

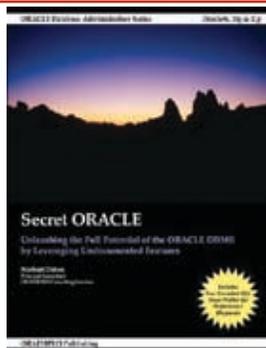
# Secret ORACLE

## Unleashing the Full Potential of the ORACLE DBMS by Leveraging Undocumented Features

A Book Review by Brian Hitchcock

### Details

**Author:** Norbert Debes  
**ISBN:** 978-1-4357-0551-7  
**Pages:** 404  
**Year of Publication:** 2008  
**Edition:** 1  
**Price:** \$41.95  
**Publisher:** Lulu.com  
 (self-published)



### Summary

**Overall review:** Very good, as long as readers understand what they are getting into.  
**Target audience:** Oracle DBAs and anyone who supports Oracle databases.  
**Would you recommend to others:** Yes.  
**Who will get the most from this book:** Those with DBA experience.  
**Is this book platform specific:** No.  
**Why did I obtain this book:** It was recommended to me.

### Overall Review

The content is focused on Oracle database features that are undocumented. This raises several issues. How can the reader know if this material has been verified? If you actually use this material to make changes to your Oracle environment, will you be supported? And how do you know if the next patch you apply will change the way one or more of these undocumented features works? The answer to all of these questions is that the reader won't know. Having said that, this doesn't mean the material isn't valuable. Learning more details of how something works, even if you don't (or can't) use them, forces you to gain a better understanding of how it all works normally. The Oracle documents tell what to do but don't often tell you what you can't do. Other sources, such as this book, will tell you what you can't do, what works well, and why.

Learning about undocumented features can help you use the normal features better. A specific example is found on page 33. Knowing more about how `OS_AUTHENT_PREFIX` works allows me to improve security in my databases, even though I won't be using the undocumented feature described.

While the material is useful, remember that undocumented generally means unsupported and probably will change as patches and new versions are released. I think you will benefit from the material presented in this book, but I am not advising anyone to actually try to use these undocumented features.

This book is not a quick, easy read. It covers a lot of ground in very short bursts. It doesn't spend (waste?) your time describing a database and explaining what a tablespace is. It is assumed you want something more. This implies that the reader should be an experienced Oracle DBA, and while that is certainly an appropriate audience, it is also true that a junior DBA could benefit as well.

With a basic understanding of Oracle database structure and features, reading this book would be an excellent way to be drawn into many new areas of the Oracle RDBMS. Reading the various sections and referring to other sources for further explanation would be of value for any new DBA.

How much did I like this book? In the first 30 pages I found so many things that I didn't know but that I think are worth knowing that I can't describe them all to you in this review. That's how good this book is. This book does, however, redefine the concept of a *chapter*. Chapter 6 is two pages long, the longest chapter is 26 pages, and over half of the chapters are under 10 pages. This doesn't affect the quality of the content, but it is different and takes some getting used to.

I also found very few errors. This is interesting because I just finished reading a book from Oracle Press that had many errors, both typos and factual errors. I was surprised that a self-published book would have many fewer errors than a book from a major publisher.

Given the wide range of topics covered, I would recommend approaching this book as follows. Each chapter has Status section and a Benefits section. I would read the Status and if it interests you, read the Benefits Learned. Then decide if the entire section is relevant to your interests. Don't misunderstand, I personally recommend reading the entire book, because you don't know what you'll run across.

Since there are 37 chapters in this book, I don't break out each one; rather, I identify each of the eleven parts of the book and provide comments on each.

I also want to point out that while a book's introduction doesn't usually warrant comment, this time it does. It con-

tains the best explanation I've seen of database versus instance. Page xiii of the introduction is really worth reading.

I also liked the author's use of the word *disambiguate*—I think we need more disambiguation, and we need it as soon as possible!

### **Part 1—Initialization Parameters (Chapters 1–2)**

I didn't know the details of the unconditional auditing for SYSDBA and SYSOPER. Similarly, the combinations of possible parameters for AUDIT\_SYSLOG\_LEVEL is useful. For PGA\_AGGREGATE\_TARGET it is interesting to know what the undocumented features are, but I would follow the author's advice that this should only be considered "under rare circumstances." The description of exactly how memory is allocated and released for manual PGA memory management is good to know.

Another section that I found to be excellent was the Misconceptions about PGA\_AGGREGATE\_TARGET. Again, while this book is mainly focused on undocumented features of Oracle RDBMS, the explanations of how things really work are great and don't require you to use any of these features.

Details of the EVENT parameter relating to 10046 tracing of multiple processes is very good. The section covering OS\_AUTHENT\_PREFIX is great. I would recommend this section and especially the Lessons Learned for all DBAs.

I had no idea that there are 540, 1124, and 1627 hidden (undocumented) parameters in 9iR2, 10gR2, and 11g respectively. While interesting, remember that these are hidden for a reason, and I would not change them without a good reason such as direction from Oracle Support.

### **Part 2—Data Dictionary Base Tables (Chapters 3–4)**

Very interesting, good information about the DBA views we use all the time and the base tables they really access.

The explanation of the V\$OBJECT\_USAGE view and index usage is very good. Even if this isn't directly relevant to your database, you should read this. It illustrates an important point. Official Oracle documents don't always tell us exactly how a feature works. How much time has been spent debating how to use a feature when we really don't know how it actually works because the manuals are wrong? Specifically, the V\$ implies that this is a dynamic performance view, but it's not. Cool. Good interview question. Similarly, two of the columns of this view store dates but are of data type VARCHAR2. That's not the way we are told to do things.

### **Part 3—Events (Chapters 5–8)**

A detailed description of deadlocking is given along with using the Event 10027 to generate lots more diagnostic information. This could be very useful when developing applications and less so for operational support.

Event 10053, which outputs the decisions of the cost-based optimizer (CBO), is described in detail and output is shown. This is useful, although it would be better to see how all this could be applied to a real-world example. One example of the detail given is that if the CBO uses a cached execution plan there will be little output to see, and a workaround for this situation is given.

I had never heard of Event 10079, let alone how to use it to

document passwords being sent without encryption. Very interesting.

### **Part 4—X\$ Fixed Tables (Chapters 9–12)**

The author discusses the issues around SGA sampling tools and doesn't advocate their use. I completely agree.

The discussion of how V\$ views are used to expose information in C data structures using X\$ tables as an intermediate step is great. I have never fully understood how this works. It is good to understand what exactly the X\$ tables are: they are not database tables.

Another useful discussion focuses on the security risks of the SELECT ANY DICTIONARY system privilege. Perhaps the most useful information in the discussion of X\$ Fixed Tables is that the author believes equally useful information is available from the V\$ views versus the X\$ tables, and that the author has never needed the contents of any X\$ tables to resolve a performance issue. This is something to think about as you decide where to put your training time.

The detailed example of how to go from the V\$ View to the related GV\$ view, and eventually to the lowest level X\$ table, is very instructive and concludes with a review of code available from the book's source code depot that will generate listings of the X\$ fixed tables for any V\$ view and vice versa.

A more detailed review of X\$BH and Latch Contention demonstrates that looking at the undocumented X\$ tables isn't the best way to go about improving system performance. The author asks the question, is it worth the DBA's time to pursue this level of investigation, or does this just lead to Compulsive Tuning Disorder (CTD)? A good discussion to keep in mind when you are faced with a tuning engagement.

While poking around in the X\$ tables may not always be the best use of your time, the section covering X\$KSLED and Enhanced Session Wait Data could be very useful. Here we are shown how to use the X\$ fixed tables related to the V\$SESSION\_WAIT view to get better wait event timing data. It is shown why we only get centisecond resolution now and how to change this to get microsecond resolution, which can be used to examine very short wait events—and that this approach works for 9i as well as 10g.

The final section discusses X\$KFFXP and ASM Metadata. I don't use ASM so I can't comment on how useful this is, but we are shown how to use operating system-level commands to access ASM database and server parameter files, which may be useful for recovering data.

### **Part 5—SQL Statements (Chapters 13–16)**

The use of ALTER SESSION (SYSTEM) SET EVENTS is covered. I didn't realize that other than turning on the 10046 trace event, use of these commands is not documented. For example, we aren't officially told how to turn off 10046. This section covers how to use these statements to trace database sessions while including wait event and bind variable information.

Similarly, the ALTER SESSION SET CURRENT\_SCHEMA section is very useful. While I have oftentimes used the workaround of documenting a user's existing schema password, resetting the password, completing my work, and resetting the user password using the "by values" clause of the alter user

statement, it is cleaner to be able to actually change to the schema user you need to be. While there are some restrictions on what you can do in the user schema after using this command, and these are covered in detail, it is still a viable alternative.

The next section formally documents what we have used for a long time, namely the ALTER USER IDENTIFIED BY VALUES statement. A thorough example is given for using this. There is also an example of how to lock database accounts that offers improved security. This prevents hackers from being able to confirm even the existence of the locked account, which is often the first step to breaking in.

I didn't know about SELECT FOR UPDATE SKIP LOCKED and again, learning about an undocumented feature could help your general understanding of Oracle. This feature is used by Advanced Queuing to improve scalability of applications that try to update the same set of rows at the same time. It is also explained, and I didn't realize, that Oracle Streams is Advanced Queuing renamed.

### **Part 6—Supplied PL/SQL Packages (Chapters 17–21)**

The first package discussed is DBMS\_BACKUP\_RESTORE. I really liked this because I've always wondered how RMAN works when it connects to and works on a database. In addition to the detailed discussion of how RMAN uses this package, there is a very worthwhile explanation of the issues with the control file holding all the backup information. A vital detail is that while the RDBMS from 9i forward has a feature to automatically backup the control file, this feature is disabled by default. It is also interesting to know that unlike other PL/SQL packages, this package can be executed when the db is not mounted.

DBMS\_IJOB is used internally by DBMS\_JOB, and can be used directly to work around limitations of DBMS\_JOB such as creating jobs in other schemas. An amusing (to me, anyway) usage note is offered. It turns out that if a job is broken, the DBMS\_JOB will return the NEXT\_DATE parameter as a fixed date of Jan 1, 4000. Clearly the Y4K crisis is on the horizon. Remember, you read about it first in the *NoCOUG Journal!*

Next up is DBMS\_SCHEDULER, which is certainly well documented, but there are many more details about this package that are not, such as executing external jobs and default privileges of external jobs. I didn't know the details of how exit code handling worked and this is explained very well. Also interesting is how to use DBMS\_SCHEDULER to schedule RMAN jobs so that the database can back itself up. This requires using some of the undocumented features to control environment variables and argument passing among other issues.

I was fascinated to read the explanation of why the UNIX user "nobody" is a requirement for installing Oracle. It turns out that this is related to how external jobs not owned by SYS are executed. I recommend this section as yet another great example of learning useful security information by learning about undocumented features.

The DBMS\_SYSTEM package includes an undocumented procedure GET\_ENV. This is useful because you can use this to find out which ORACLE\_HOME an instance is running from. This is something I wish I had known about in the past;

it would have saved me a lot of time. This package has many other undocumented features, such as enabling SQL trace in any database session and changing parameters in running sessions.

While you may not ever need to use these packages as described in these sections—and perhaps you shouldn't use or rely on unsupported features—I learned a number of useful things that help me understand what is going on in the database.

### **Part 7—Application Development (Chapters 22–23)**

The first section is all about Perl DBI and DBD::Oracle. I'm not a Perl user so I can't comment on any of the technical issues covered here. Basically this tells us that ORACLE\_HOME comes with a Perl installation and this includes the modules DBI and DBD, which can be used to develop applications that access the Oracle database. Specific issues addressed include Perl programming in an Oracle environment to deal with named bind variables, methods to connect to the database, and many more. This is very detailed and looks like it could be very useful if you need to develop your own database tools to track performance, etc.

Next we see material discussing Application Instrumentation. This includes the TRCSESS utility. There is a very detailed explanation of the problems caused by a shared server process. In short, tracing a shared server process may generate incorrect results because multiple sessions are being serviced by the single shared server process. It details like this that are not covered in the official Oracle documents. Also discussed are issues relating to the meaning of the SQL\*Net message from client wait event, a topic that has been discussed many times by many experts. I found this review to be worth reading, and it reinforces my perception that performance tuning is complicated and that you need to be very experienced with real-world experience to be sure what you are doing. The author is clear: ignoring this wait event is a mistake.

A case study using JDBC end-to-end metrics and TRCSESS is described in great detail. I'm not experienced with these features, and while I found the case study interesting, I can't comment on how useful it may be to other readers.

### **Part 8—Performance (Chapters 24–28)**

Here we see discussion of trace files and how to read them. This is very detailed and will be useful if you spend (or plan to spend) lots of time reading the raw trace files. Even the casual observer will learn some things that will help reading trace files once in a while. It is the format of the SQL trace file that is undocumented. Users generally format these trace files using TKPROF and TRCSESS. We are told that reading these files in their raw state is useful because processing (with TKPROF for example) can obscure important information.

I'm not sure how useful this level of detail will be unless you are working with trace files all the time. I found it worthwhile to read because it is interesting to see what is in the trace files.

I've heard for a long time about how good performance tuning can be when you have a fully instrumented application. The problem is, I've never seen such an application; but then, perhaps I'm just unfortunate. The author offers this

comment on properly instrumented applications, “. . . something that has eluded me in my career as a DBA.” So perhaps it isn’t just me after all.

The STATSPACK chapter appealed to me more than some of the others because I have used Statspack before. While it would be difficult to say that Statspack is an undocumented feature, the author discusses undocumented report parameters, the structure of the Statspack repository, and how to use the analytic function LAG to examine large numbers of snapshots. I found it interesting to read about a specific bug that affects versions before 11g, a bug that omits wait events which happen only in the end snapshot.

Also interesting is the licensing issue for Active Workload Repository (AWR). AWR is bundled with the Database Diagnostics Pack, which costs extra. Yes, the AWR is installed and enabled by default, but that doesn’t mean you’re licensed to use it.

Next is how to integrate Extended SQL Trace and AWR, which is useful for finding all the execution plans used for a specific SQL statement over time. Further, some plans may not be captured normally because they don’t get sent to the trace file until the cursor closes. A detailed example shows how to retrieve such plans from the AWR.

A free profiler for extended SQL trace files, ESQLTRCPROF, is discussed. Note that this departs somewhat from the theme of the book, i.e., undocumented Oracle features. This profiler is not an Oracle product; it was written by the author and it appears to be documented. On the other hand, the author shows how to make good use of this utility. I haven’t tried to use this tool so I can’t comment on its usefulness. The examples in this section are very detailed and make clear that this utility is designed to be a replacement for TKPROF.

This is followed by a discussion of the MERITS Performance Optimization Method, an optimization method that relies on ESQLTRCPROF and uses undocumented features. I have not used this method so I can’t comment on it from personal experience. **It does rely on AWR and ASH, which raises the licensing issues for AWR mentioned previously.**<sup>1</sup> The case study is interesting and is described in great detail, which should offer readers more than enough information to determine if this method is something they want to try. Two things stood out for me. **First, you must have a test system that is identical to the original environment.**<sup>2</sup> I agree, it makes sense, but I’ve never seen the resources spent to make this happen. Second, an excellent example of just how hard it is to really know what is happening inside your database: specifically, cursors without associated SQL statements. Turns out this is due to poorly instrumented LOB routines. This brings up again the issue of needing, but not having, fully instrumented application code to do the best tuning.

### Part 9—Oracle Net (Chapters 29–32)

This was my favorite part, wherein I think I learned the most things that I can actually use in my environments. I had never heard of IP=FIRST in the listener.ora file. This feature can be used to control which network adapter is used on sys-

<sup>1</sup> Book author’s clarification: The MERITS method also works with Statspack. If someone does not have a license for AWR, they should use the script sp\_capture.sql instead of awr\_capture.sql.

<sup>2</sup> Book author’s correction: This is not correct. I wrote (quote from page 300) [.. the following factors may need to be identical between the original environment and the test environment: - hardware capability ..]. So in the book it says that the hardware capability may need to be identical to reproduce the performance problem. There is no limitation that the hardware capability between test and production environment must be identical in order to use the MERITS Method. The MERITS Method works on any Oracle DBMS server.

tems that have more than one. I learned a lot reading how IP address binding works; about multihome systems; and how tnspsing can be used with an IP address, not just a Net service name.

Valid Node Checking is a feature I have worked with in my Oracle Applications 11i environments. It allows configuring a list of hosts that may or may not connect to the database. While this is documented, the fact that you can change this dynamically for 10g and 11g is not. This means you can change or enable/disable this feature without restarting the listener.

While I probably won’t need to configure any of this, knowing that these things exist and how they work may help me troubleshoot issues.

### Part 10—Real Application Clusters (Chapters 33–34)

I don’t support RAC environments. However, I learned some good things here, for example, the details of how Transparent Application Failover (TAF) works. One of the undocumented parts here is that you can use ALTER SYSTEM DISCONNECT SESSION to explicitly disconnect a session and have TAF take affect. The details of what TAF does and doesn’t do were new to me, specifically, that TAF for select statements “may fail.”

It turns out that you can remove the RAC option and this is undocumented. This could be useful in certain failure scenarios where you have to start the database but can’t repair RAC. The case study walks through all the steps needed to do this conversion.

### Part 11—Utilities (Chapters 35–37)

These sections describe undocumented aspects of the OERR script, which provides error messages for Oracle errors, RMAN Pipe Interface, and how to use ORADEBUG for performance and hanging issues. I would stay away from using ORADEBUG to modify SGA structures unless Oracle Support told me to.

### Conclusion

Change is good, change is scary, change is inevitable. The way Oracle books are published is changing as well. There was a time when authors had to submit a proposal to a major publisher and hope that their ideas would be found worthy of publication. Now, however, an author can choose to become published without the approval or oversight of anyone else. There are services that will take your content and offer it for sale on Amazon.com and print the actual book when someone orders it. No editor, no publisher, no technical review. Is this a good thing or just an inevitable change?

This is the first time I’ve read an Oracle book that was not associated in some way with a publisher. And this book is better in many ways than most I’ve read that came from major publishers.

Here is a specific example from my own experience: I read a book on Oracle Applications, a book that was not from a formal publisher. The information was very detailed and very relevant to someone that supports an Oracle 11i environment. From this book I learned some things that you must

(continued on page 21)

---

*(continued from page 14)*

not do when installing a new 11*i* environment—things that are not specifically identified in the official Oracle documents as unsupported. I don't think it was a coincidence that some of the most useful information I have found about supporting 11*i* came from a book that was not from a major publisher.

Now that it is apparent that self-publishing of Oracle books is really practical and being done, we need to change the way we look for them. This self-published book was only available on Amazon.com. This tells me that some of the best, most timely, and unique books won't ever make an appearance in your local bookstore. Also note that when you search Amazon.com, Oracle books from the major publishers will appear first. If you search for "Oracle," you will find *Secret Oracle* as the 222nd result, or on the 19th page of search results. You have to realize that these books are out there and you have to go find them. ▲

---

*Brian Hitchcock has worked at Sun Microsystems in Newark, California, for the past 11 years. He is a member of a DBA team that supports 2400+ databases for many different applications at Sun. He frequently handles issues involving tuning, character sets, and Oracle applications. Other interests include Formula One racing, finishing his second Tiffany Wisteria lamp, Springbok puzzles, Märklin model trains, Corel Painter 8, and watching TV (TiVo rules!). Previous book reviews by Brian and his contact information are available at [www.brianhitchcock.net](http://www.brianhitchcock.net).*

Copyright © 2008, Brian Hitchcock