate” the effectiveness of your database, you are likely to end up joining the ranks of those who have cried:

“I upgraded the database and my favorite performance metric just got worse.” ▲

### Acknowledgments

The material in this white paper and the accompanying presentation was originally presented to the UKOUG annual conference in December 2001, and subsequently at OW Copenhagen 2002.

### Further Reading

Connor McDonald: “Custom Hit Ratio”  
[www.oracledba.co.uk](http://www.oracledba.co.uk)

Jonathan Lewis: “Old Wives Tales”  
[www.jlcomp.demon.co.uk](http://www.jlcomp.demon.co.uk)

*This article is reprinted with permission from the IOUG Live! Conference, April 2003.*



## TECH TIPS

# NYOUG Technical Journal Archives

The New York Oracle Users Group has developed an online collection of technical papers that have been published in their newsletter, “The Technical Journal.” Here is a quick sampling from the NYOUG Technical Journal archives:

- ***Applying a Blockcentric Approach to Oracle Tuning***

In this paper, Dan Fink presents a new approach to Oracle application tuning that “shifts the focus away from rows and towards blocks.” See what Dan has to say at [www.nyoug.org/200303fink.pdf](http://www.nyoug.org/200303fink.pdf).

- ***Cats, Dogs, and Ora-01555***

Tim Gorman takes a look at the sometimes tenuous relationships between DBAs and developers, starting off with some differing perspectives on the “snapshot too old” error. Read this entertaining paper from Tim at [www.nyoug.org/200303gorman.pdf](http://www.nyoug.org/200303gorman.pdf).

Other topics include:

- **Digging Deeper: Segment Level Statistics in Oracle 9iR2**
- **Getting Started with Oracle 9i’s XML DB Functionality**
- **Query Tuning Using Advanced Hints**

You can browse the full archives at [www.nyoug.org/techjournal.htm](http://www.nyoug.org/techjournal.htm). ▲



(APPLICATION CONFIDENCE)



"Stride Rite's databases and DBAs run better with Quest Central."

—Yusef Akyuz, CIO, Stride Rite Corporation

stride rite

**High Performance. High Availability.** With 45 discrete mission critical databases, Stride Rite cannot afford to lose ground due to poor performance or downtime. Quest Central™ allows a three-person team to manage more than a terabyte of data without running ragged. Database management tasks that used to take days now take hours. And performance issues are identified and resolved before they affect users. Let Quest Software show you how you can confidently manage more databases without breaking stride or your staffing budget. For a Free Database Health Check, call

800-306-9329 or visit [www.quest.com/nocoug](http://www.quest.com/nocoug).

FREE  
DATABASE  
HEALTH  
CHECK



© 2011 Quest Software Inc., Irvine, CA 92618. Tel: 949-754-8000 Fax: 949-754-8999

QUEST CENTRAL™: MANAGE MORE DATABASES TODAY

# NoCOUG Summer Conference

Thursday, August 21, 2003

## Session Descriptions

For up-to-date information, see [www.nocoug.org](http://www.nocoug.org)

### KEYNOTE

#### **The Burden of Proof**

Jonathan Lewis, *President, JL Computer Consultancy*

Oracle is a subtle and complex piece of software with literally thousands of features and capabilities. Despite this, there are still many so-called experts who insist on publishing quick hints and tips that are supposed to fix everybody's biggest problems. In the absence of proof - no "expert" should be trusted. Without at least a test case, no tip should be considered relevant or valid. This keynote discusses why it is important to insist on a reasonable level of proof before accepting any hint or tip as being worthwhile.

### TRACK 1

#### **How the CBO Works**

Jonathan Lewis, *President, JL Computer Consultancy*

The Cost Based Optimizer has been around for many years, but still suffers a lot of criticism because there rarely seems to be any connection between the cost of a query, and the speed with which it performs. This presentation looks at the underlying mechanisms of the CBO and explains why this is the case. We start with a basic example in Oracle 8, and then move on to some of the Oracle 9 features that are designed to make the issues of earlier versions of Oracle non-existent. Time and legal restrictions permitting, we will then have a quick look at any interesting details in Oracle 10 that might make a difference.

#### **More Examples of Interpreting Wait Events to Boost System Performance**

Roger Schrag and Terry Sutton, *Database Specialists, Inc.*

At any given moment, every Oracle server process is either busy servicing a request or waiting for something specific to happen—a condition we call a "wait event." The Oracle instance maintains detailed statistics of all wait events—how many times each session has waited for a specific event, total time waited, and exactly what events are being waited on right now. This information can be used to gain insight into the performance of both your application and database. In this presentation, we will quickly define what wait events are, detail how to extract wait event information from the instance, and discuss enhancements to the wait event interface in Oracle9i.



We'll spend the bulk of the session learning how to interpret the wait event data we collect by way of new examples. Many of us have heard of wait events and know how to query the v\$ views, but in this presentation we'll walk through real life examples of how to use wait event data to diagnose problems and boost system performance.

#### **Understanding and Leveraging Oracle 9i Advisories**

Azeem Mohamed, *Product Marketing Manager, Quest Software Inc.*

In Oracle 9i, new features called Advisories help a DBA to choose the optimal sizes for the required memory components. There are two advisories for dynamic SGA tuning: the Buffer cache advisory and the Shared pool advisory. Another advisory relates to the PGA target size.

Additionally, there are several new parameters in Oracle 9i that help a DBA to manage the memory structures of Oracle dynamically. This session will investigate the new features of Oracle 9i relative to memory management and give insight on how a DBA can leverage these new available performance options. The listener will learn about the dynamic SGA, new memory measurement units, as well as view SGA statistics, track resize operations, understand the new advisories, monitor SQL memory usage, enable automatic SQL memory management and other related memory management information. Specific examples and scripts will be included to highlight use case scenarios of these advisory options.

### TRACK 2

#### **Database Development with Oracle 9i JDeveloper**

Brian Fry, *Senior Product Manager, Application Development Tools, Oracle Corporation*

In addition to its support for Java and XML, Oracle9i JDeveloper is a complete environment for SQL and PL/SQL development and debugging. This session explores the database development aspects of Oracle9i JDeveloper—including creating database objects, editing and compiling PL/SQL code, running and debugging PL/SQL and Java stored procedures along with executing and tuning SQL statements. DBAs and database developers will learn how to use Oracle9i JDeveloper for their development work.

#### **Introduction to Using Oracle with Perl and Perl's Database Modules**

Randal L. Schwartz, *Stonehenge Consulting*

Randal L. Schwartz introduces the Perl DBI database interface. Prior knowledge of Perl is very helpful. Knowledge of basic database concepts such as SQL and tables will be presumed.

## Converting Oracle Forms and/or PL/SQL to Java

Jennifer McNeill, *President & CEO, CipherSoft*

With the emergence of Oracle 9i and Oracle 9iAS, users are trying to understand how to migrate their applications into the new “web enabled” environment. As well, many of these application developers are finding that this migration can be time consuming and overwhelming. As they move their applications into the Java environment they are looking for the ability to facilitate a smooth transition. Users want to move quickly to take advantage of new technology without having to spend enormous time and financial resources.

This presentation discusses the technical aspects and challenges of moving Oracle Forms and PL/SQL into Java and XML and reviews the pros and cons of migrating the applications. It also reviews automated tools that are available to perform this conversion. Detailed discussion is provided around the benefits of utilizing the new technology available in 9i and 9iAS.

### TRACK 3

#### How to Perform a Robust Online Re-org Using Materialized Views

Noel Reed, *Bradmark Technologies*

#### Managing Performance Through Versioning of Statistics

Claudia Fernandez, *Technical Services Manager, LECCO Technology, Inc.*


Learn how to use versioning of statistics in Oracle 9i to manage and enhance application performance. This presentation will exemplify how statistics between different environments such as development and production can be imported/exported to simulate production data volumes and manage performance changes.

#### Supporting Multiple Development Databases Without Multiple DBAs

Brian Hitchcock, *DBA, SUN Microsystems*

Development databases tend to be smaller and less critical than production databases, or at least that is the general perception. For developers working on a critical release date, their perception is somewhat different. The number of DBAs dedicated to support development database is usually limited and in many cases, there aren't any at all. Using a set of simple cron scripts and email, it is easy to continuously

# Mark Your Calendars!



**The NoCOUG  
Fall Conference  
will take place  
on Thursday,  
November 13,  
2003 at PG&E in downtown San  
Francisco. You won't want to  
miss it!**

monitor, backup and gather performance statistics for multiple development databases. While this approach lacks sex appeal, it is also free. The process to install the scripts and use the results is covered in detail from real-world experience of using this process for the last year. The scripts reviewed are provided.

#### ETL—There is a new Sheriff in Town—Oracle

Michael Schrader, *Principal, Apex Solutions, Inc.*

ETL (Extract, Transformation and Loading) is having a paradigm shift. Previously, ETL was done with tools outside the database. Ab Initio, Data Stage and Informatica were kings of the ETL world. However, there were major disadvantages to this paradigm. Third party ETL tools required a separate box because they had no controls on the consumption of CPU resources. Third party ETL tools were not ingrated into the database. This meant that ETL developers became experts in running the tool; however, had no knowledge on how the tool related to the database. In order to create fast and efficient ETL process, you needed to know both. Oracle 9i is changing the ETL paradigm. This presentation will cover these Oracle 9i ETL features and explain why you want to join the paradigm shift and do ETL in the database. ▲

## Lockheed Martin



A big “Thank You” goes out to NoCOUG member, Dave Erwin, of Lockheed Martin for being instrumental in coordinating our Spring Conference at Lockheed Martin in Sunnyvale. Another huge “Thank You” goes to Eric Vilez of Lockheed Martin’s Conference Center Operations & Services for his outstanding assistance during our event there.

# NoCOUG Summer Conference

Thursday, August 21, 2003

**Location:** The NoCOUG Summer Conference will be held on Thursday, August 21, 2003 at Chevron-Texaco, 6101 Bollinger Canyon Road San Ramon, CA 94583. See complete directions to the meeting location at [www.nocoug.org](http://www.nocoug.org).  
**Note:** Please bring two forms of photo identification. See website for details.

- 8:00 AM     **Registration and welcome. Refreshments served.**
- 9:00–9:30   **General Session.**
- 9:30–10:30 **Keynote: "The Burden of Proof" by Jonathan Lewis**
- 11:00–12:00 **Parallel Sessions #1**  
**Track 1: "How the CBO Works"**  
**Track 2: "Database Development with Oracle9i JDeveloper"**  
**Track 3: "How to Perform a Robust Online Re-org Using Materialized Views"**
- 12:00–1:00 **Lunch Break**
- 1:00–2:00   **Parallel Sessions #2**  
**Track 1: "More Examples of Interpreting Wait Events to Boost System Performance" (Part 1)**  
**Track 2: "Introduction to using Oracle with Perl and Perl's Database Modules" (Part 1)**  
**Track 3: "Managing Performance Through Versioning of Statistics"**
- 2:00–2:15   **Break**
- 2:15–3:15   **Parallel Session #3:**  
**Track 1: "More Examples of Interpreting Wait Events to Boost System Performance" (Part 2)**  
**Track 2: "Introduction to using Oracle with Perl and Perl's Database Modules" (Part 2)**  
**Track 3: "Supporting Multiple Development Databases Without Multiple DBAs"**
- 3:15–3:45   **Afternoon Break and Raffle**
- 03:45–4:45 **Parallel Sessions #4**  
**Track 1: "Understanding and Leveraging Oracle 9i Advisories"**  
**Track 2: "Converting Oracle Forms and/or PL/SQL to Java"**  
**Track 3: "ETL—There is a New Sheriff in Town—Oracle"**
- 4:45–??     **Networking/Happy Hour at O'Kane's Irish Pub – 200 Montgomery St., San Ramon**
- Cost:       **\$40 admission fee for non-members. Members free. Includes lunch voucher.**

Keep  
checking the  
NoCOUG website  
for the most  
updated Summer  
Conference  
schedule!

View session  
descriptions on  
page 26 – 27.

**RSVP online at [www.nocoug.org/rsvp.htm](http://www.nocoug.org/rsvp.htm)**

**NoCOUG**  
P.O. Box 3282  
Danville, CA 94526

FIRST-CLASS MAIL  
U.S. POSTAGE  
PAID  
SAN FRANCISCO, CA  
PERMIT NO. 11882

**Thank you to Chevron-Texaco,  
our meeting sponsor.**