Using SQLPlus with rlwrap on MS Windows

By: Michael Paddock

At this year's HOTSOS Symposium Tanel Poder presented a training day that really highlighted some very practical ways to tailor your environment so that common tasks will be easier. One tip that I have really appreciated is how to use the Unix/Linux package rlwrap (readline wrapper). Using this with a few parameters and in conjunction with SQL*Plus makes a huge difference. It is now possible to have a command history and tab-enabled auto-completion of keywords available in SQL*Plus.

To simplify things even more aliases can be created to shorten a lengthy command. These aliases can be set to connect to specific databases so that Oracle's Easy Connect strings or TNS aliases don't have to be entered every time SQL*Plus is started. Here's an example of what you would store in your ~/.bash_aliases file:

alias sqltestdb='rlwrap -D2 -irc -b'\"@(){}[],+=&^%#;|\'\" -f ~/sql/wordfile_11gR2.txt $ORACLE_HOME/bin/sqlplus system/PASSWORD@test.mydomain.com/testdb.mydomian.com'

Now all your would have to enter is sqltesetdb to start the session. I won't break down all of the parameters but the wordfile_11gR2.txt is important because that's where the keywords are stored that the auto-complete capability uses. This can be handmade or (even better) you can get a copy of it along with Tanel's very useful SQL library at http://files.e2sn.com/scripts/tpt_public.zip.

This works beautifully in any Unix/Linux environment where the rlwrap utility is installed. But, what if you carry around a Windows laptop and want to keep all of your aliases with you? The easiest option is to set things up on a central Unix/Linux server that you can log in to and use as your alias' home. However, there is another option.

Windows has a "shell" environment called PowerShell that is flexible enough to allow similar functionality. This can be downloaded and installed in your current Windows environment.

PowerShell – http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=6ccb7e0d-8f1d-4b97-a397-47bcc8ba3806

First though, you need to install Cygwin and make sure that you specify rlwrap as one of the packages to be included. Then make sure that you add the c:\cygwin\bin directory to your Windows PATH.

Cygwin – cygwin.com

Now it's simply a matter of creating a few aliases and a function that is similar to the Unix/Linux version. PowerShell aliases are managed using these commands:

```
set-alias rw "c:\cygwin\bin\rlwrap.exe"

set-alias sqlplus "c:\oracle\11.2.0\bin\sqlplus.exe"
```

The function can be created using a DOS path to the wordfile_11gR2.txt file(c:\sql). But, PowerShell prefers a POSIX equivalent (/cygdrive/c/sql) for functions. Once this is complete simply execute "sqltestdb" to start SQL*Plus using the rlwrap capabilities. Using this modular approach allows you to have a separate function for each database.

```
function sqltestdb {rw -irc -f /cygdrive/c/sql/wordfile_11gR2.txt sqlplus
mpaddock/PASSWORD@testsrv.mydomain.com/testdb.mydomain.com}
set-alias sqltestdb
```

If you want to get rid of it:
remove-item function:sqltestdb:

To list the function definitions use this command:
get-content function:\FUNCTION NAME (You can replace FUNCTION NAME with * to list all functions).

It would be a good idea to list these and save the results to a text file as a backup.

The functions are persistent between functions so they won't need to be recreated. To make sure that the aliases for rw and sqlplus remain include the definitions in your $profile. You can find this file by typing $profile at the command prompt.

```
$profile
C:\Documents and Settings\mpaddock\My
Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
```

This a simple text file that can be edited be typing "notepad $profile". You can add the set-alias commands here and they will be available every time you start PowerShell.

PowerShell has a lot of interesting capabilities you can dig in to and explore. For me, I'm just looking to simplify things. The easier the command, the better off I am.