Oraenv For Windows

Contents

Introduction

The Oracle DBAs amongst us, who are used to working with Unix/Linux, are *reasonably* well provided with tools to make our Oracle lives easier - `oraenv` for example is used to clear down the existing Oracle environment for a given `$ORACLE_SID` and to set up a new one based on a potentially new `$ORACLE_SID` being supplied.

On Windows, it's not so easy. At Oracle 9i, there used to be an `Oracle Home Selector` utility installed with the client/RDBMS software but it seems to have vanished just as quickly at release 10g. What to do?

In my current contract I've been suffering from Windows. It's not incurable, but it's a damned annoying environment to work with - nothing is as it seems, and setting up an Oracle environment appears to be a little more like hard work than I'm used to on Unix/Linux systems, where I've spent the vast majority of my DBA life.

I did have a previous version of oraenv for Windows out there in the wild, but it *didn't do everything*. It couldn't adjust the %PATH% for example, which made it pretty useless to be honest. However, as I only had one %ORACLE_HOME% to work with on my servers, that wasn't too big a problem for me. Now we have patching, which can mean running a number of %ORACLE_HOME%s on the same server. Suddenly, I needed to *up my game* a little and come up with an oraenv utility that actually worked in such an environment. It wasn't easy, and I discovered a few Windows foibles/features along the way. Bugs in other words!

So here we are, a Windows oraenv that works pretty much identically to the one we are used to in Unix which can set the %ORACLE_HOME% and more importantly, clean up and correctly set the %PATH% in a manner that the Unix utility of the same name does.

Enjoy.

- Norman Dunbar
- April 2017.

Requirements

You need the following files, somewhere on your %PATH%:

- oraenv.cmd - the main batch file for Windows.
- DBHome.exe - A utility to extract the correct %ORACLE_HOME% for a given %ORACLE_SID%.
- DBPath.exe - A utility to remove all folders and sub-folders of the current %ORACLE_HOME% from the current %PATH%.
- TidyPath.exe - A utility to ensure that %PATH% is correctly set with any paths which have spaces or brackets present, properly double quoted.

You will also need to set up an ORATAB file in one of the following manners:

- Setting the environment variable %ORATAB% to the *full path* to the file to be used. In this case, the file can have any name you wish and need not always be oratab.txt.

```
set ORATAB=c:\Oracle\oratab.txt
```

- Creating a file named oratab.txt in the location pointed to by the environment variable %ORACLE_BASE%. You should test if %ORACLE_BASE% is set by running:

```
set ORACLE_BASE
```

  You are expecting *not* to see the message *Environment variable oracle_base not defined* returned.

- Creating a file named oratab.txt in the same folder as the DBHome.exe file.

The above options are also how the utility DBHome.exe searches for the appropriate file.

ORATAB File Format

The ORATAB file is a plain text file. It contains optional comments, and a list of ORACLE_SIDs and corresponding ORACLE_HOMEs. The format is as follows:

Comments

A comment is introduced by a hash/pound character (#) and the remainder of the line is ignored. For example:

```
#-------------------------------------------------------------------------
# Format is:
#-------------------------------------------------------------------------
# ORACLE_SID | ORACLE_HOME |# Optional Comment.
#-------------------------------------------------------------------------
```

Entries

Each entry consist of an ORACLE_SID a pipe separator (|) and an ORACLE_HOME path. For example:

```
client_32 | c:\Oracle\clients\11204\client_32
client_64 | c:\Oracle\clients\11204\client_64
```

And so on. Spaces *are* permitted on either side of the pipe character - as it makes the file a little easier to read by humans. As this is windows, the letter case of the ORACLE_SID and/or ORACLE_HOME is ignored. Tab characters, on the other hand, are not allowed in the file.

> **Note:** We have to use the pipe as it is not likely to appear in a file path on Windows, and unlike on Unix/Linux, the colon *is* used in paths - it's part of the drive specifier, so we can't use the colon, much as I would like to, for consistency!

Entry Comments

Each entry may have a trailing comment. This is indicated by a pipe character *immediately* followed by a hash. (|#) Spaces are not permitted *between* those two characters, but are allowed before and after them. For example:

```
client_32 | c:\Oracle\clients\11204\client_32 |# 32 bit 11204 client.
client_64 | c:\Oracle\clients\11204\client_64 |# 64 bit 11204 client.
abc123    | c:\Oracle\product\11.2.0\dbhome_1 |# Production database.
```

Calling Conventions

The oraenv.cmd utility adjusts its behaviour according to the setting of the environment variable %ORAENV_ASK%, as follows:

ORAENV_ASK set to YES

```
set oraenv_ask=yes
```

Execution Without a Parameter

In this case, running:

```
oraenv
```

will prompt interactively for a new %ORACLE_SID%. The current %ORACLE_SID% is offered as a default. If there currently is not an %ORACLE_SID% then you will see NOT_SET and you cannot accept the default. The utility will loop around until you supply a new %ORACLE_SID% in this case.

Execution With a Parameter

Running the utility with a new %ORACLE_SID% supplied on the command line, like this:

```
oraenv abc123
```

will set the Oracle environment to that of abc123 without asking, provided that abc123 is found in the ORATAB file in use.

ORAENV_ASK set to NO

```
set oraenv_ask=no
```

Execution Without a Parameter

In this case, running:

```
oraenv
```

simply displays current environment.

Execution With a Parameter

Running the utility with a new %ORACLE_SID% supplied on the command line, like this:

```
oraenv abc123
```

will set the Oracle environment to that of abc123 without asking, provided that abc123 is found in the ORATAB file in use.

Additional Utility Programs

The main utility is oraenv.cmd as described above. However, it is unable to perform with a support crew made up of the following executable files:

- DBHome.exe
- DBPath.exe
- TidyPath.exe

These are explained below.

DBHome

This utility simply finds the desired %ORACLE_SID%'s %ORACLE_HOME% by searching the ORATAB file for the supplied %ORACLE_SID%. For example:

```
DBHome abc123
```

will display, on stdout, the oracle_home for SID abc123.

This is picked up by oraenv.cmd and passed to DBPath.exe to remove the existing %ORACLE_HOME% from the %PATH% and to add the new %ORACLE_HOME% to the %PATH%.

Error Codes

The following error codes (in %ERRORLEVEL%) can be returned by the utility:

- 0: No error. Everything worked fine.
- 1: No %ORACLE_SID% was supplied on the command line.
- 2: Cannot open an ORATAB file.
- 3: The supplied %ORACLE_SID% cannot be found in the ORATAB file.
- 4: Memory allocation problem. (Highly unlikely!)

DBPath

This utility requires to be passed a path. That particular path is searched for in %PATH% and if found, *all* occurrences are removed from %PATH% to give the effect of removing the said path, usually representing an %ORACLE_HOME% from the current %PATH%. As the utility cannot adjust the caller's %PATH%, it simply displays the new path setting on stdout.

This is picked up by oraenv.cmd and used by SET to set a new value for %PATH% without any of the previous %ORACLE_SID%'s paths being present. The new %ORACLE_SID%'s \bin folder will subsequently be added to %PATH% to update the Oracle environment to match the new %ORACLE_SID%.

For example:

Running the code:

```
DBPath c:\Oracle\11.2.0
```

will remove any part of %PATH% which has c:\Oracle\11.2.0 present, so in this example, the following would all be removed from %PATH%, if they were present:

- c:\Oracle\11.2.0\
- c:\Oracle\11.2.0\bin
- c:\Oracle\11.2.0\OPatch
- c:\Oracle\11.2.0\perl\bin

Or anything else that has `c:\Oracle\11.2.0` in it's path name.

Error Codes

The following error codes (in `%ERRORLEVEL%`) can be returned by the utility:

- 0: No error. Everything worked fine.
- 1: No ORACLE_HOME supplied on the command line.
- 2: `%PATH%` is not defined in the caller.

TidyPath

Windows Control Panel allows you to set up various paths, in %PATH%, which may or may not have spaces, open or closing brackets etc in the path names. This is fine as it stands, however, in batch scripts - like `oraenv.cmd`, attempting to `set path=%ORACLE_HOME%\bin;%PATH%` when a path has spaces etc in it, *sometimes* results in an error similar to "\Microsoft not expected here". This was tracked to a path where the word "\Microsoft" was preceded by a space, and this caused the call to set the new `%PATH%` to fail. Nothing like consistency is there?

This utility is able to read the current `%PATH%` setting and tidies it up by scanning each part of it to find any unquoted paths with spaces, opening '(' or closing ')' brackets present. It outputs a new setting for `%PATH%` on stdout so that `oraenv.cmd` can pick it up and use it to set `%PATH%` correctly.

Error Codes

The following error codes (in `%ERRORLEVEL%`) can be returned by the utility:

- 0: No error. Everything worked fine.
- 1: `%PATH%` is not defined in the caller.