VMware Horizon Session Recording:

Fling Documentation, Version 2.0.x

Author: Andrew Morgan / (@andyjmorgan) / mandrew@vmware.com
Version: 2.0.x
Fling: https://flings.vmware.com/horizon-session-recording
Revised Date: May 18th, 2020.

Table of Contents

VMware Horizon Session Recording

Overview:

The VMware Horizon Session Recording solution contains is built with two items:

- The Recording Agent.
- The Recording Server.

The Agent is installed on all Horizon Machines from which you wish to record sessions. Once the Agent begins to record a user's session, it registers the session with the Server and begins to upload recording data.

The Server is installed on one (or more, for High Availability) machines to collect information about the session along with the raw recording data for storage and playback.

Recording Server:

The Session Recording server component comprises of 3 items:

- A Database for session information storage along with configuration.
- A windows NTFS folder for recording storage.
- A web service for collecting recording data, along with administration and playback.

Deployment types.

Depending on your installation use case and requirements, there are two deployment models to consider.

Stand Alone Deployment:
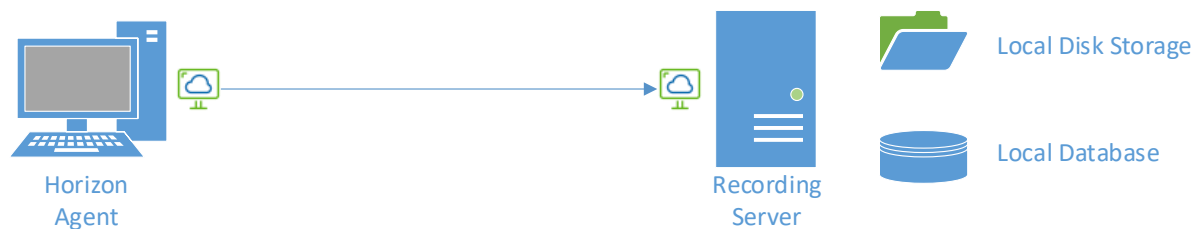
For a Proof of concept, or small setup which does not require redundancy, a single server is sufficient, leveraging a local SQLite database type and a local NTFS folder.

This database ships as standard and is stored in the installation directory, called "local.db".

Highly Available Deployment

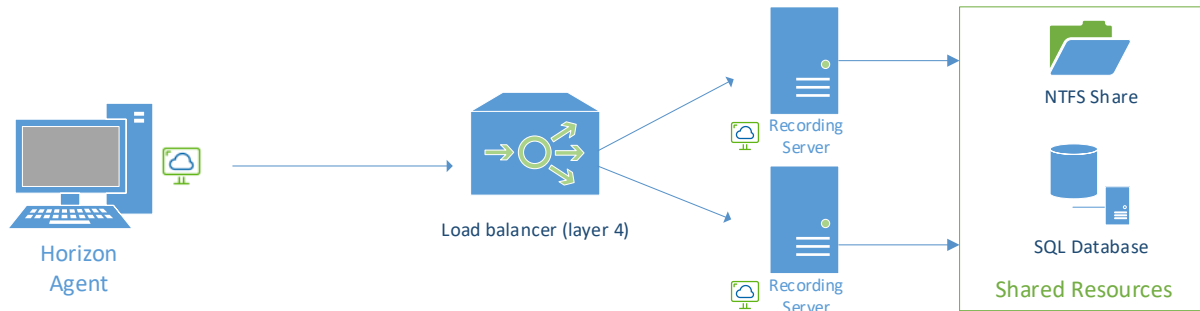For a larger, production deployment, you can load balance multiple recording servers behind a load balancer to service recording for your environment.

To deploy on more than one server, you can leverage the following database types:

- Microsoft SQL
- PostgreSQL

These databases can be leveraged on multiple servers in order to share configuration and session data. But it's vital that you also configure a shared NTFS Folder on all servers for storing recording data.

Pre-Requisites:

The Session Recording is built in .Net Core, so no framework is required.

Simply installed it onto a 64-bit (x64) desktop or server operating system which is a member of the domain if you wish to use group extraction.

Installing the Session Recording Server:

1. Download the HorizonSessionRecordingServer.exe file from the flings site and copy it to a local folder on the server.
2. Launch the installer and complete the steps.
3. Once complete, the server should now be available on https://<localservername>:9443

Upgrading the Session Recording Server:

The upgrade process is identical to the installation, but note that the existing settings, recordings and database will remain unaffected.

Uninstalling the Session Recording Server:

Use Add remove programs (appwiz.cpl) to remove the session recording server binaries. By default, the following components must be cleaned up manually:

1. the local database (local.db)
2. the logs (C:\programdata\VMware\Horizon Session Recording)
3. the recordings (if local, installation directory\recordings)

Resetting the installation:

If you need to reset the installation, for whatever reason. Follow the below steps:

- Stop the Horizon Session Recording Server service.
- Delete the recordings folder.
- Delete the servicesettings.json file in the start-up directory.
- Delete the local.db file in the start-up directory.
- Start the Horizon Session Recording Server service.

Securing the Recording Folder:
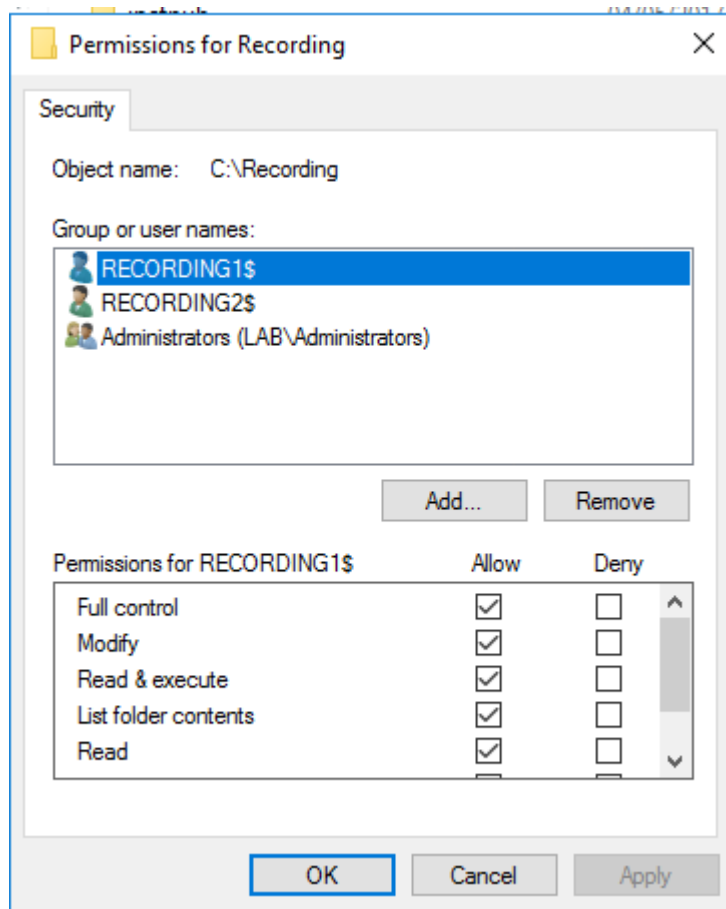
(the computer accounts for the computers in the cluster, recording1 & recording2 have full control)

By default, depending on the location of the recordings storage folder (local or ntfs share), the permissions on this directory may not be secure.

To properly secure the folders, ensure only the recording servers active directory accounts have access to the folder. This will ensure only the Recording Servers will have the ability to add, modify or delete recordings. This action is manual and is highly recommended.

Default Credentials:

The following credentials are stored in the database by default, the administrator has full control of the service settings, while the viewer can simply find and watch and download recordings.

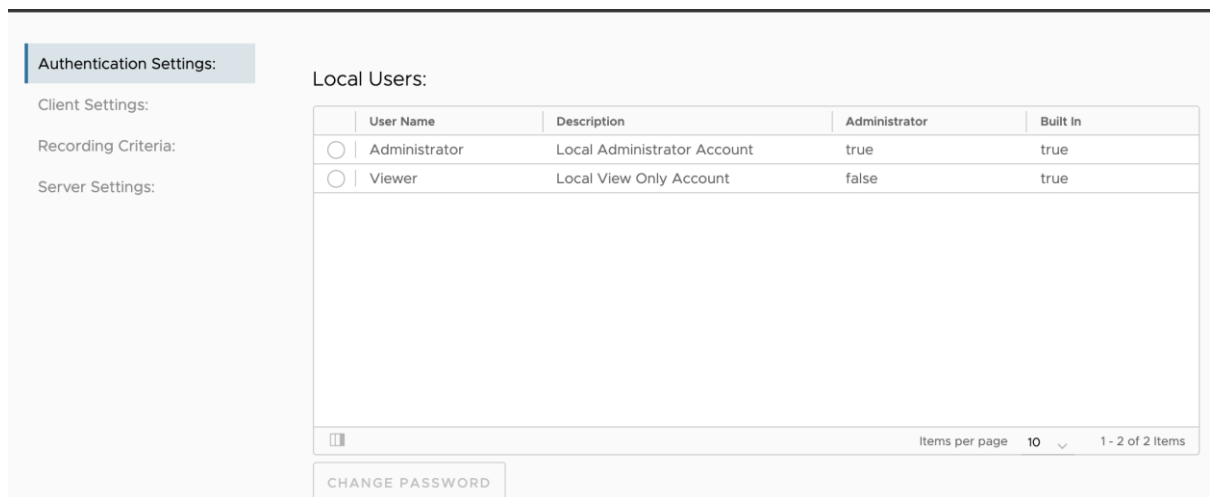**Username:** administrator
Password: Recording123

Username: viewer
Password: Recording456

These passwords can be changed from the Server settings dialog:

Modifying the Database connection:

To modify the database connection of the local server, you must use the Edit Deployment wizard. Please note this will only affect the local server and will initiate a local server restart.

Modifying the Recording Storage Folder:

To Modify the local storage folder for all servers in the cluster, you must use the edit deployment wizard.

<mark>Warning:</mark> This should only EVER be attempted when all, but one server is offline and there are NO ACTIVE recording sessions.

Modifying the SSL certificate:

The SSL certificate used by the Horizon Session Recording Server is retrieved from the local machines certificate store.

The Service will look for a certificate that has a friendly name of HorizonSessionRecordingServer and also has a private key. If the SSL certificate does not exist, it will be created on service start up.

To install a custom certificate:

<mark>Warning: the SSL certificate MUST be trusted by the agent if you change it after deployment, otherwise the agent's trusted thumbprints you may have added during the installation will not contain the new certificates thumbprint.</mark>

1. rename the existing one (in case you make a mistake).
2. Install the desired certificate to the machine store (Ensure the private key is marked as exportable)
3. Set the friendly name to "HorizonSessionRecordingServer"
4. Restart the Horizon Session Recording Server service.

Server Settings:

The service settings can be accessed via the web portal, when authenticated as the administrator account.
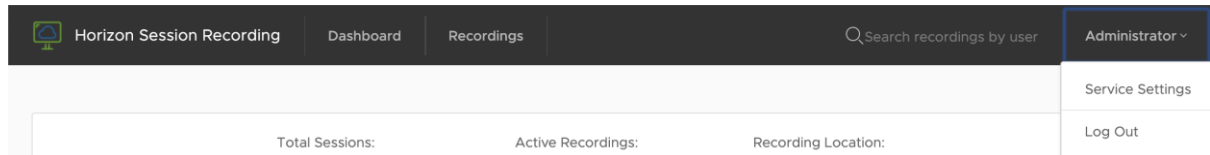
The following settings are configurable on the Web Service portal and broken down into 4 categories:

- Authentication
- Client Settings
- Recording Criteria
- Server Settings

Please note, all agent related settings require an agent service restart to update. (e.g. a reboot).

Authentication:

From this page, you can modify the default credentials associated with the deployment.

Client Settings:

These settings affect the behaviour of the Session Recording Agent.

Enable Notification	Instructs the agent whether to notify the users that the session is being recorded, or not.
Notification Message	The message that will be displayed to the users if notifications are enabled.
Chunk size:	The size of the buffer on the agent for each recording screen. Once this buffer fills, the data will be submitted to the server.
Upload Interval:	The length of time an active recording chunk (see above) will be uploaded to the server if the buffer has not yet filled. E.g. if the interval is set to 5 minutes, even if the buffer has yet to fill, once this timespan has elapsed, the current recording screen data is uploaded to the server and a new chunk is created. This is to ensure there is a smaller window for missed recordings if the recording agent encounters a failure.

Recording Criteria:

These settings dictate the types of sessions that will be recorded:

Record Local Sessions	This instructs the agent to record all sessions that are brokered from internal connection servers on the LAN.
Record Remote Sessions.	This instructs the agent to record all sessions that are brokered via a Unified Access Gateway or Security server.
Groups to record:	If you wish to specify that only certain groups of users should be recorded (with the above criteria) you can specify groups here. If this is left blank / empty, all users will be evaluated for recording.

Server Settings:

The server settings page is split into two categories, local server settings (only affecting the server you are currently connected to) along with cluster settings (affecting all servers connected to the database.

Note: When changing local server settings, you must run the edit deployment wizard. When changing the cluster storage location, you must run the edit deployment wizard.

Database Type:	This setting is only relevant to the local server and is read from the servicesettings.json file on the local server. There are 3 types of database: • SQLite (local file) • MSSQL (Microsoft SQL) • PostgreSQL This can only be edited during a Deployment Modification.
Database Connection Type:	To reduce the sheer amount of fencing and gating I have to do around SQL setup, you will need to add the connection by Database Connection String. If you are unsure what to do with this, ask your SQL DBA.

	This can only be edited during a Deployment Modification.
Recording Storage folder:	This is the folder (NTFS Share or Local, depending on deployment) the server will storage active recording data or finished recordings. This folder must be secured manually (see above). This can only be edited during a Deployment Modification.
Conversion Chunk Size:	This is the size of the buffer used to convert active recordings to finished recordings. Don't change this unless instructed to do so.
Conversion Wait Time:	This is the amount of time a pending recording conversion (from active to finished) will wait before attempting to convert the files. Don't change this unless instructed to do so.
Delete Raw Files:	This setting affects both the agent and the server. This instructs the agent whether to delete the recording when it's uploaded the data to the server successfully or not. This instructs the server to retain the raw files post conversion to mp4. Disabling this value greatly compromises security of the files and should only be disabled if instructed to do so.
Retention Settings:	This value instructs the web service to delete recordings only than this value in days. Note: locked recordings are excluded from the retention rules and remain infinitely.

Further Settings:

Addition settings, such as the JWT token timeout, along with the SSL / TLS port are configurable in the servicesettings.json file in the installation directory.

Note, the service should be stopped before modifying this file and these settings will only affect the server you modify. These settings MUST match on all recording servers participating in the cluster.

```
ServiceSettings.JSON - Notepad

File  Edit  Format  View  Help
{
    "HTTPSPort": 9443,
    "DBConnectionString": "GlmrgliM0TpWt5nbClRQyjoG]
    "JWTTokenLifeSpan": 60,
    "DBType": 1,
    "ConnectionStringIsEncrypted": true
}
```

Editing the Deployment Model:

Modifying the deployment model of the Session Recording Server will affect both the local server settings (e.g. the database connection) along with the Clusters storage location. Once the wizard completes, the local web service will be restarted to update the database, any changes to the storage location will affect all servers in the cluster immediately.

Warning: data is never migrated when modifying the deployment, Only the service configuration and user authentication details are migrated. Ensure to backup all recordings in the database by downloading the recordings, then delete the sessions in advance to avoid having unwanted data lying around.

Migrating the database connection:

When modifying the local server's database connection, it is largely safe enough to do this assuming there are no active sessions in progress (in a standalone mode).

Choose the database type and connection string you require. These settings will be stored in the servicesettings.json file, in the server installation directory and will be encrypted.

If the user specified in the connection string has the permissions to create the database, it will do so. Failing that, pre-create an empty database with the desired name and ensure the account provided has the ability to create the tables required.

This account should have the ability to make database layout changes for future upgrades where the database schema may change, and the web service will automatically perform the migrations.

The webservice will be restarted once the database has been instantiated and the local users / settings have been migrated.

Migrating the storage location:

When modifying the clusters storage location, this will immediately affect all servers in the cluster and should NEVER be performed while there are active recordings in flight. As above, NO data is migrated during this change.

Recording Agent:

The Session Recording Agent is an x64, .Net Framework 4.6.1 Service that communicates with the blast process, starting and stopping recordings as required.

Prerequisites:

The Agent requires:

- Horizon 7.9 or newer.
- An x64 operating system.
- Microsoft .Net Framework 4.6.1 or newer.
- Windows 7 Service Pack 1 or Windows 10.

Installing the Session Recording Agent:

1. Download the HorizonSessionRecordingAgent.exe file from the flings site and copy it to a local folder on the server.
2. Launch the installer and complete the steps.
3. When prompted for the connection, be careful to include the correct URL.
4. If the SSL certificate is not trusted, ensure to add the thumbprint carefully.
5. Once complete, the server should now be available on https://<localservername>:9443

Silently Installing the Session Recording Agent:

horizonsessionrecordingagent.exe /qb
SERVERADDRESSPROP=https://yoururl.domain.local:9443

Upgrading the Session Recording Agent:

The upgrade process is identical to the installation, ensure to also add the connection and thumbprint as the installation may overwrite the existing values.

## Uninstalling the Session Recording Agent:

Use Add remove programs (appwiz.cpl) to remove the session recording server binaries. By default, the following components must be cleaned up manually:

1. the logs (C:\programdata\VMware\Horizon Session Recording)
2. any pending recordings (if local, installation directory\recordings)

Modifying the connection manually:

The registry settings used by the agent are stored in:

HKLM\software\VMware, Inc.\VMware Blast\SessionRecordingAgent
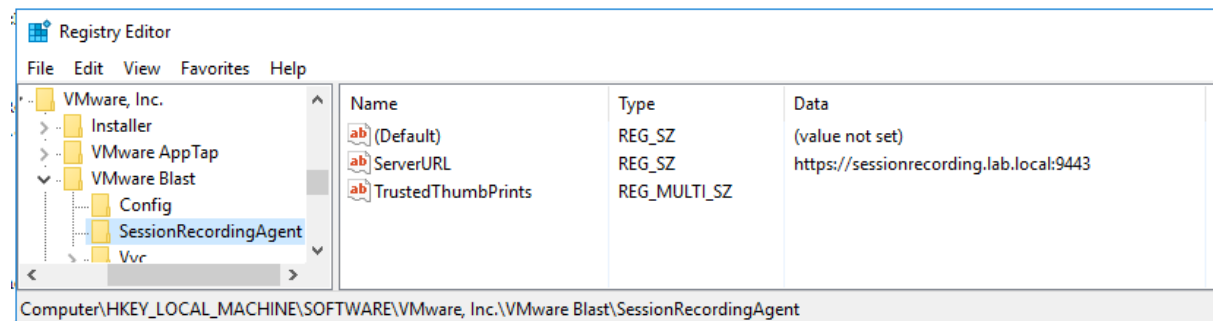
Troubleshooting:

Logging Location:

By Default, both the Agent and Server will log to:

 c:\programdata\VMware\ Horizon Session Recording

Changing the logging level:

In order to troubleshoot any undesired behaviour, logging can be extended on the agent and server by modifying the NLog.config file with a text editor.

Note: Changing the minlevel to "Debug" or "Trace" will dramatically increase the size of the log files.

Nlog.config:
…
<rules>
  <logger name="*" minlevel="Trace" writeTo="WebServiceLogging" />
 </rules>
…