

NAME

zonecfg – set up zone configuration

SYNOPSIS

zonecfg -z *zonename*

zonecfg -z *zonename subcommand*

zonecfg -z *zonename -f command_file*

zonecfg help

DESCRIPTION

The **zonecfg** utility creates and modifies the configuration of a zone. Zone configuration consists of a number of resources and properties.

To simplify the user interface, **zonecfg** utilizes the concept of a scope. The default scope is global.

The following synopsis of the **zonecfg** command is for interactive usage:

zonecfg -z *zonename subcommand*

Resources

The following resource types are supported:

fs file-system

inherit-pkg-dir

Directory inherited from the global zone. Software packages whose contents have been transferred into that directory are inherited in read-only mode by the non-global zone and the non-global zone's packaging database is updated to reflect those packages. Such resources are not modifiable or removable once a zone has been installed with **zoneadm**.

net Network interface.

device Device.

rctl Resource control.

attr Generic attribute.

Properties

Each resource type has one or more properties. There are also some global properties, that is, properties of the configuration as a whole, rather than of some particular resource.

The following properties are supported:

(global)
zonepath

(global)
autoboot

(global)
pool

fs dir, special, type, options

inherit-pkg-dir
dir

net address, physical

device match

rctl name, value

attr name, type, value

As for the property values which are paired with these names, they are either simple, complex, or lists. The type allowed is property specific. Simple values are strings, optionally enclosed within quotation marks. Complex values have the syntax:

```
(<name>=<value>,<name>=<value>,...)
```

where each <value> is simple, and the <name> strings are unique within a given property. Lists have the syntax:

```
[<value>,...]
```

where each <value> is either simple or complex. A list of a single value (either simple or complex) is equivalent to specifying that value without the list syntax. That is, "foo" is equivalent to "[foo]". A list can be empty (denoted by "[]").

The property types are described as follows:

global: zonepath

Path to zone's file system.

global: autoboot

Boolean indicating that a zone should be booted automatically at system boot.

global: pool

Name of the resource pool that this zone must be bound to when booted.

fs: dir, special, type, options

Values needed to determine how, where, and so forth to mount file systems. See **mount(1M)**, **mount(2)** and **vfstab(4)**.

inherit-pkg-dir: dir

The directory path.

net: address, physical

The IP address and physical interface name of the network interface.

device: match

Device name to match.

rctl: name, value

The name and priv/limit/action triple of a resource control.. See **prctl(1M)** and **rctladm(1M)**.

attr: name, type, value

The name, type and value of a generic attribute. The **type** must be one of **int**, **uint**, **boolean** or **string**, and the value must be of that type. **uint** means unsigned, that is, a non-negative integer.

The following table summarizes resources, property-names and types:

resource	property-name	type
(global)	zonepath	simple
(global)	autoboot	simple
(global)	pool	simple
fs	dir	simple
	special	simple
	type	simple
	options	list of simple
inherit-pkg-dir	dir	simple
net	address	simple
	physical	simple
device	match	simple
rctl	name	simple
	value	list of complex
attr	name	simple

type	simple
value	simple

To further specify things, the breakdown of the complex property "value" of the "rectl" resource type, it consists of three name/value pairs, the names being "priv", "limit" and "action", each of which takes a simple value. The "name" property of an "attr" resource is syntactically restricted in a similar fashion to zone names: it must begin with an alphanumeric, and can contain alphanumerics plus the hyphen (-) and period (.) characters. Attribute names beginning with "zone." are reserved for use by the system. Finally, the "autoboot" global property must have a value of "true" or "false".

OPTIONS

The following options are supported:

-f *command_file*

Specify the name of **zonecfg** command file. *command_file* is a text file of **zonecfg** subcommands, one per line.

-z *zonename*

Specify the name of a zone. Zone names are case sensitive. Zone names must begin with an alphanumeric character and can contain alphanumeric characters, the underscore (_) and the hyphen (-). The name global and all names beginning with **SUNW** are reserved and cannot be used.

SUBCOMMANDS

You can use the **add** and **select** subcommands to select a specific resource, at which point the scope changes to that resource. The **end** and **cancel** subcommands are used to complete the resource specification, at which time the scope is reverted back to global. Certain subcommands, such as **add**, **remove** and **set**, have different semantics in each scope.

Subcommands which can result in destructive actions or loss of work have a **-F** option to force the action. If input is from a terminal device, the user is prompted when appropriate if such a command is given without the **-F** option otherwise, if such a command is given without the **-F** option, the action is disallowed, with a diagnostic message written to standard error.

The following subcommands are supported:

add *resource-type* (global scope)

add *property-name property-value* (resource scope)

In the global scope, begin the specification for a given resource type. The scope is changed to that resource type.

In the resource scope, add a property of the given name with the given value. The syntax for property values varies with different property types. In general, it is a simple value or a list of simple values enclosed in square brackets, separated by commas (**[foo,bar,baz]**). See **PROPERTIES**.

cancel End the resource specification and reset scope to global. Abandons any partially specified resources. **cancel** is only applicable in the resource scope.

commit

Commit the current configuration from memory to stable storage. The configuration must be committed to be used by **zoneadm**. Until the in-memory configuration is committed, you can remove changes with the **revert** subcommand. The **commit** operation is attempted automatically upon completion of a **zonecfg** session. Since a configuration must be correct to be committed, this operation automatically does a verify.

create **[-F]** **[-b | -t template]**

Create an in-memory configuration for the specified zone. Use **create** to begin to configure a new zone. See **commit** for saving this to stable storage.

If you are overwriting an existing configuration, specify the **-F** option to force the action. Specify the **-t** *template* option to create a configuration identical to *template*, where *template* is the name of a configured zone. Use the **-b** to create a blank configuration. Without arguments, **create** applies the Sun default settings.

delete [-F]

Delete the specified configuration from memory and stable storage. This action is instantaneous, no commit is necessary. A deleted configuration cannot be reverted.

Specify the **-F** option to force the action.

end End the resource specification. This subcommand is only applicable in the resource scope. **zonecfg** checks to make sure the current resource is completely specified. If so, it is added to the in-memory configuration (see **commit** for saving this to stable storage) and the scope reverts to global. If the specification is incomplete, it issues an appropriate error message.

export [-f output-file]

Print configuration to standard output. Use the **-f** option to print the configuration to *output-file*. This option produces output in a form suitable for use in a command file.

help [usage] [subcommand] [syntax] [command-name]

Print general help or help about given topic.

info zonepath | autoboot | pool**info [resource-type [property-name=property-value]*]**

Display information about the current configuration. If resource-type is specified, displays only information about resources of the relevant type. If any property name-value pairs are specified, displays only information about resources meeting the given criteria. In the resource scope, any arguments are ignored, and info displays information about the resource which is currently being added or modified.

remove resource-type { property-name=property-value}(global scope)**remove property-name**

property-value (resource scope)" 6 In the global scope, removes the specified resource. The {} syntax means 1 or more of whatever is inside the curly braces. You must specify enough property name-value pairs for the resource to be uniquely identified.

In the resource scope, removes the given property name from the current resource.

select resource-type {property-name=property-value}

Select the resource of the given type which matches the given *property-name property-value* pair criteria, for modification. This subcommand is applicable only in the global scope. The scope is changed to that resource type. The {} syntax means 1 or more of whatever is inside the curly braces. You must specify enough *property -name property-value* pairs for the resource to be uniquely identified.

set property-name=property-value

Set a given property name to the given value. Some properties (for example, **zonepath**) are global while others are resource-specific. This subcommand is applicable in both the global and resource scopes.

verify Verify the current configuration for correctness:

- All resources have all of their required properties specified.
- A **zonepath** is specified.

revert [-F]

Revert the configuration back to the last committed state. The **-F** option can be used to force the action.

exit [-F]

Exit the **zonecfg** session. A commit is automatically attempted if needed. You can also use an EOF character to exit **zonecfg**. The **-F** option can be used to force the action.

EXAMPLES**Example 1: Creating the Environment for a New Zone**

In the following example, **zonecfg** creates the environment for a new zone. **/usr/local** is loopback mounted from the global zone into **/opt/local**. **/opt/sfw** is loopback mounted from the global zone, three logical network interfaces, and resource controls for the maximum number of light-weight processes. The example also shows how to select a given resource for modification.

```
example# zonecfg -z my-zone3
zonecfg:my-zone3> create
zonecfg:my-zone3> set zonepath=/export/home/my-zone3
zonecfg:my-zone3> set autoboot=true
zonecfg:my-zone3> add fs
zonecfg:my-zone3:fs> set dir=/usr/local
zonecfg:my-zone3:fs> set special=/opt/local
zonecfg:my-zone3:fs> set type=lofs
zonecfg:my-zone3:fs> add options [ro,nodevices]
zonecfg:my-zone3:fs> end
zonecfg:my-zone3> add inherit-pkg-dir
zonecfg:my-zone3:inherit-pkg-dir> set dir=/opt/sfw
zonecfg:my-zone3:inherit-pkg-dir> end
zonecfg:my-zone3> add net
zonecfg:my-zone3:net> set address=192.168.0.1
zonecfg:my-zone3:net> set physical=eri0
zonecfg:my-zone3:net> end
zonecfg:my-zone3> add net
zonecfg:my-zone3:net> set address=192.168.0.2
zonecfg:my-zone3:net> set physical=eri0
zonecfg:my-zone3:net> end
zonecfg:my-zone3> add net
zonecfg:my-zone3:net> set address=192.168.0.3
zonecfg:my-zone3:net> set physical=eri0
zonecfg:my-zone3:net> end
zonecfg:my-zone3> add rctl
zonecfg:my-zone3:rctl> set name=zone.cpu-shares
zonecfg:my-zone3:rctl> add value
(priv=privileged,limit=5,action=none)
zonecfg:my-zone3:rctl> end
zonecfg:my-zone3> select rctl name=zone.cpu-shares
zonecfg:my-zone3:rctl> remove value
(priv=privileged,limit=5,action=none)
zonecfg:my-zone3:rctl> add value
(priv=privileged,limit=10,action=deny)
zonecfg:my-zone3:rctl> end
zonecfg:my-zone3> exit
example#
```

Example 2: Associating a Zone with a Resource Pool

The following example shows how to associate an existing zone with an existing resource pool:

```
$ zonecfg -z myzone
zonecfg:myzone> set pool=mypool
zonecfg:myzone> exit
```

For more information about resource pools, see **pooladm(1M)** and **poolcfg(1M)**.

EXIT STATUS

The following exit values are returned:

- 0** Successful completion.
- 1** An error occurred.
- 2** Invalid usage.

ATTRIBUTES

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWzoneu
Interface Stability	Evolving

SEE ALSO

zlogin(1), **mount(1M)**, **pooladm(1M)**, **poolcfg(1M)**, **prctl(1M)**, **rctladm(1M)**, **zoneadm(1M)**, **vfstab(4)**, **attributes(5)**