Solution-Soft®

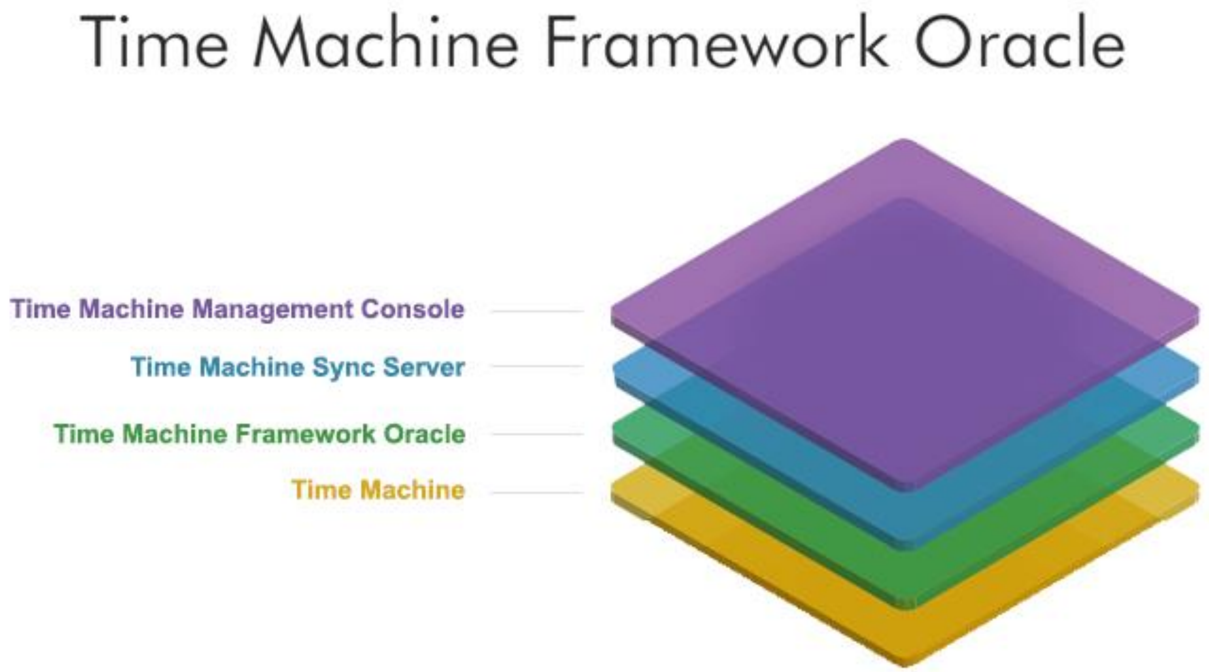
WHITE PAPER

TIME MACHINE FRAMEWORK FOR ORACLE 18c (TMFO) MULTITENANT DATABASES

TIME MACHINE FRAMEWORK FOR ORACLE (TMFO)

Time Machine Framework for Oracle (TMFO) is part of the Time Machine stack that provides more granularity of time travel within an Oracle instance, so multiple different virtual clocks can co-exist. TMFO enables customers to monitor all connections to an Oracle instance and to time travel specific connection with its own virtual clock with the console. Customers can also define, enable or disable rules, so virtual clock will be automatically set upon connection if the rule matches the filters on database user, host, or program.

TMFO has unique feature to time travel separate Oracle12c or Oracle18c tenant databases of a single Oracle instance independently thus providing more flexibility and independence in QA testing of applications which use separate tenant databases in a single QA Oracle instance.


Time Machine Framework Oracle
Time Machine Management Console
Time Machine Sync Server
Time Machine Framework Oracle
Time Machine

GRAPHIC 1

TM BENEFITS & KEY FEATURES:

- Enables multiple virtual clocks with the same Oracle instance
- Time travel by connection, program or database user within an Oracle instance
- Automate date and time sensitive testing by defining simple rules
- Intuitive and easy to use GUI
- Supports Linux, Windows, Solaris, AIX, HP-UX
- Supports all Oracle releases from 11g up to 18c
- Support independent time travel of tenant Oracle12c and Oracle18c databases
- Support of cloud Oracle deployments in AWS, Oracle Cloud, etc
- Supports PL/SQL API for easy scripting

This white paper provides detailed information about how to configure TMFO in separate Oracle18c tenant databases and time travel tenant databases to different virtual times within a single Oracle instance.

TMFO REQUIREMENTS

Before installing TMFO in Oracle18c, the following requirements must be satisfied:

- Time Machine (TM) must be installed on a host operating system before you can proceed to TMFO installation. You must ensure that Time Machine is successfully activated with valid license codes and the Time Machine service runs without any issues.
- Ensure that Time Machine installation directory is contained in PATH environment variable.
- The following TM versions are supported:
    - AIX 64 bit - Time Machine version 10 or later
    - HPUX Itanium - Time Machine version 10 or later
    - Linux x64 - Time Machine version 11.1R9 or later
    - Solaris SPARC - Time Machine version 10 or later
    - Windows x64 - Time Machine version 11.1R12 or later

TMFO INSTALLATION IN ORACLE 18C TENANT DATABASES

TMFO installation for Oracle18 multi-tenant databases includes three major steps:

- Install TMFO modules in the host where Oracle18c database is running
- Install TMFO database objects in first Oracle18c tenant database
- Install TMFO database objects in all subsequent Oracle18c tenant databases

TMFO installation in Oracle18c tenant databases is different for first tenant and all remaining tenants within a single Oracle instance.

TMFO MODULES INSTALLATION IN DATABASE HOST OPERATING SYSTEM

To install TMFO in a first Oracle18c tenant database, follow the steps below:

- Login to Oracle database server host as Oracle database software owner user. For Unix/Linux platforms, this user is called usually oracle.
- Create a new directory owned by Oracle database software owner user. This directory will be used as installation and working directory of TMFO product. This directory will be referred to as TMFO_DIR. For example, on Linux create tmfo folder by the command:

  ```
  $mkdir /home/oracle/tmfo
  ```

- Extract the TMFO distribution archive to the TMFO_DIR folder.
- Go to TMFO_DIR folder, open tmWorkPck.sql script file in any editor program and configure the following required parameters:
  - *tmOwner* - name of the Oracle tenant database schema which will be used to store necessary database objects and TMFO PL/SQL API. This parameter should be set to the schema name (in upper case). In most cases, TMFO schema name is TM.

For example,

```
tmOwner            varchar2(100) := 'TM';
```

- *tmUserProgram* - absolute path to Time Machine tmuser program executable;

For example, on Unix/Linux platforms:

```
tmUserProgram       varchar2(1000) := '/etc/ssstm/tmuser';
```

- *tmWorkDir* - absolute path to TMFO_DIR folder.

For example, on Unix/Linux platforms:

```
tmWorkDir          varchar2(1000) := '/home/oracle/tmfo';
```

- Proceed to TMFO installation in a first Oracle18c tenant database.

TMFO INSTALLATION IN FIRST TENANT DATABASE

If you install TMFO in a first tenant database, please follow the steps below which provide example steps for tenant database PDB1. Otherwise, refer to the topic “TMFO Installation in Subsequent Tenant Databases”.

- Go to the TMFO_DIR folder.
- Run SQL*Plus and connect to a first Oracle18c tenant database as SYS user.
- Create a new schema which will be used to store necessary database objects and TMFO PL/SQL API. The schema name must be set to the name configured in *tmOwner* parameter in tmWorkPck.sql script. For example, run SQL statements as SYS user in the PDB1 tenant database:

```
$ sqlplus sys as sysdba

-- connect to first Oracle18c tenant database PDB1

SQL> ALTER SESSION SET CONTAINER = PDB1;
Session altered.

SQL> CREATE USER TM IDENTIFIED BY pwd;
User created.

SQL> GRANT CONNECT,RESOURCE,UNLIMITED TABLESPACE to tm;
Grant succeeded.
```

- Run install.sql script in first tenant database PDB1:

```
SQL> show con_name

CON_NAME
------------------------------
PDB1

SQL> @install.sql
```

- When prompted, enter the name of TMFO schema name configured by the parameter *tmOwner*.
- Wait until script execution completes and verify that no errors occurred in the log file install.text created in the TMFO_DIR.
- If no errors were generated, TMFO framework is installed successfully and ready to use in the first tenant database.
- Optionally, you can grant EXECUTE privilege on TMWORK PL/SQL package to other database users who will use TMFO product.

For example, login in SQL*Plus to the tenant database under the TMFO user and execute the following SQL statement to grant EXECUTE privilege to a database user USER1:

```
$ sqlplus tm/tm@PDB1

SQL> show con_name

CON_NAME
------------------------------
PDB1

SQL> grant execute on tmwork to user1;
Grant succeeded.
```

TMFO INSTALLATION IN SUBSEQUENT TENANT DATABASES

After successful TMFO installation in the first tenant database, you can optionally install TMFO in the remaining tenant databases. Below steps provide installation example for tenant database PDB2:

- Go to the TMFO_DIR folder.
- Run SQL*Plus and connect to a target Oracle18c tenant database as SYS user.
- Create a new schema which will be used to store necessary database objects and TMFO PL/SQL API in this tenant. The schema name must be set to the name configured in *tmOwner* parameter in tmWorkPck.sql script. For example, run SQL statements as SYS user in the PDB2 tenant database:

```
$ sqlplus sys as sysdba

-- connect to the Oracle18c tenant database PDB2

SQL> ALTER SESSION SET CONTAINER = PDB2;
Session altered.

SQL> CREATE USER TM IDENTIFIED BY pwd;
User created.

SQL> GRANT CONNECT,RESOURCE,UNLIMITED TABLESPACE to tm;
Grant succeeded.
```

- Run addDB.sql script in the tenant database PDB2:

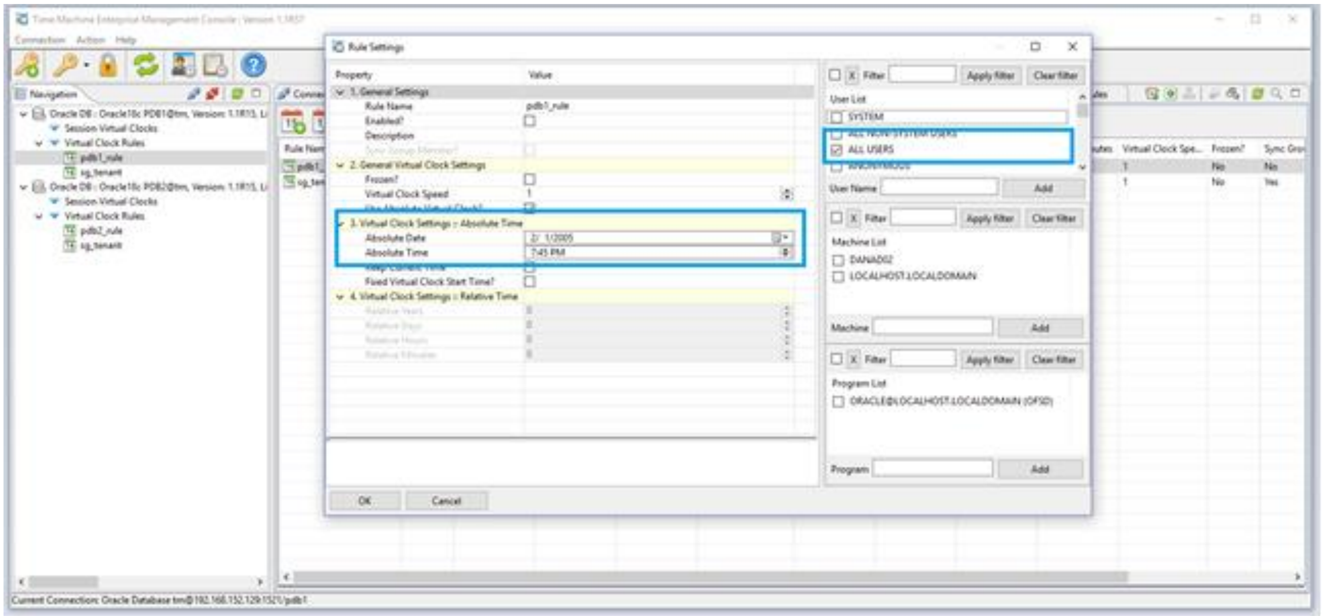
```
SQL> show con_name

CON_NAME
------------------------------
PDB2

SQL> @addDB.sql
```

- When prompted, enter the name of TMFO schema name configured by the parameter *tmOwner*.
- Wait until script execution completes and verify that no errors occurred in the log file addDB.text created in the TMFO_DIR.
- If no errors were generated, TMFO framework is installed successfully and ready to use in the tenant database.

TIME TRAVEL ORACLE18C TENANT DATABASE

TMFO allows to setup different virtual times for different Oracle18c tenant databases. This enables customers to easily perform independent time shift QA testing in cloud based databases. This white paper shows provides example of how to configure different virtual times for 2 Oracle 18c tenant databases PDB1 and PDB2 with TMFO and TMConsole Enterprise products.

- In TMConsole Enterprise connect to the TMFO repository installed in the tenant database PDB1.
- In the “Oracle Database Virtual Clock Rules” configure a new rule “pdb1_rule” to enable virtual time for the tenant PDB1:



Rule Settings

Property	Value
1. General Settings	
Rule Name	pdb1_rule
Enabled?	□
Description	
2. General Virtual Clock Settings	
Frozen?	□
Virtual Clock Speed	1
3. Virtual Clock Settings :: Absolute Time	
Absolute Date	2/ 1/2005
Absolute Time	7:45 PM
Fixed Virtual Clock Start Time?	□
4. Virtual Clock Settings :: Relative Time	

User List: □ SYSTEM, □ ALL NON-SYSTEM USERS, ☑ ALL USERS, □ ANONYMOUS

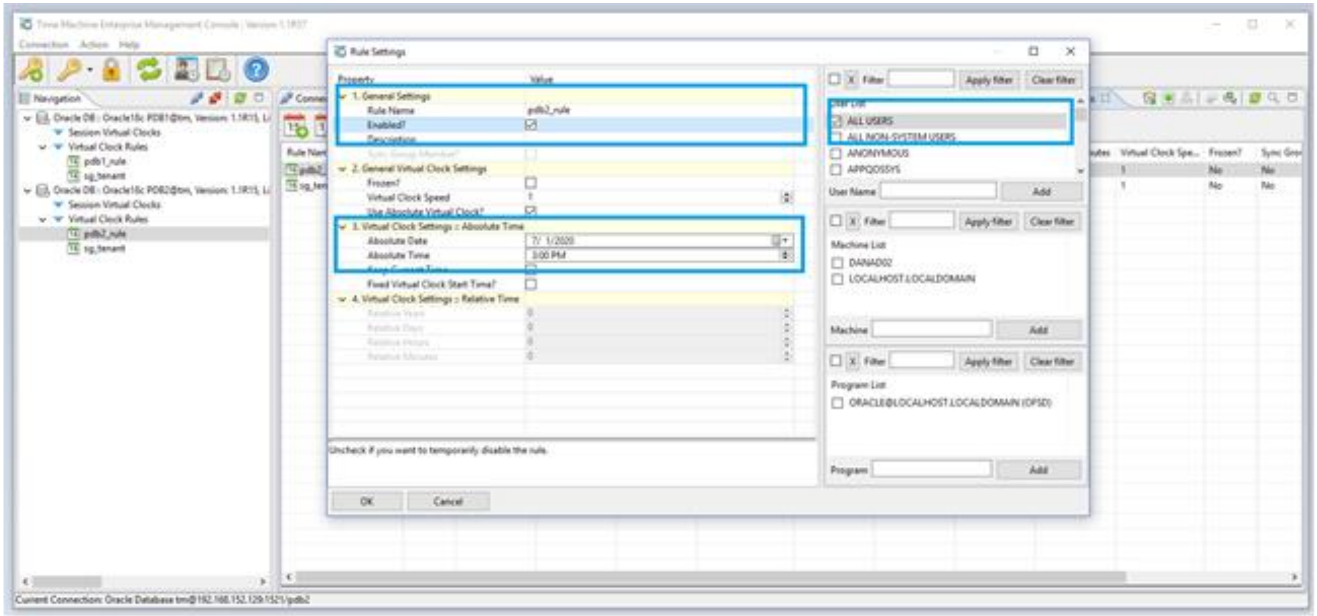
Machine List: □ DANAD02, □ LOCALHOST.LOCALDOMAIN

Program List: □ ORACLE@LOCALHOST.LOCALDOMAIN (OFSD)

Current Connection: Oracle Database tm@192.168.152.129:1521/pdb1

GRAPHIC 2

- In the “User List” you can either check “ALL USERS” to enable virtual clock for all tenant database users or select particular users which are specific for your application.
- You can optionally configure additional rule filters based on client host names and application names.
- Check “Enabled?” check box and press OK to activate virtual clock for the tenant PDB1.
- Connect to the tenant PDB2 and configure rule for this tenant:



Rule Settings

Property	Value
1. General Settings	
Rule Name	pdb2_rule
Enabled?	☑
Description	
2. General Virtual Clock Settings	
Frozen?	□
Virtual Clock Speed	1
Use Absolute Virtual Clock?	☑
3. Virtual Clock Settings :: Absolute Time	
Absolute Date	7/ 1/2020
Absolute Time	3:00 PM
Fixed Virtual Clock Start Time?	□
4. Virtual Clock Settings :: Relative Time	

Uncheck if you want to temporarily disable the rule.

User List: ☑ ALL USERS, □ ALL NON-SYSTEM USERS, □ ANONYMOUS, □ APPQOSSYS

Machine List: □ DANAD02, □ LOCALHOST.LOCALDOMAIN

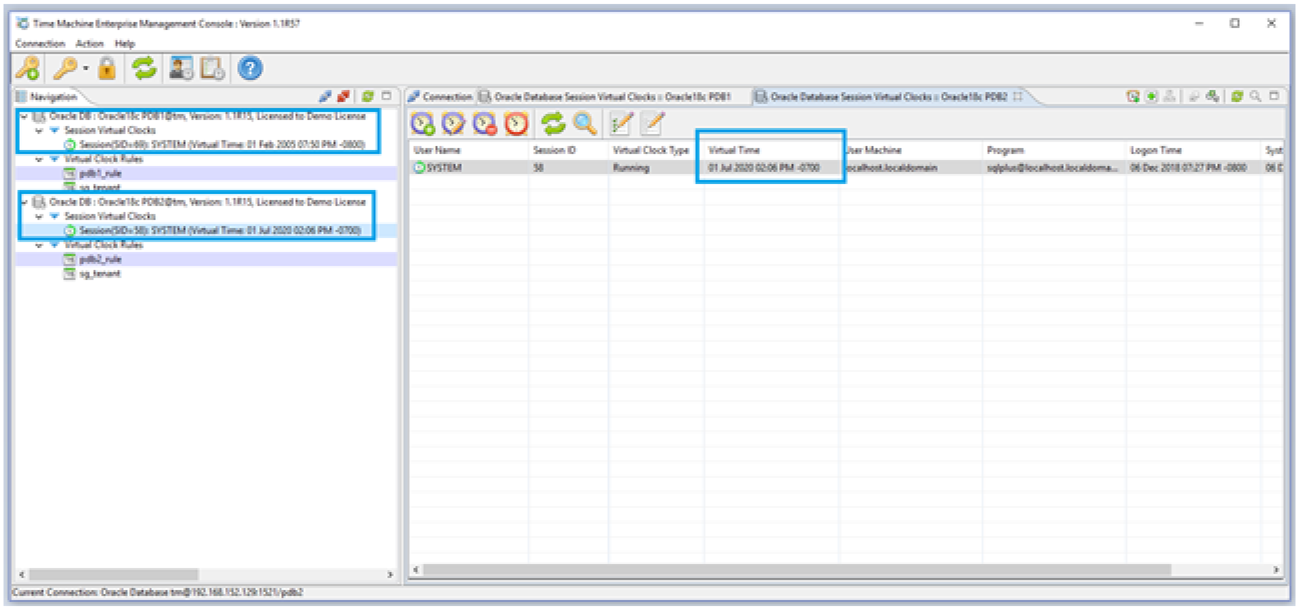
Program List: □ ORACLE@LOCALHOST.LOCALDOMAIN (OFSD)

Current Connection: Oracle Database tm@192.168.152.129:1521/pdb2

GRAPHIC 3

- You can optionally configure additional rule filters based on client host names and application names. Save your changes.
- Save your changes.

Now, you have configured different virtual times for two tenant databases PDB1 and PDB2. Let's connect to both tenants and verify if applications see correct virtual times in every tenant. In TMConsole Enterprise, it is also possible to verify actual virtual times for all user sessions in each tenant database:


Time Machine Enterprise Management Console : Version 1.1R57
Connection Action Help
Navigation
Connection
Oracle Database Session Virtual Clocks :: Oracle18c PDB1
Oracle Database Session Virtual Clocks :: Oracle18c PDB2

Oracle DB : Oracle18c PDB1@tm, Version: 1.1R15, Licensed to Demo License
Session Virtual Clocks
Session(SID=60): SYSTEM (Virtual Time: 01 Feb 2005 07:50 PM -0800)
Virtual Clock Rules
pdb1_rule
sg_tenant
Oracle DB : Oracle18c PDB2@tm, Version: 1.1R15, Licensed to Demo License
Session Virtual Clocks
Session(SID=58): SYSTEM (Virtual Time: 01 Jul 2020 02:06 PM -0700)
Virtual Clock Rules
pdb2_rule
sg_tenant

User Name	Session ID	Virtual Clock Type	Virtual Time	User Machine	Program	Logon Time	Syst
SYSTEM	58	Running	01 Jul 2020 02:06 PM -0700	localhost.localdomain	sqlplus@localhost.localdoma...	06 Dec 2018 07:27 PM -0800	06 D

Current Connection: Oracle Database tm@192.168.152.129:1521/pdb2

GRAPHIC 4

Oracle Cloud Ready

The Time Machine Framework for Oracle was developed and tested in the Oracle Database Cloud Service and is listed in the Oracle Cloud Marketplace.

Visit our page in the *Oracle Cloud Marketplace*

Solution-Soft®

SolutionSoft Systems, Inc.
2350 Mission College Blvd., Suite #777
Santa Clara, CA 95054, U.S.A.

Phone: 1.408.346.1400
Fax: 1.408.346.1499
Sales: 1.408.346.1415

www.solution-soft.com

www.facebook.com/solution-soft
www.twitter.com/solution-soft
www.linkedin.com/solution-soft

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.
Time Machine and Solution-Soft are registered trademarks of SolutionSoft Systems, Inc.
All other trademarks are properties of their respective owners.