![Sun Microsystems logo]

Sun Java System Administration Server 5.2 2005Q1 Man Page Reference

Adobe PostScript™

041125@10536

Contents

Preface

Both novice users and those familar with the Solaris Operating System can use online man pages to obtain information about their SPARC™ or x86 based system and its features.

Note – In this document, the term "x86" refers to the Intel 32-bit family of microprocessor chips and compatible microprocessor chips made by AMD.

A man page is intended to answer concisely the question "What does it do?" The man pages in general comprise a reference manual. They are not intended to be a tutorial.

Note – Sun Cluster software runs on two platforms, SPARC and x86. The information in this document pertains to both platforms unless otherwise specified in a special chapter, section, note, bulleted item, figure, table, or example.

Overview

The following contains a brief description of each man page section and the information it references:

- Section 1 describes, in alphabetical order, commands available with the operating system.
- Section 1M describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes.
- Section 2 describes all of the system calls. Most of these calls have one or more error returns. An error condition is indicated by an otherwise impossible returned value.

- Section 3 describes functions found in various libraries, other than those functions that directly invoke UNIX system primitives, which are described in Section 2.
- Section 4 outlines the formats of various files. The C structure declarations for the file formats are given where applicable.
- Section 5 contains miscellaneous documentation such as character-set tables.
- Section 6 contains available games and demos.
- Section 7 describes various special files that refer to specific hardware peripherals and device drivers. STREAMS software drivers, modules and the STREAMS-generic set of system calls are also described.
- Section 9 provides reference information needed to write device drivers in the kernel environment. It describes two device driver interface specifications: the Device Driver Interface (DDI) and the Driver/Kernel Interface (DKI).
- Section 9E describes the DDI/DKI, DDI-only, and DKI-only entry-point routines a developer can include in a device driver.
- Section 9F describes the kernel functions available for use by device drivers.
- Section 9S describes the data structures used by drivers to share information between the driver and the kernel.

Below is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. For example, if there are no bugs to report, there is no BUGS section. See the intro pages for more information and detail about each section, and man(1) for more information about man pages in general.

NAME   This section gives the names of the commands or functions documented, followed by a brief description of what they do.

SYNOPSIS   This section shows the syntax of commands or functions. When a command or file does not exist in the standard path, its full path name is shown. Options and arguments are alphabetized, with single letter arguments first, and options with arguments next, unless a different argument order is required.

                        The following special characters are used in this section:

                        [ ]   Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.

. . .	Ellipses. Several values can be provided for the previous argument, or the previous argument can be specified multiple times, for example, "`filename ...`".
|	Separator. Only one of the arguments separated by this character can be specified at a time.
{ }	Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.
PROTOCOL	This section occurs only in subsection 3R to indicate the protocol description file.
DESCRIPTION	This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, and functions are described under USAGE.
IOCTL	This section appears on pages in Section 7 only. Only the device class that supplies appropriate parameters to the `ioctl`(2) system call is called `ioctl` and generates its own heading. `ioctl` calls for a specific device are listed alphabetically (on the man page for that specific device). `ioctl` calls are used for a particular class of devices all of which have an io ending, such as `mtio`(7I).
OPTIONS	This secton lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.
OPERANDS	This section lists the command operands and describes how they affect the actions of the command.
OUTPUT	This section describes the output – standard output, standard error, or output files – generated by the command.

RETURN VALUES	If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or –1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions declared void do not return values, so they are not discussed in RETURN VALUES.
ERRORS	On failure, most functions place an error code in the global variable `errno` indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.
USAGE	This section lists special rules, features, and commands that require in-depth explanations. The subsections listed here are used to explain built-in functionality:
	Commands Modifiers Variables Expressions Input Grammar
EXAMPLES	This section provides examples of usage or of how to use a command or function. Wherever possible a complete example including command-line entry and machine response is shown. Whenever an example is given, the prompt is shown as `example%`, or if the user must be superuser, `example#`. Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS, and USAGE sections.
ENVIRONMENT VARIABLES	This section lists any environment variables that the command or function affects, followed by a brief description of the effect.

EXIT STATUS	This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion, and values other than zero for various error conditions.
FILES	This section lists all file names referred to by the man page, files of interest, and files created or required by commands. Each is followed by a descriptive summary or explanation.
ATTRIBUTES	This section lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. See `attributes`(5) for more information.
SEE ALSO	This section lists references to other man pages, in-house documentation, and outside publications.
DIAGNOSTICS	This section lists diagnostic messages with a brief explanation of the condition causing the error.
WARNINGS	This section lists warnings about special conditions which could seriously affect your working conditions. This is not a list of diagnostics.
NOTES	This section lists additional information that does not belong anywhere else on the page. It takes the form of an aside to the user, covering points of special interest. Critical information is never covered here.
BUGS	This section describes known bugs and, wherever possible, suggests workarounds.

AS 1m

NAME	magt – start the Directory Server master agent
SYNOPSIS	**mpsadmserver magt** [-p *portnum*] *CONFIG parameters_file*
	/*ServerRoot*/**plugins/snmp/magt/magt** [-p *portnum*] *CONFIG parameters_file*
DESCRIPTION	The magt command starts the SNMP master agent, which runs as a daemon process and listens by default to the User Datagram Protocol (UDP) port 161 for SNMP requests. The master agent waits for subagents to register, sends requests to and receives responses from subagents, and traps notifications from subagents.
	When it is invoked, the master agent reads its configuration file, determines the subtree Object Identifier (OID) for various subagents, and populates its own Management Information Bases (MIBs).
	To run the master agent on a port other than 161, modify the *configuration_file* file before starting the agent from the command line.
OPTIONS	The following options are supported:

-p *portnum*	Specifies the port number to which clients must connect to use the magt snmp command.
CONFIG	The configuration file is located in *ServerRoot*/plugins/snmp/magt. It defines the community and the manager that the master agent works with. Specify the manager value as a valid system name or an IP address. An invalid manager name in the CONFIG file causes the master agent start-up to fail. For more information, see the *Administration Server Administration Guide*.
parameters_file	The path to the parameters file. This file contains information from the MIB-II system group, including system location and contact information. If it does not already exist, starting the master agent for the first time will create it. An invalid manager name in the *parameters_file* prevents the master agent from starting.

EXIT STATUS	The following exit values are returned:
	0 Successful completion.
	1 An error occurred.
ATTRIBUTES	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWdsvu
Stability Level	Committed Private

SEE ALSO	mpsadmserver(1M), sagt(1M)

NAME | mpsadmconfig – configure Administration Server

SYNOPSIS | **/usr/sbin/mpsadmconfig** [options] *subcommand* [args]
[*subcommand*-args...]

DESCRIPTION | The mpsadmconfig command provides a single command-line tool to configure Administration Server.

OPTIONS | The following options are supported:

-con[tinueOnError]
Finishes any remaining tasks, even when an error occurs. The default behavior is to quit when any task fails, without running the remaining tasks.

-enc[ryption]
Uses encrypted HTTP (HTTPS) to connect to the server. The default protocol is HTTP.

-h[elp] [*task*]
Displays general usage information. To display the usage information for a particular task, specify the task as the argument to -h without the leading hyphen, for example, -hrestart.

-i[nputFile] *filename*
Reads options and tasks from the specified file. You can specify additional options on the command line. If an option is present on the command line and in the specified file, the command-line settings are used. If the -inputFile option is present in the specified file, it is ignored to prevent mpsadmconfig from reading multiple sets of options.

-ser[ver] [*host*] : *port*
Connects to the server on the specified host and port. If no host is specified, the local host is used. The server port number (preceded by a colon) is required.

-u[ser] [*uid*] : [*pwd*]
Connects to the server using the specified username and password. If a user name is not specified, you are prompted for the current user's password. The password appears on screen when it is typed, so if security is a concern, use the -inputFile option and list the user name and password in a file with suitable permissions. Note that if the -user option is specified, then, at a minimum, the colon must be specified. If the -user option is not specified, the user is prompted for both the user name and password.

-verb[ose] [*0-9*]
Sets the level of screen output (9=full output, 0=no output). The default level is 5.

-vers[ion]
Displays the version and copyright information.

The [options] can be specified using the terse single character form such as -u if applicable, or they can be specified using the longer but more descriptive form such as -user. The complete option name need not be specified. For example, -us will work just as well as -user. The option name must be specified so that it is not ambiguous among other [options] or the tasks described in the following section.

Note that [options] are not case sensitive. Thus, -USER and -User are both accepted as the -user option.

Multiple tasks can be run from the same invocation of mpsadmconfig. Tasks specified in an input file are run first. Tasks are run in the order specified in the input file and on the command line.

TASKS   The following tasks are supported:

-countA[ccessLogEntries]
Counts the number of entries in the access log file. Run this task before
-viewAccessLogEntries to determine the number of entries in the access log.

-viewA[ccessLogEntries]
Lets you view the specified entries in the access log file.

-countE[rrorLogEntries]
Counts the number of entries in the error log file. Run this task prior to
-viewErrorLogEntries to determine the number of entries in the error log.

-viewE[rrorLogEntries]
Lets you view the specified entries in the error log file.

-enableE[ndUserAccess]
End user access to the Administration Server has been removed from
Administration Server 5.2.

-disableE[ndUserAccess]
End user access to the Administration Server has been removed from
Administration Server 5.2.

-getAc[cessLog]
Retrieves the path for the access log file for this instance of Administration Server.

-setAc[cessLog]
Specifies the path for the access log file for this instance of Administration Server.

-getAdd[ressess]
Lets you view the IP addresses from which connections are allowed.

-setAdd[ressess]
Specifies the IP addresses from which connections are allowed.

-getAdminUI[D]
Retrieves the Administration Server administrator's user name.

-setAdminUI[D]
Specifies the Administration Server administrator's user name.

-setAdminP[wd]
Specifies the Administration Server administrator's password.

-getAdminUs[ers]
Retrieves the path of the adminusers file.

`-setAdminUs[ers]`
  Specifies the path of the `adminusers` file.

`-getCa[cheLifetime]`
  Displays the period of time for which a user authentication is cached.

`-setCa[cheLifetime]`
  Specifies the period of time to cache a user authentication.

`-getCl[assname]`
  Retrieves the Java classname for this instance of Administration Server.

`-setCl[assname]`
  Enables you to set a new Java classname for this instance of Administration Server.

`-getDe[faultAcceptLanguage]`
  Displays the default language for this instance of Administration Server.

`-setDe[faultAcceptLanguage]`
  Specifies the default language for this instance of Administration Server.

`-getDS[Config]`
  Retrieves the current LDAP server host, port, and base DN, and identifies whether the LDAP server is running SSL.

`-setDS[Config]`
  Specifies the LDAP server host, port, and base DN, and specifies whether the LDAP server is running SSL.

`-getE[rrorLog]`
  Retrieves the path for the server error log file.

`-setE[rrorLog]`
  Specifies the path for the server error log file.

`-getH[osts]`
  Lets you view the host names from which connections are allowed.

`-setH[osts]`
  Specifies the host names from which connections are allowed.

`-getO[neACLDir]`
  Retrieves the path for the ACL folder.

`-setO[neACLDir]`
  Specifies the path for the ACL folder.

`-getPo[rt]`
  Lets you view the port number that this instance of Administration Server is using.

`-setPo[rt]`
  Specifies the port number that this instance of Administration Server should use.

`-getSe[rverAddress]`
  Retrieves the IP address of this instance of Administration Server.

-setSe[rverAddress]
    Specifies the IP address that this instance of Administration Server should use.

-getSy[ystemUser]
    Retrieves the UNIX user name that this instance of Administration Server runs as.

-setSy[ystemUser]
    Specifies the UNIX user name that this instance of Administration Server should
    run as.

-getU[GDSConfig]
    Retrieves the current user and group LDAP server information, including the host,
    port, base DN, and authentication DN.

-setU[GDSConfig]
    Specifies the host, port, base DN, authentication DN, and authentication password
    for the instance of Directory Server containing the user and group directory. You
    can invoke -setUGDSConfig either with or without arguments. If you invoke this
    task without any arguments, the Directory Server configuration is reset to the
    installation defaults.

-r[estart]
    Restarts this instance of Administration Server.

-st[op]
    Stops this instance of Administration Server.

The task to be done can be specified using the abbreviated name such as -r for
restarting the server or using the complete name -restart. The task name must be
specified so that it is not ambiguous among other tasks or the [options] described
previously. The task keywords are also case-insensitive, so both -RESTART and
-Restart are accepted as the -restart task.

Multiple tasks can be run from the same invocation of mpsadmconfig. Tasks
specified in an input file are run first. Tasks are run in the order specified in the input
file and on the command line.

EXAMPLES   **EXAMPLE 1** Changing the port number

The following example changes the port number to 33333 and restarts Administration
Server. Note that the verbose level will be set to 5.

```
% mpsadmconfig -server eastcoast.example.com:22222 -user phlee:password
        -verbose 5 -setPort 33333 -restart
```

EXAMPLE 2 Retrieving host information

The following example retrieves the hosts from which connections are allowed. Note
that the verbose level will be set to 9.

```
% mpsadmconfig -server eastcoast.example.com:33333 -u phlee:password
    -verb 9 -geth
```

EXAMPLE 3 Displaying the help

The following example displays the help information for restarting Administration
Server.

```
% mpsadmconfig -h restart
```

EXIT STATUS The following exit values are returned:

0   Successful completion.

1   An error occurred.

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWasvu
Stability Level	Obsolete

NOTES End user access has been removed from Administration Server 5.2.

SEE ALSO mpsadmserver(1m), mpsconsole(1m), directoryserver(1m)

NAME | mpsadmserver – front end for Administration Server

SYNOPSIS | **/usr/sbin/mpsadmserver** help [subcommand]

/usr/sbin/mpsadmserver { configure [-f *configuration_file*]
    | unconfigure -f *configuration_file* }

/usr/sbin/mpsadmserver {start | stop | restart | startconsole}

/usr/sbin/mpsadmserver { magt | sagt } {options-and-arguments}

/usr/sbin/mpsadmserver {admin_ip} {options-and-arguments}

/usr/sbin/mpsadmserver {sync_cds} [5.2]
    [-f *{absolute path to credential's file}*] [-l *{logfile absolute path}*]

DESCRIPTION | The mpsadmserver command is a comprehensive front end to the utility programs provided by Administration Server.

SUBCOMMANDS | The following subcommands are supported:

admin_ip *args*
    Deprecated subcommand. Use the Server Console to update the local Administration Server configuration file and configuration directory instead. When your computer system's IP address changes, you must change to the new IP address, as failing to do so would prevent you from being able to start Administration Server.

configure *args*
    Configures Administration Server. This subcommand creates a basic configuration for Administration Server that is used to manage a set of Sun Java System products, referenced in a specific configuration directory.

    The configure subcommand has two modes of operation. You can invoke it with a GUI-based interaction to gather input. Alternatively, you can provide input in a configuration file using the -f option.

    The configure subcommand supports the following option:

    -f *configuration_file*

    Specifies the configuration file for silent configuration.

help [*subcommand*]
    Display mpsadmserver usage message or subcommand specific usage message. To obtain a usage message for the mpsadmserver command itself, type mpsadmserver help and to obtain a usage message for a mpsadmserver subcommand, type mpsadmserver *subcommand* help.

magt *args*
    Starts the SNMP master agent.

restart
    Restarts Administration Server.

sagt *args*

Starts the proxy SNMP agent. For more information, see the *Administration Server Administration Guide*.

start

Starts Administration Server.

startconsole

Starts Sun Java System Server Console which enables you to manage certain installed Sun Java System servers. However, prefer use of the mpsconsole(1m) command.

stop

Stops Administration Server.

sync_cds *args*

Synchronizes upgraded Administration Server settings with those stored in the configuration directory. The configuration directory server must be available for this step to complete properly.

unconfigure *args*

Unconfigures Administration Server. This subcommand stops Administration Server and the SNMP master agent and removes all changes created by the configure command.

The unconfigure subcommand has two modes of operation. You can invoke it with a GUI-based interaction to gather input. Alternatively, you can provide input in a configuration file using the -f option.

The unconfigure subcommand supports the following option:

-f *unconfiguration_file*

Specifies the configuration file for silent unconfiguration.

EXAMPLES   **EXAMPLE 1** Configuring Administration Server

The following command configures Administration Server using a configuration file:

```
example% /usr/sbin/mpsadmserver configure -f /usr/admserv/setup_data/typical.ins
```

EXIT STATUS   The following exit values are returned:

0   Successful completion.

1   An error occurred.

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWasvu

mpsadmserver(1M)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Stability Level	Stable

SEE ALSO   `directoryserver(1m)`, `mpsconsole(1m)`, `mpsadmconfig(1m)`

Administration Server Administration Guide

Directory Server Administration Guide

Directory Server Administration Reference

NAME	mpsconsole – start Sun Java System Server Console
SYNOPSIS	**mpsconsole** [-a *adminURL*] [-f *filename*] [-h] [-l *languageCode*] [-u *userID*] [-w *password*] [-x *extraOptions*]
DESCRIPTION	The mpsconsole command starts Sun Java System Server Console, a stand-alone Java application that works in conjunction with an instance of Directory Server and an instance of Administration Server.
OPTIONS	The following options are supported:

-a *adminURL*	Specifies a base URL for the instance of Administration Server that you want to log in to.
-f *filename*	Captures errors and system messages to the specified file.
-h	Displays the help for the mpsconsole command.
-l *languageCode*	Specifies the language that Server Console should use. Possible values for *languageCode* are en, ja, and fr.
-u *userID*	The user ID with which to log into Server Console.
-w *password*	The password for the user entered with the -u option.
-x *extraOptions*	Specifies that you want to use additional options.
	Possible values for *extraOptions* are nologo and nowinpos. If you specify the nologo option, the console splash screen is not displayed. If you specify the nowinpos option, the console window is placed in the upper left corner of the screen. To specify both options, separate them with a comma, for example:

```
% mpsconsole -x nologo,nowinpos
```

EXAMPLES

EXAMPLE 1 Capturing errors and messages to a file called system.out

% mpsconsole -f system.out

EXAMPLE 2 Logging into a base URL

% mpsconsole -a http://france.example.com:1389

EXAMPLE 3 Logging in with a user ID and password

% mpsconsole -u bjensen -w mypword

EXIT STATUS

The following exit values are returned:

0	Successful completion.
1	An error occurred.

mpsconsole(1M)

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWasvc
Stability Level	Stable

SEE ALSO mpsadmserver(1m), mpsadmconfig(1m), directoryserver(1m)

NAME | restart-admin – restart Administration Server

SYNOPSIS | **mpsadmserver restart-admin**

*ServerRoot***/restart-admin**

DESCRIPTION | The restart-admin command restarts Administration Server.

OPTIONS | There are no options for the restart-admin command.

EXIT STATUS | The following exit values are returned:

0 Server successfully restarted.

1 Server could not abe restarted.

ATTRIBUTES | See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWdsvu
Stability Level	Committed Private

SEE ALSO | mpsadmserver(1M), start-admin(1M), stop-admin(1M)

NAME	sagt – start the SNMP proxy agent
SYNOPSIS	**mpsadmserver sagt** [-c *CONFIG*]
	*ServerRoot***/plugins/snmp/sagt/sagt** [-c *CONFIG*]
DESCRIPTION	The sagt command starts the SNMP proxy agent.
OPTIONS	The following option is supported:

-c *CONFIG*   Specifies a file including the SNMP port on which the daemon listens, and the MIB trees and traps that the proxy SNMP agent forwards. By default, the file is located in *ServerRoot*/plugins/snmp/sagt

EXIT STATUS	The following exit values are returned:

0   Successful completion.

1   An error occurred.

FILES	The CONFIG file which specifies the SNMP port on which the daemon listens, and the MIB trees and traps that the proxy SNMP agent forwards. This file is located in *ServerRoot*/plugins/snmp/sagt.

0   Successful completion.

1   An error occurred.

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWdsvu
Stability Level	Committed Private

SEE ALSO	mpsadmserver(1M),magt(1M)

NAME	start-admin – start Administration Server
SYNOPSIS	**mpsadmserver start-admin**
	*ServerRoot***/start-admin**
DESCRIPTION	The start-admin command starts Administration Server.
OPTIONS	There are no options for the start-admin command.
EXIT STATUS	The following exit values are returned:

0   Server started successfully.

1   Server could not be started.

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWdsvu
Stability Level	Committed Private

SEE ALSO   mpsadmserver(1M), stop-admin(1M), restart-admin(1M)

NAME	startconsole – start the console
SYNOPSIS	**mpsconsole startconsole**

*ServerRoot***/startconsole**

DESCRIPTION The `startconsole` command starts the console, a stand-alone Java application that works in conjunction with an instance of Directory Server and an instance of Administration Server.

OPTIONS The following options are supported:

-a *adminURL*	Specifies a base URL for the instance of Administration Server that you want to log in to.
-f *filename*	Captures errors and system messages to the specified file.
-h	Displays the help for the `startconsole` command.
-l *languageCode*	Specifies the language that Server Console should use. Possible values for *languageCode* are en, ja, and fr.
-u *userID*	The user ID with which to log into Server Console.
-w *password*	The password for the user entered with the -u option.
-x *extraOptions*	Specifies that you want to use additional options.
	Possible values for *extraOptions* are `nologo` and `nowinpos`. If you specify the `nologo` option, the console splash screen is not displayed. If you specify the `nowinpos` option, the console window is placed in the upper left corner of the screen. To specify both options, separate them with a comma, for example:

% startconsole -x nologo,nowinpos

EXIT STATUS The following exit values are returned:

0	Console started successfully.
1	Console could not be started.

ATTRIBUTES See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWdsvu
Stability Level	Committed Private

SEE ALSO mpsconsole(1M)

NAME	stop-admin – stop Administration Server
SYNOPSIS	**mpsadmserver stop-admin**
	*ServerRoot***/ stop-admin**
DESCRIPTION	The `stop-admin` command stops the Administration Server
OPTIONS	There are no options for the `stop-admin` command.
EXIT STATUS	The following exit values are returned:

0   Server stopped successfully.

1   Server could not be stopped.

ATTRIBUTES See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWdsvu
Stability Level	Committed Private

SEE ALSO mpsadmserver(1M), start-admin(1M), restart-admin(1M)

stop-admin(1M)