

How Should We Manage All These Databases??

Karen E. Langley

Information Technology Services
Lockheed Martin Energy Systems

Oracle Open World 98
San Francisco, CA
November 1998

"The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. DE-AC05-84OR21400. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes."

How Should We Manage All These Databases??

Karen Langley, Lockheed Martin Energy Systems

Introduction

In an organization where there are many DBAs working with many instances and databases on many machines with many developers – how do you manage all of this without total chaos? This paper will outline how the central Database Support organization at Lockheed Martin Energy Systems in Oak Ridge, TN manages more than 250 instances on more than 90 systems with a variety of operating systems. This discussion will include how tasks and responsibilities are divided between System DBAs, Application Project DBAs, and developers. The use of standards as well as local routines to maintain the systems will be discussed. Information on the type of communications used to keep the different groups informed and up-to-date will also be presented.

Database Support Organization

The Database Support group is a central organization providing services to all three plants in Oak Ridge as well as to subcontractors. The group consists of two co-supervisors and fourteen DBAs, and two administrative support personnel. The supervisors also perform technical work as well as attending to administrative tasks. One supervisor is the Technical Project Leader for databases on OpenVMS systems whereas the other is the Technical Project Leader for databases on UNIX systems. At this time, the NT and Windows95 work is shared between the supervisors. The fourteen DBAs all work on Oracle databases. Many also have skills in other databases (DB2, CA-IDMS, CA-DB, Informix, SQL Server) to enable Database Support to provide services in those areas as well.

In the Database Support group, each person has Oracle experience in the UNIX environment (including AIX, HP-UX, Digital Unix, SunSolaris, SunOS, MPE/iX, SCO, and IRIX). Several people have OpenVMS experience and several are knowledgeable in the NT environment. For each database system maintained by Database Support, one person is assigned as the 'primary' DBA and then two or three people are assigned as 'secondary' DBAs. This provides coverage of all systems at all times. The ability to manage a large number of systems requires standards, teamwork, and communication.

With the more than 250 instances and 80 systems, there are more than 300 developers. The sizes of the systems vary – anywhere from an 8 user license on a departmental machine to a 250 user license on a central system with more than 7500 accounts for a 'time card' system. Database Support provides consulting services to the developers in addition to providing

installation, upgrade, backup & recovery, hot standby databases, performance tuning of the database systems. Database Support is the interface with Oracle Customer Support via phone and Metalink.

Roles and Responsibilities

The different groups involved with Oracle databases include: system DBAs (a.k.a. database system administrators), application project DBAs, developers, end users and operating system managers. System DBAs are the Database Support personnel. Application project DBAs are developers with elevated privileges in the database and responsible for configuration control of their application. The end users are the ultimate users of an application. The operating system manager is responsible for the operating system software. All of these groups must work together.

Typically System DBAs (Database Support personnel) are responsible for Oracle software installation and upgrades, database creation, creating application tablespaces, database shutdown and startup scripts, login scripts, as well as backup and recovery procedures. In addition, the System DBAs are responsible for implementing procedures to maintain users, complying with security and auditing requirements, and database monitoring and tuning. Coordination with operating system managers and tool installers/developers must be considered when planning upgrades. Database Support provides 24x7 support for production systems – this includes each person having a pager and computer operations having a list of systems with primary and secondary DBAs listed. On a daily basis, Database Support works with application project DBAs, developers, and users to solve problems, provide solutions to programming questions, and offer consulting services.

Application project DBAs are people from the application development group with access to the application project userid. Each project is given a userid for the project that will remain for the life of the project. This resolves an issue when a developer leaves the project - the database objects do not have to be recreated under a different user. In a production environment three developers have access to the project userid to facilitate configuration control. The application project userid is typically an ops\$ account to enable batch jobs to run without need of a userid/password to be embedded in the job. The application project DBA creates, deletes and alters tables, indexes and other database objects needed by the application. Their responsibilities also include defining and creating roles, access management (assigning roles to users), and monitoring application performance and tuning. The application project DBAs consult with System DBAs as well as their developers and end users. Other responsibilities include activities specific to the particular application.

Database Standards

To enable Database Support to manage as many systems as we do, standards have been implemented. These standards involve SID and database names, tablespace names and disk/datafile names. Databases are named prod, qa and devl for production, quality assurance (migration), and development databases respectively. Non-application tablespaces are the same for all databases (i.e. temp_ts, users_ts, rbs_ts). For application tablespaces the application name/acronym is part of the

tablespace name. Wherever possible, the disks are similarly named across systems. In several situations, the computer systems predated the use of Oracle, so disks names do vary. However, the directory naming convention is consistent. The datafile names include the name of the database, tablespace and then a sequential number. This makes it easy to see how many files are used in each tablespace for each database. All of these standards combined make it much easier for someone to work on a system they normally don't manage.

SQL*Net aliases are determined by concatenating the nodename with the database name. For node **goofy** and database **prod** the SQL*Net alias would be **goofyprod**. This makes it very easy to remember what the alias is and where you are going when connecting remotely.

Every new account is given a role, `gen_user`, that provides basic privileges (create session, create table – see appendix for detailed listing). The application project accounts are also given the `proj_dba` role for additional privileges above the `gen_user` role. In addition, the `dbsupport_role` role is created for the System DBA accounts in the databases. Each user is also given a profile, `gen_profile`, which limits `idle_time` to 15 minutes.

To provide assistance to the developers, a series of views have been developed by Database Support and implemented in each database. Some of these views include `my_roles`, `my_space`, `my_extents`, and `tab_privs_granted`. When these views are used from the application project `userid` account (which has the `proj_dba` role), only information on their application is shown. For example, `my_roles` will show for roles the account is the administrator of, who has the role and whether it is a default role or not for that user. The convention is only non-passworded roles will be made default roles. Details on these views can be found in the appendix.

Standard Routines

UCAMS

UCAMS (Universal Computer Access Management System) is an internally written set of routines to manage accounts across computer systems, databases, NT domains, etc. at Lockheed Martin Energy Systems. A central UCAMS system maintains information for each user – `userid`, name, location, and authorized systems. On each computer and each Oracle system, a UCAMS daemon waits for transactions from the central system. This enables each person to have one password for all systems they access. UCAMS provides a mechanism for accounts to be authorized, users to pick randomly generated passwords, accounts to be suspended (due to inactivity, leave the company, etc.), as well as accounts added and deleted. There is also auditing capabilities to ensure that the accounts on the system match the information in UCAMS. UCAMS itself uses an Oracle database with a web front end for the application. Oracle/UCAMS, the UCAMS code that works with the Oracle database, has been written such that it will work on all platforms (several flavors of UNIX, OpenVMS and NT) – only

one version of the code to maintain. The program is written in C with several PL/SQL procedures. UCAMS will grant the `gen_user` role when creating an account as well as any other roles listed in a specific table. Since the `gen_user` role has the create session privilege, it is simply a matter of revoking the role to effectively suspend a user. Likewise, to unsuspend a user, the role just needs to be granted back as a default role.

Set-Default-Roles

The Set-Default-Roles routine is a set of PL/SQL procedures and a C program. The PL/SQL procedures enable developers to enter a userid into a table that is read by the C program. The C program, via a batch job, will look at the table and then set all non-passworded roles to be default for that user. Thus, a developer can create a new role and then grant it to a user. Then if they want to have that role be made default, they just use the set-default-role procedure. Generally this is sufficient for the user and developer. However, if they are testing, or for whatever reason the role needs to be made default immediately, the developer can call Database Support and we will go into SQL*Plus and update the user's default roles with the 'alter user' command. All non-passworded roles will be made default by the UCAMS transactions as well as Set-Default-Roles. Passworded roles are never made default (defeats the purpose of the password).

Sniper

One of our security requirements is to disable a user after a specific number of invalid logins. Since Oracle7 did not provide that capability, Database Support wrote Sniper to fulfill this requirement. Sniper is a set of procedures that are run every few minutes. By using auditing, valid and invalid logins are recorded in `sys.aud$`. Sniper will then look at this information to determine if a user had too many invalid logins within a period of time. If that is true, the account will be suspended (by revoking the `gen_user` role), a suspension recorded, and then an email message sent to the user and Database Support indicating the suspension. If there are an excessive number of invalid logins for a particular account, Computer Security is also notified by email. The account is not reinstated until confirmation by the user that they had invalid logins.

Database Monitoring

To provide a method to routinely monitor database performance of systems maintained by Database Support, a series of routines were written. These PL/SQL scripts reside on one computer system for ease of maintenance. Configuration files specific to each system are setup enabling different thresholds to be set. The scripts are run daily or weekly, depending upon the system and the routine, via batch (cron) jobs. If a threshold is exceeded, an email message is generated and sent to the `dbsystem` mail account (more on the mail accounts under Communications below). Then the System DBA can evaluate the situation and react accordingly. This allows us to be more proactive in our tasks. Items monitored include tablespace free space, object extents, compliance with security requirements, and information on SGA, rollback segments, etc. An additional monitor routine, `db_dead`, determines if the database is up. If not, then the primary DBA is notified by a pager message.

Communications

Email Accounts

Database Support has two email accounts – *dbsupport* and *dbsystem*. These two accounts reside on a UNIX system and each person in Database Support is able to set user (su command) to the accounts associated with these mail names. The *dbsupport* account is advertised as the location for developers, end user, and operating system managers, to contact Database Support. This provides a central location for all questions and requests. In addition, if the primary DBA is unavailable, a secondary DBA, or for that matter anyone in Database Support, is able to respond to the request. This provides more timely support to our 'customers'. The *dbsystem* email account is used for the Database Support routines – backup, Oracle/UCAMS, Sniper, and the monitor routines. With the two accounts, we are able to distinguish between 'customer' messages and 'system' messages. The system message include machine name, database name and routine in the subject to make it easier to know which mail messages need to be read by each Database Support person. Read messages are re-filed or deleted to prevent the same message from being read, and worked, by several people.

Database Support

Even though Database Support manages database systems at a variety of locations, all of the offices are located together. This facilitates direct communications amongst the group. Once a month Database Support personnel get together for about an hour for a roundtable type meeting. In these meetings we discuss standards, techniques, problems, and anything else we feel each person in the group should be aware of. These meetings are very productive and highlights of the meeting are written up and emailed to each person in case someone misses a meeting and for future reference. In addition to these communications, a notebook is maintained with installation and configuration information for each computer system with an Oracle installation. Information maintained includes primary and secondary DBAs, IP name, operating system and hardware versions, Oracle software installed and versions, databases installed, times for system and database backups, and application information. This notebook has proved invaluable many times especially when working on a system you don't normally work with.

Web Pages

Database Support has developed a set of web pages on the intranet. Information on the web pages include the types of databases supported, services provided, personnel, meeting information, on-site training available, and documentation. There are several sections with helpful hints, sample routines, and frequently asked questions. For the developers there are sections on topics such as 'Useful Views and Tables', 'Running Batch Jobs', and 'Export/Import Jobs'. There are also links to other sites from these web pages. In a 'Database Support Only' area, documentation for Database Support personnel is maintained including instructions for setting up a standard system and installing local software (Oracle/UCAMS, Sniper). Using the web has enabled us to always obtain the latest information even from a location away from the office.

Meetings

With the sheer number of personnel at Lockheed Martin in Oak Ridge using Oracle, two internal user groups have emerged. Database Support plans the agenda, obtains speakers, and manage the meetings which are usually 60-90 minutes in length. The Oracle User's Group meeting is generally less technical in nature of the two groups. At these meetings, held about twice a year with 50-75 attendees, new products or applications are demonstrated or general information is provided. Speakers have been personnel at Lockheed Martin and from Oracle. The Oracle Project DBA User's Group meeting is much more technical in nature and is geared to the developers and application project DBAs. The majority of these meetings are conducted by Database Support personnel providing information on standards, useful hints in developing applications, or tuning. On occasion, an application developer will present a specific technique they have used in their application. These meetings are held every 2-3 months and attended by about 40-60 developers. In addition to the meeting, Database Support will write up a 'paper' on the subject and make it available on the Database Support web pages. This provides the information to all the developers even if they were unable to attend the meeting.

Conclusion

It is possible to manage many database systems with many developers on a variety of platforms with a central Database Support staff. To do so, teamwork, communications, and standards are required. In Database Support at Lockheed Martin Energy Systems, we have implemented these ideas with a defined set of roles and responsibilities among the various personnel – system DBAs, application project DBAs, developers, end users, and operating system managers. This also enables to us to minimize chaos and to respond quickly when requested to establish a new system.

Appendix

Gen_user Role

```
rem create_gen_user_role.sql
rem creates gen_user role and grants privileges
create role gen_user;
grant
    alter session, create cluster, create database link, create procedure,
    create sequence, create session, create snapshot, create synonym,
    create table, create trigger, create view
    to gen_user;
```

Proj_dba Role

```
rem create_proj_dba_role.sql
rem create proj_dba and grants privileges
create role proj_dba;
grant
    monitorer,
    create public database link, drop public database link,
    create public synonym, drop public synonym,
    create role
```

```
to proj_dba;
```

My_roles View

```
rem my_roles.sql
rem run as sys

create view my_roles
as
select granted_role role, grantee username,
       admin_option adm, default_role def,
       password_required pwd
from sys.dba_role_privs, sys.dba_roles
where granted_role in
(select granted_role
 from sys.dba_role_privs
 where user = dba_role_privs.grantee
 and dba_role_privs.admin_option = 'YES')
and granted_role = role;

create public synonym my_roles for sys.my_roles;
grant select on my_roles to proj_dba;
```

My_space View

```
rem my_space.sql
rem adopted from freespa7.sql script
rem run as sys
rem
rem set up user_data_files view

drop view user_data_files;
create view user_data_files (tablespace_name, k_bytes)
as
select dba_data_files.tablespace_name, sum(bytes)/1024 from dba_data_files,
user_tablespaces where user_tablespaces.tablespace_name = dba_data_files.tablespace_name
group by dba_data_files.tablespace_name;
drop public synonym user_data_files;
create public synonym user_data_files for sys.user_data_files;
grant select on user_data_files to proj_dba;
rem
drop view user_tablespace_free_space;
create view user_tablespace_free_space
(tablespace_name, k_free, k_max, num_extents)
as select TABLESPACE_NAME,
       sum(BYTES)/1024,
       max(BYTES)/1024,
       count(BYTES)
from      user_free_space
group by TABLESPACE_NAME;

drop view my_space;
create view my_space
(tablespace_name, k_bytes, k_free, pct_used, num_extents, k_bigext)
as select
       substr(a.TABLESPACE_NAME,1,20),
       a.k_bytes,
       b.k_free,
       (a.k_bytes-b.k_free)*100/a.k_bytes,
       b.num_extents,
       b.k_max
from      user_data_files a,
       user_tablespace_free_space b
where a.tablespace_name = b.tablespace_name;
```

```
drop public synonym my_space;  
create public synonym my_space for sys.my_space;  
grant select on my_space to proj_dba;
```

My_extents View

```
rem my_extents.sql  
rem run as sys  
  
drop view my_extents;  
create view my_extents  
(segment_name, segment_type, k_bytes, extents, max_extents)  
as  
select substr(segment_name,1,30),  
       substr(segment_type,1,10),  
       bytes/1024,  
       extents,  
       max_extents  
from user_segments;  
  
drop public synonym my_extents;  
create public synonym my_extents for sys.my_extents;  
grant select on my_extents to proj_dba;
```

Tab_privs_granted View

```
rem tab_privs_granted.sql  
rem run as sys  
  
create view tab_privs_granted  
as  
select table_name name,  
       grantee      username,  
       grantable   adm,  
       privilege   privilege  
from sys.dba_tab_privs  
where owner=user;  
  
create public synonym tab_privs_granted for sys.tab_privs_granted;  
grant select on tab_privs_granted to proj_dba;
```

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.