

**NAME**

**vlmcsd.ini** – vlmcsd KMS emulator configuration file

**SYNOPSIS**

**vlmcsd.ini**

**DESCRIPTION**

**vlmcsd.ini** (or simply called the "ini file") is a configuration file for **vlmcsd**(8). By default vlmcsd does not use a configuration file. It is completely optional and for advanced users only. You must use the **-i** option on the vlmcsd command line to use an ini file. There is no default name or default location for the ini file.

Everything, that can be configured in the ini file, may also be specified on the command line. Any configuration option specified on the command line takes precedence over the respective configuration line in the ini file.

**Benefits of a configuration file**

While you can use the configuration file to simply modify the default behavior of vlmcsd, it can also be used to change the configuration of vlmcsd after you sent a HUP **signal**(7). Whenever you send SIGHUP, the configuration file will be re-read. Any changes you made to the ini file will be reflected after vlmcsd received the hangup signal.

**Differences between command line and configuration file**

If you specify an illegal option or option argument on the command line, vlmcsd displays help and exits. If you specify an incorrect *keyword* or *argument* in the ini file, vlmcsd displays a warning with some information, ignores the respective line and continues. This is intentional and prevents vlmcsd from aborting after a SIGHUP if the configuration was modified incorrectly.

**SYNTAX**

vlmcsd.ini is a UTF-8 encoded text file with each line being in the format *keyword* = *argument*. The *keyword* is not case-sensitive. The *argument* is treated literally. It is neither required nor allowed to enclose the *argument* in any form of quote characters except when quote characters are part of the argument itself. Whitespace characters are ignored only

- at the beginning of a line
- between the *keyword* and '='
- between '=' and the *argument*

Lines, that start with '#' or ';' are treated as comments. Empty lines are ignored as well. If a *keyword* is repeated in another line, vlmcsd will use the *argument* of the last occurrence of the *keyword*. An exception to this is the Listen *keyword* which can be specified multiple times and causes vlmcsd to listen on more than one IP address and/or port.

Some *arguments* are binary arguments that need to be either TRUE or FALSE. You can use "Yes", "On" or "1" as an alias for TRUE and "No", "Off" or "0" as an alias for FALSE. Binary arguments are case-insensitive.

**KEYWORDS**

The following *keywords* are defined (not all keywords may be available depending on the operating system and the options used when **vlmcsd**(8) was compiled):

**Listen** This defines on what combinations of IP addresses and ports vlmcsd should listen. **Listen** can be specified more than once. The *argument* has the form *ipaddress*[:*port*]. If you omit the *port*, the default port of 1688 is used. If the *ipaddress* contains colons and a *port* is used, you must enclose the *ipaddress* in brackets. The default is to listen to 0.0.0.0:1688 and [::]:1688 which means listen

to all IPv4 and all IPv6 addresses. See the **-L** option in **vlmcsd(8)** for more info about the syntax. If you use **-L** or **-P** on the command line, all **Listen** keywords in the ini file will be ignored. The **Listen** keyword cannot be used if **vlmcsd** has been compiled to use Microsoft RPC (Windows and Cygwin only) or simple sockets.

Examples:

```
Listen = 192.168.1.123:1688
```

```
Listen = 0.0.0.0:1234
```

```
Listen = [fe80::1721:12ff:fe81:d36b%eth0]:1688
```

**Port** Can only be used if **vlmcsd** has been compiled to use simple sockets or on Windows and Cygwin if **vlmcsd(8)** has been compiled to use Microsoft RPC. Otherwise you must use **Listen** instead. Causes **vlmcsd** to listen on that port instead of 1688.

#### **FreeBind**

Can be TRUE or FALSE. If TRUE, you can use the **Listen** keyword with IP addresses that are currently not defined on your system. **vlmcsd(8)** will start listening on these IP addresses as soon as they become available. This keyword is only available under Linux and FreeBSD because no other OS currently supports that feature. FreeBSD supports this only for IPv4 and requires the `PRIV_NETINET_BINDANY` privilege which is normally assigned to processes of the root user.

#### **PublicIPProtectionLevel**

Set the level of protection against KMS activations from public IP addresses.

0 = No protection (default)

1 = Listen on private IP addresses only (plus those specified by one or more **Listen** statements)

2 = Disconnect clients with public IP addresses without activating

3 = Combines 1 and 2

For details on public IP protection levels see **vlmcsd(8)** command line option **-o**.

#### **UseNDR64**

Can be TRUE or FALSE. Specifies whether you want to use the NDR64 transfer syntax. See options **-n0** and **-n1** in **vlmcsd(8)**. The default is TRUE.

#### **UseBTFN**

Can be TRUE or FALSE. Specifies whether you want to use bind time feature negotiation in RPC. See options **-b0** and **-b1** in **vlmcsd(8)**. The default is TRUE.

#### **RandomizationLevel**

The *argument* must 0, 1 or 2. This specifies the ePID randomization level. See options **-r0**, **-r1** and **-r2** in **vlmcsd(8)**. The default randomization level is 1. A **RandomizationLevel** of 2 is not recommended and should be treated as a debugging level.

**LCID** Use a specific culture id (LCID) even if the ePID is randomized. The *argument* must be a number between 1 and 32767. While any number in that range is valid, you should use an official LCID. A list of assigned LCIDs can be found at <http://msdn.microsoft.com/en-us/goglobal/bb964664.aspx>. On the command line you control this setting with option **-C**.

**MaxWorkers**

The *argument* specifies the maximum number of worker processes or threads that will be used to serve activation requests concurrently. This is the same as specifying **-m** on the command line. Minimum is 1. The maximum is platform specific and is at least 32767 but is likely to be greater on most systems. The default is no limit.

**ConnectionTimeout**

Used to control when the vlmcsd disconnects idle TPC connections. The default is 30 seconds. This is the same setting as **-t** on the command line.

**DisconnectClientsImmediately**

Set this to TRUE to disconnect a client after it got an activation response regardless whether a timeout has occurred or not. The default is FALSE. Setting this to TRUE is non-standard behavior. Use only if you are experiencing DoS or DDoS attacks. On the command line you control this behavior with options **-d** and **-k**.

**PidFile** Write a pid file. The *argument* is the full pathname of a pid file. The pid file contains a single line containing the process id of the vlmcsd process. It can be used to stop (SIGTERM) or restart (SIGHUP) vlmcsd. This directive can be overridden using **-p** on the command line.

**LogFile**

Write a log file. The *argument* is the full pathname of a log file. On a unixoid OS and with Cygwin you can use the special filename 'syslog' to log to the syslog facility. This is the same as specifying **-l** on the command line.

**LogDateAndTime**

Can be TRUE or FALSE. The default is TRUE. If set to FALSE, logging output does not include date and time. This is useful if you log to **stdout(3)** which is redirected to another logging mechanism that already includes date and time in its output, for instance **systemd-journald(8)**. If you log to **syslog(3)**, **LogDateAndTime** is ignored and date and time will never be included in the output sent to **syslog(3)**. Using the command line you control this setting with options **-T0** and **-T1**.

**LogVerbose**

Set this to either TRUE or FALSE. The default is FALSE. If set to TRUE, more details of each activation will be logged. You use **-v** and **-q** in the command line to control this setting. **LogVerbose** has an effect only if you specify a log file or redirect logging to **stdout(3)**.

**WhitelistingLevel**

Can be 0, 1, 2 or 3. The default is 0. Sets the whitelisting level to determine which products vlmcsd activates or refuses.

**0:** activate all products with an unknown, retail or beta/preview KMS ID.

**1:** activate products with a retail or beta/preview KMS ID but refuse to activate products with an unknown KMS ID.

**2:** activate products with an unknown KMS ID but refuse products with a retail or beta/preview KMS ID.

**3:** activate only products with a known volume license RTM KMS ID and refuse all others.

The SKU ID is not checked. Like a genuine KMS server vlmcsd activates a product that has a random or unknown SKU ID. If you select **1** or **3**, vlmcsd also checks the Application ID for correctness. If Microsoft introduces a new KMS ID for a new product, you cannot activate it if you used **1**

or **3** until a new version of `vlmcsd` is available.

### CheckClientTime

Can be TRUE or FALSE. The default is FALSE. If you set this to TRUE `vlmcsd(8)` checks if the client time differs no more than four hours from the system time. This is useful to prevent emulator detection. A client that tries to detect an emulator could simply send two subsequent request with two time stamps that differ more than four hours from each other. If both requests succeed, the server is an emulator. If you set this to TRUE on a system with no reliable time source, activations will fail. It is ok to set the correct system time after you started `vlmcsd(8)`.

### MaintainClients

Can be TRUE or FALSE (the default). Disables (FALSE) or enables (TRUE) maintaining a list of client machine IDs (CMIDs). TRUE is useful to prevent emulator detection. By maintaining a CMID list, `vlmcsd(8)` reports current active clients exactly like a genuine KMS emulator. This includes bug compatibility to the extent that you can permanently kill a genuine KMS emulator by sending an "overcharge request" with a required client count of 376 or more and then request activation for 671 clients. `vlmcsd(8)` can be reset from this condition by restarting it. If FALSE is used, `vlmcsd(8)` reports current active clients as good as possible. If no client sends an "overcharge request", it is not possible to detect `vlmcsd(8)` as an emulator with `MaintainClients = FALSE`. Maintaining clients requires the allocation of a buffer that is about 50 kB in size. On hardware with few memory resources use it only if you really need it.

If you start `vlmcsd(8)` from an internet superserver, this setting cannot be used. Since `vlmcsd(8)` exits after each activation, it cannot maintain any state in memory.

### StartEmpty

This setting is ignored if you do not also specify `MaintainClients = TRUE`. If you specify FALSE (the default), `vlmcsd(8)` starts up as a fully "charged" KMS server. Clients activate immediately. `StartEmpty = TRUE` lets you start up `vlmcsd(8)` with an empty CMID list. Activation will start when the required minimum clients (25 for Windows Client OSses, 5 for Windows Server OSses and Office) have registered with the KMS server. As long as the minimum client count has not been reached, clients end up in HRESULT 0xC004F038 "The count reported by your Key Management Service (KMS) is insufficient. Please contact your system administrator". You may use `vlmcs(1)` or another KMS client emulator to "charge" `vlmcsd(8)`. Setting this parameter to TRUE does not improve emulator detection prevention. It's primary purpose is to help developers of KMS clients to test "charging" a KMS server.

### ActivationInterval

This is the same as specifying `-A` on the command line. See `vlmcsd(8)` for details. The default is 2 hours. Example: `ActivationInterval = 1h`

### RenewalInterval

This is the same as specifying `-R` on the command line. See `vlmcsd(8)` for details. The default is 7 days. Example: `RenewalInterval = 3d`. Please note that the KMS client decides itself when to renew activation. Even though `vlmcsd` sends the renewal interval you specify, it is no more than some kind of recommendation to the client. Older KMS clients did follow the recommendation from a KMS server or emulator. Newer clients do not.

**User** Run `vlmcsd` as another, preferably less privileged, user. The *argument* can be a user name or a numeric user id. You must have the required privileges (capabilities on Linux) to change the security context of a process without providing any credentials (a password in most cases). On most unixoid OSses 'root' is the only user who has these privileges in the default configuration. This

setting is not available in the native Windows version of `vlmcsd`. See **-u** in `vlmcsd(8)`. This setting cannot be changed on the fly by sending `SIGHUP` to `vlmcsd`.

**Group** Run `vlmcsd` as another, preferably less privileged, group. The *argument* can be a group name or a numeric group id. You must have the required privileges (capabilities on Linux) to change the security context of a process without providing any credentials (a password in most cases). On most unixoid OSses 'root' is the only user who has these privileges in the default configuration. This setting is not available in the native Windows version of `vlmcsd`. See **-g** in `vlmcsd(8)`. This setting cannot be changed on the fly by sending `SIGHUP` to `vlmcsd`.

### Windows

The *argument* has the form `ePID [ / HwId ]`. Always use `ePID` and `HwId` for Windows activations. If specified, **RandomizationLevel** for Windows activations will be ignored.

### Office2010

The *argument* has the form `ePID [ / HwId ]`. Always use `ePID` and `HwId` for Office 2010 activations. If specified, **RandomizationLevel** for Office 2010 activations will be ignored.

### Office2013

The *argument* has the form `ePID [ / HwId ]`. Always use `ePID` and `HwId` for Office 2013 activations. If specified, **RandomizationLevel** for Office 2013 activations will be ignored.

### Office2016

The *argument* has the form `ePID [ / HwId ]`. Always use `ePID` and `HwId` for Office 2016 activations. If specified, **RandomizationLevel** for Office 2016 activations will be ignored.

## VALID EPIDS

The ePID is currently a comment only. You can specify any string up to 63 bytes. In Windows 7 Microsoft has blacklisted few ( < 10 ) ePIDs that were used in KMSv5 versions of the "Ratiborus Virtual Machine". Microsoft has given up on blacklisting when KMS emulators appeared in the wild.

Even if you can use "Activated by cool hacker guys" as an ePID, you may wish to use ePIDs that cannot be detected as non-MS ePIDs. If you don't know how these "valid" ePIDs look like exactly, do not use GUIDS in `vlmcsd.ini`. `vlmcsd` provides internal mechanisms to generate valid ePIDs.

If you use non-ASCII characters in your ePID (you shouldn't do anyway), these must be in UTF-8 format. This is especially important when you run `vlmcsd` on Windows or cygwin because UTF-8 is not the default encoding for most editors.

If you are specifying an optional HWID it follows the same syntax as in the **-H** option in `vlmcsd(8)` except that you must not enclose a HWID in quotes even if it contains spaces.

## FILES

`vlmcsd.ini(5)`

## AUTHOR

`vlmcsd(8)` was written by crony12, Hotbird64 and vityan666. With contributions from DougQaid.

## CREDITS

Thanks to CODYQX4, deagles, eIcn, mikmik38, nosferati87, qad, Ratiborus, ...

**SEE ALSO**

**vlmcsd(8), vlmcsd(7), vlmcs(1), vlmcsdmulti(1)**