

NAME

ovs-ctl – OVS startup helper script

SYNOPSIS

```

ovs-ctl --system-id=random|uuid [options] start
ovs-ctl stop
ovs-ctl --system-id=random|uuid [options] restart
ovs-ctl status
ovs-ctl version
ovs-ctl [options] load-kmod
ovs-ctl --system-id=random|uuid [options] force-reload-kmod
ovs-ctl [--protocol=protocol] [--sport=sport] [--dport=dport] enable-protocol
ovs-ctl help | -h | --help
ovs-ctl --version

```

DESCRIPTION

The **ovs-ctl** program starts, stops, and checks the status of Open vSwitch daemons. It is not meant to be invoked directly by system administrators but to be called internally by system startup scripts.

Each of **ovs-ctl**'s commands is described separately below.

The “start” command

The **start** command starts Open vSwitch. It performs the following tasks:

1. Loads the Open vSwitch kernel module. If this fails, and the Linux bridge module is loaded but no bridges exist, it tries to unload the bridge module and tries loading the Open vSwitch kernel module again. (This is because the Open vSwitch kernel module cannot coexist with the Linux bridge module before 2.6.37.)

The **start** command skips the following steps if **ovsdb-server** is already running:

2. If the Open vSwitch database file does not exist, it creates it. If the database does exist, but it has an obsolete version, it upgrades it to the latest schema.
3. Starts **ovsdb-server**.
4. Initializes a few values inside the database.
5. If the **--delete-bridges** option was used, deletes all of the bridges from the database.

The **start** command skips the following step if **ovs-vswitchd** is already running:

6. Starts **ovs-vswitchd**.

Options

Several command-line options influence the **start** command's behavior. Some form of the following option should ordinarily be specified:

--system-id=uuid

--system-id=random

This specifies a unique system identifier to store into **external-ids:system-id** in the database's **Open_vSwitch** table. Remote managers that talk to the Open vSwitch database server over network protocols use this value to identify and distinguish Open vSwitch instances, so it should be unique (at least) within OVS instances that will connect to a single controller.

When **random** is specified, **ovs-ctl** will generate a random ID that persists from one run to another (stored in a file). When another string is specified **ovs-ctl** uses it literally.

The following options should be specified if the defaults are not suitable:

--system-type=type

--system-version=version

Sets the value to store in the **system-type** and **system-version** columns, respectively, in the database's **Open_vSwitch** table. Remote managers may use these values to determine the kind of system to which they are connected (primarily for display to human administrators).

When not specified, **ovs-ctl** uses values from the optional **system-type.conf** and **system-version.conf** files (see section **FILES**) or it uses the **lsb_release** program, if present, to provide reasonable defaults.

The following options are also likely to be useful:

--external-id="name=value"

Sets **external-ids:name** to *value* in the database's **Open_vSwitch** table. Specifying this option multiple times adds multiple key-value pairs.

--delete-bridges

Ordinarily Open vSwitch bridges persist from one system boot to the next, as long as the database is preserved. Some environments instead expect to re-create all of the bridges and other configuration state on every boot. This option supports that, by deleting all Open vSwitch bridges after starting **ovsdb-server** but before starting **ovs-vswitchd**.

The following options are less important:

--daemon-cwd=directory

Specifies the current working directory that the OVS daemons should run from. The default is / (the root directory) if this option is not specified. (This option is useful because most systems create core files in a process's current working directory and because a file system that is in use as a process's current working directory cannot be unmounted.)

--no-force-corefiles

By default, **ovs-ctl** enables core dumps for the OVS daemons. This option disables that behavior.

--no-mlockall

By default **ovs-ctl** passes **--mlockall** to **ovs-vswitchd**, requesting that it lock all of its virtual memory, preventing it from being paged to disk. This option suppresses that behavior.

--ovsdb-server-priority=nice_{ness}

--ovs-vswitchd-priority=nice_{ness}

Sets the **nice**(1) level used for each daemon. All of them default to **-10**.

--ovsdb-server-wrapper=wrapper

--ovs-vswitchd-wrapper=wrapper

Configures the specified daemon to run under *wrapper*, which is one of the following:

valgrind

Run the daemon under **valgrind**(1), if it is installed, logging to *daemon.valgrind.log.pid* in the log directory.

strace Run the daemon under **strace**(1), if it is installed, logging to *daemon.strace.log.pid* in the log directory.

glibc Enable GNU C library features designed to find memory errors.

By default, no wrapper is used.

Each of the wrappers can expose bugs in Open vSwitch that lead to incorrect operation, including crashes. The **valgrind** and **strace** wrappers greatly slow daemon operations so they should not be used in production. They also produce voluminous logs that can quickly fill small disk partitions. The **glibc** wrapper is less resource-intensive but still somewhat slows the daemons.

The following options control file locations. They should only be used if the default locations cannot be used. See **FILES**, below, for more information.

--db-file=file

Overrides the file name for the OVS database.

--db-sock=socket

Overrides the file name for the Unix domain socket used to connect to **ovsdb-server**.

--db-schema=*schema*

Overrides the file name for the OVS database schema.

--extra-dbs=*file*

Adds *file* as an extra database for **ovsdb-server** to serve out. Multiple space-separated file names may also be specified. *file* should begin with */*; if it does not, then it will be taken as relative to *dbdir*.

The “stop” command

The **stop** command does not unload the Open vSwitch kernel modules.

This command does nothing and finishes successfully if the OVS daemons aren’t running.

The “restart” command

The **restart** command performs a **stop** followed by a **start** command. The command can take the same options as that of the **start** command. In addition, it saves and restores OpenFlow flows for each individual bridge.

The “status” command

The **status** command checks whether the OVS daemons **ovs-vswitchd** and **ovsdb-server** are running and prints messages with that information. It exits with status 0 if the daemons are running, 1 otherwise.

The “version” command

The **version** command runs **ovsdb-server --version** and **ovs-vswitchd --version**.

The “force-reload-kmod” command

The **force-reload-kmod** command allows upgrading the Open vSwitch kernel module without rebooting. It performs the following tasks:

1. Gets a list of OVS “internal” interfaces, that is, network devices implemented by Open vSwitch. The most common examples of these are bridge “local ports”.
2. Saves the OpenFlow flows of each bridge.
3. Stops the Open vSwitch daemons, as if by a call to **ovs-ctl stop**.
4. Saves the kernel configuration state of the OVS internal interfaces listed in step 1, including IP and IPv6 addresses and routing table entries.
5. Unloads the Open vSwitch kernel module (including the bridge compatibility module if it is loaded).
6. Starts OVS back up, as if by a call to **ovs-ctl start**. This reloads the kernel module, restarts the OVS daemons and finally restores the saved OpenFlow flows.
7. Restores the kernel configuration state that was saved in step 4.
8. Checks for daemons that may need to be restarted because they have packet sockets that are listening on old instances of Open vSwitch kernel interfaces and, if it finds any, prints a warning on std-out. DHCP is a common example: if the ISC DHCP client is running on an OVS internal interface, then it will have to be restarted after completing the above procedure. (It would be nice if **ovs-ctl** could restart daemons automatically, but the details are far too specific to a particular distribution and installation.)

force-kmod-reload internally stops and starts OVS, so it accepts all of the options accepted by the **start** command.

The “load-kmod” command

The **load-kmod** command loads the openvswitch kernel modules if they are not already loaded. This operation also occurs as part of the **start** command. The motivation for providing the **load-kmod** command is to allow errors when loading modules to be handled separately from other errors that may occur when running the **start** command.

By default the **load-kmod** command attempts to load the openvswitch kernel module.

The “enable-protocol” command

The **enable-protocol** command checks for rules related to a specified protocol in the system’s **iptables(8)** configuration. If there are no rules specifically related to that protocol, then it inserts a rule to accept the specified protocol.

More specifically:

- If **iptables** is not installed or not enabled, this command does nothing, assuming that lack of filtering means that the protocol is enabled.
- If the **INPUT** chain has a rule that matches the specified protocol, then this command does nothing, assuming that whatever rule is installed reflects the system administrator’s decisions.
- Otherwise, this command installs a rule that accepts traffic of the specified protocol.

This command normally completes successfully, even if it does nothing. Only the failure of an attempt to insert a rule normally causes it to return an exit code other than 0. The following options control the protocol to be enabled:

--protocol=protocol

The name of the IP protocol to be enabled, such as **gre** or **tcp**. The default is **gre**.

--sport=sport

--dport=dport

TCP or UDP source or destination port to match. These are optional and allowed only with **--protocol=tcp** or **--protocol=udp**.

The “help” command

Prints a usage message and exits successfully.

OPTIONS

In addition to the options listed for each command above, this option controls the behavior of several of **ovs-ctl**’s commands.

EXIT STATUS

ovs-ctl exits with status 0 on success and nonzero on failure. The **start** command is considered to succeed if OVS is already started; the **stop** command is considered to succeed if OVS is already stopped.

ENVIRONMENT

The following environment variables affect **ovs-ctl**:

PATH **ovs-ctl** does not hardcode the location of any of the programs that it runs. **ovs-ctl** will add the *sbindir* and *bindir* that were specified at **configure** time to **PATH**, if they are not already present.

OVS_LOGDIR

OVS_RUNDIR

OVS_DBDIR

OVS_SYSCONFDIR

OVS_PKGDATADIR

OVS_BINDIR

OVS_SBINDIR

Setting one of these variables in the environment overrides the respective **configure** option, both for **ovs-ctl** itself and for the other Open vSwitch programs that it runs.

FILES

ovs-ctl uses the following files:

ovs-lib Shell function library used internally by **ovs-ctl**. It must be installed in the same directory as **ovs-ctl**.

logdir/daemon.log

Per-daemon logfiles.

rundir/daemon.pid

Per-daemon pidfiles to track whether a daemon is running and with what process ID.

pkgdatadir/vswitch.ovsschema

The OVS database schema used to initialize the database (use **—db-schema to override this location**).

dbdir/conf.db

The OVS database (use **—db-file** to override this location).

rundir/openvswitch/db.sock

The Unix domain socket used for local communication with **ovsdb-server** (use **—db-sock** to override this location).

sysconfdir/openvswitch/system-id.conf

The persistent system UUID created and read by **—system-id=random**.

sysconfdir/openvswitch/system-type.conf***sysconfdir/openvswitch/system-version.conf***

The **system-type** and **system-version** values stored in the database's **Open_vSwitch** table when not specified as a command-line option.

EXAMPLE

The files **debian/openvswitch-switch.init** and **xenserver/etc_init.d_openvswitch** in the Open vSwitch source distribution are good examples of how to use **ovs-ctl**.

SEE ALSO

README.md, **INSTALL.Linux.md**, **ovsdb-server(8)**, **ovs-vswitchd(8)**.