

Deploying PL/SQL applications, building Rome in a day

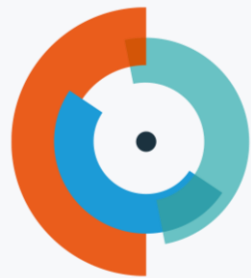
Alan Arentsen



Deploying PL/SQL applications, building Rome in a day

Alan Arentsen





Database Consultancy



ORACLE
ACE



alanarentsen.blogspot.com

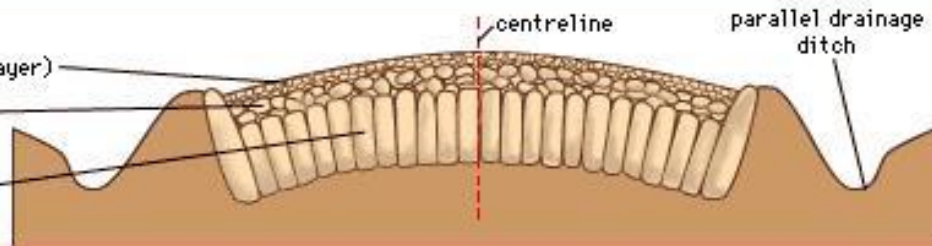


@alanarentsen



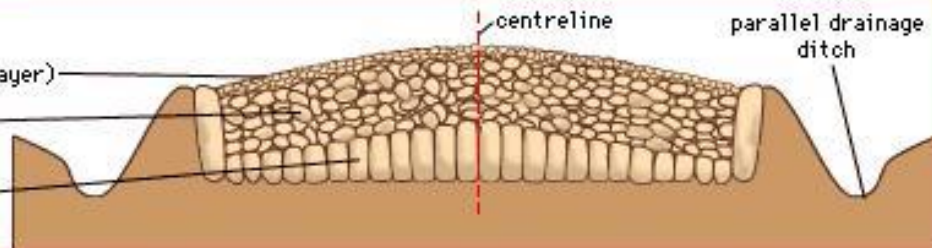
Trésaguet

- gravel or broken stone (1-inch layer)
- broken stone (2-inch layer)
- foundation layer (8 inches)



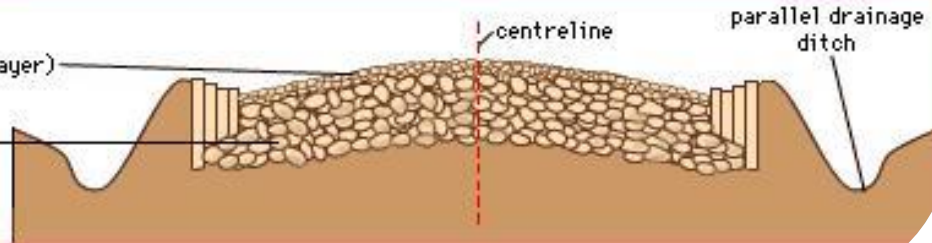
Telford

- 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

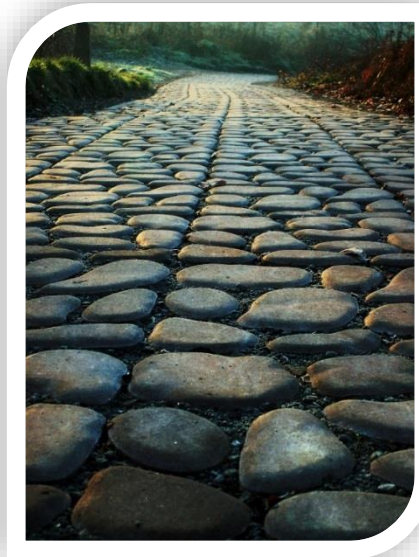


Upgrades



Final thoughts

Developing





file



database



git



HUDSON

Extensible continuous
integration server





Risk of losing your code changes



Can't rollback the changes you made



git



Never lose your code changes



Rollback the changes you made
Use the blame function







"No, it's MY data!"

Jim



owner of sourcecode

**Developer is responsible for
updating version control system**

**Temporary object should stay out
of the installation**

**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**



owner of sourcecode



organisation results in automation









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.



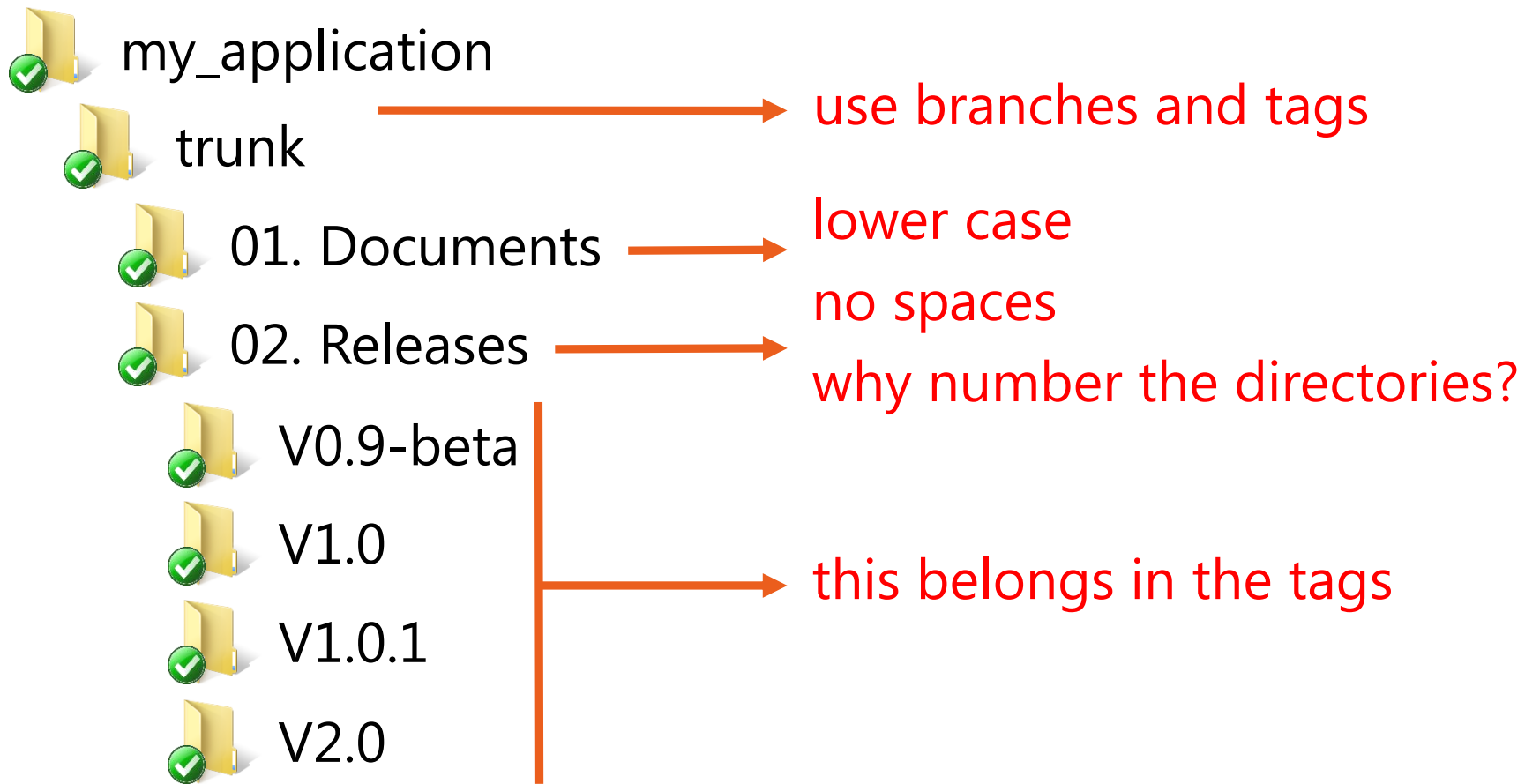
branches

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.



Don't forget to merge with the trunk!!







Inspirations



Developing



Installation



Upgrades



Final thoughts

Installation



1

2

3

install.bat

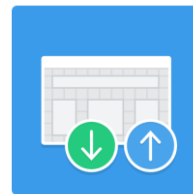
.\install_apex.sql

.\apex\install-remove\install_apex.sql

.\install_db.sql

.\database\install-remove\remove.sql

.\database\install-remove\install.sql



Export / Import

ORACLE
SQL Developer Command Line
SQLcl 4.2



```
java oracle.apex.APEXExport  
  -db @//!dbhost!:1521/!servicename!  
  -user !dbschema!  
  -password !password!  
  -applicationid !appid!
```

f100.sql



REMOVE



INSTALL



REMOVE

- ❌ **queue's**
- ❌ **scheduler jobs**
- ❌ **tables**
- ❌ **sequences**
- ❌ **views**
- ❌ **materialized views**
- ❌ **packages**
- ❌ **types**
- ❌ **logger**



INSTALL

- ✔ **sequences**
- ✔ **journaling**
- ✔ **extra objects**
- ✔ **packages specs**
- ✔ **materialized views**
- ✔ **queue's**
- ✔ **tables**
- ✔ **triggers**
- ✔ **types**
- ✔ **views**
- ✔ **packages bodies**
- ✔ **scheduler jobs**

RECOMPILE all database objects!



datamodel



sequences.sql



logger



journalling.sql



materialized_views



extra_objects.sql



packages



tables.sql



queues



scheduler



types



views

 datamodel

 logger

 materialized_views

 packages 

 dummy_pkg.pks

 queues

 dummy_pkg.pkb

 scheduler

 types

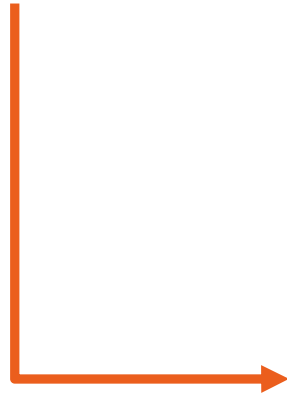
 views



dummy_pkg.pks



dummy_pkg.pkb



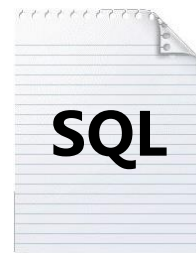


new_dummy_pkg.pks



new_dummy_pkg.pkb





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`



BASH

THE BOURNE-AGAIN SHELL



```
@ECHO OFF
```

```
REM =====
```

```
REM == Prepare the Command Processor ==
```

```
REM =====
```

```
SETLOCAL ENABLEEXTENSIONS
```

```
SETLOCAL ENABLEDELAYEDEXPANSION
```

```
SETLOCAL
```

-
-
-
-
-
-

```
ENDLOCAL
```

```
@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.

```
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!
```

```
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!
```

```
-----  
-- This file was generated on: zo 06-11-2016 --  
-----
```

```
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
```

 datamodel

 logger

 materialized_views

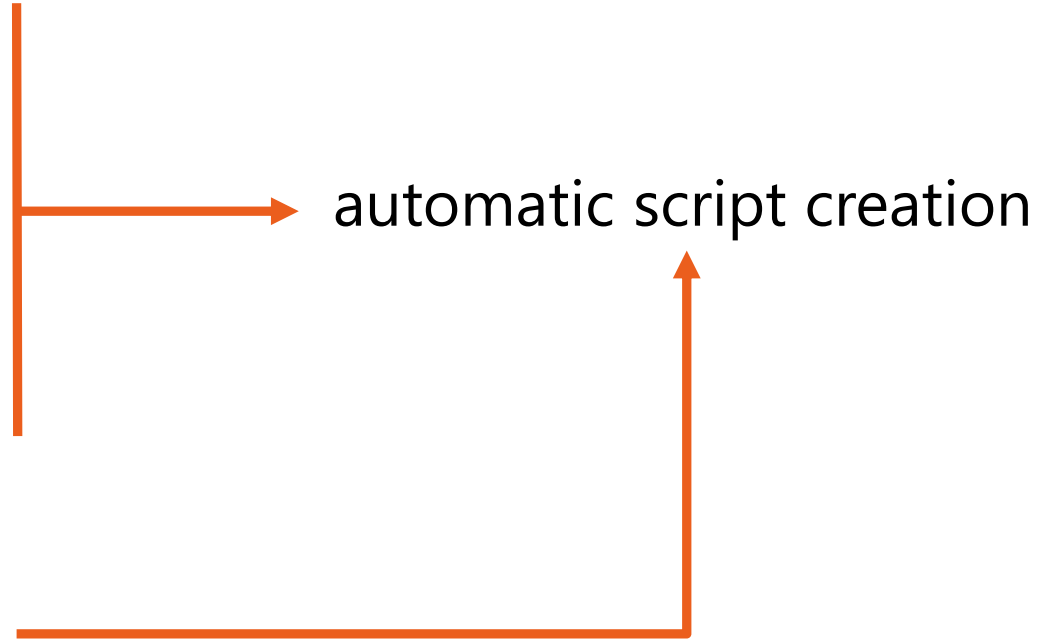
 packages

 queues

 scheduler

 types

 views





scripting tips

```
-----  
-- This file was generated on: zo 06-11-2016 --  
-----
```

```
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
```

```
-----  
-- This file was generated on: zo 06-11-2016 --  
-----
```

```
set define on @@dummy_pkg.pks
```

```
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
```

```
-----  
-- This file was generated on: zo 06-11-2016 --  
-----
```

```
set define on @@dummy_pkg.pks
```

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

```
prompt @@dummy_pkg.pks  
@@dummy_pkg.pks
```

```
prompt Running new_dummy_pkg.pks  
@@new_dummy_pkg.pks
```

```
set define on
```

 trunk

```
sqlplus oracle/oracle@orcl @install_db.sql
```

 trunk\database\install-remove

```
@database/install-remove/install.sql
```

@invalid.sql  trunk

@@invalid.sql  trunk\database\install-remove



scripting tips



trunk

install.bat



tools

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:
```



HUDSON
Extensible continuous
integration server

.\tools\install.bat

```
=====
=====                               Installatie                               =====
=====                               Menu                               =====
=====

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

Z Instellingen naar default zetten

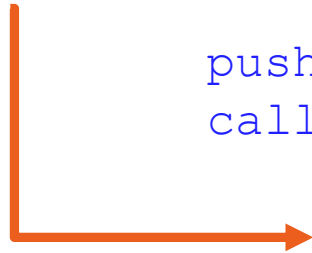
Uitvoeren:
1 Installatie schema en applicatie

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



.\tools\install.bat

```
pushd ..\db\my_app  
call install.bat
```



.\db\my_app\install.bat



datamodel



logger



materialized_views



packages



queues



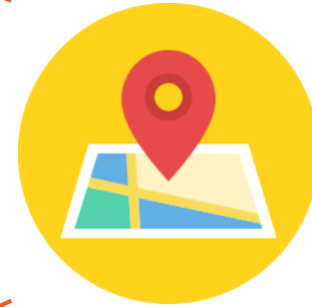
scheduler



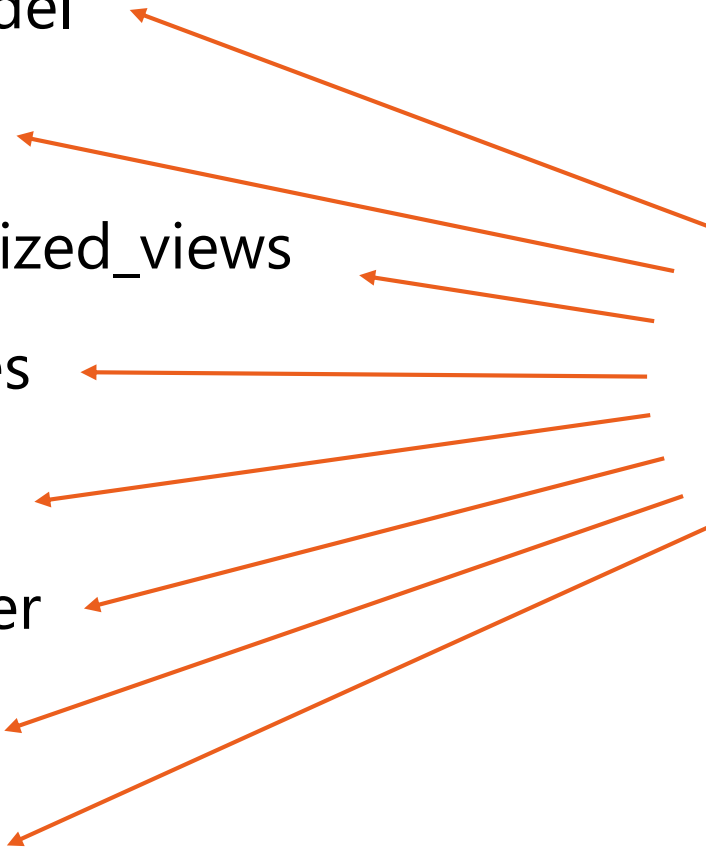
types



views



install.sql





```
@database/install-remove/install.sql
```



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



```
@@&sqlsubdir/../../datamodel/tables.sql
```



scripting tips



```
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

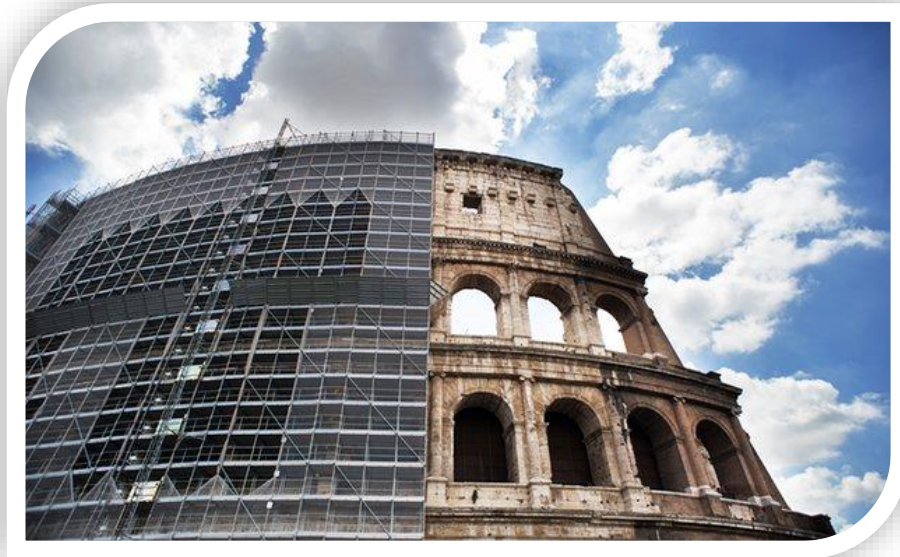


Upgrades



Final thoughts

Upgrades



1

2

3

upgrade.bat

.\install_apex.sql

.\apex\install-remove\install_apex.sql

.\upgrade_db.sql

.\database\upgrade\remove.sql

.\database\upgrade\upgrade.sql

- ✔ datamodel
- ✔ logger
- ✘ materialized views
- ✘ packages
- ✔ queues
- ✘ scheduler
- ✘ types
- ✘ views

'Always' delete objects which do not involve data



or

Create scripts to delete every removed object



datamodel



my_table_update.sql



logger



my_functional_update.sql



queues

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

 my_application

 branches

 tags

 v0.9-beta

 v1.0

 v1.0.1

 **v2.0**

 trunk



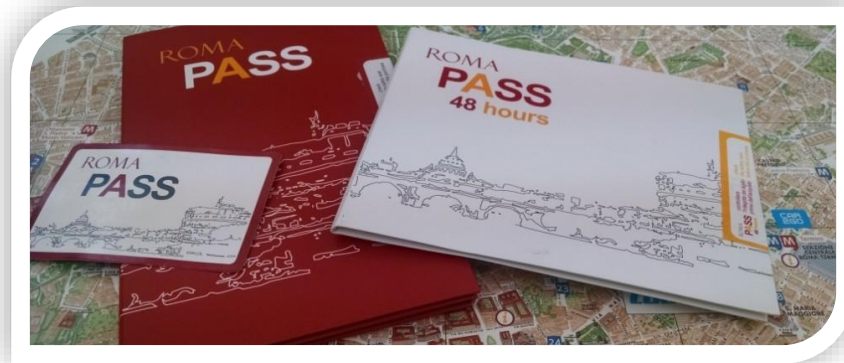
Develop organized

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



RIGHT



*Toilet Paper
Orientation*



WRONG

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

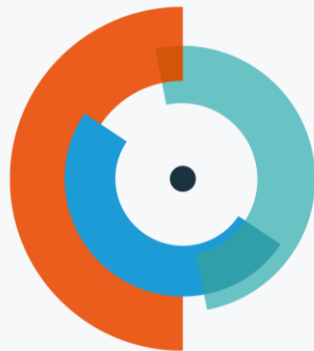




SEEK
TRUTH







Alan Arentsen
alan.arentsen@database-consultancy.nl

alanarentsen.blogspot.com

www.database-consultancy.nl
