

**NAME**

**systemd-analyze** – Analyze system boot-up performance

**SYNOPSIS**

**systemd-analyze** [OPTIONS...] [time]  
**systemd-analyze** [OPTIONS...] blame  
**systemd-analyze** [OPTIONS...] critical-chain [UNIT...]  
**systemd-analyze** [OPTIONS...] plot [> file.svg]  
**systemd-analyze** [OPTIONS...] dot [PATTERN...] [> file.dot]  
**systemd-analyze** [OPTIONS...] dump  
**systemd-analyze** [OPTIONS...] set-log-level *LEVEL*  
**systemd-analyze** [OPTIONS...] set-log-target *TARGET*  
**systemd-analyze** [OPTIONS...] verify [FILES...]

**DESCRIPTION**

**systemd-analyze** may be used to determine system boot-up performance statistics and retrieve other state and tracing information from the system and service manager, and to verify the correctness of unit files.

**systemd-analyze time** prints the time spent in the kernel before userspace has been reached, the time spent in the initial RAM disk (initrd) before normal system userspace has been reached, and the time normal system userspace took to initialize. Note that these measurements simply measure the time passed up to the point where all system services have been spawned, but not necessarily until they fully finished initialization or the disk is idle.

**systemd-analyze blame** prints a list of all running units, ordered by the time they took to initialize. This information may be used to optimize boot-up times. Note that the output might be misleading as the initialization of one service might be slow simply because it waits for the initialization of another service to complete.

**systemd-analyze critical-chain** [UNIT...] prints a tree of the time-critical chain of units (for each of the specified *UNIT*s or for the default target otherwise). The time after the unit is active or started is printed after the "@" character. The time the unit takes to start is printed after the "+" character. Note that the output might be misleading as the initialization of one service might depend on socket activation and because of the parallel execution of units.

**systemd-analyze plot** prints an SVG graphic detailing which system services have been started at what time, highlighting the time they spent on initialization.

**systemd-analyze dot** generates textual dependency graph description in dot format for further processing with the GraphViz **dot**(1) tool. Use a command line like **systemd-analyze dot | dot -Tsvg > systemd.svg** to generate a graphical dependency tree. Unless **--order** or **--require** is passed, the generated graph will show both ordering and requirement dependencies. Optional pattern globbing style specifications (e.g. \*.target) may be given at the end. A unit dependency is included in the graph if any of these patterns match either the origin or destination node.

**systemd-analyze dump** outputs a (usually very long) human-readable serialization of the complete server state. Its format is subject to change without notice and should not be parsed by applications.

**systemd-analyze set-log-level** *LEVEL* changes the current log level of the **systemd** daemon to *LEVEL* (accepts the same values as **--log-level=** described in **systemd**(1)).

**systemd-analyze set-log-target** *TARGET* changes the current log target of the **systemd** daemon to *TARGET* (accepts the same values as **--log-target=**, described in **systemd**(1)).

**systemd-analyze verify** will load unit files and print warnings if any errors are detected. Files specified on the command line will be loaded, but also any other units referenced by them. This command works by prepending the directories for all command line arguments at the beginning of the unit load path, which means that all units files found in those directories will be used in preference to the unit files found in the

standard locations, even if not listed explicitly.

If no command is passed, **systemd-analyze time** is implied.

## OPTIONS

The following options are understood:

### **--user**

Operates on the user systemd instance.

### **--system**

Operates on the system systemd instance. This is the implied default.

### **--order, --require**

When used in conjunction with the **dot** command (see above), selects which dependencies are shown in the dependency graph. If **--order** is passed, only dependencies of type *After=* or *Before=* are shown. If **--require** is passed, only dependencies of type *Requires=*, *Requisite=*, *Wants=* and *Conflicts=* are shown. If neither is passed, this shows dependencies of all these types.

### **--from-pattern=, --to-pattern=**

When used in conjunction with the **dot** command (see above), this selects which relationships are shown in the dependency graph. Both options require a **glob(7)** pattern as an argument, which will be matched against the left-hand and the right-hand, respectively, nodes of a relationship.

Each of these can be used more than once, in which case the unit name must match one of the values. When tests for both sides of the relation are present, a relation must pass both tests to be shown. When patterns are also specified as positional arguments, they must match at least one side of the relation. In other words, patterns specified with those two options will trim the list of edges matched by the positional arguments, if any are given, and fully determine the list of edges shown otherwise.

### **--fuzz=timespan**

When used in conjunction with the **critical-chain** command (see above), also show units, which finished *timespan* earlier, than the latest unit in the same level. The unit of *timespan* is seconds unless specified with a different unit, e.g. "50ms".

### **--no-man**

Do not invoke man to verify the existence of man pages listed in *Documentation=*.

### **-H, --host=**

Execute the operation remotely. Specify a hostname, or a username and hostname separated by "@", to connect to. The hostname may optionally be suffixed by a container name, separated by ":", which connects directly to a specific container on the specified host. This will use SSH to talk to the remote machine manager instance. Container names may be enumerated with **machinectl -H HOST**.

### **-M, --machine=**

Execute operation on a local container. Specify a container name to connect to.

### **-h, --help**

Print a short help text and exit.

### **--version**

Print a short version string and exit.

### **--no-pager**

Do not pipe output into a pager.

## EXIT STATUS

On success, 0 is returned, a non-zero failure code otherwise.

## EXAMPLES FOR DOT

**Example 1. Plots all dependencies of any unit whose name starts with "avahi-daemon"**

```
$ systemd-analyze dot 'avahi-daemon.*' | dot -Tsvg > avahi.svg
```

```
$ eog avahi.svg
```

### Example 2. Plots the dependencies between all known target units

```
systemd-analyze dot --to-pattern='*.target' --from-pattern='*.target' | dot -Tsvg > targets.svg
$ eog targets.svg
```

## EXAMPLES FOR VERIFY

The following errors are currently detected:

- unknown sections and directives,
- missing dependencies which are required to start the given unit,
- man pages listed in *Documentation=* which are not found in the system,
- commands listed in *ExecStart=* and similar which are not found in the system or not executable.

### Example 3. Misspelt directives

```
$ cat ./user.slice
[Unit]
WhatIsThis=11
Documentation=man:nosuchfile(1)
Requires=different.service

[Service]
Desription=x

$ systemd-analyze verify ./user.slice
[./user.slice:9] Unknown lvalue 'WhatIsThis' in section 'Unit'
[./user.slice:13] Unknown section 'Service'. Ignoring.
Error: org.freedesktop.systemd1.LoadFailed:
  Unit different.service failed to load:
  No such file or directory.
Failed to create user.slice/start: Invalid argument
user.slice: man nosuchfile(1) command failed with code 16
```

### Example 4. Missing service units

```
$ tail ./a.socket ./b.socket
==> ./a.socket <==
[Socket]
ListenStream=100

==> ./b.socket <==
[Socket]
ListenStream=100
Accept=yes

$ systemd-analyze verify ./a.socket ./b.socket
Service a.service not loaded, a.socket cannot be started.
Service b@0.service not loaded, b.socket cannot be started.
```

## ENVIRONMENT

**\$SYSTEMD\_PAGER**

Pager to use when **—no-pager** is not given; overrides **\$PAGER**. Setting this to an empty string or the value "cat" is equivalent to passing **—no-pager**.

*\$SYSTEMD\_LESS*

Override the default options passed to **less** ("FRSXMK").

**SEE ALSO**

**systemd**(1), **systemctl**(1)