Special NoCOUG Training Day

Don't misss this rare opportunity!
See page 27.

The Case of the Missing Kanji

Read about the database adventures of Brian Hitchcock.
See page 13.

A Conversation with Gaja Krishna Vaidyanatha

Author of Oracle Performance Tuning 101
See page 5.

BAARF – It's Not What You Think!

The scoop on why some people are saying, "Enough is enough."
See page 10.

Your Career Takes Off with NoCOUG in 2004!

We have some fantastic things planned in 2004 that will help your career take off! With NoCOUG, you always have the opportunity to network, learn, and share information with your peers. There are great conferences filled with technical sessions and roundtable discussions. Plus, there are plenty of volunteer opportunities to get involved in: speaking, serving on the board of directors, writing for the *NoCOUG Journal*, volunteering at the conferences. Just contact us at board@nocoug.org to get involved in 2004.

Mark your calendars now!

➤ Winter Conference on February 19, Oracle Conference Center, Redwood Shores
➤ Training Day on April 8, Dublin Library, Dublin
➤ Spring Conference on May 13, Lockheed Martin, Sunnyvale
➤ Summer Conference on August 19, Chevron/Texaco in San Ramon
➤ Fall Conference: TBD

Why Should You Make a Presentation at a NoCOUG Conference? See page 24.

Editor's Note

When assistant journal editor Laurie Robbins and I sat down to plan this issue, we decided to make a few changes for 2004. We hope you enjoy the changes to the *NoCOUG Journal* as much as we enjoyed planning it. We also hope that you will consider contributing to the *NoCOUG Journal* in some way in 2004. An easy way to start is by submitting a tech tip. These are typically just half a page or less, so it's a great way of learning the process, researching a small article, and seeing your name in print when it's all done. Give them a try in 2004! Also, feel free to write to us with any comments or suggestions you may have about the *NoCOUG Journal*. We look forward to hearing from you at journal@nocoug.org.

—Lisa Loper, *Journal Editor*

Table of Contents

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.

The submission deadline for the upcoming May issue is April 1, 2004. 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.*

2004: Will It Be the Year of Oracle 10g?

by Roger Schrag

Roger Schrag

Oracle 10g is on the minds of a lot of Oracle DBAs and developers these days. What can it do? How is it different from Oracle 8i or 9i? Will the grid paradigm really take hold? How soon do I need to be up to speed on 10g or fear being left behind?

The cynic in me answers that last question: "Look how many people are still running Oracle 8i in production!" The fact that Oracle extended full error correction support of Oracle 8i for another year speaks volumes on the subject. Indeed, a lot of shops are still working on getting their databases upgraded to Oracle 9i—and don't have time to even begin thinking about 10g.

So will 2004 really be the year of Oracle 10g? I suspect so. Regardless of all the folks still running Oracle 8i in production and regardless of whether or not the grid paradigm gains momentum, Oracle 10g will still become ubiquitous in 2004. Everyone will ask about it. New companies—and established companies beginning new projects—will try Oracle 10g. Oracle DBAs and developers everywhere will be asking their colleagues, "So, have you tried 10g yet?"

A colleague of mine sent a resume to a placement firm last November, looking for a job as an Oracle DBA. A recruiter called him into the office for a preliminary interview and asked, "Do you have any Oracle 10g experience?" My colleague explained to the recruiter that Oracle 10g had not yet been released in production. The recruiter responded, "Well, these folks are looking for a DBA with 10g experience. Do you have any?"

Okay, so that recruiter might have jumped the gun a little. But I'm sure this sort of thing will happen more and more in 2004. Oracle 10g will generate buzz in the database world, just like every prior release of Oracle has. In this sense, 2004 will indeed be the year of Oracle 10g.

And, speaking of 2004 . . . Start off on the right foot by attending the NoCOUG Winter Conference on Thursday, February 19th at Oracle's corporate headquarters in Redwood Shores. This event will be sponsored by Oracle, and there will be plenty of Oracle employees on hand to answer our questions and teach us all a thing or two about Oracle 10g. As always, there will also be speakers from other organizations and end users giving technical presentations on a variety of topics ranging from Java application development to database security. You'll have lots of learning and networking opportunities at the NoCOUG Winter Conference on February 19.

NoCOUG puts on one full-day conference each quarter. Admission is free for NoCOUG members. That's four days of valuable Oracle education per year. The cost of this education—just $70 for an individual membership—makes Oracle education affordable for just about everyone. If your employer won't pay for your NoCOUG membership, consider paying for it yourself. Think of the $70 as an investment in your career. You'll get a lot of bang for your buck. You can join NoCOUG online at **www.nocoug.org** and pay by check or credit card.

In 2004, NoCOUG is offering something new—a Training Day with Gaja Krishna Vaidyanatha. The Training Day, to be held April 8, is an intensive, full-day seminar on the subject of Oracle performance management. There is a separate fee to attend this event, but it will be well worth it if Oracle performance is a topic of interest to you.

As we usher in the new year, I'd like to take a moment to reflect back on 2003. I, and the rest of the NoCOUG board of directors, would like to thank you for being a part of the NoCOUG family. In 2003 we brought you four conferences featuring such highly regarded Oracle experts as Tom Kyte, Jonathan Lewis, Craig Shallahamer, and Gary Goodman. We brought you the quarterly *NoCOUG Journal* publication with its articles and valuable resources, and the NoCOUG website **www.nocoug.org** with its event updates and presentations available for free download.

We'd also like to thank the organizations that sponsor NoCOUG conferences. Thank you Oracle Corporation, Lockheed Martin, ChevronTexaco, and Pacific Gas and Electric for your generous support in 2003.

Here is one parting thought: Do you know somebody brilliant whom you would like to see at a future NoCOUG conference? I'd like to hear your ideas! Please write to me at **rschrag@dbspecialists.com** and let me know. Tom Kyte was one of our most popular conference speakers in 2003. Do you know how it happened? Somebody sent me an email saying, "You ought to get Tom Kyte to speak at NoCOUG." I asked Tom, and he said, "Sure!" It can be that easy. So please, write to me with your ideas.

See you on February 19 at the NoCOUG Winter Conference! ▲

A Conversation with Gaja Krishna Vaidyanatha

Author of *Oracle Performance Tuning 101*
Gaja will be featured at NoCOUG's Training Day on April 8, 2004

by Laurie Robbins

NoCOUG: Gaja, thank you for taking the time out of your schedule to talk to the *NoCOUG Journal*. I think with your upcoming NoCOUG Training Day on April 8, many readers would enjoy getting to know you a little better.

NoCOUG: *You have given many presentations, including Training Day events, seminars, and presentations at Oracle User's Groups, and as a trainer at Oracle Corporation. Do you enjoy interacting with others in presentation format?*

Gaja: One of the things I try to do whenever I talk is to provide insight into some information, which people are not readily aware of, or to offer a different perspective. I also try to make it kind of fun and entertaining, not to put on a dog and pony show, but I like to involve the audience so no one falls asleep, and I think people remember me for that.

The other thing that's a real energizer for me is sharing the knowledge. I recall a famous quote, "Share your knowledge and you will attain immortality." Not that I am trying to become immortal! I love interacting with people. I think that's the reason I love doing seminars.

I want the audience to feel their time with me includes some of the best technical training they've ever had. When I do something, I put 100% into it because I'd rather not do something if I have to do it half-hearted. There have been many times when I've gone to an ex-manager and said, "I don't want to do it," because I know I can't put my heart into it, and if I don't put my heart into it, it just doesn't come out right. People notice, and I want to be known as someone who is passionate about what I do, and good at what I do.

NoCOUG: *The humor you incorporate does add entertainment value. Do you find giving presentations and adding humor comes naturally to you or does it require a great deal of preparation?*

Gaja: Adding humor to a presentation is just me, it's no effort at all. It's just when I stand in front of people, I literally let my guard down, and I'm just myself. And if you see the humor, that's the kind of person I want to be known as.

> *"I want to be known as someone who is passionate about what I do, and good at what I do."*

Gaja Krishna Vaidyanatha

NoCOUG: *Your humor is also evident in the book,* Oracle Performance Tuning 101. *How did you become interested in writing a book on the subject of performance tuning from the wait interface perspective?*

Gaja: The book was done with a charter and it is going to be different in many ways: one, it's not going to have one additional page than what's required; there are enough door stoppers in the market. Also, we wanted it to be something different, something more than just dry technical data because I've read books that were great in technical content, but after about 10 pages I start to fall asleep. So, we made a conscious effort to write the book using a different style. The way I wrote the chapters is the way I would have done a presentation. I approached each chapter as if I were writing a script for a presentation. Although when I do my presentations, it's mostly extemporaneous.

I also wanted the beginner to be able to pick it up and start reading. I think the title of the book is a little misleading. The title *Oracle Performance Tuning 101* created a perception that the content is very basic. But the book offers the rudimentary concepts and the fundamentals of performance optimization on Oracle systems. And I kind of joke about it in my presentations that there's only one way to tune, and that's the right way. You want to use the wait interface to look at the symptoms, which will then

lead you to the bottlenecks within your Oracle system.

However, it all started by accident. I think it was at IOUG of 2001 and while I was walking in the exhibit hall, I passed by the Osborne booth and my dear friend Rachel Carmichael, who co-wrote the DBA 101 book, literally pulled me off the carpet and presented me to one of the editors and said, "Here's the guy you need to write your tuning book." I'm looking at Rachel and saying, "Thanks for that, but why do I need to write a tuning book. There are tons of tuning books out there?"

Rachel had attended the Oracle Performance Management presentation I had given at Oracle World and IOUG that year or the previous year. In that presentation, I mentioned tongue in cheek that any book on performance tuning should be 40 pages or less, and, oh, by the way, the paper for this presentation is 36 pages. So, I said OK and that's how it began.

NoCOUG: *What was the first introduction you had to the wait interface tuning method?*

Gaja: Let me start by saying that I was introduced to the buffer cache ratio methodology while I was at Oracle Education because that's what we taught in the performance management class. During that time, I kept thinking the hit ratios didn't really tell me anything earth-shattering, didn't make much sense, but I didn't know what would make sense until I saw the YAPP Method (Yet Another Performance Profiling Method) paper, which put a different perspective on Oracle tuning. Gotta thank my friend and colleague Anjo Kolk for that. Also, being employed by Oracle at the time gave me access to some of the internal forums where DBA-related questions where asked. There were also a lot of good people that contributed to my learning process and in those days many people said, "Yeah, you can look at ratios, but if you really want to optimize your database, session waits are the keys to the kingdom."

It was more out of curiosity that I went to look at session waits, and said, "Wow, that's pretty neat." The unfortunate thing was that documentation was very skimpy, and even if you looked at the Oracle education class notes (in the mid-1990s), I think there were scores of pages on ratios and about two to three pages on session waits—it seemed almost like an afterthought.

NoCOUG: *Going back in time a little, what part of your formal education do you think contributes the most to your skills working with Oracle software?*

Gaja: I think I have a unique or interesting background, depending on how you look at it. My bachelor's is in business and my master's is in computer science. What I bring from the business degree, is some of the management stuff and all of my statistical courses. Part of the thinking process in performance optimization is data analysis, and the statistical background helps me understand some of these concepts, which I think might have taken me longer without that training.

While obtaining my master's degree, I took computer science courses in operating systems. These courses con-

tribute to my day-to-day job and have become very valuable in performance optimization. But to really understand how Oracle works, you have to keep up with the technology and new releases; you've got to get your hands dirty. I mean you've got to build a database and play with it. Recently, it seems the most productive time to explore the Oracle software is during long transatlantic flights. I find I'm very productive at 41,000 feet. Yeah, I get a heck of a lot of work done; 37,000 to 41,000 feet is my zone.

NoCOUG: *If you were so fascinated by operating systems, why didn't you end up in that field?*

"Yeah, I get a heck of a lot of work done; 37,000 to 41,000 feet is my zone."

Gaja: My first position in the Oracle world was at Owens Corning in Toledo, Ohio, in 1992. On my first day, I walked in as a C programmer and my interest was in writing operating system routines like in my graduate school days. So my first day, my manager—a brilliant guy named Mark Amos—walked up and said, "Hey, welcome aboard, Gaja. You will learn Oracle and here's a set of 12 diskettes, have at it." I didn't even know how to spell Oracle that day; so a bunch of books and a bunch of diskettes later was how I got started. After three months I started building my first application using SQL.

Mark was a great manager. He really knew how to hire and retain people. I still remember a conversation with Mark where I asked him, "Can you draw a picture for me on how client server really works?" And he just drew a stack on the client side and a stack on the server side and correlated the client server to what I call the seven-layer burrito, which was the seven layers in the OSI model and it was fascinating. Everything just started to fall into place. I even saved that picture and have it today. I think that's when I really started understanding how applications worked in the client server environment and why the networking component is required and how it's used and how TCP/IP works.

I never thought I would be a developer and a database administrator and a systems administrator, which I ended up doing while at Owens Corning. At the end of my term there, I was managing six people and I put together backup and disaster recovery plans for some of their systems. I think one of the senior managers was very proud that one of our systems had not gone down in over a year and a half because we just did it right.

NoCOUG: *It seems you enjoy your work so much, it doesn't seem like it's a job to you.*

Gaja: Especially with my current position at Veritas. I don't feel like it's a job, I feel it's my life. I have felt extremely constrained and limited in a couple of my previous jobs. I think I'm very blessed to have the position I have with this company and the empowerment that is given to me to get the job done.

NoCOUG: *It sounds like you may have been frustrated in other positions but what you do technically is part of who you are. Do you know at what age you knew what that was going to be? Did you have that same passion in business?*

Gaja: No, the reason I didn't have that same passion is because I majored in accounting and I hated accounting. My dad thought it was a good idea given the market position those days. So I kind of gave in and said okay, fine, I'll give it a shot. But I hated it; it was just not me.

A strange thing happened in my first internship as an accountant. We did this audit for a good-sized company, a state-owned company back in India, and in the first three weeks I was

> *"A strange thing happened in my first internship as an accountant."*

there, we uncovered a great deal of fraud in the books, which kind of put me in the hot seat. Things got a little bit out of hand with people tailing me and chasing me around. I actually got rear-ended on my motorcycle twice within three days by the same person. They wanted me off the case. There were no threats, they were trying to physically cause damage so that I would be off the case; I had two falls on my motorcycle. So, badly bruised and hurt, I went back to my parents and said, "I don't like it, I want to do something else."

I remember I took my first computer course which used dbase III Plus in 1989. It was a PC-based database. Programming was very natural and very intuitive. I didn't have to think to do my homework and my assignments, and I did very well. Then I decided I wanted to take the full-blown system design course, so I did, and that laid the foundation for me to come to the United States.

NoCOUG: *In the Performance Management Seminar you recently gave at Veritas, you talked a lot about 10g. It would be interesting to learn more about 10g from your experience. What do you think will be the most significant new features in 10g?*

Gaja: Oracle Corporation did a survey a couple years ago at one of the IOUG's to find out where people spend most of their time, and as a result, launched into this effort called Server Manageability. The whole focus behind Server Manageability was to automate all of the mundane tasks and the things that people have to do manually. So the biggest benefit of 10g is automation of some of the mundane database administration tasks.

I think in every new version of Oracle there's this hype that DBAs are going to lose their jobs, but people that have been around for a few releases know better. If anything, I think your job gets more secure because there is more functionality that's added to each release, and therefore there's more to learn. A typical DBA, say, 10 years

A Great Training Day for You this Spring

I am very pleased to share with you that the NoCOUG will be hosting a full-day training for its members, and all other interested Oracle professionals, in Northern California on Thursday, April 8, 2004, at Dublin Library's Community Room in Dublin. Gaja Vaidyanatha, featured in this issue of the Journal, will conduct this training. The core content of this event revolves around Oracle database tuning, which is mainly geared toward DBAs.

A myriad of tuning techniques, tools, and topics such as application tuning, instance tuning, contention tuning, optimization techniques, Oracle 10g, and Gaja's own "radical approach" will be discussed in detail.

Database designers and developers should also gain a great deal of valuable tuning knowledge from this event. They would also be able to apply the learned methods and techniques, and know when to avoid pitfalls that may serve as barriers to what would otherwise be a performing database early on in a database design or development phase.

Your interest and participation level in this program will be very important to the NoCOUG board and the small team directly in charge of promoting, planning, and undertaking any such future training days.

Please visit www.nocoug.org for further details, updates, directions, and registration. Also, please try to register early as seats are limited.

See you all on April 8.
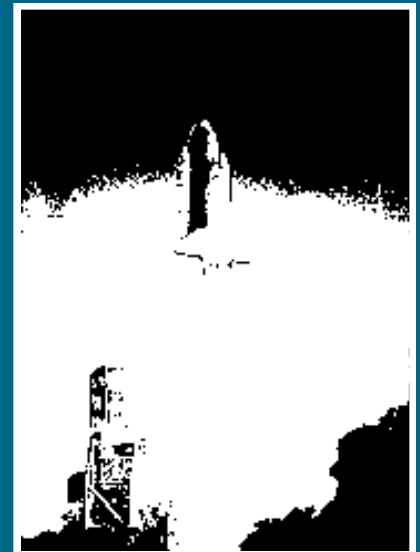
Take off with NoCOUG's great educational opportunities in 2004!

Vilin Roufchaie
Program Coordinator, Board Member At Large,
NoCOUG

ago, maintained a few databases. Today it's not uncommon to find a DBA that manages 50, 100, or more databases. So anything that you can do to make their life easier with the mundane administration tasks is definitely beneficial. You are making the DBA more productive by automating the mundane tasks, but there are still certain things where the knowledge and expertise of a DBA are required, and I don't think that's going to change.

NoCOUG: *What do you consider some of the most mundane DBA tasks?*

Gaja: Rollback segment management, which was fixed with 9i with undo segments. It's a pain! One of the mundane tasks was to determine if you had enough rollback seg-

> *"If I were a DBA manager, the last thing I would want is to have my DBAs constantly fiddle with the database buffer cache or the shared pool."*

ments, size them, and keep track of them. I think it's great that Oracle came up with Undo Management, which will figure out how many segments need to be created, and how they need to be sized, etc.

10g takes it to the next level by actually providing an Undo Advisor that allows you to make a very good estimate as to how large your Undo Tablespace needs to be.

The other mundane part of a DBA job is in space management. Part of that was addressed in 8i with the introduction of locally managed tablespaces, but 10g takes that to the next level by providing the ability to shrink segments in place and online. I would have killed to have that functionality six or seven years ago because when you had to go and shrink a table or reorganize a table, you literally had to go create a copy of the table, and sometimes it was really difficult depending on the size of the table.

The backup functionality has also been improved. I believe there is also the ability to do incomplete recovery at the tablespace level.

On the recoverability side, I believe there is an UNDROP command, as a DROP in 10g will only "mark the object in the dictionary" for the drop. So long as you do not PURGE, you can actually get the table back after you have dropped it. Cool feature!

If I were a DBA manager, the last thing I would want is to have my DBAs constantly fiddle with the database buffer cache or the shared pool. All of that is getting automated: Just the way we had PGA_AGGREGATE_SIZE in 9i, 10g comes along with the concept of providing an "SGA aggregate size" parameter for the SGA. Basically, by setting this one parameter you tell Oracle that it has XGB of memory, and it may use it as it pleases. So the database buffer cache acts like a broker and dishes out memory to the shared pool, to the large pool, and to the Java pool as required. This will be especially useful during batch jobs

where you require more large pools when compared to the OLTP part of a day. Previously, customers had to shutdown and restart the database for the batch window to change the large pool size because it couldn't be changed dynamically. With 10g you don't have to do that with the automated memory management.

In my mind, those are all mundane tasks. As a DBA, I would never want to keep fiddling around with memory parameters. The more Oracle can automate SGA memory management, the better. But I think there is still significant ground that needs to be covered in the application performance management space. It's still not a science and anyone who's built a product or built functionality for performance diagnostics and/or performance monitoring will tell you that.

NoCOUG: *What features in 10g are you really looking forward to exploring?*

Gaja: For me, personally, there are two features that I probably want to dig into: one is the online segment shrink and the other is auto SGA tuning. I think the automatic SGA tuning interests me a lot because I've been a DBA in environments where they've had significant batch job windows where, like I mentioned before, you had to take the system down and bring it back up and start it with a new init.ora. You needed two init.ora files, an init.ora and an initbatch.ora, where for the batch window, the large_pool size was larger. The interesting part of that feature is that I want to see how it actually works for very large systems—SGAs that are in the, excess of 8, 10, 16GB and also environments where there are batch windows every hour. You know there are a lot of applications out there where the first 15 minutes of every hour is like a mini-batch window, so I want to see how this memory management actually works in these scenarios. Online segment shrinks allow you to re-organize your objects "in place" without creating a copy of the object. I think that is very cool. 10g's other new feature I will play with is Active Session History.

NoCOUG: *Is there anything that concerns you about 10g?*

Gaja: It is big! There is a bit of a concern right now on the size of the Oracle executable. It has grown progressively since version 8.0, so you have to be aware of that and make sure you have enough memory on your machine to run the same number of sessions. Please check this out when 10g is released.

Another concern I have is the management of the SYSAUX tablespace: to watch out for the amount of history retained in the Automatic Workload Repository. Some large customers of Oracle can and will retain at least a year's worth of performance history, so given that tablespace is now part of your production database, it affects backup times, and it also affects your mean time to recover (MTTR). I don't want people to assume they can just blindly upgrade from 9i to 10g and that everything is going to be exactly the same—they may find that that's not the case.

NoCOUG: *What gives you the greatest satisfaction at work and/or on the job?*

Gaja: The ability to change, the ability to create, the ability to make things better. At the end of the day, my goal is to impact my product in a positive way so that customers get value out of it. I'm very sensitive to the lack of time people experience these days. A typical DBA may be responsible for dozens, if not hundreds, of databases. You just don't have the time to sit there and mess around with the product too long, to get it to work or to get value from it; it has to be blatantly obvious. And the most exciting thing for me is to constantly make my solution better for my customers.

NoCOUG: *So, as an instructor or presenter, your class or seminar is the product?*

Gaja: Yes, when I deliver a class or seminar, I definitely look at it as a product. I want people to be excited, I want people to be energized, and I want people to go away with the core elements of the curriculum, and that's my philosophy in presentation or information dissemination, or whatever you want to call it. I want people to not only learn something, but to remember that they also had a great time in the process. I apply that to my work with software products because I want people to identify with the product and say, yeah, this product really rocks, it's very easy to use, it's fantastic, and it makes my life so much easier. That's music to a technical product manager's ears, like mine.

NoCOUG: *At this point in your career, what are your professional goals?*

Gaja: I really would like to get to a state where I have a large team I can work with whose priorities I can affect and impact. I do work with a large team now but it's more of an individual role, however, it's my desire to have an impact on a much larger basis. I think the opportunities are tremendous, and one of my goals within the next five years is to be in a senior management position.

NoCOUG: *So, you enjoy management?*

Gaja: I absolutely do. I often like to joke and say that when managers overdo their jobs they become "damagers." When I was at Owens Corning, I had the opportunity to manage a team of six interns. I think it's very gratifying. As a manager, people look to you to provide technical leadership, and when you have the opportunity to work with a team that directly reports to you, I think the working relationship is tighter. I constantly remind myself of the "human aspect" of management.

In my prior position at Quest Software I made it a point to keep track of the birthdays of my fellow team members. Just that one email to them on that special day showed them that I cared enough to keep track of that day in their lives. And you would not believe how much that made a difference. It really showed me that it's the little things that sometimes matter the most. I enjoy it—it's truly a pleasure for me.

NoCOUG: *How do you balance the role of manager and keeping current with the technical aspects of your job?*

Gaja: I definitely do have the job of keeping up technically, and also the management side. The balancing act is very difficult at times; often you get into the management side more, then you just don't have enough time to keep abreast technically. It takes a little bit of extra effort, but I've been able to do it thus far. One way to do it is to become very intimate with your laptop on some very

"Honesty, truthfulness, the ability to learn, the ability to solve problems —these are the qualities I look for in a DBA candidate."

long transatlantic flights from California to the UK and back. If you've got enough battery power, you can spend a few hours playing around with your Oracle database, which I tend to do. Also, I think I mentioned to you, my creativity hits a zenith around 37,000 feet. Sometimes, you just have to be disciplined, and I want to do that. Another way is to block a couple hours in your week, maybe on Friday afternoon, and just say no phone calls, no emails, no nothing in these two hours, heads down and let's dig some stuff up. There never seems to be enough time.

NoCOUG: *What are the top books in your Oracle reference library?*

Gaja: *Scaling Oracle8i: Building Highly Scalable OLTP System Architectures* by James Morle; *Practical Oracle 8i,* by Jonathan Lewis; *Optimizing Oracle Performance,* by Cary Millsap; *Oracle Design,* by Dave Ensor; *Oracle8i*

Internal Services for Waits, Latches, Locks, and Memory, by Steve Adams; Tom Kyte's books (some great nuggets of technical wisdom in them); PL/SQL books by Steven Feuerstein and John Beresniewicz; *DBA 101,* by Rachel Carmichael; *Practical Performance Analyst,* by Neil Gunther; *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling,* by Raj Jain; *Oracle Backup and Recovery Handbook,* by Rama Velpuri.

NoCOUG: *Are there any other publications or resources that you use?*

Gaja: There's a fantastic distribution list: oracle-l. I get to see some very interesting things: people posting problems and the responses, and there are very good people on the list—Cary Millsap, Anjo Kolk, Jonathan Lewis, among others.

NoCOUG: *If you were hiring a DBA, what qualities would you look for?*

Gaja: If I see a resume where the word "expert" shows up, I take off a few points (laughs). I've done a few interviews in my life trying to get some good people. I look for their ability to learn, the openness and the aptitude. I don't have a problem asking a technical question for which they honestly come out and say "I don't know"—that's okay. However, if they start making up stories, that's definitely a red flag. I don't like people who lie on their resume either; I've caught people lying on their resume. I remember one interview where the candidate claimed to have been an Oracle Parallel Server DBA for eight years and my first question in the interview was, "Can you describe for me an environment where a database is supported by multiple instances?" And his response was, "Oracle doesn't have a configuration that supports multiple instances." And, that's where the interview ended.

BAARF—It's Not What You Think

If you attend NoCOUG's Training Day on April 8, featuring well-known Oracle expert and author Gaja Vaidyanatha, you'll have to ask him about BAARF. Otherwise known as "Battle Against Any Raid Five," this was the brainstorm of a couple of members (James Morle and Mogens Nørgaard) of the Oak Table Network having a beer or two back in June. They say that, "RAID-3, 4, and 5 are all included in the initiative, so the F in BAARF both stands for Free, Four, and Five."

Online they also state, "The reason for BAARF is that we've had it. Enough is enough. For 15 years a lot of the world's best database experts have been arguing back and forth with people (vendors and others) about the pros and cons of RAID-3, 4 and 5." For more BAARF news, see www.oaktable.net/fullNews.jsp?id=8.

Oak Table Network

Gaja is also a member of the Oak Table Network and has participated in some very amusing conference skits on the topic of BAARF. The Oak Table Network consists of a number of leaders in the Oracle user community. It was formed in Denmark over the oak table of Mogens Nørgaard. A major technical conference was taking place, and after hours a group of "leading lights" gathered for many discussions about Oracle topics in Mogens' dining room. The oak table was "the piece of furniture which had supported eleven laptops, 800 metres of cable and numerous bottles for those long nights." For more information— and amusing stories—check out www.oaktable.net. ▲

Honesty, truthfulness, the ability to learn, the ability to solve problems—these are the qualities I look for in a DBA candidate. I don't care much about the certification. I personally have known many folks who have the experience but are not certified. I'd rather go with the person that's got the experience because at the end of the day certification is just an exam. I've known people who have never done any real life work, studied the exam cram books, and gone and passed the tests. I think that's absurd. You have to be able to apply knowledge. It's the application of the knowledge where the difference is. Otherwise just taking exams and getting certified is completely useless.

NoCOUG: *Do you have any role models in the technology field, and who are they and why?*

Gaja: Mark Amos from my Owens Corning days and Michael Dell of Dell Computers.

Mark Amos because he was probably the best manager I've ever had. I knew exactly what I needed to do, I was measured on that, and there were no curve balls thrown at me. From an employee-manager perspective, there were no surprises. When I sat with him to do the periodic review I knew exactly the outcome. He had the technical knowledge; the man could never resist getting his hands dirty. I remember there were times I would hit a brick wall, and I could ask him for help and he knew just how to help me solve the problem. There were times he would fix the code, or send me on a path that would help me eventually resolve the problem. The technical aspect was great, but more importantly, he was just a good person. He is a man with a good heart. He was able to motivate me and he never took me for granted as in a body shop. He was very concerned for my well being and progress. He had terrific skills in communication, technical knowledge, technical leadership, and the ability to motivate from a human standpoint. I've known very few managers who invited the whole team to their homes for dinner. Hopefully, I'll be a manager like him someday.

Michael Dell because he is an absolute visionary in the way he built a computer hardware company, the way he managed the company and how it grew. The ability and insight in meeting the customer needs, managing his inventory in a stellar fashion, delivering products on time and at a reasonable price are great skills. Sometimes people end up building a Caesar's Palace when the customer requested a courtyard. You have to understand what the customer really wants and you don't have to go on this huge project of trying to be everything for everyone. It seems like a simple concept, but many people just don't get it. Michael Dell has got that right!

NoCOUG: *What has been the greatest influence in your life?*

Gaja: The influence of my parents. They are the primary reason for my success and where I am today in my career. They made great financial sacrifices to get me to the U.S. for my master's degree. I don't believe I could have asked for better parents; they symbolize the epitome in parenthood. For them, my education and career were paramount over everything else, including their basic comforts. Such levels of sacrifice are rare—I can't thank them enough.

NoCOUG: *What do you do to relax?*

Gaja: Music helps me relax, as far as I can remember. I have studied for every exam that I know of with music in the background. It just helped me learn much better. Reading thought-provoking books is relaxing too. May I suggest *The Celestine Prophecy,* by James Redfield. He has written many sequels to that. Great books!

Also, one of the greatest relaxing experiences for me is being in the wilderness. When I go hiking I try to find a spot where I can get some quiet time for an hour or so, and just relax there listening to the birds, looking at the flora and fauna.

I'm also a scuba diver and I recharge very quickly on my dives and being one with nature. I also enjoy riding my motorcycle on weekends, riding into the mountains, say, to Big Basin, and finding a quiet spot in a park to reflect on life.

NoCOUG: *And finally, what do you think attendees of the upcoming NoCOUG Training Day will walk away with at the end of the day?*

Gaja: I want it to be the best eight hours of technical training they've ever had. The content of the presentation, the humor, networking, the environment; I want everything to stand out. The entire seminar is applicable to any version of Oracle. What I want to do is provide snippets and nuggets of information, where applicable, about what's new in 10g. However, I think this seminar is worthwhile regardless of the version of Oracle you are running. That's the beauty of the basic elements of this methodology; you can apply them with 8i, 9i, or even 10g. ▲

Gaja can be contacted at gajav@yahoo.com

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. ▲

Long-term event sponsorship:

LOCKHEED MARTIN

CHEVRONTEXACO

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

$ TREASURER'S REPORT

Judy Lyman, *Treasurer*

Beginning Balance		
October 1, 2003		**$ 54,359.27**
Revenue		
Membership Dues	1,843.55	
Meeting Fees	580.00	
Vendor Receipts	3,950.60	
Advertising	-	
Interest	22.33	
Total Revenue		**$ 6,396.48**
Expenses		
Regional Meeting	4,823.16	
Journal	5,510.18	
Membership	21.52	
Administration	1,020.00	
Website	75.00	
Board Meeting	542.45	
Miscellaneous	750.00	
IRS	-	
FTB Tax	-	
Total Expenses		**$ 12,742.31**
Ending Balance		
December 31, 2003		**$ 48,013.44**

The Case of the Missing Kanji

By Brian Hitchcock, Sun Microsystems

How did all this get started? I'll tell you how. The interviewer wore a very loud Hawaiian shirt—that's how it all started. I should have been more careful. I assumed this meant exotic job locations. It meant working out by the salt ponds, where watching salt water evaporate is a large-scale industrial concern.

I knew Sybase as well as Oracle and that meant I was the perfect "volunteer" for the migration. It all sounded so simple. They couldn't find anyone else with this strange set of qualifications. I had been chosen. I used Oracle Migration Workbench, which worked very smoothly. The migration went well, perhaps too well.

And then it happened. Of all the DBA joints in all the corporate world, she walked into mine. Christine said little. "Brian, the Kanji data are missing." I knew I was in trouble. Up to that point, no one had mentioned the minor detail that there was multi-byte Japanese Kanji in this innocent looking single-byte database.

The "expert" DBAs at corporate analyzed the situation in the predictable way. "Brian, a single-byte database can't support multi-byte Kanji." And that was that. I was stuck, in the middle of nowhere, between Christine's need for her Kanji, and the experts' declaration that the Kanji had never existed.

This left only one small detail. How to get the Kanji back?

Before the upgrade to Oracle, Christine's application had been working for years without a problem. The application retrieved data from the single-byte Sybase database and sent the data to the user's browser where the Kanji were displayed. Everyone was happy. When I moved the data from Sybase to Oracle, I had no reason to worry. I created the Oracle database using the WE8ISO8859P1 character set to match the ISO1 character set of the existing Sybase database.

The way the application worked is shown in Figure 1. The source system was sending Japanese Kanji characters into the Sybase application database. The application would retrieve data from the Sybase database to generate HTML that was sent to the end user's browser. Notice that in the lower left of the figure, I have included a single Kanji character. It would turn out later that the Japanese Kanji data for this application were from the EUC-JP character set. I have included the byte code for this specific Kanji character in the EUC-JP character set, which in hexadecimal is B0A1. Notice that this byte code for this specific Kanji character was stored as B0A1 in the source system and was sent to the Sybase database as B0A1. The application retrieved this double-byte byte code and sent it to the end user's browser. The Netscape browser examined this byte code and then displayed the proper Kanji character. Notice that the Netscape browser was configured to use the character set called Japanese auto detect (under the View menu, Character Set). The Netscape browser looked at each byte to be displayed, and if the eighth bit was set to 1, it assumed that this was the first byte of a double-byte byte code, a byte code for a Japanese character. By following this specific byte code for this specific Kanji character as it moves through the application, we can see how the problem developed.

Brian Hitchcock

I moved the Sybase data to the Oracle database using Oracle Migration Workbench, which generated SQL*Loader scripts for each table in the Sybase database. I then executed these scripts. Everything ran well with no errors. I created the Oracle database using the WE8ISO8859P1 character set to match the ISO1 character set of the existing Sybase database. The SQL*Loader scripts were executed and I didn't worry about character sets, which meant the client environment where SQL*Loader was executing used the default character set, which for Oracle is US7ASCII. The US7ASCII character set uses a single byte to represent each character. Within each byte, only seven bits are used to represent the
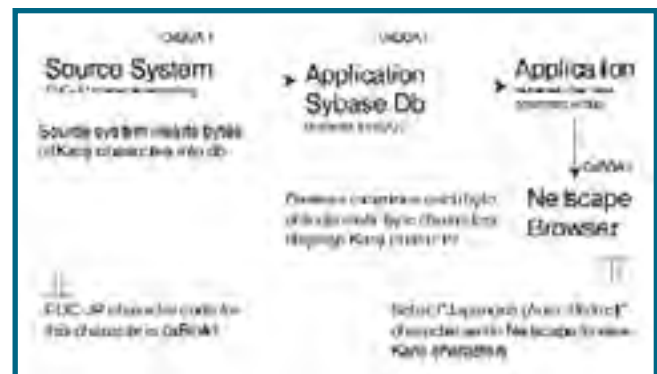

Figure 1. Application Before Upgrade to Oracle

binary code for each character. The highest bit, the eighth bit, is set to zero for every character in the character set. Another way to look at this is that the decimal value for all the US7ASCII characters is in the range 0 to 127.

When you use any Oracle utility, such as SQL*Plus and SQL*Loader, the environment parameter NLS_LANG determines the character set used by the client utility program. In this case, since I didn't set NLS_LANG, it defaulted to US7ASCII.

However, it turns out that the Sybase database had also been created with the default character set, which for Sybase is called ISO1. Notice that different vendors have very different names for character sets that may turn out to be the same. In this case, the Sybase character set called ISO1 is the same as the Oracle character set called WE8ISO8859P1. This character set uses eight bits to represent each character. The Oracle database where I was loading the data had been created with the WE8ISO8859P1 (Oracle version of the 8859-1) character set. This was done simply to match the existing production databases.

Notice that the existing Sybase database had the byte code for our Kanji character of B0A1. This byte code was then moved into the flat file produced by the Sybase BCP utility. This flat file was loaded by the Oracle utility SQL*Loader into the Oracle database and the byte code changed from B0A1 to 3021. Once this byte code was stored in the Oracle database, the application retrieved the two bytes of this byte code, generated the HTML, and the byte code 3021 was displayed in the Netscape browser as two characters (two ASCII characters) zero followed by an exclamation point. Clearly things were not correct.

What was happening was that ASCII data was being moved into a database that was created with the WE8ISO8859P1 character set. Therefore, when I executed Oracle SQL*Loader with NLS_LANG defaulting to US7ASCII, the Oracle software looked at the data in the SQL*Loader datafile and treated each byte that was loaded as if it came from the US7ASCII character set. This is exactly what Oracle was being told to do. It also meant that Oracle was setting the eighth bit of each byte to zero before loading it into the Oracle database. There were no error messages generated. As far as Oracle was concerned, everything was fine. Any byte in the Sybase database that had the eighth bit set to 1 was loaded into the Oracle database with the eighth bit set to zero.
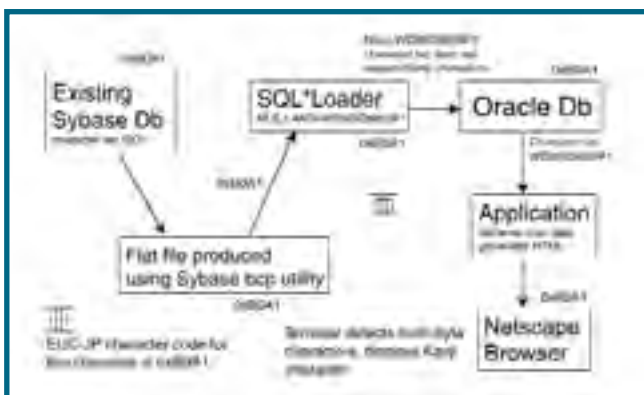

Figure 2. Fix—Sybase Data to Oracle

The bit-by-bit details of what was happening are tedious, but worth following. Starting with the correct two-byte byte code in hexadecimal for our selected Kanji character (B0A1), the binary digit equivalent of B0A1 is 1011 0000 1010 0001. When we change the eighth or highest order bit to zero for each of the two bytes, the resulting binary digits are 0011 0000 0010 0001, and the hexadecimal equivalent of these modified binary digits is 3021. The byte code 3021 corresponds with the character zero followed by an exclamation point.

This wouldn't have been an issue, except for the fact that we had multi-byte Kanji stored in the single-byte Sybase database. Each Kanji character is represented (in this specific case of the EUC-JP character set) by two bytes. The way the Kanji characters are encoded into two bytes causes the eighth bit of both bytes to be set to 1. (Each byte has a decimal value of 128 to 255.) As far as Sybase was concerned, these weren't Kanji characters at all, they were simply two bytes in a single-byte database. Somehow, these two bytes were retrieved by the application and sent to the user's browser and the Kanji character was displayed. Note that the Sybase character set ISO1 uses all eight bits for each character, so the bytes stored in the Sybase database were both correct ISO1 characters. The trick was that when these two bytes were sent to the user's browser, they were somehow combined and instead of two ISO1 characters (a-z, A-Z and some special characters for European languages) a single Kanji character was displayed.

When these two bytes were loaded into the US7ASCII Oracle database, the eighth bit of both bytes was set to zero. The two bytes of the Kanji character now appeared to be two characters from the US7ASCII character set. The user's browser was not able to display a Kanji character from these two seven-bit bytes. This is how the Kanji were lost.

Now that I could explain what had happened to the Kanji, the immediate concern was how to fix it. The answer comes from understanding how Oracle treats character data when moving between SQL*Loader and the database.

Oracle looks at NLS_LANG in the client environment where SQL*Loader is executed, and compares that with the character set of the database. If the character set and NLS_LANG both specify the same character set, Oracle doesn't do anything to the bytes, it simply loads them into the database. Note that it doesn't matter if the bytes represent correct characters in the specified character set. Oracle doesn't compare the bytes to the character set of the database or the client, it simply loads the bytes. In this case, SQL*Loader was using US7ASCII while the database had been created with the WE8ISO8859P1 character set. Therefore, the fix was simply to modify the client environment so that the ISO1 character set was specified. For Oracle, the equivalent character set to Sybase ISO1 is WE8ISO8859P1. Once this change was made, Oracle detected the same character set at the client and at the database and no changes were made to any of the bytes.

Figure 2 shows the situation after the fix was made. In the client environment where the SQL*Loader utility was executed, NLS_LANG was set to WE8ISO8859P1. Now that the character sets were the same between the SQL*Loader client environment and the Oracle database that SQL*Loader was
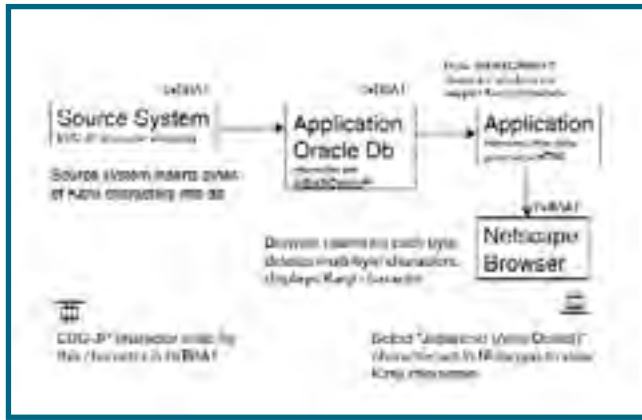
Figure 3. Current Oracle Production

Figure 4. Existing Application

connected to, Oracle made no attempt to examine or convert the bytes as they moved from the flat file through SQL*Loader into the Oracle database. The byte code of our selected Kanji character (B0A1) moves through the system without being changed, and therefore was loaded into the Oracle database as B0A1. The application could then retrieve this byte code and send it to the Netscape browser where it was displayed as the correct Kanji character.

At this point, the bytes that represented the Kanji characters were loaded into the Oracle database. The bytes of each Kanji character that had been in the Sybase database were now in the Oracle database. Remember that both the Sybase ISO1 and the Oracle WE8ISO8859P1 character sets are single-byte character sets. Neither of these character sets supports (officially) multi-byte character sets. On the other hand, since the application was able to retrieve Kanji characters from both databases, it was clear that something else was happening.

After the migration from Sybase to Oracle, the production application was working and everyone was happy. The fact that the database character set didn't support Kanji (multi-byte character data) was not a concern to anyone, except for someone out by the salt ponds, but no one listened to him anyway.

Figure 3 shows the production system after the Sybase data had been successfully loaded into the new Oracle database. The byte codes for individual Kanji characters were moving through the entire application and were correctly displayed in the end-user's Netscape browser.

As exciting as it is to watch the salt brine evaporate, leaving salt behind, I was curious how the existing application was able to do what the experts said couldn't be done. We were in production with a system that didn't officially support Kanji, but they were being stored and displayed on a daily basis. Since I wasn't an expert, this bothered me.

To figure out what was going on required examining all the pieces of the application. The multi-byte Kanji characters were inserted into the Oracle database by another application. Further proof that Oracle (and Sybase) doesn't examine the bytes of character data as they move into and out of the database. The application code itself is Java. The Java code uses JDBC to retrieve the character data from the Oracle database. Java converts the character data to UCS2 (Unicode, two bytes per character) and generates HTML that is sent to
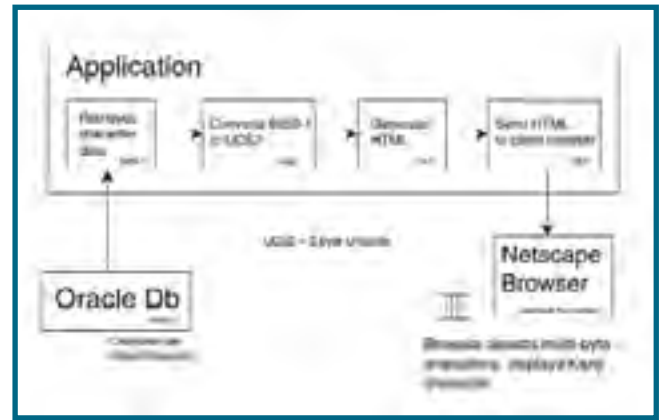
the user's browser. The Netscape browser then examines the bytes sent to it. You can choose a character set within the Netscape browser, and for this application, the user always uses the "Japanese (Auto-Detect)" character set. This means that the browser examines the bytes of the character data. If it "detects" bytes where the eighth bit is not -zero, then it assumes that these bytes are part of a multi-byte character. Since the browser is "detecting" Japanese characters, it assumes that any multi-byte characters are Japanese, and displays them as Kanji. Also note that while Netscape was able to "detect" Japanese multi-byte character data, this only worked because all the multi-byte character data was Japanese. If there had been, for example, Chinese character data, which would also be multi-byte, Netscape wouldn't be able to "detect" Chinese versus Japanese character data and the application would not have worked. Figure 4 shows all the pieces of the application, and the various character sets that were being used by each piece. Notice that each component of the overall application was either configured for a specific character set or was defaulting to a specific character set.

I was also curious about the source data itself. What exactly was the character set used when the original Kanji character data was inserted into the database? And how could I determine this? I wanted to know what exactly had happened to get us into this mess. I looked at the actual bytes for a sample of the data in the database. I had to learn how to convert between the many different character sets that support Japanese Kanji characters. I also looked at the actual Kanji characters displayed by the application for the sample data I was examining. I then found the Kanji characters in a Japanese character dictionary, which offers the row-cell values for each Kanji character. I then converted the row-cell values into the equivalent bytes for several character sets that support Kanji and compared them with the bytes of my sample data. I then verified that the source data came from the EUC-JP character set. Note that in order to do this, someone had to go through all these steps. There are no Oracle utilities that would help me.

After this, I could see what had happened. The original Kanji characters were encoded in the multi-byte EUC-JP character set. The bytes of the Kanji data were inserted into the Oracle database (as they were before in the Sybase database). The application retrieved these bytes, and the client

browser looked at the bytes and detected multi-byte data.

Perhaps I was done? Perhaps Christine would be content and I could return to the salt ponds? No. At the corporate level (where the experts live) the decision had been made that this application, as well as many others, should be converted to support multiple languages. The application was to be "internationalized." The standard way to do this is to convert the Oracle database to the UTF8 character set which has unique bytecodes for the characters of many, if not all, the world's languages. The UTF8 character set is Oracle's implementation of the UTF-8 encoding scheme for the Unicode character set. A good idea, but this was coming from the same experts that said there couldn't be any multi-byte data in the existing single-byte databases.

The standard way to convert an Oracle database from one character set to another is simple. I would export the existing Oracle database (character set WE8ISO8859P1) and import into another database that had been created using the Oracle UTF8 character set. The experts all agreed that this was all that would be needed.

After this process was complete the application was tested and it was able to retrieve and display the existing Kanji data. Everyone was happy. Everyone, except for that someone out by the salt ponds.

Since I had been deceived before (the original Sybase database didn't contain any multi-byte data, remember?), I decided to look at what the export/import process would do to the bytes of a Kanji character.

I needed to examine the process that is used to encode a character in the UTF8 character set. I had to learn that the term "UTF8" means many things. UTF-8 is an international standard for encoding character sets. This means you can create any set of characters you like, and use the UTF-8 encoding method to encode your set of characters into bytes for storage in a computer. This means that the term UTF-8 doesn't refer to any specific character set. At the same time, the Oracle 8i Unicode character set is also called UTF8 and refers to a very specific set of characters.

The UTF-8 encoding process takes the Unicode code point (the hexadecimal number of the character) and encodes that value into one, two, or three bytes. For ASCII characters, UTF8 uses one byte. Two bytes are used to encode the Unicode characters up to U+00FF, and three bytes are used for Unicode values above that. UTF8 uses three bytes for

Figure 5. Conversion Issue

Asian characters. Note that UTF8 doesn't use more bytes for the ASCII data. Converting to UTF8 character set will not increase the size of the ASCII character data. The bit-by-bit process of converting the EUC-JP byte code of our selected Kanji character to the new UTF8 byte code for the same character is as follows: We start with the EUC-JP byte code (two bytes), which is B0A1. The Unicode code point (byte code) for this same Kanji character is 4E9C. You can review various Oracle Metalink documents for the details of this conversion process. The Unicode byte code of 4E9C requires three bytes in the UTF8 encoding process. The binary digits for 4E9C are 0100 1110 1001 1100, and when encoded into UTF8 this becomes 11100100 10111010 10011100 or E4BA9C. For the same Kanji character, the bytes for EUC-JP are B0A1 and in UTF8 they are E4BA9C. Note that the byte code was two bytes in EUC-JP, and it became three bytes in new UTF8.

There are multiple character sets for Japanese and other Asian languages. The same Kanji character has a different byte code in each of these character sets. I had to figure this out to know what the bytes should be in the UTF8 database. Details of this process are covered later on. I used this to compare with the bytes of the original Kanji character data to determine that they came from the EUC-JP character set.

When the existing data was exported from the WE8ISO8859P1 database, the export utility assumed that each single byte exported represented a single character. As each byte was inserted into the UTF8 database, Oracle converted each single byte into the equivalent UTF8 byte or bytes. This means that the two bytes for a Kanji character in the existing database were converted one byte at a time. Each byte became two bytes in the UTF8 database. The correctly encoded UTF8 bytes for the same Kanji character consist of three bytes. I now had evidence that simply exporting from WE and importing into UTF8 did not produce the correct results. The bytes in the UTF8 database were not the correct bytes for the Kanji in the UTF8 character set.

I had to go through the conversion manually to see what was happening as the import process moved the character data into the UTF8 database. Oracle took each single byte that was exported from the WE8ISO8859P1 database, determined the Unicode character and then applied the UTF8 encoding process. This is very different from what should have been done. The two bytes of the Kanji character in the WE8ISO8859P1 database should be exported together. The Unicode character for these two bytes should be determined. Then, this Unicode character should be put through the UTF8 encoding process to generate the correct UTF8 bytes (three bytes) for the Kanji character.

Manually generating the UTF8 bytes for each single byte of the existing two bytes of a single Kanji character demonstrates exactly what happened. Starting with the two bytes of our EUC-JP Kanji character, B0A1, the import process took each of the two bytes one at a time. The first byte, B0 was converted to UTF8, which results in C2B0, and the second byte A1 becomes C2A1. The result of importing into the UTF8 database was to change B0A1 into four bytes C2B0C2A1. Notice that the byte code for our selected Kanji character, B0A1, which is two bytes, became four bytes after the export file was imported into the Oracle database. This
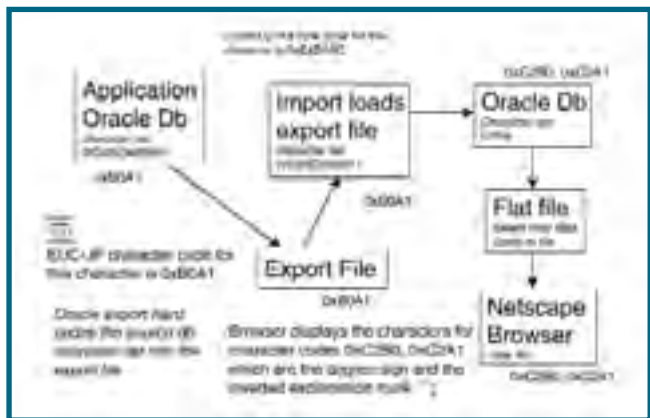
process is shown in Figure 5.

This explains how the Kanji data would be corrupted (lost) in the UTF8 database. The two bytes of the Kanji character in the WE8ISO8859P1 database were converted by export/import into four bytes in the UTF8 database. These four bytes represented two characters in the UTF8 database. The correct (manually generated) UTF8 encoding for the same Kanji character has three bytes E4BA9C.

What had happened? Oracle did exactly what it was told to do. The export utility looked at the database it was exporting from and saw the single-byte character set WE8ISO8859P1. Oracle correctly determined that each byte coming from this database represented a single character. The import utility inserted each byte into the UTF8 database. Oracle compared the character set of the export file to the character set of the UTF8 database and knew that character set conversion was required. Oracle then took each single byte from the export file and converted that single byte to the correct UTF8 set of bytes. But, this wasn't the correct set of UTF8 bytes for a Kanji character.

But, Christine was happy to report that the application was working just fine with the existing data that had been exported from the WE8ISO8859P1 database and imported into the new UTF8 database.

I had to tell her that maybe not today, maybe not tomorrow, but someday, during the rest of the life of this application she'd regret it.

Christine was not amused. Why couldn't it work? Why not for us? Why worry about the future? Couldn't we just be friends?

Because I'm not an expert, I had compared the correct UTF-8 bytes for a single Kanji character with the bytes that resulted from export/import for the same character. The bytes were not the same. But so what? Who cares?

The problem comes in the future. Up to this point Christine and I had only been concerned with the existing data. After the existing data was exported and imported into the UTF8 database, the application worked fine. But, what would happen when, in the future, some process inserted the correct UTF8 bytes for this same Kanji character into the UTF8 database? At that point, the UTF8 database would contain, for the same Kanji character, four bytes from existing data, and three bytes from newly inserted data. How would the application know which was which? Further, recall that the existing application works only because the Netscape browser is configured to use the Japanese (Auto-Detect) character set. Is the application really "UT-F8" if it requires using a specific Japanese character set in the browser? Is the application really "UTF8" if it has both correctly and incorrectly encoded bytes for the same character?

How could the existing application work with what I believed was corrupted (four bytes versus three bytes) Kanji character data? I had to review how the application worked to see the answer. The application uses Java and JDBC to retrieve the bytes from the UTF8 database. Java then converts these bytes to the Unicode character, generates HTML, and sends the HTML to the user's browser as shown in Figure 6.

We now had four bytes in the UTF8 database that repre-sented the UTF8 conversion of each of two bytes of the original Kanji character data in the EUC-JP character set. When Java converted these four bytes into Unicode, it got the correct European characters for each of the pairs of bytes. Note that these two characters are exactly the same characters that were inserted into the original Oracle WE8ISO8859P1 database. The application sent these two characters in HTML to the browser. The browser then detected these eight-bit bytes and displayed the correct Kanji character!

What had happened was that the bytes of each Kanji character were converted to UTF8 (one byte at a time) when they were inserted into the UTF8 database. At this point, the original two bytes of a single Kanji had been encoded to UTF8. Java then reversed this process, which simply returned the original two bytes. The browser then detected the two bytes and displayed the correct Kanji character.

Overall, the export/import to the UTF8 database hadn't really changed anything. The application still delivered the same two bytes to the browser that it did before UTF8 had entered the process.

I decided to test the existing application by inserting into the UTF8 database the three bytes that represent the cor-rectly encoded UTF-8 for a specific Kanji character. The application didn't work!

The existing application was able to work with the existing data that had been exported from WE8ISO8859P1 and imported into the UTF8 database. But, the existing application didn't work when it tried to work with correct UTF8 data. I even tried changing the browser character set to UTF8, but still, the browser did not display any Kanji characters.

At this point, what had I really accomplished? When I inserted correct UTF8 bytes for a specific Kanji into the UTF8 database, the application didn't work. Simply export-ing data from the WE8ISO8859P1 database and importing into a UTF8 database resulted in corrupted (incorrect) bytes for Kanji characters. I needed to understand what the appli-cation was doing to figure out how to correct it. I believed the application should have been able to display Kanji if the correct bytes for UTF8-encoded Kanji were in the database.

How could I debug the application? The application was far too complex to attack directly. Instead, I simply avoided the problem by asking for help from one of the developers. Angela (the Java Diva—yes, she tours and has an entourage) wrote a simple Java servlet. This servlet simply retrieves the
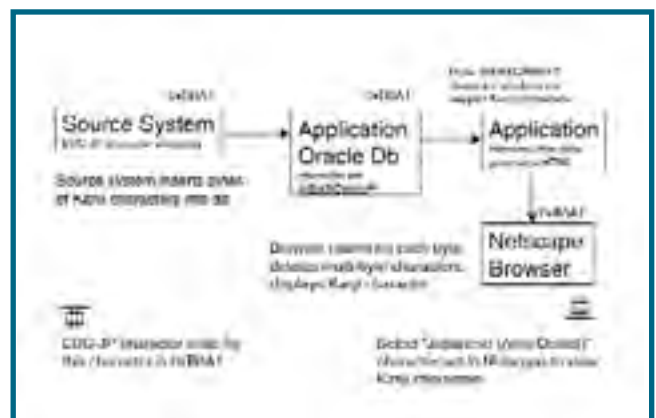

Figure 6.

bytes (character data) from the UTF8 database and then generates and sends HTML to the browser. If I could fix the servlet so that it *did* work with correct UTF8 bytes, then I would (perhaps) know how to fix the application.

I needed help modifying this servlet to work with UTF8. I appealed to the Java community and they had the answer. It turns out I needed to modify the servlet code that generates the HTML. The usual suspects were rounded up. Examples of specific modifications to the servlet code to make it work correctly with UTF8 character data are shown below.

```
res.setContentType("text/html;charset=UTF-8");
PrintWriter out = new PrintWriter(new
OutputStreamWriter(res.getOutputStream(),
"UTF-8"),true);
out.println("<META HTTP-EQUIV=" + DQ +
"Content-Type" + DQ + " CONTENT=" + DQ +
"text/html; charset=utf-8" + DQ + ">");
```

The Java code, just like the database and so many other pieces of the application, was using a default character set, which was eight-bit and used a single byte for each character. The fix was to specify UTF8 as the character set. The modified servlet code then worked correctly. The modified servlet code would retrieve the bytes that correctly encoded the Kanji character, and the correct Kanji was displayed in the user's browser using the Netscape UTF8 character set.

The next step was to fix the application. The similar portions of the application code were located and the same modifications were made. Now the modified application would work with correct UTF8 bytes and they were displayed properly. However, I then discovered that the application also retrieves web content from yet another system, and that system didn't have a character set specified so it was defaulting as well. This other system had to be reconfigured for UTF8 and everything worked.

But now the question was, had I really fixed the application? I had a modified application that worked properly with correctly encoded UTF8 data, but what about the existing data? Using the modified application, existing data (data that was exported from WE8ISO8859P1 database, imported into UTF8 database) was not displayed correctly. Further proof that the existing data was not correctly encoded UTF8 for Kanji characters.

I needed to fix the whole application, including the application data in the UTF8 database. I traced the flow of the

Figure 7.

existing data through the modified application, as shown in Figure 7. The four bytes that were in the UTF8 database that were the UTF8 encoding of each of the two bytes of the original Kanji were retrieved by the Java code. The Java code converts them to Unicode, which results in the two bytes that were originally inserted into the WE8ISO8859P1 database as the EUC-JP bytes for the Kanji character.

The modified application then generates HTML, but, since the application now specifies UTF8 for the character data being sent in the HTML, the two bytes in Java are converted to UTF8. The same four bytes that were in the UTF8 database for the Kanji character were sent to the browser. The browser is configured to display UTF8. The UTF8 characters displayed are the degree sign and the upside down exclamation point, very different from the correct Kanji character.

While the application itself was fixed, the existing data in the UTF8 database was not correctly encoded UTF8 character data. How would I go about fixing the existing data in the database? I had a plan. I had the application working with correct UTF8 data. The only thing left was to correct the existing data in the UTF8 database. I knew how the original Kanji data that was from the EUC-JP character set had been stored in the WE8ISO8859P1 database, and how these bytes were converted (corrupted) as they were imported into the UTF8 database. I needed some way to convert the existing bytes in the UTF8 database back to what they had been (EUC-JP) and then convert them correctly into UTF8.

It was time to review the various character set encoding schemes that we had seen during this project. The original EUC-JP data had two bytes for a single Kanji character, and there were two bytes in the WE8ISO8859P1 database. The two bytes became four bytes when imported into the UTF8 database. The correctly encoded UTF8 for the Kanji character had three bytes.

I needed to convert the existing data, but how? I could see several possibilities to fix this. Why not fix the problem in the WE8ISO8859P1 database before exporting? The problem with this is that there really wasn't much point. Whatever I did to the data in the WE8IOS8859P1 database, the export would still be an export from a single-byte database. This means import would still load one byte at a time, and each individual byte would be converted to UTF8 as it was imported into the UTF8 database. This would corrupt the bytes no matter what I did in the WE8ISO8859P1 database.

Another option would be to not export/import at all. Instead, use SQL select statements to generate a flat file for each table in the WE8ISO8859P1 database. When each of these flat files is loaded into the UTF8 database using SQL*Loader, I could use the SQL*Loader option CHARACTERSET JA16EUC, where JA16EUC is the Oracle character set that uses EUC-JP encoding. As SQL*Loader loads the bytes, they would be read in using the JA16EUC character set. The two bytes in the flat file for a Kanji character would be correctly recognized as a multi-byte character and would be correctly converted to UTF8 in the UTF8 database. The reason I didn't like this option is that I would have to repeat this for every table.

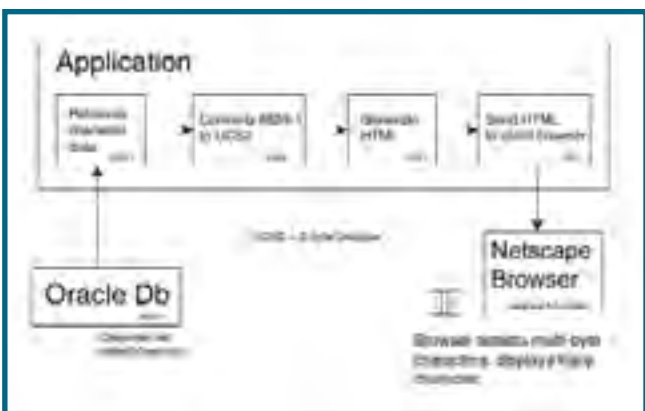Yet another option would be to wait until after the data

was in the UTF8 database. I could fix the data after the export and import process had been completed. This option is illustrated in Figure 8. This would retain the simplicity of a single export and import to get all the database objects from the WE8ISO8859P1 database into the UTF8 database. I found the Oracle SQL CONVERT ( ) function that could do this. The CONVERT ( ) function can convert character data between any two of the many Oracle character sets. The CONVERT ( ) function works on one column of one table at a time, although a single update statement could work on multiple columns at once. I did test the CONVERT ( ) function to demonstrate that it did indeed correctly convert the existing four bytes in the UTF8 data (the four bytes of a Kanji character that had been imported into the UTF8 database) into the correct three bytes.

Now I had a process to correct the data. Export from the WE8SIO8859P1 database, import into the UTF8 database, and after the import, use the CONVERT ( ) function to convert each column of each table that contained multi-byte data.

The Oracle SQL CONVERT( ) function can make one conversion at a time. I needed to make two conversions. The first conversion needed to get from the existing UTF8 character set back to WE8SIO8859P1. This had the effect of changing the data that had been imported into the UTF8 database back to the bytes as they existed in the WE8ISO8859P1 database. The second time CONVERT( ) was used to convert from WE8ISO8859P1 to UTF8. If something were to interrupt this process, it is important that CONVERT( ) not be re-used. It is easy to corrupt the
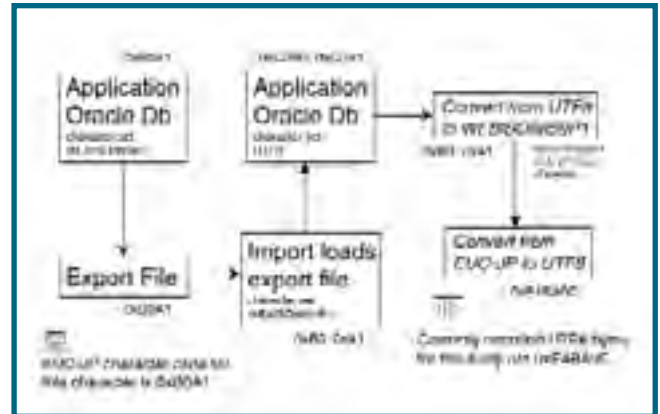
Figure 8. Fix After Import

data if CONVERT( ) is run multiple times on the same data. If needed, I would have to regenerate (export, import) the data and re-run the complete CONVERT( ) process to ensure the correct result.

With this done, I had a complete process to convert the existing WE8ISO8859P1 database to UTF8. The only additional steps were to widen the columns in the UTF8 database that would hold multi-byte data. I had seen that UTF8 could use up to three bytes to store a single Unicode character. Therefore, I widened any column that would contain UTF8 to three times the original size. It is important to realize that Oracle stores bytes, not characters. If a column was created to store 10 characters and was sized as 10 bytes before, it should be resized to 30 bytes as part of moving to UTF8.

The conversion process would then be like this: export

the WE8ISO8859P1 database, import into the UTF8 database. For this import, use the ROWS=N option. This means import will create all the database objects, but won't import the table data. I would then widen the columns as needed and import again but using the IGNORE=Y option. This causes import to ignore the errors caused when import tries to create a table that already exists, and the table data would get loaded. I would then run the two-step CONVERT( ) process to correct the data.

Note that the conversion process also includes modifying the application code and any other pieces of the application that could have an affect on the character set used for processing character data.

One of several details that haven't been explained is how I knew which of the many Japanese character sets had been used to generate the Japanese character data that was in the original Sybase database. I had figured out that the original Kanji character data was from the EUC-JP character set. How did I determine this? In this crazy world how could anyone really know anything? What I did was to take a sample of the Kanji data from the original (existing) application. I identified one record in the existing Sybase database. I displayed the Kanji for this record using the original application. I then found the first Kanji character in a Japanese character dictionary (count the radicals!). This dictionary also listed the row-cell location of this Kanji in the JIS-0208 character set. Using multiple sources, I had to learn how to convert from JIS-0208 to other character sets. I generated the bytes for several of these character sets and compared them to the bytes stored in the Sybase database. This showed me that the original Kanji were from the EUC-JP character set.

Note that there is no Oracle utility that helped me with this. I had to figure this out on my own. The Oracle NLS manual provided the bytes for multiple character sets for a single Kanji, which I used as a check of my work. This was my Rosetta stone.

I needed several reference books as I worked to identify the Japanese character set that had been used. In addition to the Japanese character dictionary, I used the Unicode standard and the O'Reilly book, *CJKV Information Processing*, to learn how to do the needed character set conversions.

When it was all over, what did I learn from my relationship with Christine?

1) Oracle (and Sybase) don't store characters, they store bytes and strings of bytes.

2) Normally, Oracle does no checking of character sets, bytes inserted may or may not represent a character in the database character set.

3) Only under specific circumstances does Oracle "apply" a character se—when the NLS_LANG of the import session doesn't match the character set of the database, for example.

4) Changing character sets affects more than just the database.

5) Bytes of a character from any character set can be stored in a database created with any character set—EUC-JP data stored in WE8ISO8859P1 db, bytes aren't "correct" in the db character set.

6) Any character set conversion may corrupt the character data.

7) Simply exporting and importing to UTF8 does not solve all the problems of moving to UTF8.

8) Testing requires generating correctly encoded character data in the new character set — no utilities do this for you.

9) Every piece of an application makes some decision about the character set or has a default character set.

10) If all the existing data in a database really is in the database character set, then exporting and importing to a database with a different character set would work.

11) You need to be able to see the original data to verify the character set conversion has been done correctly — I had to know what the original Kanji looked like before the conversion — the same set of bytes can represent many different characters in different character sets.
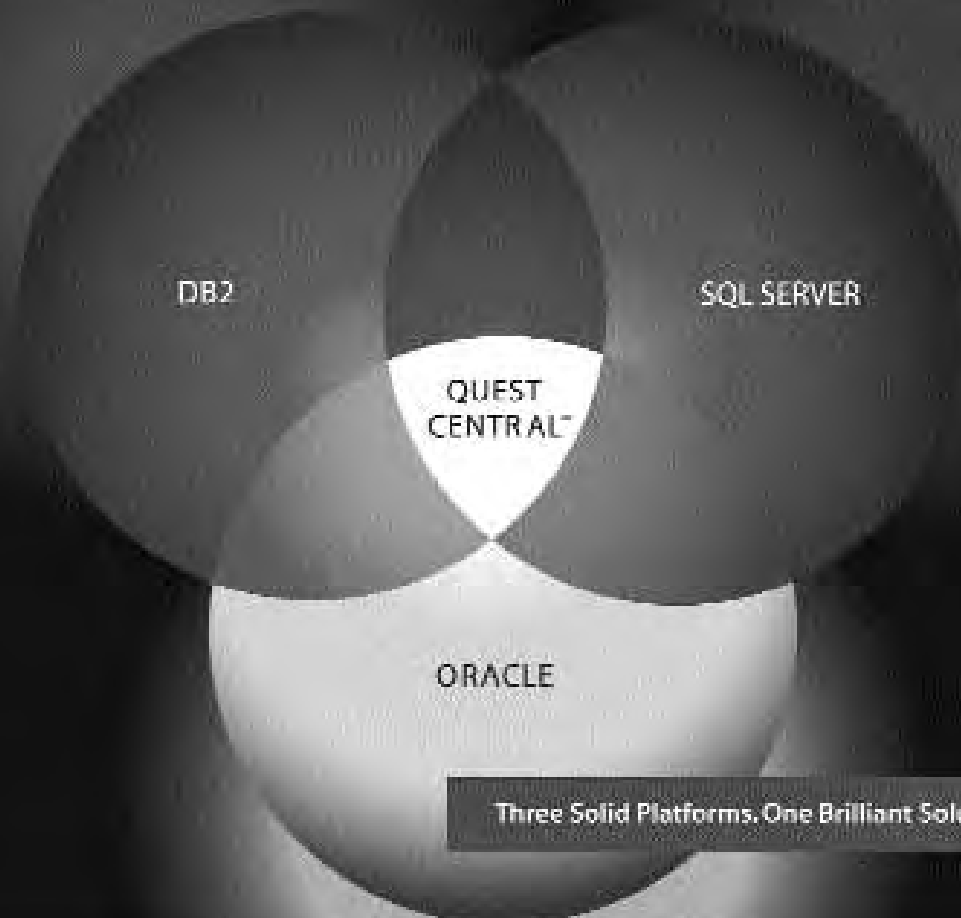
12) Beware of interviewers in Hawaiian shirts, especially when they promise exotic locales (I was misinformed!).

There is a greater single lesson to be learned from all of this. Anytime you are changing the character set of your database, you need to realize that the existing database may contain character data that was inserted at some time in the past from a character set different than that of the existing database. In the specific case of our story here, there was Japanese character data that was originally inserted from the EUC-JP character set into an Oracle database whose character set was WE8ISO8859P1. Anytime you are converting a database to UTF8 for example, you need to decide how you will determine if all of the character data in your existing database is really from the character set that is specified for the existing database. The conversion from a character set to the UTF8 character set can be done very easily with export and import. But this does not address the general issue of whether or not your existing database contains only character data that was originally from the character set of the existing database. Using our story again as an example: the conversion to UTF8 went smoothly, but the real issue was that some of the existing character data, when it was inserted, was not coming from a source that used the same character set as that of the existing database. When the Japanese data was inserted into the original Sybase database, it was inserted from a source system that was using the EUC-JP character set. The data from that source was inserted into a Western European (ISO1, single-byte) Sybase database. Anytime you are converting database character sets, you need to take this possibility into consideration. ▲

Brian Hitchcock has worked at Sun Microsystems in Newark, California, for the past nine years. Before that he worked at Sybase in Emeryville, California. Even further into the past he spent 12 years at Lockheed in Sunnyvale, California, designing microwave antennas for spacecraft. He supports development databases for many different applications at Sun, handling installation, tuning, NLS, and character set issues as well as Unicode conversions. Other interests include Formula One racing, finishing a Tiffany Wisteria lamp, Springbok puzzles, Marklin model trains, Corel Painter 8, and watching TV (TiVo® rules!). Brian can be reached at brian.hitchcock@aol.com or brhora@aol.com.

NoCOUG Conference Reporter:
Fall Conference 2003 at PG&E

Interview Conducted by Jen Hong, Director of Marketing, NoCOUG

Jen Hong

Member Profile: Randy Samberg

JH: How long have you been an Oracle professional?

I have been an Oracle database administrator for the last seven to eight years.

JH: How long have you been attending NoCOUG conferences?

This is my second NoCOUG conference.

Randy Samberg

JH: How did you find out about the NoCOUG conference?

Well, I found out from the NoCOUG website. I was looking for some networking group at the time, and I found NoCOUG from the website search.

JH: How do you enjoy the conference so far?

I love it; I think there is a lot of information offered at the conference. There are a lot of networking opportunities with other Oracle professionals, and training sessions give me the opportunity to talk to the vendors.

JH: Which part of the conference do you enjoy?

I think I enjoy the various sessions the most. It is probably the most valuable to me.

JH: What, in particular, brings you back to NoCOUG? What are one or two things that attract you back to the conference?

Educational opportunities would be the most appealing to me.

JH: Do you have any recommendation to the NoCOUG board?

One thing I see is that the NoCOUG conference is very well organized. The sessions and speakers are very well organized. I don't have a recommendation at this time.

JH: As an Oracle professional, do you have any recommendations for books for fellow NoCOUG members?

I have recently bought an Oracle Press book, *Oracle 9i DBA Handbook*, by Kevin Loney and Marlene Theriault.

JH: What do you think about the conference agenda and schedule?

It is pretty good; this time is better than last time because it was less vendor oriented and there was more technical information. The first speaker was really good; I went to "Managing an Oracle Optimization Project" by Gary Goodman.

JH: What are your priorities as an Oracle professional?

My priority is to learn as much as I can. I feel my experiences in the past have been kind of limited in certain areas. Right now I am interested in learning things like performance tuning and data warehousing. In my current job, we did very limited things using Oracle Forms and Report and I need to learn more about that.

JH: Much success to your pursuit of Oracle… ▲

Member Profile: Mary Lee

JH: How long have you been an Oracle professional?

Since Oracle 7, about seven to eight years.

JH: How long how you have been attending NoCOUG conferences?

I've been attending NoCOUG for a long time, for the past seven years now.

Mary Lee

JH: What do you like about NoCOUG conferences?

I like the technical sessions that are offered, where I can learn something rather than being sold products to. I like things that are technical in nature. For example, I learned how to set events to detect performance problems.

JH: So this is a big part of coming to NoCOUG? Is there anything else that attracts you to NoCOUG events?

Well, yes, technical training is very important to me; however, I come to NoCOUG to see friends. I have made many friend-

ships throughout the years by coming to NoCOUG. The networking part of NoCOUG is also very important to me.

JH: What are some of the things you like about what NoCOUG offers?

Knowledge sharing is very important to me. Here people can share their experiences with Oracle products and issues they have at work.

JH: What kind of recommendations would you give to a NoCOUG board member about the conferences?

I enjoy conferences that are knowledge sharing based on experiences of resolving real-life problems and technical know-how.

JH: Have you had positive experiences attending NoCOUG conferences?

Yes, very much, so I keep on attending the conferences. In fact, my manager knows that I have been coming to NoCOUG and in the past years I have paid out of my own pocket. She acknowledges the benefits of coming to NoCOUG and the variety of topics that it offers; now she is sponsoring my membership with NoCOUG via the corporate account.

JH: What kinds of hobbies and interests do you have?

I like hiking and traveling, and in my spare time reading some Oracle books.

JH: Why do you think it is important to keep up with Oracle technology?

Well, because things change rapidly. There are always better things as far as improvements of Oracle performance; how to better tune the Oracle system to improve performance. Learning new performance tuning methodologies, rather than staying with the old methods. So, it is important to keep yourself up to date.

JH: Why do you think it is important to attend conferences like NoCOUG? How does it help you?

By attending NoCOUG conferences I learn new information about how to better use Oracle technology, and how to update and improve my skills. Since the conference is just once a quarter, I find myself taking the materials back to do more research.

JH: Thank you for talking with the NoCOUG Reporter. ▲

DDL Generation– Oracle's Answer to Save You Time and Money

James Koopman

By James Koopmann

The New Method

The days of building your own DDL extraction utilities are almost gone. While Oracle has given us a simplistic approach to look at the DDL, it isn't quite everything we would hope for.

DBMS_METADATA.GET_DDL

This is the new package and function that will produce the DDL for you. In its simplest form, all you need to do is provide an object_type and an object_name.

The following example will generate the DLL for the DBA_TABLES view.

```
set pagesize 0
set long 90000
SELECT DBMS_METADATA.GET_DDL('VIEW','DBA_TABLES') FROM dual;
```

If you want to get fancy, you can generate all of the views for a particular user by just joining the function to the DBA_TABLES view. Here is an example that will generate all the DDL for all views owned by the user 'SYS'.

```
set pagesize 0
set long 90000
SELECT DBMS_METADATA.GET_DDL(object_type, object_name, owner)
FROM DBA_OBJECTS WHERE OBJECT_TYPE = 'VIEW' AND OWNER = 'SYS';
```

As you can see, this new functionality is very powerful.

Suppose you want to generate all 'CREATE USER' statements for all the users in your system. Just issue the following:

```
SELECT DBMS_METADATA.GET_DDL('USER', username) FROM DBA_users;
```

Or maybe you want all the 'CREATE TABLESPACE' statements. Try this:

```
SELECT DBMS_METADATA.GET_DDL('TABLESPACE', tablespace_name)
FROM DBA_tablespaces;
```

The power is almost endless.

Why Should You Make a Presentation at a NoCOUG Conference?

by Brian Hitchcock

There are many reasons why you should consider making a presentation at NoCOUG. Being able to prepare and give a presentation is one of the most valuable job skills you can have. Whether you are in software development, database administration, production support, or any form of management, being able to make a presentation will help you with your career. I know this from personal experience. Over my many years in both the aerospace and commercial software worlds, making presentations has helped me find and get assigned to more interesting work. It also hasn't hurt what I get paid, as it results in better reviews each year at work.

Workplace Visibility

Once you have prepared and given a presentation you can present it again to different audiences. For example, the presentation that I gave at NoCOUG covering various National Language Support (NLS) issues has been presented three more times at Sun. After the initial presentation, a group in Europe heard about it and asked that I present it at a time more convenient to them. Later on I was assigned to a new project that was having NLS issues and the project team had many questions for me. I told them I had a presentation that covered many of the issues and they requested that I present to team. The effort that I put into preparing the presentation was a onetime investment. And after that investment was made, I was able to give the presentation several times to different audiences with very little additional effort.

Having a presentation already prepared made the transfer of experience and information much more efficient. If I had been asked to simply answer questions from the individual project team members, I wouldn't have been able to cover so much material in an organized manner. With a prepared presentation I didn't have to try to remember all that I had learned, and I could focus more on the unique questions and observations that come up each time I make the presentation.

Management Perspective

Management is always looking for ways to leverage the existing knowledge in their organizations. One of the ways to do this is to encourage the employees to share the knowledge they already have. Presenting what you have learned to your peers at work (and at NoCOUG) is a great way to improve the learning of all involved. At Sun there are many groups looking internally for help answering their specific Oracle questions. These groups are always looking for someone who will prepare and present

Brian Hitchcock

material to them within Sun. It's a great way to get better known within your company. Future employers of yours may be other managers within your current company. Making a presentation is sort of like a job interview. You are marketing yourself to future employers every time you make a presentation. Even if members of the group that are present aren't employers, they may well be your peers at some future job. As hundreds of resumes cross a manager's desk, it will help you immensely if someone within that company or group recognizes your name. People will remember who you are because you have given a presentation.

Your Continuing Education

Another benefit of presenting to multiple audiences is that you get a range of questions from the participants. These questions allow you to review your work and figure

out how you might do things differently (and hopefully better) in the future. Critical review of your work by your peers improves your thinking, which ultimately improves your work. The questions you will get during your presentation will be very much like the questions you will face in any job interview. Nothing prepares you for a job interview like questions during a presentation. Being able to field questions about your work will make you better prepared for your next interview.

Once you have a presentation prepared, you can use the charts of your presentation as the basis of a journal article for either NoCOUG or for other users groups, or for internal publications at work. Making a presentation also allows you to review your work, study the way you attacked the problems, and question the assumptions you made as your work progressed. This process results in a more thorough analysis of what the problems were when you started, how you solved them, and how you worked with others. You will learn more detail about the subject you are presenting by making the presentation. You may think you know quite a bit about a given subject, and you may be right, but through the process of preparing and presenting your material, you'll find there are many details that you had either forgotten or that you never fully explored. Incorporating this level of detail in your presentation makes you much more knowledgeable about the subject and makes you much more confident about dealing with this subject in the future.

Prospective Employers

Making a presentation will also improve your resume. Once you've given a presentation, you should immediately add that fact to your resume. One advantage of making a presentation at NoCOUG is that it will then be posted on the NoCOUG website. Your resume can then reflect the fact that examples of your work are available on the Internet. This is very impressive to prospective employers as it gives them something concrete to look at as an example of your competence before an interview. As employers get more and more resumes for every open position, this sort of differentiation can only help you stand out from the others. Having your work on the Internet provides a very specific answer to the question of what you've done lately, what problems you've solved, and how you solved them. All of these issues are important to employers. In an era of layoffs, downsizing, outsourcing and general malaise, you have the ability to market yourself by making a presentation. You don't need a lavish training budget, you don't need to travel anywhere—all you need to do is think about what you have done and be willing to tell it to the rest of us.

Your Current Employer

At the same time, your current employer is also a prospective employer when it comes to keeping the job you currently have. When your annual review comes around you want to have a good answer to the question, "Whatt did you do this year to demonstrate continued competence in the position?" It doesn't hurt at all that your answer to this question includes a presentation at NoCOUG. I have found that my managers like this because they can tell their bosses that one of their direct reports did something to further publicize the company and the group. The manager also needs to tell his/her boss what was done to develop their direct reports during the review period. They can also list this as training for the employee. Also, making a presentation makes it much easier to get approval for a day away from work to attend the NoCOUG conference.

Networking

While the networking benefits of attending NoCOUG are well known, making a presentation provides an added level of networking benefits. I'll give you an example from my personal experience. One of my presentations about using STATSPACK was also accepted as a presentation for Oracle World. Because the papers that are presented at Oracle world are published on the Oracle World website, you don't know who may be reviewing your work. After the Oracle World conference, I received email from a potential employer halfway across the country with comments on how much they liked what I presented and (no kidding) asking if I was interested in a job offer. While it is a cliché that networking will help you find work, it really is true. The more you put your work out into the public arena, the more you become known to a much wider group of people, people who will know where the jobs are when the time comes.

You Have What It Takes

People are often convinced that they don't know enough to make a presentation. You don't have to invent a topic or create something fantastic to make a presentation. Just answer this simple question: What have I done in the last three months? As you write down your answers you'll remember the problems, crises, victories, setbacks, and probably some amusing events. What you have actually done is what others want to hear about. Every time I sit down to start preparing a presentation about a specific topic, I find that I'm drawn into all sorts of related areas to help explain what I did and how I did it. Once you start, the presentation grows until you find you have to remove material to fit into the time allotted.

I know that all of you have stories to tell and they are worth telling.

I know that all of you have stories to tell and they are worth telling. Before and after the presentations I have given at NoCOUG, just from talking to you, I know that you have things to say that I want to hear about. You have experiences I have not had, and if you tell me about them, that expands what I know, which saves me a lot of time in the future.

If it would help, I can get you started. I'd be happy to work with anyone in NoCOUG to get a presentation created and presented to the group. Please contact me at brhpsycho@aol.com, or brian.hitchcock@sun.com with any questions I can help with. ▲

DDL Generation–Oracle's Answer to Save You Time and Money

DBMS_METADATA.GET_DEPENDENT_DDL

The final piece of the pie to generate all the DDL for a given object is the use of the GET_DEPENDENT_DDL function. This is handy for extracting DDL that is in addition to the normal object definition. These are items such as grants and referential integrity.

Suppose you want to generate all the DDL for constraints that affect a table; you could use the following SQL. Please notice that while the first half of this SQL will produce output where the table in question is the child in the relationship. The second part of the SQL query will generate DDL for the tables that reference the table in question as the parent. This means that there might be other relationships produced by the second part of this query that do not affect the table in question.

```
SELECT to_char(DBMS_METADATA.GET_DEPENDENT_DDL('REF_CONSTRAINT',

table_name, owner)) DDL

 FROM (SELECT DISTINCT b.table_name, b.owner

   FROM dba_constraints a,

   dba_constraints b

   WHERE b.r_constraint_name = a.constraint_name

   AND b.r_owner  = a.owner

   AND a.constraint_type in ('P','U')

   AND b.constraint_type = 'R'

   AND b.owner = '<owner>'

   AND b.table_name = 'table_name')

union


SELECT to_char(DBMS_METADATA.GET_DEPENDENT_DDL('REF_CONSTRAINT',

table_name, owner)) DDL

 FROM (SELECT DISTINCT a.table_name, a.owner

   FROM dba_constraints a,

   dba_constraints b

   WHERE a.r_constraint_name = b.constraint_name

   AND a.r_owner  = b.owner

   AND b.constraint_type in ('P','U')

   AND a.constraint_type = 'R'

   AND b.owner = 'owner'

   AND b.table_name = 'table_name')
```

Supported Object Types

Here is a glimpse of the supported object types that can be input to the two previous function calls. Look at the Oracle documentation to determine the granularity you can achieve.

ASSOCIATION	associate statistics
AUDIT	audits of SQL statements
AUDIT_OBJ	audits of schema objects
CLUSTER	clusters
COMMENT	comments
CONSTRAINT	constraints
CONTEXT	application contexts
DB_LINK	database links
DEFAULT_ROLE	default roles
DIMENSION	dimensions
DIRECTORY	directories
FUNCTION	stored functions
INDEX	indexes
INDEXTYPE	index types
JAVA_SOURCE	Java sources
LIBRARY	external procedure libraries
MATERIALIZED_VIEW	materialized views
MATERIALIZED_VIEW_LOG	materialized view logs
OBJECT_GRANT	object grants
OPERATOR	operators
OUTLINE	stored outlines
PACKAGE	stored packages
PACKAGE_SPEC	package specifications
PACKAGE_BODY	package bodies
PROCEDURE	stored procedures
PROFILE	profiles
PROXY	proxy authentications
REF_CONSTRAINT	referential constraint
ROLE	roles
ROLE_GRANT	role grants
ROLLBACK_SEGMENT	rollback segments
SEQUENCE	sequences
SYNONYM	synonyms
SYSTEM_GRANT	system privilege grants
TABLE	tables
TABLESPACE	tablespaces
TABLESPACE_QUOTA	tablespace quotas
TRIGGER	triggers
TRUSTED_DB_LINK	trusted links
TYPE	user-defined types
TYPE_SPEC	type specifications
TYPE_BODY	type bodies
USER	users
VIEW	views
XMLSCHEMA	XML schema

James Koopmann can be reached at james_koopmann@yahoo.com.

Oracle Performance Optimization Using the Wait Interface – 7, 8i, 9i and Beyond

(Oracle Performance Tuning Seminar from NoCOUG)

Thursday, April 8, 2004

8:30 a.m. to 6:00 p.m.
Dublin Library Community Room
200 Civic Plaza, Dublin, California 94568

Presented by: Gaja Krishna Vaidyanatha
Author of *Oracle Performance Tuning 101*

Sign up now! Seating is limited!

Presentation Agenda:

➤ What is Oracle Performance Management (OPM)?
➤ The Method Behind the Madness
➤ Application Tuning–What a DBA Should Know
➤ Oracle Architecture Overview
➤ Instance Tuning
➤ Database Tuning
➤ Specialized Tuning
➤ Contention Tuning
➤ What Is New in 10g?
➤ Performance Optimization Techniques, Tools, Data Source
➤ Conclusion

Abstract:

Oracle performance tuning has developed a reputation as part science, part art, and part wizardry. This seminar will impart the core principles of performance management by sharing a step-by-step process of iteratively investigating, determining, and implementing tuning solutions using a proven methodology. This time-tested and field-proven methodology looks at performance management as having a system-wide scope. Further it steers you away from the old bad habit of simply throwing more memory at Oracle's shared memory areas. This seminar propagates the idea that the bulk of possible performance gains can be achieved quickly and with minimal effort, if you truly know how to identify and understand the nature of your system and, more importantly, its bottlenecks. And we will do this without checking one infamous cache hit ratio within Oracle.

Registration Costs Include:

➤ Full-day presentation
➤ *Oracle Performance Tuning 101* book
➤ Handouts
➤ Lunch and refreshments

Register by:

February 19 . . . . $170 members; $200 nonmembers if you register at the February 19, 2004 Conference. Bring a check!!!!
Until February 29 . . . $195 members; $225 nonmembers
March 1–31 . . . . $215 members; $245 nonmembers

For any further details, see www.nocoug.org or please contact Vilin Roufchaie (board@nocoug.org).

It's Registration Time Again!

I know the year just zoomed by. I'm sure it's due to the great 2003 NoCOUG program. The board has another stellar year planned for you. Please register early to continue getting this "award winning" *Journal* and streamlining your conference registration. For our 2003 members who have not yet registered for 2004, this issue of the *NoCOUG Journal* is complimentary. We know you plan to register.

Check the address label on the back of the *Journal*. If there is an asterisk (*) affixed to your name, you are already registered for 2004. Keep in mind that the mailing list was pulled on 1/22/04. The (*) status indicator will only be used on this issue.

All 2003 members should have received an email or U.S. mail reminder by now. That will include your NoCOUG ID for online registration at www.nocoug.org. Contact me with any questions at membership@nocoug.org.

Hope to see you at the coming conferences!

Joel Rosingana
NoCOUG Membership

NoCOUG Winter Conference

Thursday, February 19, 2004

Location: Oracle Conference Center, Redwood Shores, CA. See complete directions to the meeting location at www.nocoug.org.

8:00 a.m.	**Registration and Continental Breakfast**
9:00–9:30	**General Session and Welcome**
9:30–10:30	**Keynote: Ken Jacobs, Oracle Corporation**
10:30–11:00	**Break**
11:00–12:00	**Parallel Sessions #1**

> **Track 1:** *Exploiting Data for the Benefit of the Business* by Michael Scofield, Professor/Author, Loma Linda University

> **Track 2:** *Forms to J2EE, Not Java: Architectural Challenges and Strategies for Migration of Oracle Forms to J2EE* by Sri Rajan, Churchill Software

> **Track 3:** *Enterprise Manager (EM) 10g Grid Control* by David Leroy, Oracle Corporation

12:00–1:00	**Lunch**
1:00–2:00	**Roundtable Discussions**
2:00–2:15	**Break**
2:15–3:15	**Parallel Sessions #2**

> **Track 1:** *Making DW More Relevant to the Business* by Michael Scofield, Professor/Author, Loma Linda University

> **Track 2:** *Securing an Oracle Database* by Noel Yuhanna, Forrester Research

> **Track 3:** *Recent Enhancements in Query Processing in Oracle* by Hakan Jakobsson, Oracle Corporation

3:15–3:45	**Break and Raffle**
3:45–4:45	**Parallel Sessions #3**

> **Track 1:** *Building a Poor Performing J2EE Application Is Not Your Developer's Fault* by Dave Martin, Wiley Technology

> **Track 2:** *Information Lifecycle Management Strategies for Oracle Applications Data* by Erik Jarlstrom, Princeton Softtech

> **Track 3:** *Oracle 10g Manageability* by Michael Sit, Oracle Corporation

4:45 -??	NoCOUG Networking and Happy Hour
Cost:	$40 admission fee for non-members. Members free. Includes lunch voucher.

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

Session descriptions available at www.nocoug.org.

RSVP online at www.nocoug.org/rsvp.htm

NoCOUG
P.O. Box 3282
Danville, CA 94526

Thank you to Oracle Corporation, our meeting sponsor.