

# **Oracle9i: Kiss Your init.ora Goodbye!**

**Dan Norris**

**[norris@celeritas.com](mailto:norris@celeritas.com)**

**Celeritas Technologies, LLC**

**IOUG Live! 2002**

**Paper #520**



# Definitions

- Instance: The background processes and shared memory that comprise the Oracle server (SGA)
- Database: Control files, data files, and online redo logs
- Pfile: “normal” (before 9i, at least) text parameter file (a.k.a. init.ora file)
- Spfile: the binary form of the parameter file (a.k.a. server parameter file)

# Oracle9i Server Parameter File

- The server parameter file (spfile) is a binary file that contains instance parameters
- Created by DBCA automatically (unless you tell it otherwise)
- Replaces the text init.ora file
- Modified with ALTER SYSTEM statements (No need to be on the server to modify it!)

# The Hunt for Initialization Parameters

1) The first place to look is:

```
$ORACLE_HOME/dbs/spfile$ORACLE_SID.ora
```

2) If that file does not exist, look for a shared spfile at:

```
$ORACLE_HOME/dbs/spfile.ora
```

3) When all else fails, look for the “old” init.ora file:

```
$ORACLE_HOME/dbs/init$ORACLE_SID.ora
```

**Note: On NT systems, ‘dbs’ is replaced with ‘database’ and search step #2 above (shared spfile) is skipped.**

# Overriding the default location

- There is no `spfile=“...”` parameter to the startup command (like there is for `pfile`)
- Create a one-line `init.ora` file that contains `spfile='/path/to/spfile.ora'` and optionally use the `pfile='...'` option of the startup command to specify `init.ora` location

# What's In the spfile?

- The spfile contains the same things as the init.ora file with some added functionality.
- A prefix is added to each parameter to denote which instance it applies to (“\*” means all instances). This is intended for RAC, but can be used by anyone (even when not using RAC).
- V\$SPPARAMETER view also shows spfile contents.

# For the Old Timers (Creating an spfile From Scratch)

- If you'd like to create the database yourself and would like to create an spfile, create a normal init.ora file and then issue:

```
CREATE SPFILE FROM  
PFILE=' /path/to/init.ora' ;
```

# Modifying the spfile

```
ALTER SYSTEM SET parameter_name = value  
  [ SCOPE = SPFILE | MEMORY | BOTH ];
```

- SCOPE determines what is updated: the spfile, just running instance, or both
- Default for SCOPE depends on whether the instance was started with an spfile or not:
  - If yes, BOTH is default.
  - If no, MEMORY is default (and only option allowed).

# Modifying the spfile

```
ALTER SYSTEM RESET parameter_name  
  [ SCOPE = SPFILE | MEMORY | BOTH ]  
  [ SID = sid ];
```

- You need to use the RESET clause to remove an entry from the spfile.
- Can be used for an instance-specific or shared parameter.

# Backup and Recovery

- The spfile is not guaranteed to be consistent and can not be backed up directly (just like controlfiles).
- Backup is done by converting the spfile to an init.ora file using:

```
CREATE PFILE=' /path/to/backup/init.ora'  
FROM SPFILE;
```

and backing that up.

- Recovery is done by restoring the init.ora file and running:

```
CREATE SPFILE FROM PFILE;
```

# Server Parameter File Advantages

- Instance parameters modified dynamically (with ALTER SYSTEM) are persistent without modifying a file
- Self-tuning will persist across server restarts
- Remote startup (from OEM, for example) can utilize the spfile--no local init.ora needed for startup
- Possibility of sharing one parameter file

# Server Parameter File Disadvantages

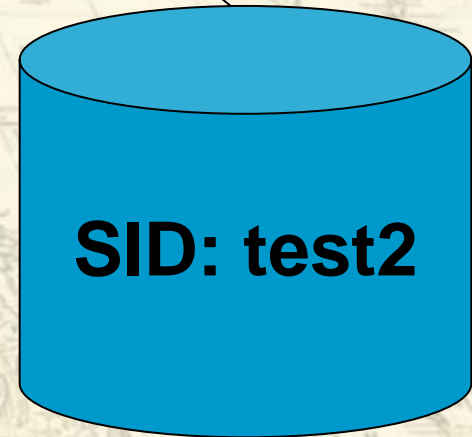
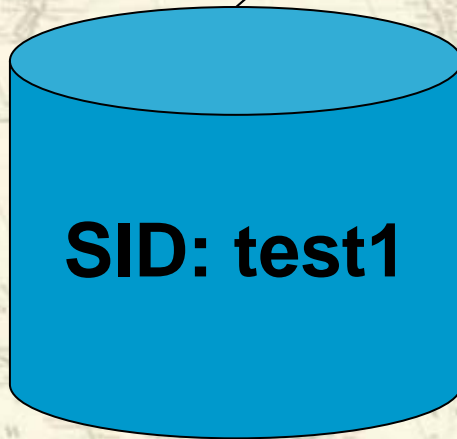
- When instance is down, changes are more difficult than init.ora
- Backup and recovery each involve multiple steps

# Shared Server Parameter Files

`$ORACLE_HOME/dbs/spfile.ora`

**SID: test1**

**SID: test2**



# Sample Shared spfile Contents

```
*.db_block_size=8192
*.sort_area_size=1048576
test1.db_name=test1
test2.db_name=test2
test1.control_files=( '/u01/oradata/test1/ctl1.ctl' )
test2.control_files=( '/u02/oradata/test2/ctl1.ctl' )
test1.db_cache_size=163840000
test2.db_cache_size=81920000
*.shared_pool_size=52428800
*.nls_date_format='mm/dd/yyyy'
...
```

# Shared spfile Advantages

- Can view and modify parameters of other instances from any instance
- Global changes (i.e. db\_domain) can be changed once and affect many instances
- Enables consistency by using global parameters (nls\_date\_format)

# Shared spfile Disadvantages

- Can view and modify parameters of other instances from any instance
- Global changes good, global mistakes bad
- No concept of privileges, any ALTER-SYSTEM privileged user on any instance is able to modify any other instance's parameters if they share a server parameter file

# Real Application Clusters (RAC) & Shared spfiles

- Shared spfiles were meant for RAC
- Finally can share one parameter file for all instances in the cluster
- Spfile is a binary file and so it can reside on a raw volume to enable sharing
- Instance-specific and global parameters managed in one globally-available file

# A Few of the (never before) Dynamically Modifiable Parameters in *9i*

- `shared_pool_size` (yes, really; great for batch/OLTP changes)
- `db_cache_size` (aka `db_block_buffers` in *8i*)
- `open_cursors`
- `service_names`

# Demonstration

- Creating spfile
- Viewing parameters
- Modifying parameters in MEMORY and SPFILE

# Looking Ahead

- Just my opinion, but I'd be looking for Oracle to desupport the "old" init.ora file in favor of the spfile in a future release of the server. Remember checkpoint\_process?

Questions fall into two categories:  
Good Questions  
Interesting Questions

Slides available at <http://www.celeritas.com>

Evaluation Form Information:

**Oracle9i: Kiss Your init.ora Goodbye!**

**Dan Norris, [norris@celeritas.com](mailto:norris@celeritas.com)**

**Paper #520**

