

# RDSH SESSION LOAD-BALANCING FOR DESKTOPS AND APPLICATIONS IN VMWARE HORIZON 6 AND VMWARE HORIZON 7

VMware Horizon 6 version 6.2 and Later

## Table of Contents

Introduction .....	3
Load-Balancing RDSH Sessions .....	4
Perfmon-Based New-Session Load Balancing .....	4
Load Value Scripts .....	4
CPU Load Script .....	5
Memory Load Script .....	5
Setting Up Load Balancing for an RDSH Farm .....	5
Configure the VMware Horizon View Script Host Service .....	5
Configure the Load Value Script .....	5
Monitoring and Troubleshooting RDSH Load Balancing .....	5
Monitoring the Load-Balancing Status .....	5
Troubleshooting Incorrect Configurations .....	7
Application Anti-Affinity-Based New-Session Placement .....	9
Defining Anti-Affinity Rules .....	9
How the Anti-Affinity Rule Works .....	9
Anti-Affinity Patterns .....	9
New-Session Placement Process for Remote Applications .....	10
Session Placement Examples .....	10
Farm Without Active Sessions .....	10
Farm with Active User Sessions .....	11
Advanced Configurations .....	12
Restrictions and Considerations of RDSH Load Balancing .....	13
Additional Resources .....	14
About the Author and Contributors .....	14

## Introduction

VMware Horizon® 6 and VMware Horizon 7 include View, the VMware virtual desktop infrastructure (VDI) software that delivers a Windows desktop experience remotely to a user's endpoint device from centralized enterprise servers. In comparison to physical desktops, delivering virtual desktops from centralized VMware vSphere® servers enhances security and improves IT responsiveness.

VMware Horizon 6 version 6.2 includes key enhancements to Remote Desktop Session Host (RDSH) features and scope that address the scalability and availability of applications across data centers in different locations. This white paper provides best practices for using these new features.

- **Availability** – You can configure Horizon Agent on RDS hosts and application pools to ensure optimum availability of applications.
- **Scalability** – You can load-balance RDSH application sessions to improve traffic flow.

Horizon 6 and Horizon 7 also offer the flexibility of automating RDSH farm deployment by enabling image management capabilities. For more information on deploying RDSH farms, see [Publishing Applications with VMware Horizon 7](#). This guide also covers how to deploy automated RDSH farms using instant clones, including processes for changing scale and performing maintenance.

This white paper is for administrators who want to provide end users with RDS-hosted Windows applications on a large scale with optimum availability.

## Load-Balancing RDSH Sessions

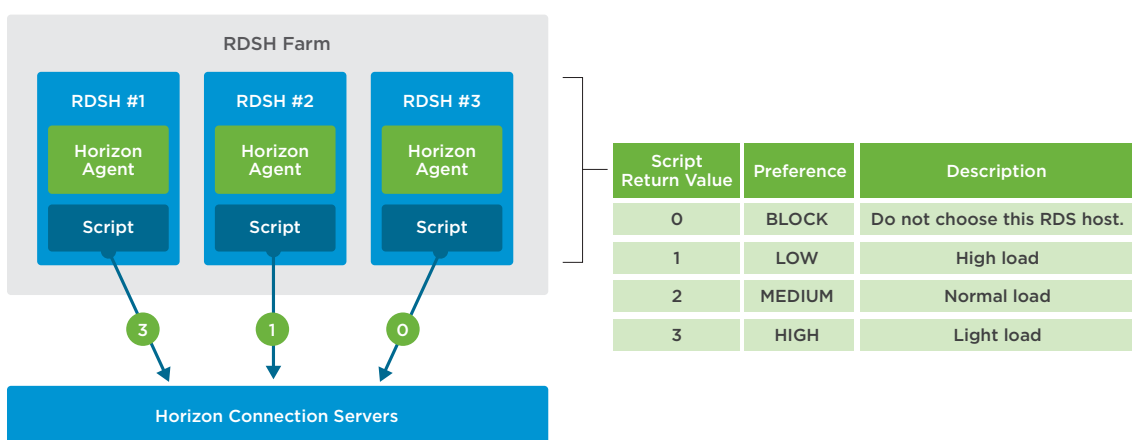
Until the release of version 6.2, VMware Horizon 6 had only one method for establishing a session, which used the current session count and a specified limit value to determine the RDS host on which to place a new session. The server with the fewest sessions was identified as the most appropriate server to host a new session.

With the introduction of the new load-balancing feature in Horizon 6 version 6.2, one or both of the following approaches can be used to establish a new session:

- [Perfmon-based](#) load balancing
- Application anti-affinity-based placement

## Perfmon-Based New-Session Load Balancing

The Perfmon-based method uses the Microsoft Windows Perfmon tool to collect resource utilization matrices from RDSH servers in a farm and then it reports a preference value to the load-balancing module of the Horizon Connection Server. The Horizon Connection Server takes appropriate action when it receives new session requests.



**Figure 1:** Perfmon-Based New-Session Load Balancing

The load value can be measured against CPU utilization, memory utilization, or other matrices that can be collected through custom-defined scripts. Be sure to collect load values from all the RDSH virtual machines (VMs) in a farm and that all load values are of the same type. For example, within an RDSH farm, do not mix CPU load values with memory load values nor exclude the values of any RDSH VMs.

### Load Value Scripts

The Horizon Agent installer provides two sample scripts that you can use to collect CPU and memory load values. When you install or upgrade the Horizon Agent, the scripts are placed in this directory:

`C:\Program Files\VMware\VMware View\Agent\scripts\`

As an administrator, you must configure all RDSH VMs in a farm to either use one of the provided scripts from this directory or a custom-defined script from a different path.

The load value scripts calculate the machine load in terms of CPU or memory utilization. The scripts report a load value as an exit code number. Based on the load value, a preference value is mapped (LBPREFERENCE), which decides whether to allow or block a new session.

**CPU Load Script**

You can collect the LBPREFERENCE value based on CPU load with the `cpuutilisation.vbs` script in the `C:\Program Files\VMware\VMware View\Agent\scripts` directory. The script returns the average percentage of CPU utilization for the last 5 minutes:

Load Value → Average (% CPU Utilization for Last 5 Minutes)

The Perfmon tool does not have a dedicated counter that averages the percentage of CPU utilization over a period of time. Therefore, the script reads a registry value that the Horizon Agent updates. The script takes a sample of instantaneous values and averages them over a period of time. A sample of CPU numbers is not leveraged here, because the numbers fluctuate dramatically over time.

**Memory Load Script**

You can collect the LBPREFERENCE value based on memory load with the `memoryutilisation.vbs` script in the `C:\Program Files\VMware\VMware View\Agent\scripts` directory. The script reads the available bytes and total bytes and computes the percentage of memory being used, which is then mapped to an LBPREFERENCE value:

Load Value → % Committed Bytes in Use

**Setting Up Load Balancing for an RDSH Farm**

To set up load balancing for a farm, you must configure each Horizon Agent in the farm to use one of the load value scripts. The scripts use the VMware Horizon View Script Host service, so in each RDSH VM you must configure the script service to auto-start and set a Windows registry key to load the desired script, as follows.

**Configure the VMware Horizon View Script Host Service**

You must configure the VMware Horizon View Script Host service to start automatically each time the RDSH VM starts. You can use either the Windows Service Manager on the machine or a GPO setting.

- To use the Windows Service Manager, see *Enable the VMware Horizon View Script Host Service on an RDS Host* in the [View Administration](#) guide.
- To use a GPO setting, see the Microsoft TechNet article [Configure a Service Item](#).

These options give you additional control when some farms need to use load balancing, and other farms need to use the older session count method. You can disable the feature per farm.

**Configure the Load Value Script**

You can configure load balancing on RDS hosts by leveraging either CPU utilization or memory utilization. This configuration is done at the RDSH VM's registry level. For more information, see *Configure a Load Balancing Script on an RDS Host* in the [View Administration](#) guide.

**Monitoring and Troubleshooting RDSH Load Balancing**

The Horizon Administrator dashboard provides information about the load-balancing status and some configuration errors.

**Monitoring the Load-Balancing Status**

After the new configuration takes effect, it can take 5-10 minutes to calculate the load values and display the correct status information. After the script configuration is reflected in the status results, the values are reported every 1 to 2 minutes with asynchronous session notifications.

Both the Horizon Administrator dashboard and the debug logs provide information about load balancing. When load balancing is properly configured and an RDSH VM has little load, all new sessions are sent to it. At this stage, the VM publishes entries such as the following in the debug log file.

```
DEBUG (0434-0510) <MessageFrameWorkDispatch> [ws_scripts] ScriptEventName 'RdshLoad',
script 'LoadScript', starting cmdline 'cscript.exe "C:\Program Files\VMware\VMware View\
Agent\scripts\cpuutilisation.vbs"'
```

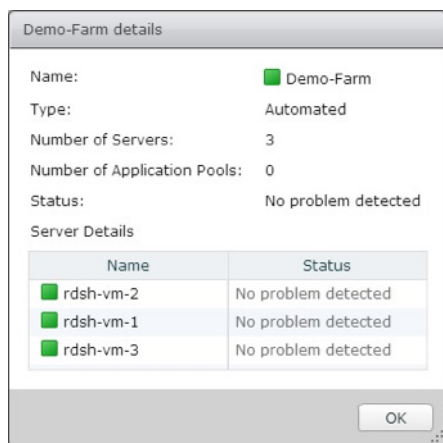
```
DEBUG (0434-0F2C) <ScriptEventTimeoutThread> [ws_scripts] ScriptEventName 'RdshLoad',
script 'LoadScript', exitCode=3
```

```
DEBUG (02B4-0154) <MessageFrameWorkShare> [wsnm_desktop] Using return code value of 3
```

```
DEBUG (02B4-0154) <MessageFrameWorkShare> [wsnm_desktop] New LBPREFERENCE value: HIGH
```

Based on the exit code, the LBPREFERENCE value is updated at certain intervals. In the preceding log sample, the script for using the CPU load value has been executed, and exit code **3** has been returned. The LBPREFERENCE has now been updated to **HIGH**, which means that the RDSH VM can accept new connections.

A farm dashboard provides the basic health status of the farm and each RDSH VM's status, as shown in Figure 2.



**Figure 2:** Farm Dashboard with Load Balancing

Click an item to expand the farm dashboard's inventory and check an RDSH VM's status and server load value.

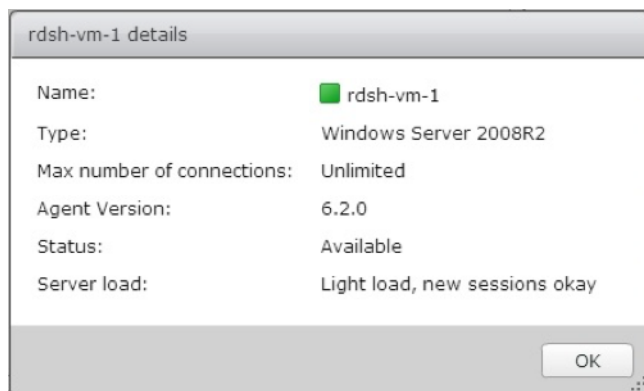
As in the previous log example, which updates the LBPREFERENCE value to **HIGH** for exit code **3**, this value would be reflected in the dashboard.

In Figure 3, the health dashboard reports the server load status for one of the RDSH VMs from the Demo-Farm farm. The load value status displayed in the dashboard is based on the corresponding exit code referenced in Table 1.

EXIT CODE VALUE	LBPREFERENCE VALUE	HEALTH DASHBOARD LOAD STATUS
3	HIGH	Light load, new sessions okay
2	MED	Normal load, new sessions okay
1	LOW	Heavy load, new sessions avoided
0	BLOCK	New sessions blocked

**Table 1:** Exit Codes and Corresponding Load-Balancing Status

**Note:** The default load value for an RDS host is **MED**.



**Figure 3:** RDSH Health Status Reporting Minimum Load Value

#### Troubleshooting Incorrect Configurations

If a configuration is not correct, load balancing uses the RDS host with the lowest utilization based on session counts. The dashboard provides troubleshooting information for these configuration problems.

- **No load value reported** - If the dashboard health status is **No load value reported**, either the script is not referenced correctly in the Windows Registry key, or the script host service is not running. To diagnose, check the debug logs for the LBPREFERENCE value. It might take a while to reflect the status in the dashboard. If you cannot find LBPREFERENCE log entries, check that the VMware Horizon View Script Host service is running.

Another cause could be errors in the registry setting. If multiple string values are defined under the specified key path, the script is not hooked properly. Make sure that only one string value is defined and that you have entered the correct path to the load value script in the Value data text box.

- **Mixed load values** - When the load balancing configuration is not identical across the RDSH VMs in a farm, the dashboard shows the following status warning: **One or more server(s) is either in warning or error state, or servers have a mixed set of reported and unreported load values.**

This message indicates that one or more RDS hosts do not have the registry key configured to point to the script, so it is not reporting the load value. Check the load value status for each RDS host. Also verify that the **RdshLoad** key and Value data text box in the registry are set correctly and that these settings are identical for all hosts in the farm.



## Application Anti-Affinity-Based New-Session Placement

Certain CPU- or memory-intensive applications do not consume full resources while running without content or data loaded. For example, AutoCAD does not consume much CPU or memory before you load a model. In such conditions, load-balancing based on CPU or memory utilization might not be effective. Moreover, if RDSH servers are running inside VMs, there is no way to control the number of instances of such resource-intensive applications running per vSphere host. You can control the number of RDSH VMs per vSphere host only by taking appropriate administrative measures during deployment.

To control the number of instances of these applications, you can use a control mechanism through anti-affinity patterns. The mechanism counts the number of instances of an application per RDS host and lets you set a threshold for the number of new connections within the farm. This way, you can pre-allocate the capacity for specific applications. However, with this method, you cannot strictly control the number of sessions. Instead, the setting is treated as a hint to the system.

### Defining Anti-Affinity Rules

You can limit the number of application instances when you create or edit an application pool. To limit the number of application instances running on a server, you define an anti-affinity rule for the application pool. This policy consists of:

- One or more anti-affinity patterns, where each match represents one instance of the process
- A maximum count

For example, to prevent creating more than two instances of AutoCAD per RDS host, define the following rule at the application pool level:

- **Anti-Affinity Pattern** - autocad.exe
- **Anti-Affinity Count** - 2

### How the Anti-Affinity Rule Works

When a new session request is initiated for which an anti-affinity rule is configured, the Horizon Connection Server passes the anti-affinity rules to the Horizon Agent on the RDSH VM. The agent uses the patterns to count the current number of instances of the application. If the count exceeds the count specified in the rule, the StartSession or StartApp request fails. The Horizon Connection Server then tries the next possible RDS host in the farm.

### Anti-Affinity Patterns

The pattern string can consist of one or more wildcard characters. An asterisk (\*) means zero or more characters. For example, **notepad\*** matches **notepad.exe**, **notepad.bat**, or just **notepad**. A question mark (?) represents one character. For example, **notepad.?xe** matches **notepad.exe**. To use multiple patterns, separate the patterns with commas. If multiple patterns match an application within a single RDSH session, that host is counted only as a single match. For more information, see *Configure an Anti-Affinity Rule for an Application Pool* in the [View Administration](#) guide.

## New-Session Placement Process for Remote Applications

You can apply the Perfmon-based and the anti-affinity-based approaches independently to RDSH farms and application pools. If you apply both approaches to a farm or pool, a new-session placement algorithm is created by combining both approaches.

**Note:** These configurations do not affect RDSH desktop sessions in which the traditional approach based on session capacity is applied.

When the remote application is launched, the system performs the following steps:

1. Partitions the farm based on existing sessions for the user, producing two lists:
  - **Session** - RDS hosts that already have a session for the user
  - **Spare** - RDS hosts that do not have a session for the user
2. Filters each list based on the load values (if available) and removes any RDS host that has a negative load value.
3. Orders each list. If all servers in the farm provide a load value, the system uses this load value. Otherwise, the system uses the current session counts to order each list.
4. Concatenates the lists, with the session list first (if any servers remain).
5. Works down the concatenated list, calling the StartSession or StartApp request until one works. The StartSession or StartApp request might fail if anti-affinity rules prevent the RDS host from hosting the new app instance.

### Session Placement Examples

The examples show a farm without active sessions and a farm with active sessions.

#### Farm Without Active Sessions

The farm consists of six VMs named RDSH1 to RDSH6. All the VMs are available and report different preference values to the broker (Horizon Connection Server) based on running a script.



RDSH5 is discarded because its state is **BLOCK**. All remaining hosts are sorted into three buckets (logical groups) based on preference value. See Table 1 for descriptions of these values. The three buckets are grouped as follows:

- Bucket 1 contains RDSH1 and RDSH3, which report preference values of **HIGH**.
- Bucket 2 contains RDSH4, which reports a preference value of **MED**.
- Bucket 3 contains RDSH2 and RDSH6, which report preference values of **LOW**.



Because Bucket 2 has only one host, it gets combined with Bucket 1. Then the bucket order is randomized, and a new list is created. The new list can be illustrated as follows:



With the newly ordered list, RDSH4 is tried first, followed by RDSH3, and so on if anti-affinity rules prevent the session from starting on the first host.

#### Farm with Active User Sessions

This farm consists of six VMs named RDSH1 to RDSH6. All hosts report preference values, and RDSH3 has an active session.



Because RDSH6 is reporting a preference value of **BLOCK** it is not considered for new session placement. All hosts without an active session for the user are then sorted into buckets based on the reporting preference value. Any bucket with a single host gets merged with another bucket that has a higher preference.



After randomizing hosts in each bucket, a new host order is created. The host that has the user session (RDSH3) is placed at the top of the list.



When a user re-establishes a remote application session for

- **The same application** – RDS hosts containing sessions in which the user has previously run an application are reused. This behavior overrides load balancing and anti-affinity rules.
- **New applications** – The remote application is launched on an RDS host where the user already has an existing session, even if the host reports a load preference of **BLOCK**. However, anti-affinity rules can prevent this behavior.

### Advanced Configurations

You can adjust some advanced settings based on special requirements only in consultation with VMware technical support.

- **Sampling period** - The default sampling period for the load value script is 5 minutes. You can change it to a minimum of 2 minutes. However, changing this setting can have a negative performance impact on the RDS host. The following registry entry controls the setting, which is measured in seconds. The default is 300 seconds.

HKEY\_LOCAL\_MACHINE\SOFTWARE\VMware, Inc.\VMware VDM\Performance Stats\  
SamplingIntervalSeconds

- **Script run interval** - You can control the interval at which the **RdshLoad** script is executed by creating a new registry key, **RdshLoadScriptIntervalSecs**, and defining the number of seconds between each invocation of the script. If this value is not present, a default of 30 seconds is used. For example, create the following key and set the interval to 15 seconds:

[HKEY\_LOCAL\_MACHINE\SOFTWARE\VMware, Inc.\VMware VDM\Agent\Configuration]

"RdshLoadScriptIntervalSecs"=dword:0000000f

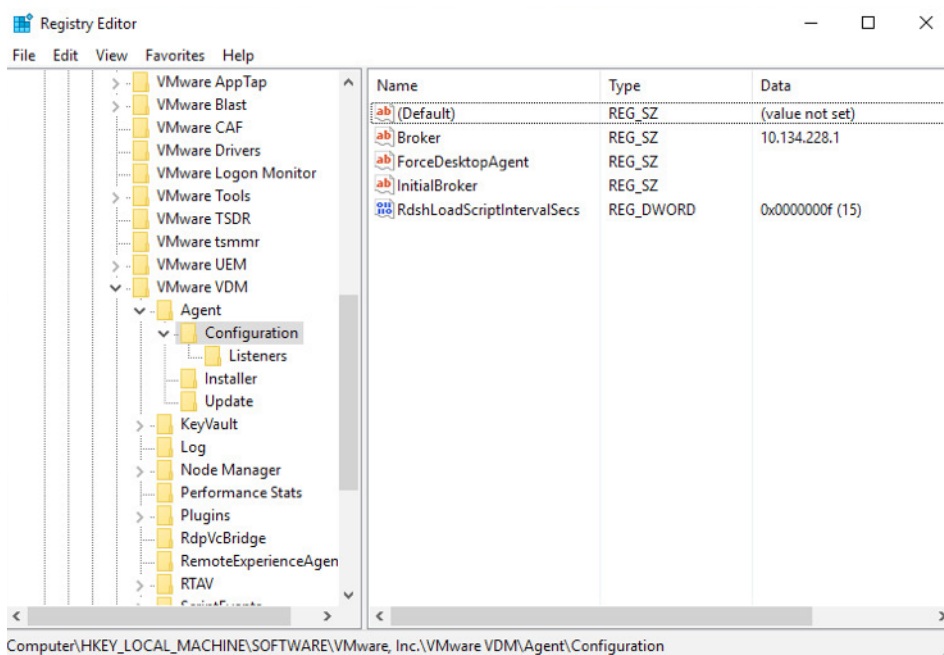


Figure 4: Creating the RdshLoadScriptIntervalSecs Key

## Restrictions and Considerations of RDSH Load Balancing

Note the following restrictions and considerations when using RDSH load balancing:

- Rules for limiting the number of application instances based on an anti-affinity count act as hints, because there is no guarantee that instances of applications on an RDS host will be restricted to the configured number. For more information, see *Anti-Affinity Feature Constraints* in the [View Administration](#) guide.
- Anti-affinity support depends on farm-level, multisession support in Horizon Clients. You must use Horizon Client 3.5 or later. Do not use mobile clients.
- Multiple sessions for RDSH desktops are not supported. RDSH desktop users are always reconnected to an existing session.
- The pre-packaged CPU utilization script uses rolling average data that is sampled every 5 minutes. Because load-value preferences are updated only once in 5 minutes, short-term high-utilization events on the RDS host might not be reflected in the returned load preferences.

## Additional Resources

Horizon 6 and Horizon 7 product documentation:

- [View Administration](#)
- [Setting Up Desktop and Application Pools in View](#)
- [Publishing Applications with VMware Horizon 7](#)
- [Overview of Remote Desktop Licensing](#)

## About the Author and Contributors

Sivaprasad Govindankutty, Staff Engineer, End-User Computing, VMware, wrote this paper.

The following people contributed to the content of this paper:

- Najaf Khan, Staff Engineer, VMware
- Graeme Gordon, Senior End-User-Computing Architect, VMware
- Jubish K. Jose, Senior Member of the Technical Staff, End-User Computing, VMware
- Alex Birch, VMware alumnus
- Satish Kumar, Senior Member of the Technical Staff, VMware

To comment on this paper, contact the VMware End-User-Computing Technical Marketing team at [euc\\_tech\\_content\\_feedback@vmware.com](mailto:euc_tech_content_feedback@vmware.com).



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 [www.vmware.com](http://www.vmware.com)

Copyright © 2017 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: VMW-TWP-RDSHLOADBALDKTPAPPS-USLTR-20170606-WEB