NAME

ovs−appctl − utility for configuring running Open vSwitch daemons

SYNOPSIS

ovs−appctl [**−−target=***target* | **−t** *target*] [**−T** *secs* | **−−timeout=***secs*] *command* [*arg*...]
ovs−appctl −−help
ovs−appctl −−version

DESCRIPTION

Open vSwitch daemons accept certain commands at runtime to control their behavior and query their settings. Every daemon accepts a common set of commands documented under **COMMON COMMANDS** below. Some daemons support additional commands documented in their own manpages. **ovs−vswitchd** in particular accepts a number of additional commands documented in **ovs−vswitchd**(8).

The **ovs−appctl** program provides a simple way to invoke these commands. The command to be sent is specified on **ovs−appctl**'s command line as non-option arguments. **ovs−appctl** sends the command and prints the daemon's response on standard output.

In normal use only a single option is accepted:

−t *target*
−−target=*target*

Tells **ovs−appctl** which daemon to contact.

If *target* begins with / it must name a Unix domain socket on which an Open vSwitch daemon is listening for control channel connections. By default, each daemon listens on a Unix domain socket named **/usr/local/var/run/openvswitch/***program***.***pid***.ctl**, where *program* is the program's name and *pid* is its process ID. For example, if **ovs−vswitchd** has PID 123, it would listen on **/usr/local/var/run/openvswitch/ovs−vswitchd.123.ctl**.

Otherwise, **ovs−appctl** looks for a pidfile, that is, a file whose contents are the process ID of a running process as a decimal number, named **/usr/local/var/run/openvswitch/***target***.pid**. (The **−−pidfile** option makes an Open vSwitch daemon create a pidfile.) **ovs−appctl** reads the pidfile, then looks for a Unix socket named **/usr/local/var/run/openvswitch/***target***.***pid***.ctl**, where *pid* is replaced by the process ID read from the pidfile, and uses that file as if it had been specified directly as the target.

On Windows, *target* can be an absolute path to a file that contains a localhost TCP port on which an Open vSwitch daemon is listening for control channel connections. By default, each daemon writes the TCP port on which it is listening for control connection into the file *program***.ctl** located inside the configured *OVS_RUNDIR* directory. If *target* is not an absolute path, **ovs−appctl** looks for a file named *target***.ctl** in the configured *OVS_RUNDIR* directory.

The default target is **ovs−vswitchd**.

−T *secs*
−−timeout=*secs*

By default, or with a *secs* of **0**, **ovs−appctl** waits forever to connect to the daemon and receive a response. This option limits runtime to approximately *secs* seconds. If the timeout expires, **ovs−appctl** exits with a **SIGALRM** signal.

COMMON COMMANDS

Every Open vSwitch daemon supports a common set of commands, which are documented in this section.

GENERAL COMMANDS

These commands display daemon-specific commands and the running version. Note that these commands are different from the **−−help** and **−−version** options that return information about the **ovs−appctl** utility itself.

list-commands

Lists the commands supported by the target.

version
Displays the version and compilation date of the target.

LOGGING COMMANDS

Open vSwitch has several log levels. The highest-severity log level is:

off No message is ever logged at this level, so setting a logging destination's log level to **off** disables logging to that destination.

The following log levels, in order of descending severity, are available:

emer A major failure forced a process to abort.

err A high-level operation or a subsystem failed. Attention is warranted.

warn A low-level operation failed, but higher-level subsystems may be able to recover.

info Information that may be useful in retrospect when investigating a problem.

dbg Information useful only to someone with intricate knowledge of the system, or that would commonly cause too-voluminous log output. Log messages at this level are not logged by default.

Every Open vSwitch daemon supports the following commands for examining and adjusting log levels.

vlog/list
Lists the known logging modules and their current levels.

vlog/list-pattern
Lists logging pattern used for each destination.

vlog/set [*spec*]
Sets logging levels. Without any *spec*, sets the log level for every module and destination to **dbg**. Otherwise, *spec* is a list of words separated by spaces or commas or colons, up to one from each category below:

- A valid module name, as displayed by the **vlog/list** command on **ovs−appctl**(8), limits the log level change to the specified module.
- **syslog**, **console**, or **file**, to limit the log level change to only to the system log, to the console, or to a file, respectively.

  On Windows platform, **syslog** is accepted as a word and is only useful if the *target* was started with the **−−syslog−target** option (the word has no effect otherwise).
- **off**, **emer**, **err**, **warn**, **info**, or **dbg**, to control the log level. Messages of the given severity or higher will be logged, and messages of lower severity will be filtered out. **off** filters out all messages.

Case is not significant within *spec*.

Regardless of the log levels set for **file**, logging to a file will not take place unless the target application was invoked with the **−−log−file** option.

For compatibility with older versions of OVS, **any** is accepted as a word but has no effect.

vlog/set PATTERN:*destination***:***pattern*
Sets the log pattern for *destination* to *pattern*. Each time a message is logged to *destination*, *pattern* determines the message's formatting. Most characters in *pattern* are copied literally to the log, but special escapes beginning with **%** are expanded as follows:

%A The name of the application logging the message, e.g. **ovs−vswitchd**.

%B The RFC5424 syslog PRI of the message.

%c The name of the module (as shown by **ovs−appctl −−list**) logging the message.

%d The current date and time in ISO 8601 format (YYYY−MM−DD HH:MM:SS).

%d{*format***}**
The current date and time in the specified *format*, which takes the same format as the *template* argument to **strftime**(3). As an extension, any **#** characters in *format* will be replaced by fractional seconds, e.g. use **%H:%M:%S.###** for the time to the nearest millisecond. Sub-second times are only approximate and currently decimal places after the third will always be reported as zero.

%D The current UTC date and time in ISO 8601 format (YYYY−MM−DD HH:MM:SS).

%D{*format***}**
The current UTC date and time in the specified *format*, which takes the same format as the *template* argument to **strftime**(3). Supports the same extension for sub-second resolution as **%d{...}**.

%E The hostname of the node running the application.

%m The message being logged.

%N A serial number for this message within this run of the program, as a decimal number. The first message a program logs has serial number 1, the second one has serial number 2, and so on.

%n A new-line.

%p The level at which the message is logged, e.g. **DBG**.

%P The program's process ID (pid), as a decimal number.

%r The number of milliseconds elapsed from the start of the application to the time the message was logged.

%t The subprogram name, that is, an identifying name for the process or thread that emitted the log message, such as **monitor** for the process used for **−−monitor** or **main** for the primary process or thread in a program.

%T The subprogram name enclosed in parentheses, e.g. **(monitor)**, or the empty string for the primary process or thread in a program.

%% A literal **%**.

A few options may appear between the **%** and the format specifier character, in this order:

− Left justify the escape's expansion within its field width. Right justification is the default.

0 Pad the field to the field width with **0**s. Padding with spaces is the default.

width A number specifies the minimum field width. If the escape expands to fewer characters than *width* then it is padded to fill the field width. (A field wider than *width* is not truncated to fit.)

The default pattern for console and file output is **%D{%Y-%m-%dT %H:%M:%SZ}|%05N|%c|%p|%m**; for syslog output, **%05N|%c|%p|%m**.

Daemons written in Python (e.g. **ovs−xapi−sync**) do not allow control over the log pattern.

vlog/set FACILITY:*facility*
Sets the RFC5424 facility of the log message. *facility* can be one of **kern**, **user**, **mail**, **daemon**, **auth**, **syslog**, **lpr**, **news**, **uucp**, **clock**, **ftp**, **ntp**, **audit**, **alert**, **clock2**, **local0**, **local1**, **local2**, **local3**, **local4**, **local5**, **local6** or **local7**.

vlog/close
Causes the daemon to close its log file, if it is open. (Use **vlog/reopen** to reopen it later.)

vlog/reopen
Causes the daemon to close its log file, if it is open, and then reopen it. (This is useful after rotating log files, to cause a new log file to be used.)

This has no effect if the target application was not invoked with the **−−log−file** option.

OPTIONS

−h
−−help Prints a brief help message to the console.

−V
−−version
Prints version information to the console.

SEE ALSO

ovs−appctl can control all Open vSwitch daemons, including: **ovs−vswitchd**(8), and **ovsdb−server**(8).