

Naming Oracle database links

By Steven Wolfe, Development Manager, Oriole Corporation..
naming.doc

The subject of naming conventions may seem trivial but anyone who has any experience of database design knows how important a clearly defined naming scheme is to maintenance and system evolution.

One highly interesting and quite often overlooked topic is how to name database links. It's all the more interesting when you consider the existence of the GLOBAL_NAMES parameter which, when set to 'true', forces database links to bear the same name (although optionally qualified) as TNS aliases. It may not be what the Oracle designers had in mind but for most DBAs this is tantamount to giving the database link either the name of a machine or of an Oracle system identifier (SID), as that is what most of us give as TNS aliases. In my opinion, naming database links after machines or SIDs is the wrong approach.

Basically, database links are needed because an application requires some reference data which is located in a different database from the one where most of what it processes is usually kept; this is a consequence in fact, of an application using data managed by another application. Unfortunately, the life expectancy of an application more often than not outlasts the loose relationship it maintains with hardware or the data containers we call databases.

For instance, the R&D department of a very large cosmetics company used to run on its main machine, an application (management of appointments when doing some large scale pre-marketing tests) quite unrelated to any others. For both performance reasons and operational constraints (the users of this application had quite different availability requirements from the others) all the tables used by the application were exported to a new database on a different machine.

Very shortly afterwards the time came to upgrade the operating system. So the main production database was also temporarily moved to the other machine, in order to have enough time to upgrade, relink all the programs and test thoroughly. Then both databases were brought back to the newly upgraded machine, in order to upgrade the second one, and the appointment database was finally taken back where it belonged. In big organizations, everybody can tell similar - or worse - stories. There were not, at this time, too many important database links.

However, had that been the case and if some GLOBAL_NAMES=TRUE had been active, the proper way would have been to drop and recreate, under a new name, database links in one or more other databases, look for synonyms using these database links and for stored procedures or even triggers which might have referenced them - Odds are very high that a few things would have slipped through the net ...

The quick and dirty way would have been to change the tnsnames.ora file; all very fine when moving databases to different machines but totally irrelevant when spawning a new database to harbor already existing data.

In fact, application names should be used both for naming table owners and database links. Let us say that the tables of applications SALES and ACCOUNTS were initially created in the same database.

The application designers were wise enough to :

- 1) have all the SALES tables, sequences, procedures, etc. owned by Oracle user SALES and likewise with ACCOUNTS,
- 2) create public synonyms for all objects in order never to have to prefix object names with owner names in programs.

Thanks to a booming business, the time comes when a new machine is purchased and separate databases PROD1 and PROD2 are created for each application.

(Note that, not knowing what tomorrow has in store, nor what other applications will someday run on each database, we were careful enough to give neutral names to our databases, instead of, say, SALE and ACCT).

ACCOUNTS obviously requires some data from SALES. Nine times out of ten, you will find ACCOUNTS using a database link named PROD1 connecting to user SALES to access, say, the PRODUCT_SALES table.

We have here two big mistakes :

- Naming the database link PROD1 just won't do!
Nothing guarantees that the SALES application will stay on database PROD1 forever.
What if we name it SALES ?
At least, referring to PRODUCT_SALES@SALES is self documenting.
- Using the SALES Oracle account to connect is opening the door wide open to unwanted access.
So define the port of entry as the application that uses it
=> ACCOUNTS
Do you want anybody who has access to ACCOUNTS to have access to the data from table PROSPECTS@SALES ?
Quite unlikely.
It's much better to create a special account on database PROD1 with some hard-to-enter password.(see: www.oriolecorp.com for details on hard to enter passwords - undocumented features section). And what better name than ACCOUNTS could we give to this user ? Once again, the privileges granted by user SALES on its objects to user ACCOUNTS will precisely document the relationship between the two applications.

In fact, this account can be considered a bridgehead of the ACCOUNTS application on the database where SALES runs.

To summarize, a database link which indicates the link between: a 'client' application (the one which requires the data) and a 'server' application (the one which provides it) should bear the name of the server application and connect to the remote database using an Oracle account which is named after the client application.

This in our opinion is the best way to protect oneself against all the hassle that is usually caused by migrating databases. Even if the two applications one day happen to be reunited on the same database, there will be virtually no impact.

The drawback, this being an imperfect world there has to be one, is that if many applications are running on the same database and if we need links to more than one, we will use more connections than are strictly required but this is a small price to pay.

The Author.

Steven Wolfe is Development Manager of Oriole Corporation, a company which provides a range of fast, easy to use, efficiency enhancing Oracle database administration tools. Full details of the range of tools and a selection of free scripts is available on the Oriole webpage at www.oriolcorp.com. Steven can be reached by e-mail at swolfe@oriolcorp.com.