



## WHITE PAPER

### TIME MACHINE® PRODUCT SUITE

#### USING VIRTUAL CLOCKS FOR TIME SHIFT TESTING

Many industries have business processes triggered by time based events. Whether you're testing settlement dates, billing cycles, or payment schedules, time based testing is an integral part of a tester's toolkit when it comes to ensuring optimal test coverage.

Test engineers frequently lack sufficient resources to complete comprehensive testing – from a test environment and a test data point of view. The commonly used method of waiting for a time trigger to occur (e.g. month-end cycle, end-of-day processing, etc.) is often not a practical way to test given the ever increasing demands of time-to-market conflicting with the desired full scope of time/date related test processes. A thorough software test process needs to involve running the applications with actual data in their target environment, executing the applications through an extensive range of past, present and future dates, and letting the test engineer and business users scrutinize the results for any functional inconsistencies. All of your existing programs and system functions should be tested in a current date/time environment while you test the future viability of the new program, in the same environment, interacting with all interfaces in an end-to-end test scenario.

Time Shift Testing (sometimes called Virtual Clock Testing) uses software virtual clocks to accomplish this goal. It makes the system being tested believe that it is the date or time needed to trigger the event we need to validate. Software virtual clocks are fast and easy to use. It is an alternative to changing the system clock on the server, which can be an extremely slow process with punishing unintended consequences. In many organizations, changing the system clock requires opening a ticket and having multiple administrators involved.

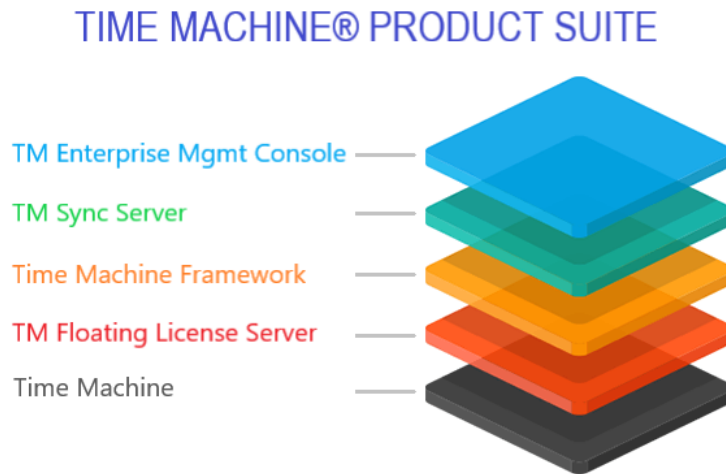
Changing the system clock in a secure network environment may also be prohibited. Active Directory and networks using Kerberos do not like to see member servers' clocks deviate more than a small specified amount of time and Group Policy Objects can be enabled not only to synchronize time on all network connected devices but also to prevent users from changing the system clock.

Time Shift Testing uses strategically placed software virtual clocks to time travel the system being validated into the past or the future. The particular approach is directly related to how the system queries the environment for the date and time and the underlying architecture of the environment itself.

Time Machine facilitates the creation of software virtual clocks. These virtual clocks enable Time Shift Testing without requiring a change to the system clock on the servers that make up the architecture of a deployed application.

## INTRODUCTION OF TIME MACHINE PRODUCT SUITE

To further enhance the experience of using Time Machine, Solution-Soft introduced the Time Machine Product Suite which additionally includes the Enterprise Management Console, Sync Server, Floating License Server, as well as a set of Frameworks for Time Machine and Time Machine for SAP Hana.



The frameworks increase time travel granularity for database instances and application pools to increase concurrency which can significantly improve overall speed of the execution in critical testing. Sync server is the key to broadcast virtual clocks to multiple systems in your environment and for test automation. Floating license Server (FLS) automates license management and is the enabler for dynamic environments in the enterprise or in the Cloud. Enterprise Management Console enables one console to monitor and manage all Time Machine Product Suite systems in the enterprise, on premise, or in the Cloud.

Today, Solution-Soft's Time Machine and its Product Suite have been utilized by DevOps teams for agile testing and test automation, besides the conventional application deployment. New Time Machine Products Suite features, including automation and dynamic broadcasting capabilities, deliver a total software testing solution for any complex Cloud migration, IT simplification, and Digital Transformation projects.

Additionally, the emergence of the Time Machine Product Suite has facilitated the evolution of Time Machine itself to fully support Docker containers and container-orchestration tools like Kubernetes, OpenShift, Docker Swarm, etc.

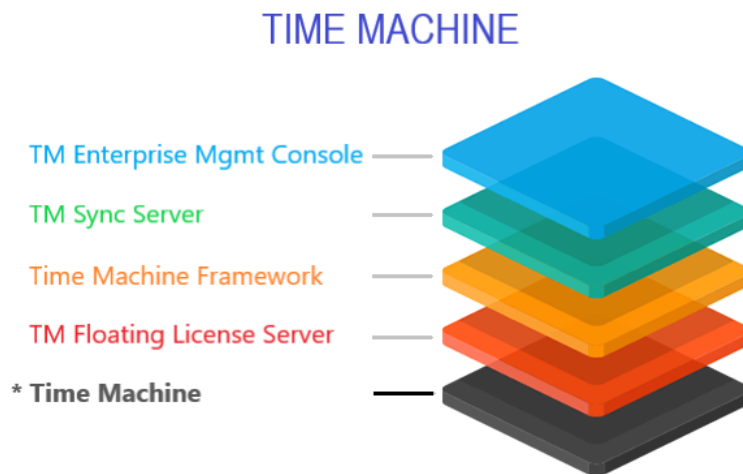
The key products from the Suite (beside Time Machine), that enable time travelling within the containers are the Floating License Server, the Sync Server, and the Enterprise Management Console. Containers check in and check out their licenses automatically and directly through the Floating License Server, getting and applying license keys without human intervention. Sync Server broadcasts virtual clock(s) to participating containers, allowing them to simultaneously time travel to the same virtual time. It also provides the URL API allowing for test automation and the ability to dynamically update the information which containers are actually participating in the time travel. Enterprise Management Console enables easy management of the virtual clocks across containers/pods and container hosts (e.g. a Kubernetes or an OpenShift cluster).

Currently, Time Machine Product Suite includes following products:

- Time Machine
- Time Machine Floating License Server
- Time Machine Frameworks for Database Servers (Oracle, MS SQL Server and PostgreSQL)
- Time Machine Frameworks for Application Servers (WebLogic, WebSphere and JBoss)
- Time Machine Sync Server
- Time Machine Enterprise Management Console

## TIME MACHINE

**Time Machine (TM)** creates software virtual clocks that allow you to time travel applications into the future or the past, and in that way perform time shift testing of date and time sensitive application logic, such as month-end, quarter-end, year-end processing, billing cycle, debt aging, regulation change date, etc.



TM is transparent to applications and databases, so no code modification is required to do time shift testing and system clock is never modified. TM eliminates the need to reset the system clock, which is time consuming, error prone and not possible under Active Directory or in a Kerberos secured environment.

The way TM works is by intercepting your file system's date and time calls. If the caller is configured with a virtual clock (and the program is not on the exclusion list), virtual time is returned, otherwise system time is returned. You can configure a virtual clock for an OS user (local or domain) where all the processes owned by that user will be affected by virtual time, OS user group or a particular process ID.

TM can provide hundreds of virtual clocks per system, each of which can be set to any time in the past, present, or future, and can run faster, slower, or at the same speed as the system clock. The variable speed clock feature allows software applications to travel through time at a user-defined pace. With this, QA or software simulation users can run their set of applications at any defined speed: up to 1,000 times faster than real-time or down to 1,000 times slower, or even create frozen virtual clocks, if needed.

TM saves resources of time and money by eliminating down time and the need to purchase duplicate systems for testing. Productivity is boosted since you are not limited to one clock at a time and you are able to expedite testing by running multiple clocks concurrently. In addition, with TM, you can automate test suites, instead of manually resetting the system clock every time.

Thousands of customers worldwide utilize TM successfully on a daily basis, including 47 of the Fortune 100 companies. Solution-Soft's customers are across all market sectors including 3M, AMEX, Australia Tax Office, BBC, Boeing, British Gas, Covered California, Discover, Fed Ex, Federal Reserve Bank, Orange, National Australia Bank, SNCF France, and Zurich Insurance.

With a recent surge for container orchestration and container workload transformation, TM has also fully adapted to support time travelling inside containers, be it a small-scale project with few containers, or a large-scale, enterprise approach to utilize TM to time travel hundreds of micro-containers to test date & time sensitive logic and run test automation.

**Time Machine for Containers** enables time travel in Docker containers by providing a collection of Docker images with Time Machine pre-installed. Container-aware Time Machine makes the virtual clock truly independent for containers, even when using the same user ID across containers.

Docker images are available on the Docker Hub, at the following location:

<https://hub.docker.com/u/solutionsoft>

If none of the currently available images are exactly what you need, you can always extend your container image with Time Machine, following the instructions from the link below:

<https://github.com/solution-soft/time-machine-dockerfiles/blob/master/README.md>

Also, if any assistance is needed with adding Time Machine to your existing Docker image, you can always contact Solution-Soft at [support@solution-soft.com](mailto:support@solution-soft.com), and Solution-Soft will actively assist you in creating the desired image.

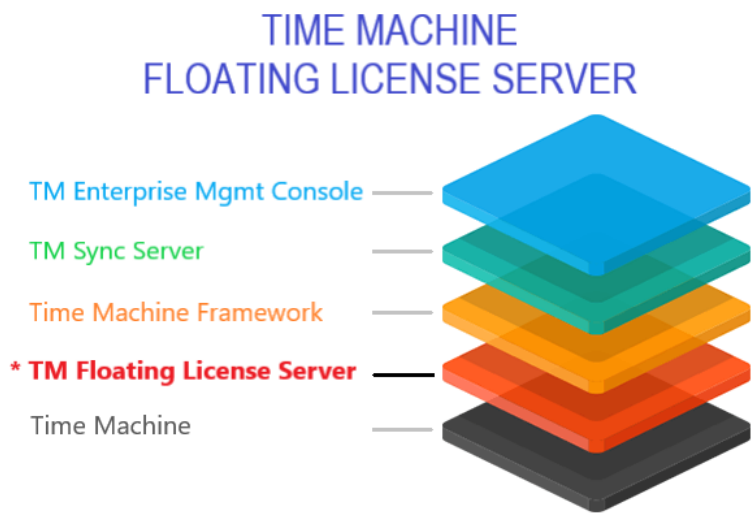
Time Machine Operator is Openshift certified and available on the Red Hat Marketplace, and Time Machine Container is certified and published in the Red Hat Container Catalog. At the same time, new capabilities and features are constantly being developed for different container use cases, in order to accommodate ever increasing customer demands.

**Time Machine for SAP HANA** is a special version of TM that enables a more granular time travel for HANA, the in-memory relational database offered by SAP. It allows to either time travel the instance as a whole, or time travel each tenant individually to different virtual clocks in parallel (similarly to TM Frameworks for DB servers that will be described later). Time Machine for SAP HANA delivers unprecedented scalability and productivity for SAP HANA migration and deployment projects, especially for the multi-tenant HANA configurations.

## TIME MACHINE FLOATING LICENSE SERVER

**Time Machine Floating License Server (TMFLS)** enables automatic license management for Time Machine across the enterprise, on premise and in the Cloud.

A pool of License Units (LUs) is hosted on TMFLS, so that the licenses can be checked out automatically by any Time Machine system when Time Machine service starts, and released once the service stops. This allows you to avoid the manual step of contacting the vendor in order to acquire a traditional node-locked license key each time when a new system comes up that needs to time travel.



As long as there are LUs left in the pool, any Time Machine system (configured for that) can grab the license automatically and time travel. As a result, LUs can be shared or floated to any Time Machine system within the enterprise or in the Cloud. So, if a test system comes up for testing for one hour, it would use the license for exactly one hour. Once it is done it would release it back to the pool for other systems to share.

The abstraction of a LU enables TMFLS to share the license across different OS platforms with any CPU core combinations. The same pool of license units can serve any Time Machine installed system on Linux, AIX, or Windows, or even in Docker containers. The Floating License Server also supports CPU core dynamic scaling, where the CPU core count might change dynamically based on the work load, so the proper number of LUs will be dynamically allocated on-the-fly, as well.

TMFLS should be hosted on a dedicated Linux or Windows server in the enterprise or in the Cloud. Any Time Machine system that would like to get a license from TMFLS would define its license host configuration file to point to TMFLS. The same Time Machine binary supports both modes of licensing. If a license host file exists, then Time Machine would get its license from TMFLS; otherwise, it would utilize the local license that is installed with a license key.

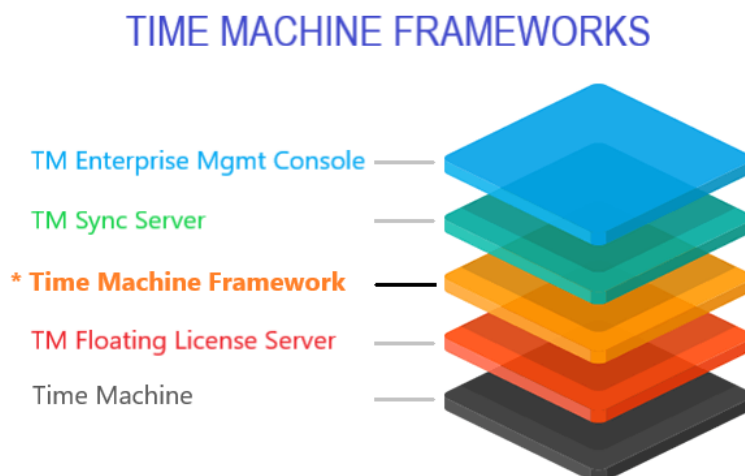
TMFLS allows administrators to see the snapshot state at real time: what is the license pool size, how many license units are utilized and free, all systems currently holding the licenses with its identity, OS, number of license units held, timestamps of first holding and last verified with the license.

There is also a separate audit trail that logs all license check-in and check-out activities. Configuration files also enable administrators to setup email notifications that alert to low license or out of license events.

TMFLS is especially powerful in the Cloud. It is common to provision systems on-the-fly as needed, so systems can come and go dynamically all the time. Without TMFLS, newly provisioned systems may need a manual step of contacting the vendor to get a license key, thus breaking the test automation.

## TIME MACHINE FRAMEWORKS FOR DATABASE SERVERS

Time Machine Frameworks for Database Servers are a part of the Time Machine stack that provide more granularity of time travelling within a database instance, so multiple different virtual clocks can co-exist.



**Time Machine Framework for Oracle (TMFO), Time Machine Framework for SQL Server (TMFSQL) and Time Machine Framework for PostgreSQL (TMFPG)** enable customers to monitor all connections to a respective database instance and to time travel specific connection with its own virtual clock via Time Machine Management Console.

In addition, 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 (and for TMFSQL and TMFPG the filter on the database used). What's more is that these rule filters can be used in combination with Time Machine Sync Server to broadcast a virtual clock to multiple targets in the sync group, such as both the database server and an application server on a different system.

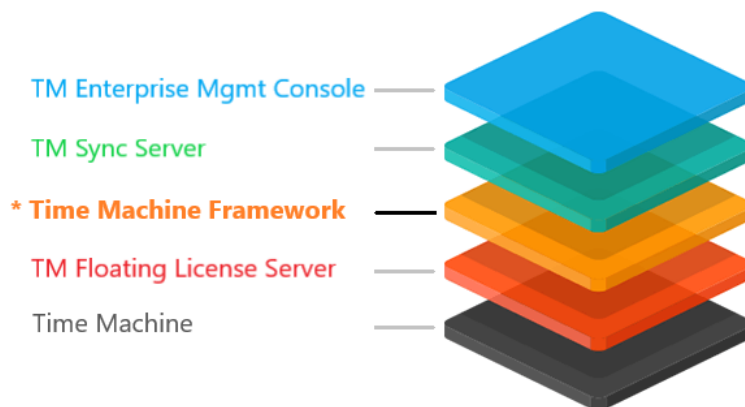
As a result, different sessions connecting to the same database instance can see different virtual clocks (or the system clock) based on the requirements and how users are sharing the instance. This is particularly useful for multi-tenant situations in the Cloud, where different customers may be sharing the same database instance and do not want to time travel to the same virtual time. You can connect to multiple databases, instances or cloud based servers simultaneously.

Also, TMFO additionally provides a PL/SQL API for functions supported by Time Machine. The Framework API can be used by developers and testers to create various test scenarios and scripts to perform time related testing and debugging of stored database routines, batch procedures, etc.

## TIME MACHINE FRAMEWORKS FOR APPLICATION SERVERS

Time Machine Frameworks for Application Servers are a part of the Time Machine stack that provide more granularity of time travelling within an application pool or domain, so multiple different virtual clocks can co-exist.

### TIME MACHINE FRAMEWORKS



**Time Machine Framework for WebLogic (TMFWL), Time Machine Framework for JBoss (TMFJB) and Time Machine Framework for WebSphere (TMFWS)** enable customers to monitor all respective managed servers and to time travel specific managed servers with their own virtual clock via TM Management Console.

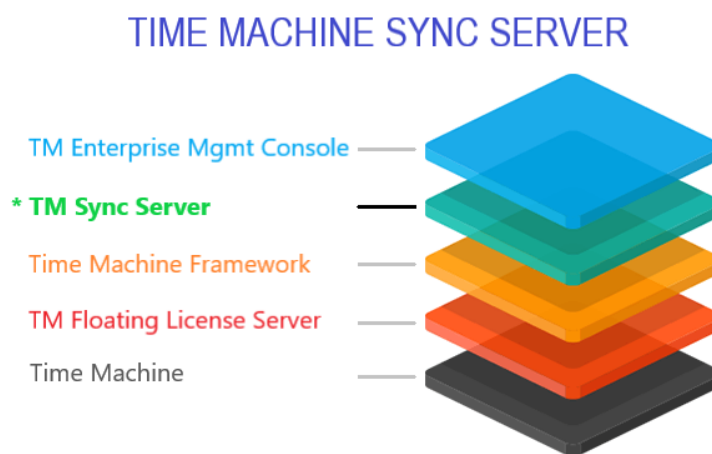
These Frameworks allow customers to use Time Machine functionality to time shift test Java applications within respective managed servers without the need to create a separate instance of the application server for testing other time related activities. The Frameworks also allow customers to configure and automatically create different virtual clocks on managed servers within the application pool, and the possibility to establish virtual clock rules automatically affecting applications deployed to the time travelled managed servers.

The Frameworks are controlled using the Time Machine Management Console, and with it, users can see the managed servers that are configured in the respective application servers, what applications are deployed and easily establish real-time virtual clocks or rules that determine exactly when a virtual clock will be created.

Also, TMFWL additionally provides a robust Java API that enables programmatic management of time shift testing functions for the WebLogic deployed applications.

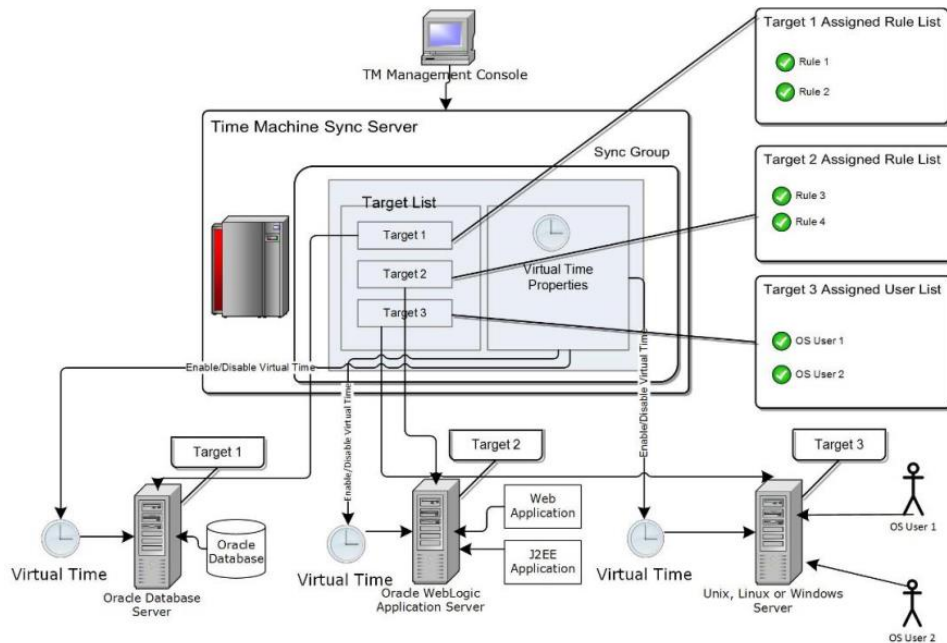
## TIME MACHINE SYNC SERVER

**Time Machine Sync Server (TMSS)** is an add-on software stack application that provides a centralized control center for broadcasting virtual clocks to all Time Machine servers in the enterprise.



TMSS uses Sync Groups to simultaneously broadcast virtual clocks to target systems that can be on premise or in the cloud, regardless of time zone or geographic location. They can be Time Machine clients running on UNIX, Linux and Windows, and can even include rules that have been defined with different Time Machine Frameworks.





TMSS can be utilized to configure password protected roles facilitating Access Control Lists, providing a certain group of users the ability to set virtual clocks and others to actually manage and configure the Sync Groups and their targets.

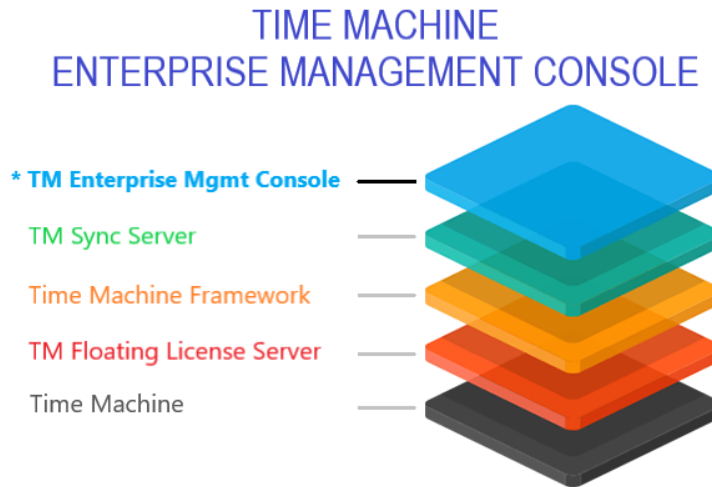
TMSS management can be done either with the GUI (Time Machine Management Console) or via built-in URL API from a test script.

TMSS has a built-in comprehensive URL API that allows you to enable, disable and set virtual time for a Sync Group programmatically. Just call the API as part of an automated testing script to enable the Sync Group, then proceed through the scripted test case(s), change the virtual time for the Sync Group to accommodate the next date centric test case and when virtual clocks are no longer needed, the Sync Group can be disabled via the API. The API calls could be initiated from industry leading tools from HP, Oracle, IBM, SmartBear, open source tools like Selenium or even in pure development in test environments.

The Time Machine Sync Server also offers the feature of virtual clock persistence. Part of the Sync Group configuration allows for the virtual clocks to persist through the restart of the servers often required for things like system maintenance and the application of security patches. If a virtual clock was set for the Sync Group and marked as persistent, then if one (or more) targets in the group is restarted, as soon as Time Machine and the Time Machine Agent start on the target(s), the virtual clock(s) will be reinstated in a way that enables it to be in sync with the other targets in the Sync Group.

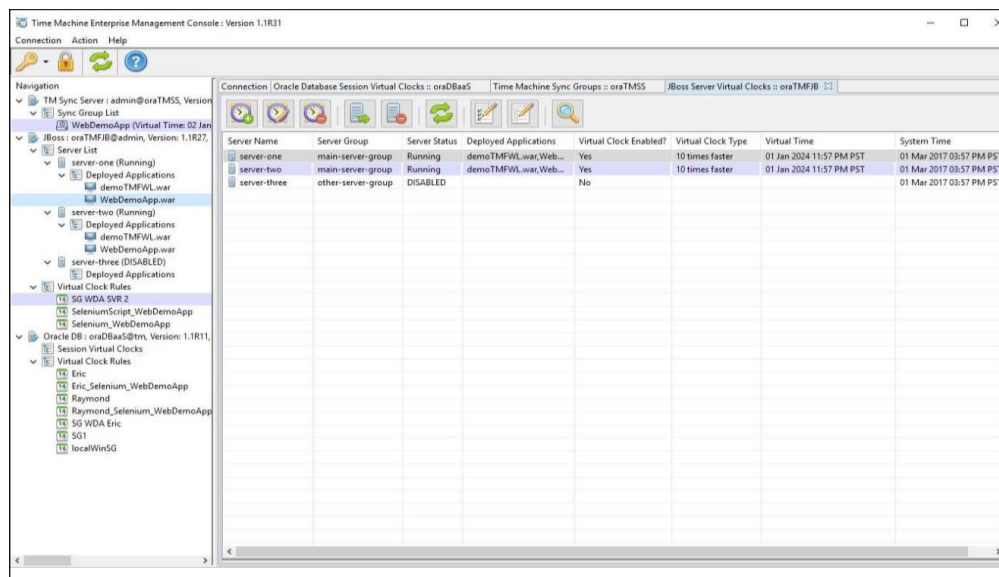
## TIME MACHINE ENTERPRISE MANAGEMENT CONSOLE

**Time Machine Enterprise Management Console (TMEMC)** provides an intuitive GUI specifically designed to navigate the Time Machine Suite of Products. If Time Machine is the engine of a vehicle, think of the TMEMC as the steering wheel.



TMEMC enables users to monitor virtual time for complex distributed applications and testing environments and manage easily and effectively all Time Machine products installed systems from a single interface, regardless whether they are on premise, or in the Cloud.

TMEMC improves the quality and convenience of complex application testing by providing a single virtual time management and configuration solution. You can configure manage and monitor virtual time for different distributed application components on UNIX, Linux, and Windows systems, from a single Management Console.



TMEMC also offers the following advanced features, which allow you to increase security in distributed enterprise environments:

- *Advanced user access management.* This feature provides the ability to configure and enable enhanced access control to TMEMC functions.

TMEMC allows you to setup authentication for Windows users which do not have administrator rights on the system where TMEMC is running. Also, it supports read-only users which are not allowed to do any changes in TMEMC.

- *Time Machine operations audit feature.* This feature provides system administrators with the possibility to audit all operations performed in TMEMC by different users who are authorized to use TMEMC on a given system.
- *Idle locking timeout.* This feature enables you to setup idle timeout for TMEMC. It automatically locks the application or the entire desktop in the case TMEMC application is idle during the configured timeout. After TMEMC is locked, users must enter a username and password again in order to unlock the application.

In addition to English, it supports following local languages: French, German, Russian, Chinese (both simplified and traditional) and Serbian.



## Solution-Soft

SolutionSoft Systems, Inc.  
2350 Mission College Blvd., Suite #777  
Santa Clara, CA 95054, U.S.A.  
Phone: 1.408.346.1400  
Sales: 1.408.346.1415

Europe  
SolutionSoft Systems, Inc.  
Trnska 8, Suite 7  
Belgrade, Serbia, 11000  
Phone: +1381 11 403 1523

[www.solution-soft.com](http://www.solution-soft.com)



[www.facebook.com/solution-soft](http://www.facebook.com/solution-soft)



[www.twitter.com/solution-soft](http://www.twitter.com/solution-soft)



[www.linkedin.com/solution-soft](http://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.

Copyright © 1993-2020 SolutionSoft Systems, Inc. All rights reserved. Time Machine and Solution-Soft are registered trademarks of SolutionSoft Systems, Inc. All other trademarks are properties of their respective owners.