

“TAMISO”

Overview



| | |
|----------------|--------------------|
| AUTORE | Mauro A. Rossolato |
| prima versione | 12.06.2016 |
| revisione | 15.03.2018 |
| sicurezza | interno |
| livello | draft |

Indice generale

| | |
|-------------------------------------|----|
| 1 OBJECTIVE..... | 4 |
| 2 WHAT IS | 4 |
| 3 WHY THIS ONE?..... | 4 |
| 3.1 ADVANTAGES..... | 4 |
| 4 ACTORS..... | 4 |
| 5 SCHEMAS..... | 5 |
| 5.1 SCHEMA REPOSITORY..... | 5 |
| 5.2 SCHEMA CLIENT..... | 5 |
| 6 MAIN TRIGGER..... | 6 |
| 6.1 TRIGGER SCOPE..... | 6 |
| 7 WORKFLOW..... | 7 |
| 8 INSTALLATION REPO..... | 7 |
| 9 AN EXAMPLE..... | 7 |
| 10 CLIENT INSTALLATION | 8 |
| 11 SCHEMA CLIENT CONFIGURATION..... | 9 |
| 11.1 NAMING CONVENTION..... | 9 |
| 11.2 CLIENT OBJECTS..... | 9 |
| 11.3 POST DEPLOY..... | 9 |
| 11.4 ADD TO REPO..... | 10 |
| 12 TMSID..... | 11 |
| 13 STRUCTURE..... | 12 |
| 13.1 TABLES..... | 12 |
| 13.2 TABLE'S TRIGGER..... | 12 |
| 14 SNIPPET..... | 13 |

Changelog

17.02.2018 – mr – inserito “rul3d”



fablabway.com
fablabway@gmail.com
+41 076 7535077

tms_note2.odt

1 OBJECTIVE

All about TAMISO: description and technical reference.



2 WHAT IS

TAMISO is a versioning platform for Oracle.

3 WHY THIS ONE?

There are several tools for script versioning: Git, SVN and so on.

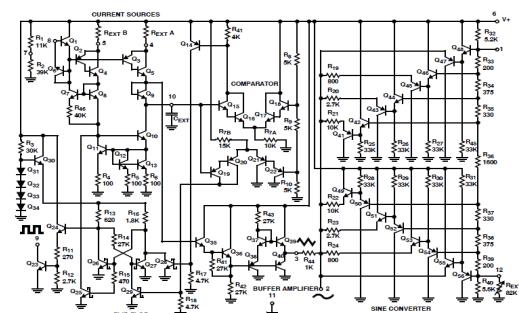
What difference makes TAMISO ?

Simple: TAMISO is built-in in your database. Each DDL will be captured before deploy, automatically. No more: "...oops, I'm sorry...".

3.1 ADVANTAGES

TAMISO is:

- easy to use
- no additional command/language is required
- save the time (automatic)
- no losses – each DDL is captured
- scalable: suitable for other languages or scripts



4 ACTORS

In the software development , there are people (or roles) as :

AD = ADministrator, project manager

AN = ANalyst

DV = DeVeloper

OP = OPerator

DA = DBA

In some circumstances a single person maintains more than a role,
e.g. the AN and DV.



5 SCHEMAS

TAMISO uses two kind of schema¹:

- repository
- client

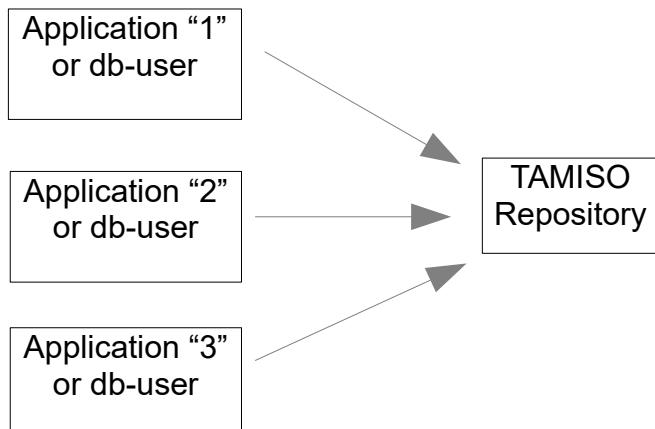
5.1 SCHEMA REPOSITORY

This schema (or db-user) contains the core tables of the TAMISO application.
The repository is used by one or more clients/application.

5.2 SCHEMA CLIENT

Is the schema managed, which contains the customer application.

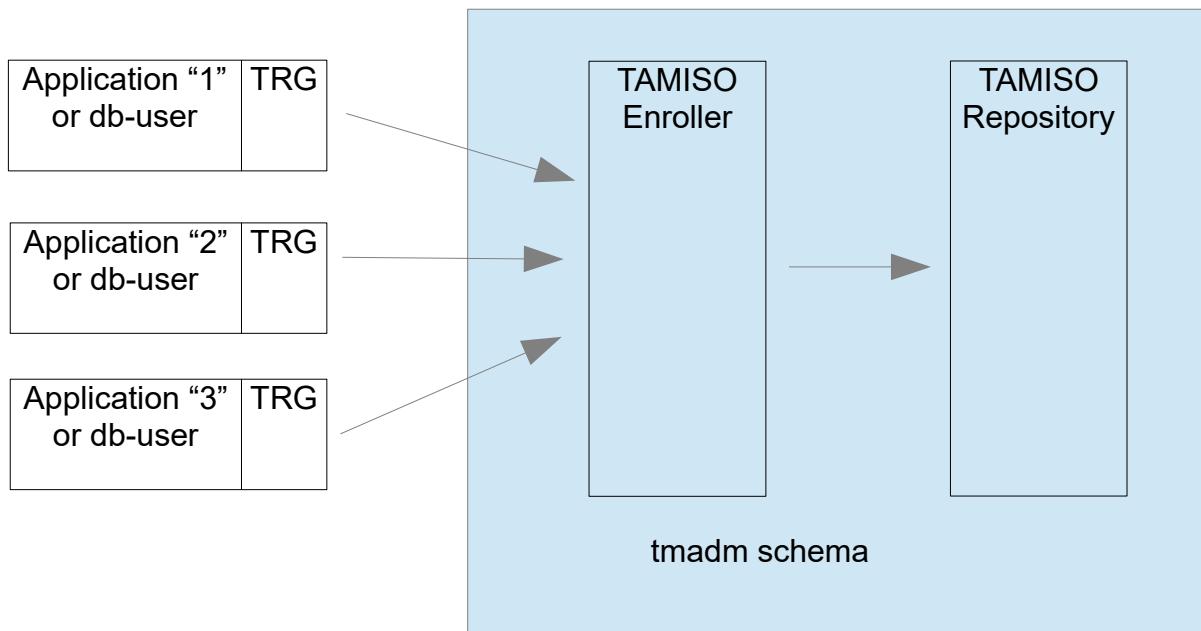
The developer works with schema-client objects. On the others words, the DV scope is limited to the schema-client.



¹ Also called “user” , on Oracle environment.

6 MAIN TRIGGER

The trigger Mo8BEFDDL_TR and TMDID represent the core of TAMISO application. This trigger, owned by the schema-client, fires for each DDL (defined) on schema-client. The automatic versioning is guaranteed by Mo8BEFDDL_TR: do not disable it!



6.1 TRIGGER SCOPE

The Mo8BEFDDL_TR is deployed into each schema-client and works at schema-level, however keep in mind that only categorized objects will be checked. On the others words the table Mo8CATOBJ is used sa “filter”. So, the Mo8BEFDDL_TR can “see” using Mo8CATOBJ.



7 WORKFLOW

The “workflow” contains the following tasks:

| ASSIGNEE | DESCRIPTION | |
|----------|--|--|
| AD | The Administrator defines a project (PJ). Resource, budget, ecc | |
| AN | The Analyst defines the object, component. Also dispatch parts o entire PJ to DV and assign TMSID to DV. Each DV has unique TMSID for the same project | |
| DV | The DeVeloper locks the objects (assigned) using the TMSID | |
| OP/DV | OP (or DV) executes the script with TMSID | |
| DV | DV unlocks the objects | |
| AD | AD closes the project (PJ) | |

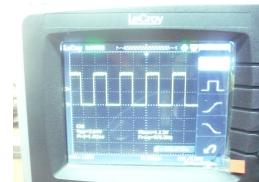
8 INSTALLATION REPO

The repository installation is very simple.

A new schema/user is required: TMSADM.

Before to launch the script, open, for editing, tmsmkrepo.sql

Change the password and tablespaces for the user. Then save and exit.



```
sqlplus /nolog
conn /as sysdba
@m08mkrepo
```

9 AN EXAMPLE

A new application is required.

The AD asks to DA for a new schema called “SELLCAR”

Now the repository is ready for new client.



10 CLIENT INSTALLATION

The client schema requires

- user
- grant on TMSADM (some objects)
- trigger

```
sqlplus /nolog  
conn /as sysdba  
@m08mkclient
```

as last step, DA creates the schema-DDL_trigger:

```
sqlplus /nolog  
SELLCAR/SELLCAR123  
@m08befddlusr
```

The DA activities ends here.

Starting from here the developer will create the required objects.



11 SCHEMA CLIENT CONFIGURATION

Before all, the AD have to create the “peoples of the project”:

```
sqlplus /nolog  
conn SELLCAR/SELLCAR123  
@m08ycmkuser
```

with these users (DV and AN) the AD can create the new project named “YOUCAR”:

```
sqlplus /nolog  
conn SELLCAR/SELLCAR123  
@m08ycmkprj
```

From here starts the DeVeloper's scenario.

First of all, DV defines the objects:

- a table for cars to sell
- a table with prospects
- a table with association cars and prospects²
- a view with cars not yet sold
- a procedure to add a new test-drive



11.1 NAMING CONVENTION

A good practice, should be to define the object name before to create them. To make so, the naming structure become:

m08 = as domain

yc = short name for YOUCAR

11.2 CLIENT OBJECTS

```
sqlplus /nolog  
conn SELLCAR/SELLCAR123  
@m08ycmkobj
```

11.3 POST DEPLOY

Obviously a simple test:

2 E.g. When a customer asks for a test drive.

```

DECLARE
WRETCODE NUMBER;
WRETMMSG VARCHAR2(100);
BEGIN
YCADDTEST ('CT215FZ',SYSDATE+3,'MAURO',
'377-2135141',WRETCODE, WRETMMSG);
DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;

SELECT * FROM YCCARTOSELL;

```

11.4 ADD TO REPO

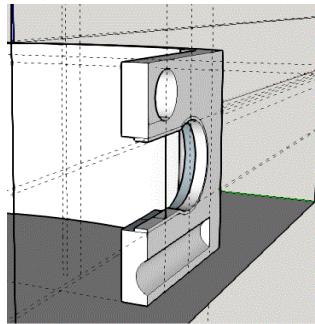
At this point TAMISO is ready to add the objects into the catalog³: in this manner TAMISO 'll be able to intercept any DDL on these objects.

```

sqlplus /nolog
conn SELLCAR/SELLCAR123
@m08yccatobj

```

The schema-client-setup is over. From now each DDL on schema⁴ will fire the trigger to preserve the previous statement.



³ The catalog resides into schema-server.

⁴ Limited to objects added into repository.

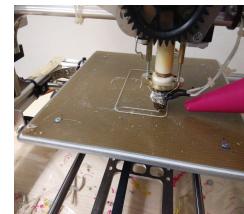
12 TMSID

The TMSID is the key attribute for TAMISO.

An object registered as work-in-progress can be modified only if TMSID is declared at compile-time.

This value is passed through the session to the procedure which stores the versions⁵.

The usage for TMSID is:

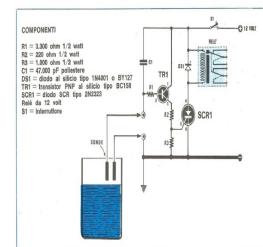


```
EXEC DBMS_APPLICATION_INFO.SET_CLIENT_INFO('YC001');
```

as you can see, the string "YC001" is issued before invoking the procedure

TMSADM.TMSMAIN.CATOBJ. Is an arbitrary string long 60 chars.

TMSID is the "passport" for objects release: without a correct TMSID the DDL will fails.



For example, when we update the object:

```
CREATE OR REPLACE VIEW YCCARTOSELL AS
SELECT T1.MMODEL,T1.MPRICE*1.184,T1.MDTONSALE,T3.WDRIVE
FROM YCCARS T1,
(SELECT FK_YCCARS,COUNT(*) WDRIVE FROM YCCARPRO WHERE 1=1 GROUP BY FK_YCCARS) T3
WHERE 1=1
AND T1.ID_YCCARS=T3.FK_YCCARS
AND T1.MDT SOLD IS NULL;
```

the previous statement will be added into Mo8DDLSRC.



⁵ TMSADM.ENROLLER procedure.

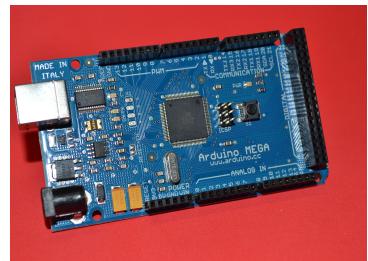
13 STRUCTURE

The TAMISO uses five tables, a package, a procedure some trigger and sequence.

13.1 TABLES

Mo8PARAM

Contains parameters specified for user.



Mo8PROJECT

List of projects

Mo8OBJCAT

List of objects that belong to project

Mo8OBJLCK

Object locked due development activity. Any objects belongs to Mo8OBJCAT.

Mo8DDLSRC

Containins the objects's code.

Mo8RELOBJ

parent of Mo8DDLSRC

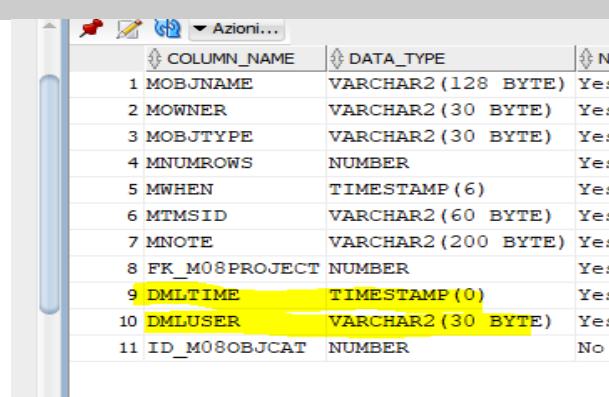
13.2 TABLE'S TRIGGER

Each tables has a couple of trigger,

- sequence
- audit (updater and timestamp)

Here is the complete list:

| | | |
|----------------|-----------------|------------|
| M08RELOBJ_PK | BEFORE EACH ROW | M08RELOBJ |
| M08RELOBJ_AUD | BEFORE EACH ROW | M08RELOBJ |
| M08PROJECT_PK | BEFORE EACH ROW | M08PROJECT |
| M08PROJECT_AUD | BEFORE EACH ROW | M08PROJECT |
| M08OBJLCK_PK | BEFORE EACH ROW | M08OBJLCK |
| M08OBJLCK_AUD | BEFORE EACH ROW | M08OBJLCK |
| M08USER_PK | BEFORE EACH ROW | M08USER |
| M08USER_AUD | BEFORE EACH ROW | M08USER |
| M08DDLSRC_PK | BEFORE EACH ROW | M08DDLSRC |
| M08DDLSRC_AUD | BEFORE EACH ROW | M08DDLSRC |
| M08OBJCAT_PK | BEFORE EACH ROW | M08OBJCAT |
| M08OBJCAT_AUD | BEFORE EACH ROW | M08OBJCAT |
| M08LOGEVT_AUD | BEFORE EACH ROW | M08LOGEVT |
| M08PARAM_PK | BEFORE EACH ROW | M08PARAM |
| M08PARAM_AUD | BEFORE EACH ROW | M08PARAM |



| COLUMN_NAME | DATA_TYPE | NOT NULL |
|-----------------|--------------------|----------|
| 1 MOBJNAME | VARCHAR2(128 BYTE) | Yes |
| 2 MOWNER | VARCHAR2(30 BYTE) | Yes |
| 3 MOBJTYPE | VARCHAR2(30 BYTE) | Yes |
| 4 MNUMROWS | NUMBER | Yes |
| 5 MWHEN | TIMESTAMP(6) | Yes |
| 6 MTMSID | VARCHAR2(60 BYTE) | Yes |
| 7 MNOTE | VARCHAR2(200 BYTE) | Yes |
| 8 FK_M08PROJECT | NUMBER | Yes |
| 9 DMLTIME | TIMESTAMP(0) | Yes |
| 10 DMLUSER | VARCHAR2(30 BYTE) | Yes |
| 11 ID_M08OBJCAT | NUMBER | No |

14 SNIPPET

For each objects of the given (like) project, how many versions?

```
SELECT T1.MPRJNAME, T2.MOWNER, T2.MOBJNAME, T2.MOBJTYPE, T3.WHOW
FROM M08PROJECT T1, M08OBJCAT T2,
(SELECT COUNT(*) WHOW,FK_M08OBJCAT FROM M08RELOBJ GROUP BY FK_M08OBJCAT) T3
WHERE 1=1
AND T1.ID_M08PROJECT=T2.FK_M08PROJECT
AND T2.ID_M08OBJCAT=T3.FK_M08OBJCAT
AND REGEXP_LIKE(T1.MPRJNAME,'YOU%', 'i');
```

