NAME

ovsdb−tool − Open vSwitch database management utility

SYNOPSIS

Database Creation Commands:

ovsdb−tool [*options*] **create** [*db* [*schema*]]
ovsdb−tool [*options*] **create−cluster** *db contents address*
ovsdb−tool [*options*] [**−−cid=***uuid*] **join−cluster** *db name local remote...*

Version Management Commands:

ovsdb−tool [*options*] **convert** [*db* [*schema* [*target*]]]
ovsdb−tool [*options*] **needs−conversion** [*db* [*schema*]]
ovsdb−tool [*options*] **db−version** [*db*]
ovsdb−tool [*options*] **schema−version** [*schema*]
ovsdb−tool [*options*] **db−cksum** [*db*]
ovsdb−tool [*options*] **schema−cksum** [*schema*]
ovsdb−tool [*options*] **compare-versions** *a op b*

Other commands:

ovsdb−tool [*options*] **compact** [*db* [*target*]]
ovsdb−tool [*options*] [**−−rbac−role=***role*] **query** [*db*] *transaction*
ovsdb−tool [*options*] [**−−rbac−role=***role*] **transact** [*db*] *transaction*
ovsdb−tool [*options*] [**−m** | **−−more**]... **show−log** [*db*]
ovsdb−tool [*options*] **check−cluster** *db...*
ovsdb−tool [*options*] **db−name** [*db*]
ovsdb−tool [*options*] **schema−name** [*schema*]
ovsdb−tool [*options*] **db−cid** *db*
ovsdb−tool [*options*] **db−sid** *db*
ovsdb−tool [*options*] **db−local−address db**
ovsdb−tool help

Logging options:

[**−v**[*module*[**:***destination*[**:***level*]]]]...
[**−−verbose[=***module*[**:***destination*[**:***level*]]]]...
[**−−log−file**[**=***file*]]

Common options:

[**−h** | **−−help**] [**−V** | **−−version**]

DESCRIPTION

The **ovsdb−tool** program is a command-line tool for managing Open vSwitch database (OVSDB) files. It does not interact directly with running Open vSwitch database servers (instead, use **ovsdb−client**). For an introduction to OVSDB and its implementation in Open vSwitch, see **ovsdb**(7).

Each command that takes an optional *db* or *schema* argument has a default file location if it is not specified.. The default *db* is **/etc/openvswitch/conf.db**. The default *schema* is **/share/openvswitch/vswitch.ovsschema**.

This OVSDB implementation supports standalone and active-backup database service models with one on-disk format and a clustered database service model with a different format. **ovsdb−tool** supports both formats, but some commands are appropriate for only one format, as documented for individual commands below. For a specification of these formats, see **ovsdb**(5). For more information on OVSDB service models, see the **Service Models** section in **ovsdb**(7).

Database Creation Commands

These commands create a new OVSDB database file. They will not overwrite an existing database file. To replace an existing database with a new one, first delete the old one.

create [*db* [*schema*]]
Use this command to create the database for controlling **ovs−vswitchd** or another standalone or active-backup database. It creates database file *db* with the given *schema*, which must be the name of a file that contains an OVSDB schema in JSON format, as specified in the OVSDB specification. The new database is initially empty. (You can use **cp** to copy a database including both its schema and data.)

create−cluster *db contents local*
Use this command to initialize the first server in a high-availability cluster of 3 (or more) database servers, e.g. for a database in an environment that cannot tolerate a single point of failure. It creates clustered database file *db* and configures the server to listen on *local*, which must take the form *protocol***:***ip***:***port*, where *protocol* is **tcp** or **ssl**, *ip* is the server's IP (either an IPv4 address or an IPv6 address enclosed in square brackets), and *port* is a TCP port number. Only one address is specified, for the first server in the cluster, ordinarily the one for the server running **create−cluster**. The address is used for communication within the cluster, not for communicating with OVSDB clients, and must not use the same port used for the OVSDB protocol.

The new database is initialized with *contents*, which must name a file that contains either an OVSDB schema in JSON format or a standalone OVSDB database. If it is a schema file, the new database will initially be empty, with the given schema. If it is a database file, the new database will have the same schema and contents.

[**−−cid=***uuid*] **join−cluster** *db name local remote*...
Use this command to initialize each server after the first one in an OVSDB high-availability cluster. It creates clustered database file *db* for a database named *name*, and configures the server to listen on *local* and to initially connect to *remote*, which must be a server that already belongs to the cluster. *local* and *remote* use the same *protocol***:***ip***:***port* syntax as **create−cluster**.

The *name* must be the name of the schema or database passed to **create−cluster**. For example, the name of the OVN Southbound database schema is **OVN_Southbound**. Use **ovsdb−tool**'s **schema−name** or **db−name** command to find out the name of a schema or database, respectively.

This command does not do any network access, which means that it cannot actually join the new server to the cluster. Instead, the *db* file that it creates prepares the server to join the cluster the first time that **ovsdb−server** serves it. As part of joining the cluster, the new server retrieves the database schema and obtains the list of all cluster members. Only after that does it become a full member of the cluster.

Optionally, more than one *remote* may be specified; for example, in a cluster that already contains multiple servers, one could specify all the existing servers. This is beneficial if some of the existing servers are down while the new server joins, but it is not otherwise needed.

By default, the *db* created by **join−cluster** will join any clustered database named *name* that is available at a *remote*. In theory, if machines go up and down and IP addresses change in the right way, it could join the wrong database cluster. To avoid this possibility, specify **−−cid=***uuid*, where *uuid* is the cluster ID of the cluster to join, as printed by **ovsdb−tool get−cid**.

Database Migration Commands
This commands will convert cluster database to standalone database.

cluster−to−standalone *db clusterdb*
Use this command to convert to standalone database from clustered database when the cluster is down and cannot be revived. It creates new standalone *db* file from the given cluster *db* file.

Version Management Commands
An OVSDB schema has a schema version number, and an OVSDB database embeds a particular version of an OVSDB schema. These version numbers take the form *x***.***y***.***z*, e.g. **1.2.3**. The OVSDB implementation does not enforce a particular version numbering scheme, but schemas managed within the Open vSwitch project use the following approach. Whenever the database schema is changed in a non-backward compatible way (e.g. deleting a column or a table), *x* is incremented (and *y* and *z* are reset to 0). When the database

schema is changed in a backward compatible way (e.g. adding a new column), *y* is incremented (and *z* is reset to 0). When the database schema is changed cosmetically (e.g. reindenting its syntax), *z* is incremented.

Some OVSDB databases and schemas, especially very old ones, do not have a version number.

Schema version numbers and Open vSwitch version numbers are independent.

These commands work with different versions of OVSDB schemas and databases.

convert [*db* [*schema* [*target*]]]
: Reads *db*, translating it into to the schema specified in *schema*, and writes out the new interpretation. If *target* is specified, the translated version is written as a new file named *target*, which must not already exist. If *target* is omitted, then the translated version of the database replaces *db* in-place. In-place conversion cannot take place if the database is currently being served by **ovsdb−server** (instead, either stop **ovsdb−server** first or use **ovsdb−client**'s **convert** command).

  This command can do simple ''upgrades'' and ''downgrades'' on a database's schema. The data in *db* must be valid when interpreted under *schema*, with only one exception: data in *db* for tables and columns that do not exist in *schema* are ignored. Columns that exist in *schema* but not in *db* are set to their default values. All of *schema*'s constraints apply in full.

  Some uses of this command can cause unrecoverable data loss. For example, converting a database from a schema that has a given column or table to one that does not will delete all data in that column or table. Back up critical databases before converting them.

  This command is for standalone and active-backup databases only. For clustered databases, use **ovsdb−client**'s **convert** command to convert them online.

needs−conversion [*db* [*schema*]]
: Reads the schema embedded in *db* and the JSON schema from *schema* and compares them. If the schemas are the same, prints **no** on stdout; if they differ, prints **yes**.

  This command is for standalone and active-backup databases only. For clustered databases, use **ovsdb−client**'s **needs-conversion** command instead.

db−version [*db*]
schema−version [*schema*]
: Prints the version number in the schema embedded within the database *db* or in the JSON schema *schema* on stdout. If *schema* or *db* was created before schema versioning was introduced, then it will not have a version number and this command will print a blank line.

  The **db−version** command is for standalone and active-backup databases only. For clustered databases, use **ovsdb−client**'s **schema−version** command instead.

db−cksum [*db*]
schema−cksum [*schema*]
: Prints the checksum in the schema embedded within the database *db* or of the JSON schema *schema* on stdout. If *schema* or *db* was created before schema checksums were introduced, then it will not have a checksum and this command will print a blank line.

  The **db−cksum** command is for standalone and active-backup databases only. For clustered databases, use **ovsdb−client**'s **schema−cksum** command instead.

compare-versions *a op b*
: Compares *a* and *b* according to *op*. Both *a* and *b* must be OVSDB schema version numbers in the form *x*.*y*.*z*, as described in **ovsdb**(7), and *op* must be one of **< <= == >= > !=**. If the comparison is true, exits with status 0; if it is false, exits with status 2. (Exit status 1 indicates an error, e.g. *a* or *b* is the wrong syntax for an OVSDB version or *op* is not a valid comparison operator.)

Other Commands

compact [*db* [*target*]]
Reads *db* and writes a compacted version. If *target* is specified, the compacted version is written as a new file named *target*, which must not already exist. If *target* is omitted, then the compacted version of the database replaces *db* in-place. This command is not needed in normal operation because **ovsdb−server** from time to time automatically compacts a database that grows much larger than its minimum size.

This command does not work if *db* is currently being served by **ovsdb−server**, or if it is otherwise locked for writing by another process. This command also does not work with clustered databases. Instead, in either case, send the **ovsdb−server/compact** command to **ovsdb−server**, via **ovs−appctl**).

[**−−rbac−role=***role*] **query** [*db*] *transaction*
Opens *db*, executes *transaction* on it, and prints the results. The *transaction* must be a JSON array in the format of the **params** array for the JSON-RPC **transact** method, as described in the OVSDB specification.

This command opens *db* for read-only access, so it may safely run concurrently with other database activity, including **ovsdb−server** and other database writers. The *transaction* may specify database modifications, but these will have no effect on *db*.

By default, the transaction is executed using the ‘‘superuser’’ RBAC role. Use **−−rbac−role** to specify a different role.

This command does not work with clustered databases. Instead, use **ovsdb-client**’s **query** command to send the query to **ovsdb−server**.

[**−−rbac−role=***role*] **transact** [*db*] *transaction*
Opens *db*, executes *transaction* on it, prints the results, and commits any changes to *db*. The *transaction* must be a JSON array in the format of the **params** array for the JSON-RPC **transact** method, as described in the OVSDB specification.

This command does not work if *db* is currently being served by **ovsdb−server**, or if it is otherwise locked for writing by another process. This command also does not work with clustered databases. Instead, in either case, use **ovsdb−client**’s **transact** command to send the query to **ovsdb−server**.

By default, the transaction is executed using the ‘‘superuser’’ RBAC role. Use **−−rbac−role** to specify a different role.

[**−m** | **−−more**]... **show−log** [*db*]
Prints a summary of the records in *db*’s log, including the time and date at which each database change occurred and any associated comment. This may be useful for debugging.

To increase the verbosity of output, add **−m** (or **−−more**) one or more times to the command line. With one **−m**, **show−log** prints a summary of the records added, deleted, or modified by each transaction. With two **−m**s, **show−log** also prints the values of the columns modified by each change to a record.

This command works with standalone and active-backup databases and with clustered databases, but the output formats are different.

check−cluster *db*...
Reads all of the records in the supplied databases, which must be collected from different servers (and ideally all the servers) in a single cluster. Checks each database for self-consistency and the set together for cross-consistency. If **ovsdb−tool** detects unusual but not necessarily incorrect content, it prints a warning or warnings on stdout. If **ovsdb−tool** find consistency errors, it prints an error on stderr and exits with status 1. Errors typically indicate bugs in **ovsdb−server**; please consider reporting them to the Open vSwitch developers.

db−name [*db*]
schema−name [*schema*]
Prints the name of the schema embedded within the database *db* or in the JSON schema *schema* on stdout.

db−cid *db*
Prints the cluster ID, which is a UUID that identifies the cluster, for *db*. If *db* is a database newly created by **ovsdb−tool cluster−join** that has not yet successfully joined its cluster, and **−−cid** was not specified on the **cluster−join** command line, then this command will output an error, and exit with status 2, because the cluster ID is not yet known. This command works only with clustered databases.

The all-zeros UUID is not a valid cluster ID.

db−sid *db*
Prints the server ID, which is a UUID that identifies the server, for *db*. This command works only with clustered databases. It works even if *db* is a database newly created by **ovsdb−tool cluster−join** that has not yet successfully joined its cluster.

db−local−address db
Prints the local address used for database clustering for *db*, in the same *protocol***:***ip***:***port* form used on **create−cluster** and **join−cluster**.

db−is−clustered *db*
db−is−standalone *db*
Tests whether *db* is a database file in clustered or standalone format, respectively. If so, exits with status 0; if not, exits with status 2. (Exit status 1 indicates an error, e.g. *db* is not an OVSDB database or does not exist.)

OPTIONS

Logging Options

−v[*spec*]
−−verbose=[*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. (If **−−detach** is specified, **ovsdb−tool** closes its standard file descriptors, so logging to the console will have no effect.)

  On Windows platform, **syslog** is accepted as a word and is only useful along 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. See **ovs−appctl**(8) for a definition of each log level.

Case is not significant within *spec*.

Regardless of the log levels set for **file**, logging to a file will not take place unless **−−log−file** is also specified (see below).

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

−v
−−verbose
Sets the maximum logging verbosity level, equivalent to **−−verbose=dbg**.

−vPATTERN:*destination***:***pattern*
−−verbose=PATTERN:*destination***:***pattern*
Sets the log pattern for *destination* to *pattern*. Refer to **ovs−appctl**(8) for a description of the valid syntax for *pattern*.

−vFACILITY:*facility*
−−verbose=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**. If this option is not specified, **daemon** is used as the default for the local system syslog and **local0** is used while sending a message to the target provided via the **−−syslog−target** option.

−−log−file[**=***file*]
Enables logging to a file. If *file* is specified, then it is used as the exact name for the log file. The default log file name used if *file* is omitted is **/var/log/openvswitch/ovsdb−tool.log**.

−−syslog−target=*host***:***port*
Send syslog messages to UDP *port* on *host*, in addition to the system syslog. The *host* must be a numerical IP address, not a hostname.

−−syslog−method=*method*
Specify *method* how syslog messages should be sent to syslog daemon. Following forms are supported:

- **libc**, use libc **syslog()** function. Downside of using this options is that libc adds fixed prefix to every message before it is actually sent to the syslog daemon over **/dev/log** UNIX domain socket.
- **unix:***file*, use UNIX domain socket directly. It is possible to specify arbitrary message format with this option. However, **rsyslogd 8.9** and older versions use hard coded parser function anyway that limits UNIX domain socket use. If you want to use arbitrary message format with older **rsyslogd** versions, then use UDP socket to localhost IP address instead.
- **udp:***ip*:*port*, use UDP socket. With this method it is possible to use arbitrary message format also with older **rsyslogd**. When sending syslog messages over UDP socket extra precaution needs to be taken into account, for example, syslog daemon needs to be configured to listen on the specified UDP port, accidental iptables rules could be interfering with local syslog traffic and there are some security considerations that apply to UDP sockets, but do not apply to UNIX domain sockets.
- **null**, discards all messages logged to syslog.

The default is taken from the **OVS_SYSLOG_METHOD** environment variable; if it is unset, the default is **libc**.

Other Options

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

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

FILES

The default *db* is **/etc/openvswitch/conf.db**. The default *schema* is **/share/openvswitch/vswitch.ovsschema**. The **help** command also displays these defaults.

SEE ALSO

ovsdb(7), **ovsdb−server**(1), **ovsdb−client**(1).