```
**********************************************************
    9860 Sat Apr 20 05:52:26 2019
new/usr/src/man/man4/system.4
10832 system.4 refers to coffexec
**********************************************************
   1 '\" te
   2 .\" Copyright (c) 2003 Sun Microsystems, Inc.  All Rights Reserved.
   3 .\" Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>
   4 .\" Copyright 2019 OmniOS Community Edition (OmniOSce) Association.
   5 .\" Copyright 2019 Peter Tribble
   6 .\" Copyright 1989 AT&T
   7 .\" The contents of this file are subject to the terms of the Common Development
   8 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   9 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
  10 .TH SYSTEM 4 "Apr 20, 2019"
   9 .TH SYSTEM 4 "Jan 29, 2019"
  11 .SH NAME
  12 system \- system configuration information file
  13 .SH DESCRIPTION
  14 .LP
  15 The \fBsystem\fR file is used for customizing the operation of the operating
  16 system kernel. The recommended procedure is to preserve the original
  17 \fBsystem\fR file before modifying it.
  18 .sp
  19 .LP
  20 It is not recommended to edit the \fB/etc/system\fR file directly but rather
  21 to deliver configuration fragments into files under \fB/etc/system.d\fR;
  22 files in this directory are combined in alphabetical order and read by the
  23 kernel before \fB/etc/system\fR is processed. Directives in \fB/etc/system\fR
  24 therefore take precedence over any settings delivered in fragment files.
  25 .sp
  26 .LP
  27 The recommended naming schema for the fragment files is to use the name of
  28 the package which is delivering the file with '\fB/\fR' characters replaced
  29 by '\fB:\fR'; file names that start with a dot (\fB.\fR) will be ignored.
  30 .sp
  31 .LP
  32 If \fB/etc/system.d/\fR exists and contains any fragment files,
  33 then the directory must also be writable or it will not be possible to
  34 create or update the system boot archive.
  35 .sp
  36 .LP
  37 The \fBsystem\fR file contains commands which are read by the kernel during
  38 initialization and used to customize the operation of your system. These
  39 commands are useful for modifying the system's treatment of its loadable kernel
  40 modules.
  41 .sp
  42 .LP
  43 The syntax of the \fBsystem\fR file consists of a list of keyword/value pairs
  44 which are recognized by the system as valid commands. Comment lines must begin
  45 with an asterisk (\fB*\fR) or a hash mark (\fB#\fR) and end with a newline
  46 character. All commands are case-insensitive except where noted.
  47 .sp
  48 .LP
  49 Commands that modify the system's operation with respect to loadable kernel
  50 modules require you to specify the module type by listing the module's
  51 namespace. The following namespaces are currently supported on all platforms:
  52 .sp
  53 .ne 2
  54 .na
  55 \fB\fBdrv\fR\fR
  56 .ad
  57 .RS 10n
  58 Modules in this namespace are device drivers.
  59 .RE
```

```
  61 .sp
  62 .ne 2
  63 .na
  64 \fB\fBexec\fR\fR
  65 .ad
  66 .RS 10n
  67 Modules in this namespace are execution format modules. The following
  68 \fBexec\fR modules are currently provided:
  69 .sp
  70 .ne 2
  71 .na
  72 \fBOnly on SPARC systems:\fR
  71 \fBOnly on SPARC system:\fR
  73 .ad
  74 .RS 28n
  75 .sp
  76 .in +2
  77 .nf
  78 aoutexec
  79 .fi
  80 .in -2
  80 .sp

  82 .RE

  84 .sp
  85 .ne 2
  86 .na
  87 \fBOnly on x86 system:\fR
  88 .ad
  89 .RS 28n
  90 .sp
  91 .in +2
  92 .nf
  93 coffexec
  94 .fi
  95 .in -2
  81 .sp

  83 .RE

  85 .sp
  86 .ne 2
  87 .na
  88 \fBOn SPARC and IA systems:\fR
  89 .ad
  90 .RS 28n
  91 .sp
  92 .in +2
  93 .nf
  94 elfexec
  95 intpexec
  96 javaexec
  97 .fi
  98 .in -2
  99 .sp

 101 .RE

 103 .RE

 105 .sp
 106 .ne 2
 107 .na
 108 \fB\fBfirmware\fR\fR
 109 .ad
```

```
.RS 10n
Raw firmware images in subdirectories, one for each device driver
module using \fBfirmload\fR(9F).
.RE

.sp
.ne 2
.na
\fB\fBfs\fR\fR
.ad
.RS 10n
These modules are filesystems.
.RE

.sp
.ne 2
.na
\fB\fBsched\fR\fR
.ad
.RS 10n
These modules implement a process scheduling algorithm.
.RE

.sp
.ne 2
.na
\fB\fBstrmod\fR\fR
.ad
.RS 10n
These modules are \fBSTREAMS\fR modules.
.RE

.sp
.ne 2
.na
\fB\fBsys\fR\fR
.ad
.RS 10n
These modules implement loadable system-call modules.
.RE

.sp
.ne 2
.na
\fB\fBmisc\fR\fR
.ad
.RS 10n
These modules do not fit into any of the above categories, so are considered
"miscellaneous" modules.
.RE

.sp
.LP
SPARC only:
.sp
.ne 2
.na
\fB\fBdacf\fR\fR
.ad
.RS 8n
These modules provide rules and actions for device auto-configuration.
.RE

.sp
.ne 2
.na
```

```
\fB\fBtod\fR\fR
.ad
.RS 8n
These modules provide support for the time of day hardware.
.RE

.sp
.ne 2
.na
\fB\fBcpu\fR\fR
.ad
.RS 8n
These modules provide \fBCPU\fR-specific kernel routines.
.RE

.sp
.LP
A description of each of the supported commands follows:
.sp
.ne 2
.na
\fB\fBexclude:\fR <\fInamespace\fR>/<\fImodulename\fR>\fR
.ad
.sp .6
.RS 4n
Do not allow the listed loadable kernel module to be loaded. \fBexclude\fR
commands are cumulative; the list of modules to \fBexclude\fR is created by
combining every \fBexclude\fR entry in the \fBsystem\fR file.
.RE

.sp
.ne 2
.na
\fB\fBinclude:\fR <\fInamespace\fR>/<\fImodulename\fR>\fR
.ad
.sp .6
.RS 4n
Include the listed loadable kernel module. This is the system’s default, so
using \fBinclude\fR does not modify the system’s operation. \fBinclude\fR
commands are cumulative.
.RE

.sp
.ne 2
.na
\fB\fBforceload:\fR <\fInamespace\fR>/<\fImodulename\fR>\fR
.ad
.sp .6
.RS 4n
Force this kernel module to be loaded during kernel initialization. The default
action is to automatically load the kernel module when its services are first
accessed. \fBforceload\fR commands are cumulative.
.RE

.sp
.ne 2
.na
\fB\fBrootdev:\fR <\fIdevice name\fR>\fR
.ad
.sp .6
.RS 4n
Set the root device to the listed value instead of using the default root
device as supplied by the boot program.
.RE

.sp
```

```
.ne 2
.na
\fB\fBrootfs:\fR <\fIroot filesystem type\fR>\fR
.ad
.sp .6
.RS 4n
Set the root filesystem type to the listed value.
.RE

.sp
.ne 2
.na
\fB\fBmoddir:\fR <\fIfirst module path\fR>[[{:, }<\fIsecond ...\fR>]...]\fR
.ad
.sp .6
.RS 4n
Set the search path for loadable kernel modules. This command operates very
much like the \fBPATH\fR shell variable. Multiple directories to search can be
listed together, delimited either by blank spaces or colons.
.RE

.sp
.ne 2
.na
\fB\fBset\fR [\fI<module>\fR:]\fI<symbol>\fR {=, |, &} [~][-]\fI<value>\fR\fR
.ad
.sp .6
.RS 4n
Set an integer or character pointer in the kernel or in the selected kernel
module to a new value. This command is used to change kernel and module
parameters and thus modify the operation of your system. Assignment operations
are not cumulative, whereas bitwise \fBAND\fR and \fBOR\fR operations are
cumulative.
.sp
Operations that are supported for modifying integer variables are: simple
assignment, inclusive bitwise \fBOR,\fR bitwise \fBAND,\fR one’s complement,
and negation. Variables in a specific loadable module can be targeted for
modification by specifying the variable name prefixed with the kernel module
name and a colon (:) separator. Values can be specified as hexadecimal (0x10),
Octal (046), or Decimal (5).
.sp
The only operation supported for modifying character pointers is simple
assignment. Static string data such as character arrays cannot be modified
using the \fBset\fR command. Use care and ensure that the variable you are
modifying is in fact a character pointer. The \fBset\fR command is very
powerful, and will likely cause problems if used carelessly. The following
escape sequences are supported within the quoted string:
.sp
.in +2
.nf
\en     (newline)
\et     (tab)
\eb     (backspace)
.fi
.in -2
.sp

.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRA sample \fBsystem\fR file.
.sp
.LP
The following is a sample \fBsystem\fR file.
```

```
.sp
.in +2
.nf
* Force the ELF exec kernel module to be loaded during kernel
* initialization. Execution type modules are in the exec namespace.
forceload: exec/elfexec
* Change the root device to /sbus@1,f8000000/esp@0,800000/sd@3,0:a.
* You can derive root device names from /devices.
* Root device names must be the fully expanded Open Boot Prom
* device name. This command is platform and configuration specific.
* This example uses the first partition (a) of the SCSI disk at
* SCSI target 3 on the esp host adapter in slot 0 (on board)
* of the SBus of the machine.
* Adapter unit-address 3,0 at sbus unit-address 0,800000.
rootdev: /sbus@1,f8000000/esp@0,800000/sd@3,0:a
* Set the filesystem type of the root to ufs. Note that
* the equal sign can be used instead of the colon.
rootfs:ufs
* Set the search path for kernel modules to look first in
* /usr/phil/mod_test for modules, then in /kernel/modules (the
* default) if not found. Useful for testing new modules.
* Note that you can delimit your module pathnames using
* colons instead of spaces: moddir:/newmodules:/kernel/modules
moddir:/usr/phil/mod_test /kernel/modules.
* Set the configuration option {_POSIX_CHOWN_RESTRICTED} :
* This configuration option is enabled by default.
set rstchown = 1
* Disable the configuration option {_POSIX_CHOWN_RESTRICTED} :
set rstchown = 0
* Turn on debugging messages in the modules mydriver. This is useful
* during driver development.
set mydriver:debug = 1
* Bitwise AND the kernel variable "moddebug" with the
* one’s complement of the hex value 0x880, and set
* "moddebug" to this new value.
set moddebug & ~0x880
* Demonstrate the cumulative effect of the SET
* bitwise AND/OR operations by further modifying "moddebug"
* by ORing it with 0x40.
set moddebug | 0x40
.fi
.in -2
.sp

.SH SEE ALSO
.LP
\fBboot\fR(1M), \fBinit\fR(1M), \fBkernel\fR(1M)
.SH WARNINGS
.LP
Use care when modifying the \fBsystem\fR file; it modifies the operation of the
kernel. If you preserved the original \fBsystem\fR file, you can boot using
\fBboot -a\fR, which will ask you to specify the path to the saved file. This
should allow the system to boot correctly. If you cannot locate a \fBsystem\fR
file that will work, you may specify \fB/dev/null\fR. This acts as an empty
\fBsystem\fR file, and the system will attempt to boot using its default
settings.
.SH NOTES
.LP
The \fBsystem\fR files are read only once, at boot time.
```