
Subject: Re: [PATCH try4, 0/3] split verbose from LOG_LEVEL

Posted by [Igor Sukhih](#) on Fri, 08 Dec 2006 14:16:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

> I think we can agree on 1 and 2.
> If so, please apply regardless of 3, which is optional.

>

> 1. Split --verbose from LOG_LEVEL

> verbose is 0 regardless of LOG_LEVEL

> --verbose will (as today) set verbose to 2

> LOG_LEVEL not affected by --verbose

>

> 2. Add optional argument to --verbose

> --verbose

> --verbose level

> --verbose=level

>

> 3. Add VERBOSE to vz.conf

>

> Regards,

>

Slightly rewritten version, sorry for single patch too lazy for splinting.

Changes from original

* fixed use case 'vzctl --verbose'

* introduced set_log_verbose() instead of init_log() modification

* if VERBOSE not set use LOG_LEVEL

--

Igor.

diff --git a/etc/vz.conf b/etc/vz.conf

index 29d13ff..a230f5a 100644

--- a/etc/vz.conf

+++ b/etc/vz.conf

@@ -8,6 +8,7 @@ ## Logging parameters

LOGGING=yes

LOGFILE=/var/log/vzctl.log

LOG_LEVEL=0

+VERBOSE=0

Disk quota parameters

DISK_QUOTA=yes

diff --git a/include/logger.h b/include/logger.h

index 67240b6..8c3d33e 100644

--- a/include/logger.h

+++ b/include/logger.h

```

@@ -21,7 +21,7 @@ #define _LOGGER_H_
#include <stdio.h>
#include "types.h"

-#define LOG_DATA log_param g_log = {NULL, 0, 1, 0, "", 0};
+#define LOG_DATA log_param g_log = {NULL, 0, 1, 0, 0, "", 0};

/** Data structure for logging.
 */
@@ -30,6 +30,7 @@ typedef struct {
    int level; /**< maximum logging level. */
    int enable; /**< enable/disable logging. */
    int quiet; /**< skip logging to stdout. */
+ int verbose; /**< Console verbosity. */
    char prog[32]; /**< program name. */
    envid_t veid;
} log_param;
@@ -66,5 +67,6 @@ int init_log(char *file, envid_t veid, i
void free_log();

void set_log_level(int level);
+void set_log_verbose(int level);

#endif
diff --git a/include/res.h b/include/res.h
index bb4ec31..b1d1252 100644
--- a/include/res.h
+++ b/include/res.h
@@ -102,9 +102,10 @@ typedef struct {
} vps_opt;

struct log_s {
-    char *log_file;
-    int level;
-    int enable;
+ char *log_file;
+ int level;
+ int enable;
+ int *verbose;
};

struct vps_param {
diff --git a/include/vzctl_param.h b/include/vzctl_param.h
index ab4790e..0037b5c 100644
--- a/include/vzctl_param.h
+++ b/include/vzctl_param.h
@@ -129,6 +129,7 @@ #define PARAM_NETIF_MAC 354
#define PARAM_NETIF_IFNAME 355

```

```

#define PARAM_NETIF_HOST_MAC 356
#define PARAM_NETIF_HOST_IFNAME 357
+#define PARAM_VERBOSE 358

#define PARAM_LINE "e:p:f:t:i:l:k:a:b:n:x:h"
#endif
diff --git a/man/vz.conf.5 b/man/vz.conf.5
index f5448ce..fc9a386 100644
--- a/man/vz.conf.5
+++ b/man/vz.conf.5
@@ -29,7 +29,9 @@ Enables or disables logging. This parame
Set location of log file, default is \f(CR/var/log/vzctl.log\fR.
.IP "\fBLOG_LEVEL\fR=\fInumber\fR"
Set the logging level. The more \fInumber\fR is, the more information will be
-displayed.
+logged.
+.IP "\fBVERBOSE\fR=\fInumber\fR"
+Set the output verbosity. Default is 0.
.SS Disk quota parameters
.IP \fBDISK_QUOTA\fR=\fByes\fR|\fBno\fR
In case the value of this parameter is set to \fBno\fR, all disk
diff --git a/src/lib/config.c b/src/lib/config.c
index 3128ce9..b8fdd23 100644
--- a/src/lib/config.c
+++ b/src/lib/config.c
@@ -53,6 +53,7 @@ static vps_config config[] = {
{"LOGGING", NULL, PARAM_LOGGING},
{"LOG_LEVEL", NULL, PARAM_LOGLEVEL},
{"LOGFILE", NULL, PARAM_LOGFILE},
+{"VERBOSE", NULL, PARAM_VERBOSE},

{"IPTABLES", NULL, PARAM_IPTABLES},
/* UB */
@@ -1806,6 +1807,14 @@ static int parse(envid_t veid, vps_param
break;
vps_p->log.level = int_id;
break;
+ case PARAM_VERBOSE:
+ if (parse_int(val, &int_id))
+ break;
+ vps_p->log.verbose = malloc(sizeof(*vps_p->log.verbose));
+ if (vps_p->log.verbose == NULL)
+ return VZ_RESOURCE_ERROR;
+ *vps_p->log.verbose = int_id;
+ break;
case PARAM_IPTABLES:
ret = parse_iptables(&vps_p->res.env, val);
break;

```

```

diff --git a/src/lib/logger.c b/src/lib/logger.c
index bd96e5e..9c483ad 100644
--- a/src/lib/logger.c
+++ b/src/lib/logger.c
@@ -53,7 +53,7 @@ void logger(int log_level, int err_no, c
    else
        out = stdout;
    va_start(ap, format);
- if (!g_log.quiet && g_log.level >= log_level) {
+ if (!g_log.quiet && g_log.verbose >= log_level) {
    va_list ap_save;

    va_copy(ap_save, ap);
@@ -108,6 +108,11 @@ void set_log_level(int level)
    g_log.level = level;
}

+void set_log_verbose(int level)
+{
+ g_log.verbose = level;
+}
+
int init_log(char *file, envuid_t veid, int enable, int level, int quiet,
char *progname)
{
@@ -118,6 +123,7 @@ int init_log(char *file, envuid_t veid, i
    return ret;
    g_log.enable = enable;
    set_log_level(level);
+ set_log_verbose(level);
    g_log.veid = veid;
    g_log.quiet = quiet;
    if (progname != NULL)
diff --git a/src/vzcfgvalidate.c b/src/vzcfgvalidate.c
diff --git a/src/vzctl.c b/src/vzctl.c
index f7007c0..e24eff0 100644
--- a/src/vzctl.c
+++ b/src/vzctl.c
@@ -102,13 +102,15 @@ int main(int argc, char *argv[], char *e
{
    int action = 0;
    int verbose = 0;
+ int verbose_tmp;
+ int verbose_custom = 0;
    int quiet = 0;
    int veid, ret, skiplock = 0;
    char buf[256];
    vps_param *gparam = NULL, *vps_p = NULL, *cmd_p = NULL;

```

```

const char *action_nm;
struct sigaction act;
- char *name = NULL;
+ char *name = NULL, *opt;

_proc_title = argv[0];
_proc_title_len = envp[0] - argv[0];
@@ -123,14 +125,32 @@ int main(int argc, char *argv[], char *e
sigaction(SIGPIPE, &act, NULL);

while (argc > 1) {
- if (!strcmp(argv[1], "--verbose")) {
- verbose = 2;
- } else if (!strcmp(argv[1], "--quiet"))
+ opt = argv[1];
+
+ if (!strcmp(opt, "--verbose")) {
+ if (argc > 2 &&
+ !parse_int(argv[2], &verbose_tmp))
+ {
+ verbose += verbose_tmp;
+ argc--; argv++;
+ } else {
+ verbose++;
+ }
+ verbose_custom = 1;
+ } else if (!strncmp(opt, "--verbose=", 10)) {
+ if (parse_int(opt + 10, &verbose_tmp)) {
+ fprintf(stderr, "Invalid value for"
+ " --verbose\n");
+ exit(VZ_INVALID_PARAMETER_VALUE);
+ }
+ verbose += verbose_tmp;
+ verbose_custom = 1;
+ } else if (!strcmp(opt, "--quiet"))
+ quiet = 1;
- else if (!strcmp(argv[1], "--version")) {
+ else if (!strcmp(opt, "--version")) {
+ version();
+ exit(0);
- } else if (!strcmp(argv[1], "--skiplock"))
+ } else if (!strcmp(opt, "--skiplock"))
+ skiplock = YES;
+ else
+ break;
@@ -228,8 +248,18 @@ int main(int argc, char *argv[], char *e
goto error;
}

```

```

    init_log(gparam->log.log_file, veid, gparam->log.enable != NO,
-   verbose ? gparam->log.level + 2 : gparam->log.level,
-   quiet, "vzctl");
+   gparam->log.level, quiet, "vzctl");
+   /* Set verbose level from global config if not overwritten
+    by --verbose
+   */
+   if (!verbose_custom && gparam->log.verbose != NULL) {
+   verbose = *gparam->log.verbose;
+   verbose_custom = 1;
+   }
+   if (verbose < -1)
+   verbose = -1;
+   if (verbose_custom)
+   set_log_verbose(verbose);
    if ((ret = parse_action_opt(veid, action, argc, argv, cmd_p,
        action_nm)))
    {

```
