Deploying PL/SQL applications, building Rome in a day
Deploying PL/SQL applications, building Rome in a day
APEX on top

Need to know about Oracle APEX

Thursday, November 5, 2014

How to get a static IP address in an Oracle VirtualBox VM

You probably know that all Oracle software is free to download and most come with a developer license. This means that you can use full versions of products to develop your applications.

Installing the software could be a bit difficult especially when you are trying new programs on a machine with pre-installed servers, sometimes without Oracle 12c and Application Express pre-installed.

Follow these links to the pre-build VMs:

Pre-Build Developer VMs (for Oracle VM VirtualBox)

VirtualBox Appliance / Virtual Machine (my current favorite)

I got started using Oracle VirtualBox images the need for more free fast in a short time you have multiple VMs with different Oracle versions.

In the past those images started becoming a bit of a problem because the images get their IP from a DHCP server, embedded in the VirtualBox software. The first VM gets the IP available, the second the next and so on. So if you start the VMs in a different order, then entries won't connect to the default database anymore.
**Frésaguet**
- Gravel or broken stone (1-inch layer)
- Broken stone (2-inch layer)
- Foundation layer (8 inches)

**Telford**
- Gravel or broken stone (1-inch layer)
- Broken stone (7-inch layer)
- Foundation layer (7 inches)

**McAdam**
- Gravel or broken stone (1-inch layer)
- Broken stone (8-inch layer)
Inspirations

Developing

Installation

Upgrades

Final thoughts
Inspirations
Exporting the Database Objects

Generating a Script for Creating

Special Considerations for Exporting Sequences and Triggers

Database 2 Day Developer's Guide

Deploying a Database Application

Generating a Script for Creating

Generating a Script for Creating
Exporting the Database Objects

Generating a Script for Creating

Special Considerations for Exporting Sequences and Triggers

Database 2 Day Developer's Guide
Deploying a Database Application

Generating a Script for Creating
Generating a Script for Creating

To run installation scripts in SQL Developer:

8 steps to success!!
instead of repeating structure it while developing
Inspirations

Developing

Installation

Upgrades

Final thoughts
Developing
Risk of losing your code changes

Can’t rollback the changes you made
Never lose your code changes

Rollback the changes you made

Use the blame function
Developer is responsible for updating version control system

Temporary object should stay out of the installation

owner of sourcecode
Do whatever you want in the database, the code is in the version control system.

Ability to re-install your database scheme from the version control system.
organisation results in automation
my_application

branches
  v1.1.1

tags
  v0.9-beta
  v1.0
  v1.1

trunk
trunk
Contains the latest version of your software. This is where you develop in.

tags
Official deliveries are secured in this directory. This is where you create your installation zip from.

branches
Patches and big changes, which are done parallel to the normal development, are made here. Changes made here should also be made in the trunk somehow, sometime.
my_application
branches
v1.0.1

tags
v0.9-beta
v1.0
v1.0.1
trunk

Don’t forget to merge with the trunk!!
my_application

trunk

01. Documents

02. Releases

V0.9-beta
V1.0
V1.0.1
V2.0

use branches and tags
lower case
no spaces
why number the directories?
this belongs in the tags
Installation
java oracle.apex.APEXExport
   -db @!/dbhost!:1521!/servicename!
   -user !dbschema!
   -password !password!
   -applicationid !appid!

f100.sql
INSTALL

RECOMPILE all database objects!

- sequences
- journalling
- extra objects
- packages specs
- materialized views
- queue’s
- tables
- triggers
- types
- views
- packages bodies
- scheduler jobs
We need a SQL script that runs every package specification / body in a specific directory.
MS-DOS command line script which creates the SQL script

MS-DOS knows the file system
SQL*Plus doesn’t
SQL*Plus can run MS-DOS files

\database\install-remove\install.sql
@ECHO OFF
REM ===================================
REM == Prepare the Command Processor ==
REM ===================================
SETLOCAL ENABLEEXTENSIONS
SETLOCAL ENABLEDELAYEDEXPANSION
SETLOCAL

ENDLOCAL
SET outputfile=%0
SET outputfile=!outputfile:~0,-4!.sql
SET workdir=%~dp0
SET currdir=%~dp0

cd /d !workdir!

.
.
.
.
.
.
.
.
.
.
.
.
.
cd /d !currdir!

ECHO file "!outputfile!" created.
SET outputfile=%0
SET outputfile=!outputfile:~0,-4!.sql
SET workdir=%~dp0
SET currdir=%~dp0

cd /d !workdir!

cd /d !currdir!

ECHO file "!outputfile!" created.
SET outputfile=%0
SET outputfile=!outputfile:~0,-4!.sql
SET workdir=%~dp0
SET currdir=%~dp0

cd /d !workdir!

.
.
.
.
.
.
.
.
.
.
cd /d !currdir!

ECHO file "!outputfile!" created.
SET outputfile=%0
SET outputfile=!outputfile:~0,-4!.sql
SET workdir=%~dp0
SET currdir=%~dp0

cd /d !workdir!

.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
cd /d !currdir!

ECHO file "!outputfile!" created.
We need a SQL script that runs every package specification / body in a specific directory.
This file was generated on: !date! --
ECHO ---------------------------------------- > !outputfile!
ECHO -- This file was generated on: !date! -- >> !outputfile!
ECHO ---------------------------------------- >> !outputfile!
ECHO set define off >> !outputfile!
ECHO. >> !outputfile!

ECHO prompt ----------------------------- >> !outputfile!
ECHO prompt === Package Specifications === >> !outputfile!
ECHO prompt ----------------------------- >> !outputfile!
ECHO. >> !outputfile!
REM ================================================================
REM == list the _pkg.pks packages ==
REM ================================================================

if exist *_pkg.pks ( 
    FOR /F %%a IN ('dir /b *_pkg.pks') DO ( 
        ECHO prompt Running %%a >> !outputfile!
        ECHO @@%%a >> !outputfile!
    )
)
ECHO. >> !outputfile!

ECHO set define '^&' >> !outputfile!
ECHO. >> !outputfile!
REM ================================
REM == list the _pkg.pks packages ==
REM ================================

if exist *_pkg.pks (  
    FOR /F %%a IN ('dir /b *_pkg.pks') DO (  
        ECHO prompt Running %%a >> !outputfile!  
        ECHO @@%%a >> !outputfile!  
    )  
)  
ECHO. >> !outputfile!

ECHO set define '^&' >> !outputfile!  
ECHO. >> !outputfile!
REM ==================================================
REM == list the _pkg.pks packages ==
REM ==============================================================

if exist *_pkg.pks ( 
    FOR /F %%a IN ('dir /b *_pkg.pks') DO ( 
        ECHO prompt Running %%a >> !outputfile!
        ECHO @@%%a >> !outputfile!
    )
)
ECHO. >> !outputfile!

ECHO set define '^&' >> !outputfile!
ECHO. >> !outputfile!
set define off

prompt ==============================================================
prompt ==== Package Specifications =====
prompt ==============================================================

prompt Running dummy_pkg.pks
@@dummy_pkg.pks
prompt Running new_dummy_pkg.pks
@@new_dummy_pkg.pks

set define on
scripting tips
set define off

prompt ==================================
prompt ==== Package Specifications ======
prompt ==================================

prompt Running dummy_pkg.pks
@@dummy_pkg.pks
prompt Running new_dummy_pkg.pks
@@new_dummy_pkg.pks

set define on
set define off

prompt ==================================
prompt ==== Package Specifications ======
prompt ==================================

prompt Running dummy_pkg.pks
@@dummy_pkg.pks

prompt Running new_dummy_pkg.pks
@@new_dummy_pkg.pks

set define on
set define off

prompt ==================
prompt ==== Package Specifications =====
prompt ==================

prompt Running dummy_pkg.pks  @@dummy_pkg.pks
prompt Running new_dummy_pkg.pks  @@new_dummy_pkg.pks

set define on
sqlplus oracle/oracle@orcl @install_db.sql

@database/install-remove/install.sql

@invalid.sql  ->  trunk
@@invalid.sql  ->  trunk\database\install-remove
scripting tips
trunk

install.bat

=== Installatie Menu ===

Instellingen:
A Login database schema (username / password) : oracle / oracle
B Connect identifier : orcl

Z Instellingen naar default zetten

Uitvoeren:
1 Installatie schema en applicatie
Maak een keuze of kies Q om het menu te verlaten:
.	ools\install.bat

Instellingen:
A Login database schema (username / password) : oracle / oracle
B Connect identifier : orcl

Z Instellingen naar default zetten

Uitvoeren:
1 Installatie schema en applicatie

Maak een keuze of kies Q om het menu te verlaten:
```bash
\tools\install.bat

pushd ..\db\my_app

call install.bat

.\db\my_app\install.bat
```
@database/install-remove/install.sql

define sqlsubdir=database/install-remove
@@sqlsubdir/install.sql

@@sqlsubdir/./datamodel/tables.sql
scripting tips
JUST ONE MORE THING
define sqlsubdir=database/install-remove
@&sqlsubdir/install.sql

sqlplus oracle/oracle@orcl @database/install-remove/install.sql

Enter value for sqlsubdir:
!! to the rescue !!
--initialize sqlsubdir variable
set termout off
column s new_value sqlsubdir

select nvl('&sqlsubdir', '.') s from dual;
set termout on
--initialize sqlsubdir variable

set termout off
column s new_value sqlsubdir

select nvl('&sqlsubdir', '.') s from dual;
set termout on
--initialize sqlsubdir variable
set termout off
column s new_value sqlsubdir

select nvl('&sqlsubdir', '.') s from dual;
set termout on
--initialize sqlsubdir variable
set termout off
column s new_value sqlsubdir

select nvl('&sqlsubdir', '.') s from dual;
set termout on

Enter value for sqlsubdir:
--initialize sqlsubdir variable
set termout off
column s new_value sqlsubdir

select nvl('&sqlsubdir', '.') s from dual;
set termout on
--initialize sqlsubdir variable
set termout off
column s new_value sqlsubdir
select null s from dual;
select nvl('&sqlsubdir', '.') s from dual;
set termout on
--initialize sqlsubdir variable
set termout off
column s new_value sqlsubdir
select null s from dual where null is not null;
select nvl('&sqlsubdir', '.') s from dual;
set termout on
--initialize sqlsubdir variable
set termout off
column s new_value sqlsubdir
select null s from dual where null is not null;
select nvl('&sqlsubdir', '.') s from dual;
set termout on

@@&sqlsubdir/../datamodel/tables.sql
export application
- Inspirations
- Developing
- Installation
- Upgrades
- Final thoughts
Upgrades
‘Always’ delete objects which do not involve data

Create scripts to delete every removed object
The order in which the scripts run is important.
All changes to objects should also be in an update script.

Run the script multiple times without error or corrupting data.

Call other scripts from this script if dependencies failed.

Comply to naming rules for automatic script recognition.

Remove upgrade scripts after the software is delivered.
testing the upgrade

install.bat

v2.0

v0.9-beta

v1.0

v1.0.1

v2.0

upgrade.bat
testing the upgrade

install.bat

v2.0

upgrade.bat

v2.0
Instellingen:
A Login database schema (username / password) : oracle / oracle
B Connect identifier : orcl

Z Instellingen naar default zetten

Uitvoeren:
1 Installatie schema en applicatie

Maak een keuze of kies Q om het menu te verlaten:
Develop organized

Be sure all your scripts work all the time

Test your installer AND your upgrade

Compare your installed and upgraded application

Use the scripts to do a daily build
Final thoughts
ALL the `.sql` files including temporary objects for developing

ALL your installation, upgrade and export scripts

ALL your test scripts including unit and other testing scripts

DATA, put ALL your test-, initial filling-, referential- data in

Default data for all the tables for testing purposes
Kscope17 Preview

Wednesday, June 21, 2017
5:00 PM to 9:30 PM

Ordina Eindhoven
Luchthavenweg 55-D, Eindhoven (edit map)