

RECORDING REMOTE DESKTOP SESSIONS

INTRODUCING RECORDTS

Brought to you by **TSFactory**



<http://www.tsfactory.com>

Copyright Notice and Trademark

© 2016 TSFactory LLC. All Rights Reserved.

RecordTS and the TSFactory logo are registered trademarks or trademarks of TSFactory LLC, or its affiliated entities.

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TSFactory LLC.

Every effort has been made to ensure the accuracy of this manual. However, TSFactory LLC makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. TSFactory LLC shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. The information in this document is subject to change without notice.

Version 1.0 – Updated September 22nd, 2016

End User License Agreement

RecordTS by TSFactory LLC is protected by an End User License Agreement. To view the agreement, visit the company website at www.tsfactory.com, under RecordTS Documentation.

Recording Remote Desktop Sessions

Abstract

With the advent of server virtualization and cloud technology, there has been a need to capture and record user activity. RecordTS is a product that does exactly that.

Back to Basics

Computing technology has come a long way. Its early beginnings started in clean rooms filled with large, ceiling high mainframe computers. A separate room lined with rows of terminals on tables was for user access. Pull up a chair and log in remotely to the mainframe for your own personal computer account. You shared the system with dozens of other users, each being allocated a slice of the processor's time and memory.



Eventually personal computers evolved and everyone had one at home and on their desk at work. Life was good, but not necessarily for the IT departments that sprung up almost overnight to handle the never ending stream of technical problems. The cost of owning and maintain a mainframe was replaced with the cost of managing hundreds of personal workstations along with the multitude of software packages and optional hardware devices.

Time moves on and personal computers become more powerful, evolving into server grade systems. Packed with processing muscle and speed, the new servers quickly overtook their predecessors, the older massive mainframes. Now it was time to move the desktops back onto the servers and off the desks of the users. Users were given terminals so they could log in remotely to their desktops. Computing technology came full circle: Virtualization was born, *again*.



As server technology became more compact, server farms were cultivated and cloud computing came into existence. Whole office networks could be moved into the cloud and IT departments redeployed and reduced in size. Cloud providers popped up overnight to accommodate the transition, but in the haste, some elements were forgotten about.

One of those elements was session recording. Due to the nature of remote desktop access, session recording was not a simple thing to implement. Software was needed to handle the new virtualized architectures and that is where RecordTS comes into the story.

Session Recording

At the very heart of remote access is the terminal server. This is a server that is configured to host user desktops using Microsoft Terminal Services, now known as Remote Desktop Services or RDS. Users log in remotely to their desktops from either a workstation or terminal using Remote Desktop Connection client software. The client software talks to the server using the Remote Desktop communications protocol or RDP. The user shares the server with other users who are all isolated and protected from each other.



In this scenario, it is possible to insert a recording service between the client and the terminal service in a typical “man-in-the-middle” configuration. In this way, the recording service intercepts all the RDP traffic and makes a copy of the session traffic. The users have no clue this is happening and their sessions are not delayed or noticeably changed in any way.

The session data is stored in a database and can be played back like a video at any time. The recording system is completely configurable and centrally managed. Each component can be located separately to accommodate almost any network configuration. Recording is limited only by the number of desktops a server can realistically host. Literally thousands of servers can be recorded in large scale environments.

It is also possible to record workstations as well as virtual machines in a hosted environment.

The RecordTS Architecture

As stated earlier, the RecordTS modules can be located separately or all together onto one machine. The main components are:

Dashboard – a web console to centrally manage the RecordTS system.

From the Dashboard you can configure the system, manage the license service and licensing of components, view stored sessions and play them back, provide basic reporting of sessions and view usage statistics.

License Service – a background service that licenses the individual components.

Licenses are obtained from the user's online account located on the TSFactory webservers. Licensing is by subscription and can be updated continuously from the website.

RecordTS is licensed by user and server. User licenses are pulled from a common pool and are considered concurrent connections. Each recorder that is installed will pull a license from the common server license pool. Dashboards are also licensed, but typically only one is required in most installations.

Recorder – the recording engine.

The recorder service installs on the terminal server or workstation being recorded and intercepts RDP traffic. Session data is streamed to the database for storage. Recorders can be configured to buffer session data to prevent loss of data in cases of intermittent network or database connectivity. Users will not experience latency (a slowdown in session response time) or loss of connection due to temporary storage issues.

Database – session data is stored here for playback and analysis.

Two types of databases are currently supported:

- Microsoft SQL Server 2012 (full version)

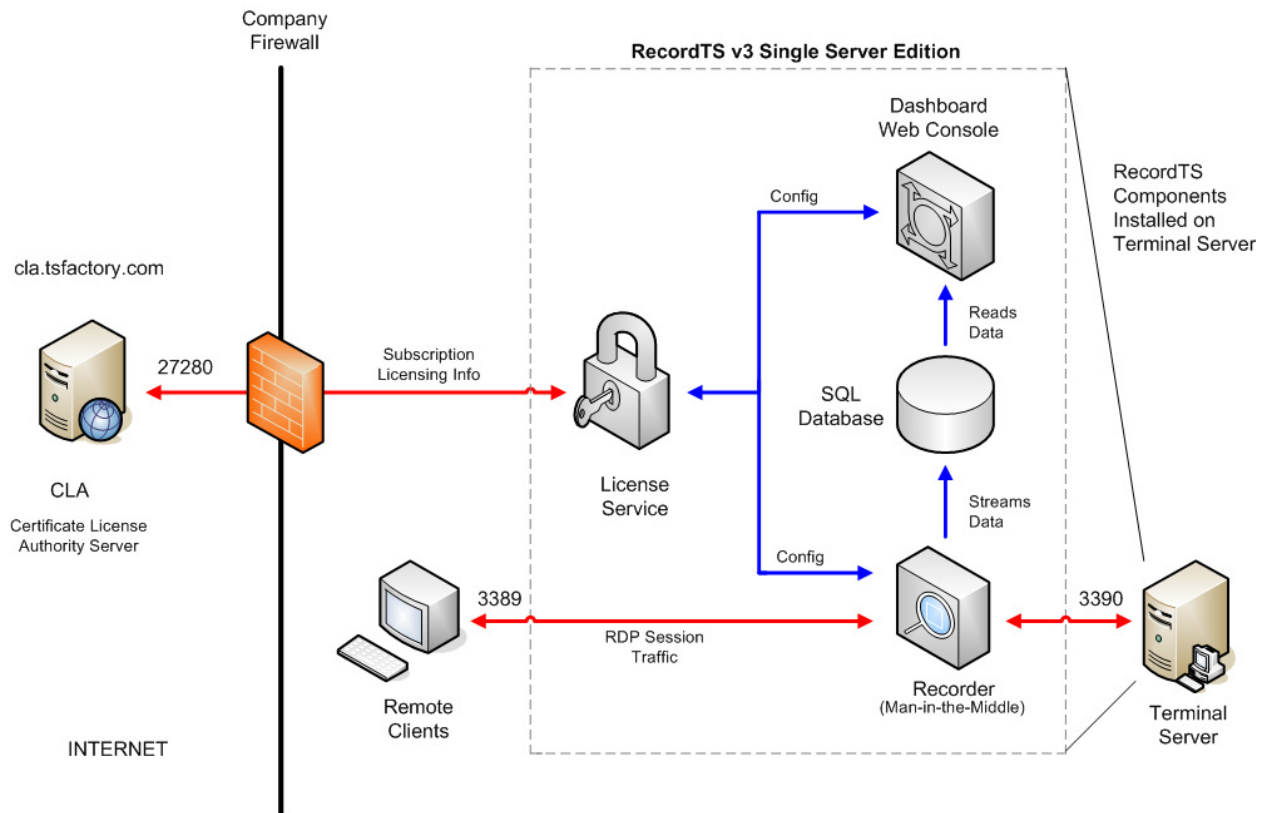
- PostgreSQL Server v9 or higher

The customer is required to install one of these databases for use with RecordTS.

Network Topologies

SMALL OFFICE, SINGLE SERVER INSTALL

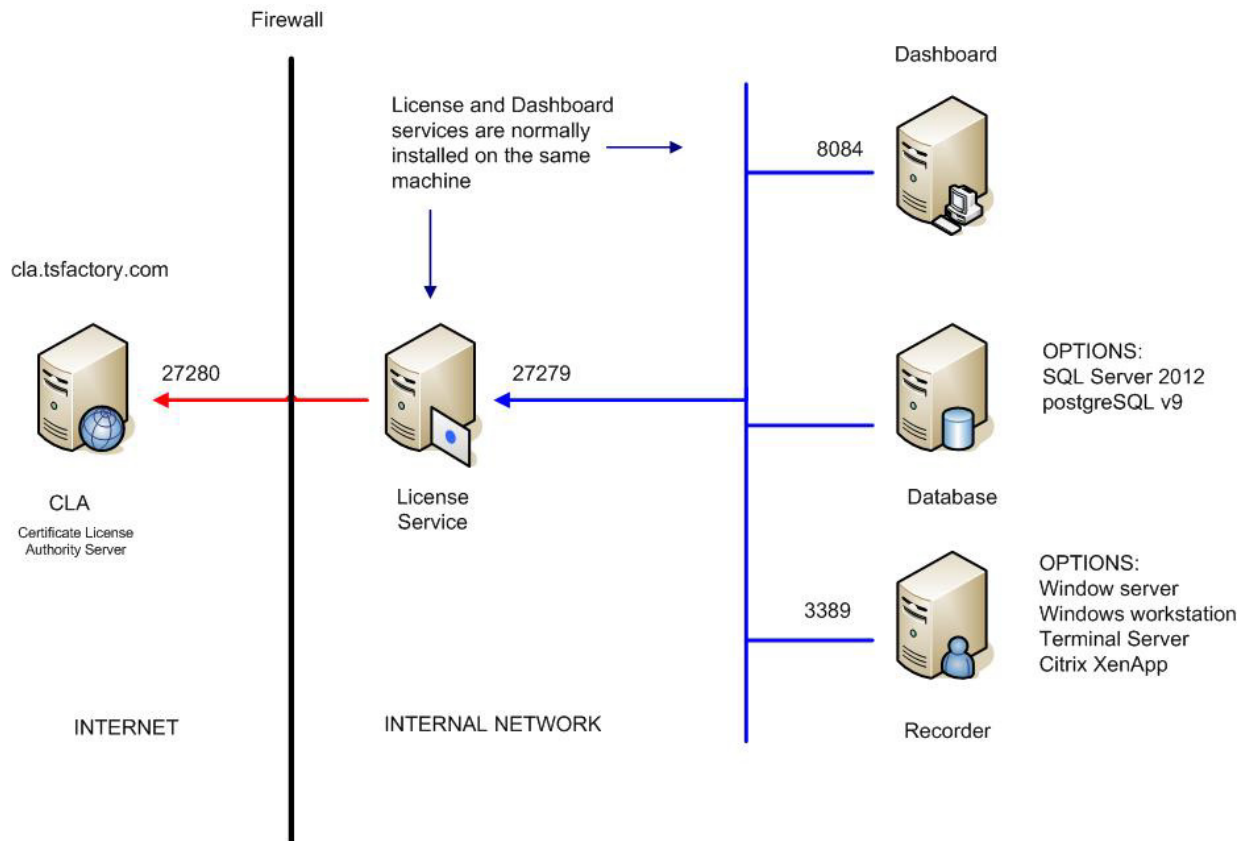
In small offices with only one server, RecordTS would be completely installed on the terminal server. This is a self-contained configuration that is easy to roll out and maintain.



RecordTS v3 Single Server Edition - internal logical framework

ENTERPRISE CONFIGURATION

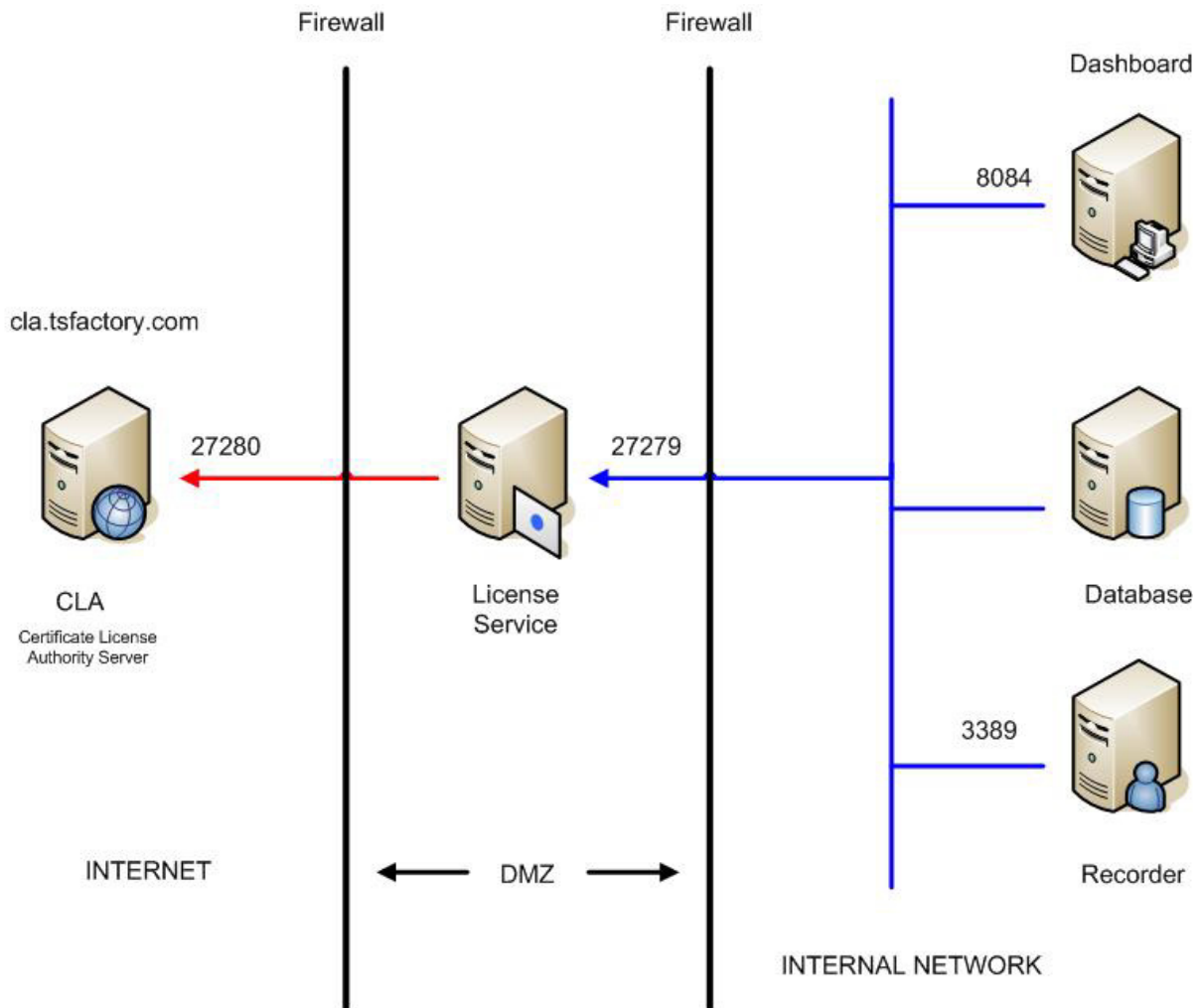
In offices with more than one server to record, RecordTS components would be installed on different machines. The license service and Dashboard should be collocated and the Recorder installed on each machine being recorded.



RecordTS v3 Network Configuration

ENTERPRISE CONFIG - LICENSE SERVER IN DMZ

The license service can be isolated into a DMZ to prevent direct access from the internet into the internal office network.



RecordTS v3 Network Configuration

License service placed in the DMZ