

***SAGE Enterprise Backup  
White Paper***



# Table of Contents

---

Notice .....	i
List of Figures .....	v
Introduction .....	1
The Purpose of This Manual .....	1
Related Documents .....	1
Notation .....	1
Problem Reporting Instructions .....	3
Introduction to SAGE .....	5
SAGE Components .....	5
SAGE Processes .....	7
SAGE System Configuration .....	7
Starting SAGE .....	10
SAGE User Commands .....	10
SAGE Command Reference .....	10
sage_control .....	11
sage_create_db .....	12
sage_dump_db .....	13
sage_find_file .....	14
sage_monitor .....	15
sage_notify .....	15
sage_restore .....	16
sage_run_command_macro .....	17
sage_save .....	17
sage_schedule .....	19
sage_show_log .....	20
sage_show_schedule .....	20
sage_supersync_admin .....	21



## List of Figures

---

Figure 1: SAGE Components .....	5
Figure 2: SAGE Processes.....	7
Figure 3: SAGE Command Reference Table .....	11
Figure 4: sage_create_db Commands .....	13
Figure 5: sage_dump_db Commands .....	14
Figure 6: sage_show_schedule Calendar.....	21



# Introduction

---

## The Purpose of This White Paper

This white paper explains the configuration and usage of Storage Admin General Executive (SAGE).

*Note: Throughout this document Stratus Continuum and V Series computers are collectively referred to as the "Stratus Host."*

*Note: For support of other Stratus machines or operating systems contact DRA.*

## Related Documents

Stratus manuals in the VOS System Administration series contain additional information related to VOS disk operations:

*VOS System Administration: Disk and Tape Administration (R284)*

*VOS System Administration: Configuring a System (R287)*

*VOS System Administration: Backing Up and Restoring Data (R285)*

*VOS System Administration: Starting Up and Shutting Down a Module or System (R282)*

*VOS System Administration: VOS Module Maintenance Guide (R343)*

*VOS System Administration: Registration and Security (R283)*

*VOS Continuum Operation and Maintenance: 600 and 1200 Series (R396) or,*

*ftServer V-Series Systems: Operation and Maintenance (R557)*

*VOS Continuum 600 and 1200 Series with PA-8000: Operation and Maintenance Guide (R445)*

The following Stratus manuals also contain important information related to VOS disk operations:

*VOS Commands Reference Manual (R098)*

*VOS Codes and Messages Reference Manual (R132)*

*VOS Reference Manual (R002)*

*Continuum Site Planning: 600 and 1200 Series (R391) or,*

*ftServer V-Series Systems: Site Planning (R543)*

## Notation

DRA has adopted several conventions which are intended to assist you in interpreting and understanding the content of this document.

*Notes alert you to special features or procedures.*

`Courier` (typewriter style) font is used for names of computer files, directories, programs, subroutines, and subsystems.

`Courier` (typewriter style) font is also used to represent text that would appear on your display screen or on a printer. This text is referred to as literal text:

```
setup_disk 4/3/1/4
```

Slanted *Courier* font (italics) is used to represent general terms that are to be replaced by literal values which you choose. These are called arguments, which are either required or optional.

Required arguments are italicized and are not surrounded by any brackets or braces. You must always supply a value for a required argument. In the following example, you must supply an actual value to replace the argument *user\_name*:

```
login user_name
```

Optional arguments are arguments for which VOS 1.) does not need a value to execute the command or 2.) for which you may choose between one or more values.

An optional argument which is italicized and surrounded by square brackets [] may take a single value, or you may omit it entirely. In the following example *description\_path* is optional.

```
create_table devices [-description_path >system>configuration]
```

An optional argument which is italicized and surrounded by braces {} may take one or more values. At least one value must be specified. In the following example *source\_path\_names* is optional, but at least one must be specified, and more than one is allowed. (*port\_name* is required.)

```
save port_name {source_path_names}
```

Single keystrokes are indicated by opposing angle brackets, e.g., <ALT>.

Cursor keys are shown as arrows, →, ↓, ←, and ↑.

Function key sequences are indicated by F followed by the function key number such as F6. This sequence is surrounded by opposing angle brackets, e.g., < F10 >, because it is a single keystroke.

# SAGE Enterprise Backup

---

## Introduction to SAGE

DRA's SAGE Enterprise Backup incorporates VOS systems into your enterprise backup solution. With SAGE, Stratus VOS administrators no longer have to manage VOS saves and restores separately.

SAGE allows VOS administrators to automate the save and restore processes. This allows VOS personnel to perform standard operations with no ongoing impact on network backup operations personnel. In turn, network backup operations personnel can manage VOS backups and restores without special knowledge of the operating system.

SAGE integrates the standard VOS save/restore process into an enterprise backup solution using a nearline storage device included with the SAGE purchase. VOS saves are done to "containers" (disk images) on the nearline storage device that appear to VOS as logical disk drives. Once the VOS save is complete the SAGE software schedules an enterprise backup of the container using the installed enterprise backup software. Restores are performed in a similar manner.

For backups, SAGE on VOS requests a disk on the nearline storage device be made available, then performs a VOS save. When the VOS save is complete, SAGE directs the enterprise backup system to move the data on the the nearline storage device to a RAID or tape silo.

For a restore, SAGE on VOS requests the enterprise backup system to restore the needed data from the RAID or tape silo to one of the the nearline storage device's shared disks. Then, when that restore is complete, the device notifies VOS so it can mount the disk and perform the VOS restore to the desired destination.

## SAGE Components

SAGE is a collection of software components that work with a nearline storage device included with the SAGE purchase to automate the enterprise backup of Stratus data. Some components run on the Stratus VOS module while others run on the nearline storage device's processor as illustrated in Figure 1.

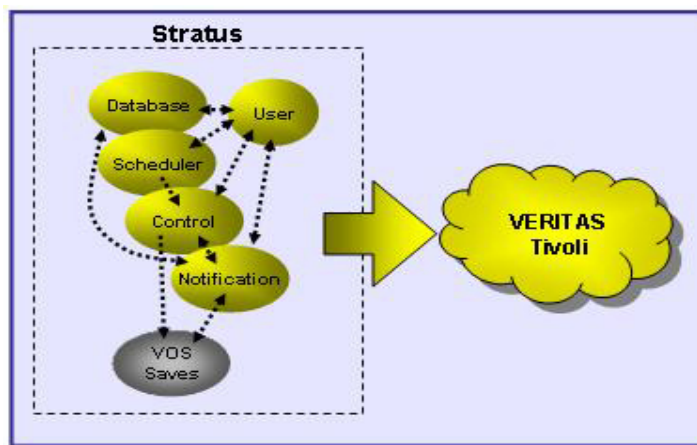


Figure 1. SAGE Components

The SAGE command line interface on VOS allows administrators a simple way to schedule saves and restores, and to query the SAGE database. Using SAGE, the system administrator does not issue a command to run a save or restore. Instead, the administrator schedules the save or restore to run now or in the future, and the SAGE system performs the operation on behalf of the administrator.

The scheduler module on VOS (`sage_schedule`) is primarily responsible for the execution of scheduled saves and restores. The scheduler implements most of the database, scheduler, and control components of the SAGE system. It allows end users to schedule VOS saves on a onetime or recurrent basis. Users can select specific files, directories, or disks to be saved.

Restores can also be scheduled with the flexibility to restore files to the same disk from which they came, or to a different disk. The scheduler module manages the mounting and dismounting of disks and the running of the save and restore commands through the use of additional modules when needed. In addition, it instructs the nearline storage device modules to run network saves and restores when necessary.

The VOS-resident database of files and backups tracks information such as what files are on a given backup, when the backup was created, and when it is to be deleted. It also tracks all saves and restores executed by the SAGE system. All saves and restores in the SAGE system are identified by a common numeric identifier, as well as by the date and time they occurred.

The notification module on VOS (`sage_notify`) is responsible for handling messages returned to VOS from the nearline storage device. These messages report the successful completion of a save or restore, or they report any error that may have occurred. It also handles the database updates necessary when the network operation completes.

On the nearline storage device, the request server module (`sage_request_server`) handles messages from the VOS modules, starts and monitors network save and restore operations, and returns result status messages to the VOS system.

In addition, the monitor process (`sage_monitor`) checks that the SAGE software is operating correctly, and logs errors and restarts processes if necessary. The control process (`sage_control`) starts and stops the system, and reports status to the user. Other programs provide database information for users.

## SAGE Processes

When SAGE is running, at least three processes are active at all times on the VOS system. These processes are named `sage_monitor`, `sage_notify`, and `sage_schedule`. In turn, these processes start others (`sage_mount`, `sage_dismount`, `sage_run_command_macro` and the actual VOS save and restore processes) as needed. Figure 7 shows the SAGE process structure. In addition, the `sage_request_server` program runs on the nearline storage device to manage and respond to save and restore requests from the VOS system.

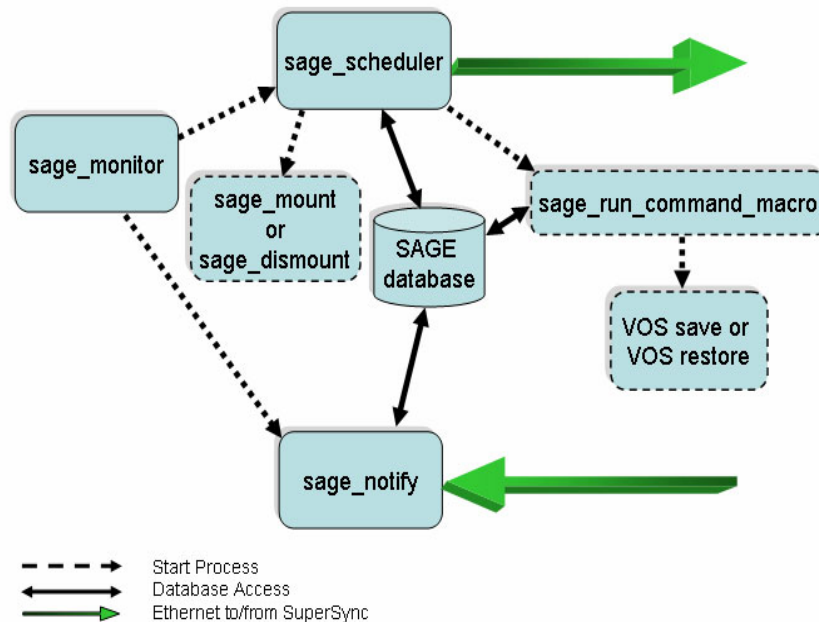


Figure 2. SAGE Processes

## SAGE System Configuration

When SAGE is installed on a system, one directory must be created on the master disk. This directory is named SAGE (all capital letters). In most installations, this directory will contain the full SAGE system; software, database, and configuration information. Some installations, however, may wish to install portions of SAGE on other disks. This is managed through a configuration file.

The SAGE system depends on the configuration file to set system-dependent parameters. The file is named `(master_disk)>SAGE>sage_config.table`, and it is created with the VOS `create_table` command, a familiar tool to system administrators. The table must be located in `(master_disk)>SAGE`. Other files and directories mentioned below can be located anywhere on the system as long as `sage_config.table` points to their correct location.

Input to the `create_table` command consists of two files: `sage_config.tin` and `sage_config.dd`. These files are usually found in `(master_disk)>SAGE>cfg`. Only `sage_config.tin` should be modified at installation.

The content of the file is:

```
/* Sample Sage configuration file. Use create_table to convert this,
   along with sage_config.dd, into a VOS table file.

/ =database_directory          '(master_disk)>SAGE>database'
  =log_directory              '(master_disk)>SAGE>logs'
  =temp_directory            '(master_disk)>SAGE>temp'
  =macro_directory           '(master_disk)>SAGE>macros'
  =supersync_ip_addr         192.168.2.2
  =hw_path                   '4/4'
  =default_expiration        '1 month'
  =purge_time                '03:00'
  =receive_port              52005
  =send_port                 52006
  =check_interval            60
  =keepalive_interval        5
  =server_timeout            60
  =no_of_simultaneous_saves  1
  =no_of_simultaneous_restores 1
  =concurrent_saves_and_restores 0
  =sage_priority             -1
  =restore_priority          -1
  =save_priority             -1
  =use_demo_mode             1
```

The entries `database_directory` through `hw_path` are required. The four directory parameters are the paths of the files used by the SAGE system. The `database_directory` contains the actual data files managed by the system. The `log_directory` contains the output logs written by the system. The `temp_directory` is used as a place to write the command macros generated by the system. And, the `macro_directory` contains optional command macros that will be run before or after SAGE save and restore operations. The development of these macros will be a joint effort of the customer and DRA.

The `supersync_ip_address` is the address of the nearline storage device used for the backup. The `hw_path` is the prefix of the VOS hardware path of the disks on the nearline storage device. On Continuum SCSI, this is the slot and port to which the nearline storage device is connected. On Continuum and V-Series Fibre Channel, this is the slot, card, and port on the card to which the nearline storage device is connected. Use `analyze_system`, then `list_boards` to help in determining this.

The `default_expiration` field specifies the length of time a save record is valid by default. This can be overridden by an option to the save command. If the expiration is 1 month, all records of the save will be removed from the database one month after the save is complete.

The `purge_time` field specifies the time of day SAGE searches for records that have expired. The default is 02:00 or 2:00 AM local time, but any time can be specified in 24-hour format. When the specified time of day is reached, SAGE examines the database for records that have reached their expiration date, and the records are removed.

The remaining entries are optional, to be used only if default values must be changed. The entries `send_port` and `receive_port` are used to establish connections between the VOS system and the nearline storage device, and default to 52005 and 52006. These ports are named from the point of view of the device server: `sage_schedule` sends messages on `receive_port` and `sage_notify` receives messages on `send_port`.

The `check_interval` parameter specifies the number of seconds SAGE sleeps between checks of the SAGE database. If this value is set to 60, for example, the program awakens every 60 seconds to check for save or restore activity that must be processed.

The `keepalive_interval` specifies the maximum number of minutes between check messages sent by VOS to the nearline storage device to verify the device is running correctly.

The `server_timeout` value specifies a base value for the number of minutes SAGE will wait for the completion of a Netbackup save or restore. This value is multiplied by the number of member disks being saved or restored. If, for example, the `server_timeout` is set to 60 minutes, and SAGE is waiting for the completion of a Netbackup of a 4-member disk, the program will wait 4 x 60 minutes, or 240 minutes, for completion.

The entries `no_of_simultaneous_saves` and `no_of_simultaneous_restores` allow system administrators to limit the number of saves and/or restores allowed to run at the same time. These entries take an integer value.

The entry `concurrent_saves_and_restores` allows or prevents a save from running at the same time as a restore and, concurrently, on same disks it prevents two saves from writing to the same disk at the same time. An entry of 1 allows concurrent saves and restores, while an entry of 0 prevents them.

The three priority options (`sage_priority`, `restore_priority`, and `save_priority`) allow the user to control the system priority of the running SAGE processes. The startup program `sage_control` starts the `sage_monitor`, `sage_notify`, and `sage_schedule` processes at the priority level specified by `sage_priority`. In addition, `sage_run_command_macro`, `sage_mount` and `sage_dismount` run at the same level. The actual VOS save and restore commands run at `save_priority` and `restore_priority` levels.

VOS priorities run from 0 to 9, with jobs running at priority 0 generally being given the fewest system resources and 9 the most. Normal jobs start at priority 5. The -1 shown here indicates the new process is to use the same priority as the process that started it.

The `-use_demo_mode` option allows SAGE to use CAC “demo mode” when dismounting disks. On VOS 15 v-series systems, a Stratus CAC call is entered every time a disk is dismounted on a VOS system. Using demo mode prevents these calls from being logged.

**WARNING:** Note that the use of demo mode puts your site at risk that a system problem may not be logged to Stratus. The risk is minimal because SAGE keeps the system in demo mode for less than 30 seconds. However, the risk is present.

After the `sage_config.tin` file is modified for the site, the `.table` file must be created and installed in the `>SAGE` directory. The command macro `sage_create_table.cm`, found in `>SAGE>cfg` performs these steps for you.

The SAGE configuration can be displayed using the `sage_dump_db -config` command.

For VERITAS users another configuration file, `sage_error_action.table`, can optionally

be placed in the SAGE directory. This file can be used to specify the action to take on specific VERITAS errors returned from the nearline storage device. The file can contain as many records as are required by a specific implementation. Each record contains three fields:

```
/ =supersync_error      150
  =error_message        termination requested by administrator
  =action                1

/ =supersync_error      157
  =error_message        suspend requested by administrator
  =action                2
```

The `supersync_error` field specifies the VERITAS error code that is returned to SAGE when a specific error occurs. The `error_message` field is the text associated with the message, and is ignored by the program, but provided for documentation purposes. The action is one of three values: 0 (the default) specifies the error is to be logged and the associated save or restore is to be terminated. 1 indicates the save or restore should be retried, while 3 indicates the error is to be ignored.

If the file does not exist, all nearline storage device errors will result in the save or restore being terminated.

After the programs are installed and the configuration is set up, the SAGE database files must be created. This is accomplished with the command `sage_create_db -all`. These commands are documented in the SAGE Command Reference section of this manual.

## Starting SAGE

Use the `sage_control -start` command to start the SAGE system. In production, this command will be placed in the `(master_disk)>system>module_start_up.cm`

SAGE is stopped using the command `sage_control -stop`.

You can determine whether SAGE is running by using the `sage_control -status` command. This command is documented in the Command Reference section.

## SAGE User Commands

SAGE provides several commands for normal user interaction. For instance, saves and restores are scheduled with `sage_save` and `sage_restore`. To locate a save containing a specific file or directory, use `sage_find_file`.

The command `sage_show_log` displays the VOS log output by a save or restore command, `sage_show_saves` lists available saves, and `sage_show_schedule` builds an iCalendar file containing details of the past or future saves and restores managed by SAGE. Following is a detailed description of each of these commands.

## SAGE Command Reference

The following pages document the arguments to each of the SAGE commands and explain how they are used. The commands are divided into five groupings.

<i>Command Group</i>	<i>Purpose</i>	<i>Commands</i>
Administrator	Commands used to control SAGE in the normal operation and maintenance of the system.	sage_control sage_supersync_admin
Background	Commands that run as background processes, started by other processes, that control SAGE during normal operations.	sage_monitor sage_notify sage_run_command_macro sage_schedule
Initialization	Commands used at the time of initial system set-up.	sage_create_db
Maintenance	Commands used for troubleshooting and debugging.	sage_dump_db sage_sequence_maint
User	Commands used to schedule SAGE saves and restores.	sage_find_file sage_restore sage_save sage_show_log sage_show_saves sage_show_schedule

**Figure 3. SAGE Command Reference Table**

*Note:* all commands have a `-version` switch, which reports the version number and date of the program and exits.

### **sage\_control**

The `sage_control` command is used to start or stop the SAGE system on the VOS module, or to report on the current status of SAGE. One option (`-start`, `-stop`, or `-status`) may be selected at a time.

```
----- sage_control -----
-start:    no
-stop:     no
-status:   no
-interval:
-force     no
```

The `-start` option is used to start SAGE. If SAGE is not already running, the program starts `sage_monitor` to activate the SAGE system.

The `-stop` option is used to stop SAGE by sending a `stop_process` signal to each of the running SAGE components. If a save or restore is in progress, `sage_control` will inform you SAGE cannot be stopped at the present time. If SAGE is running but no saves or restores are in progress, the program asks whether you really want to stop Sage.

The `-force` option overrides these checks, and immediately stops all SAGE processes

The `-status` option reports on the current status of the SAGE system. It results in a display showing current activity. Following is a typical display:

```
06-08-03 11:37:46 edt
Process Name          PID  Start Time
sage_monitor          0101caac  06-08-03 10:56:24 edt
sage_notify            0101caad  06-08-03 10:56:25 edt
sage_schedule          0101caae  06-08-03 10:56:30 edt
SageID SR State      Configuration : Disk
   47 S  NB save running  backup_weekly2 : ss3_weekly2
   48 S  Waiting for VOS save  backup_daily : ss1_daily
```

The first line of the display is the date and time. The next section shows all the SAGE processes currently running. In the example, two VOS saves are in progress simultaneously, so the list includes each save process as well as the `sage_run_command_macro` process controlling it in addition to the three main processes, `sage_monitor`, `sage_schedule`, and `sage_notify`.

The final section of the display is a list of the saves and restores actively being processed. The saves or restores are added to the list when the start time is reached, and remain on the list until finished.

Optionally, you can add `-interval n` to the `sage_control -status` command, and the display will repeat every `n` seconds.

## **sage\_create\_db**

The `sage_create_db` command is used to create the data files required for the system. The display of the command is:

```
----- sage_create_db -----
-all:          no
-detail:        no
-restore_history: no
-sage_id:       no
-save_history   no
-sequence:      no
-force:         no
```

The `-all` option instructs the command to create all files. To create individual files, select the option for the specific file. All files are located in the database directory specified in the configuration file.

<i>Option</i>	<i>Filename</i>	<i>Contents</i>
<code>-detail</code>	<code>detail.sage</code>	Name of every file saved. Used by <code>sage_find_file</code> to locate a specific file or directory to restore.
<code>-restore_history</code>	<code>restorehistory.sage</code>	Status information for completed <code>sage_restore</code> operations.
<code>-sage_id</code>	<code>sageid.sage</code>	Next available SAGE ID.
<code>-save_history</code>	<code>savehistory.sage</code>	Status information for completed <code>sage_save</code> operations.
<code>-sequence</code>	<code>sequence.sage</code>	Schedule of future save and restore operations.

**Figure 4. `sage_create_db` Commands**

If the file exists, `sage_create_db` will ask the user whether or not to delete and recreate the file. With the `-force` switch specified the program automatically deletes and recreates the file without user response.

### **sage\_dump\_db**

The `sage_dump_db` command provides a formatted dump of any or all of the SAGE database files. This command includes the following options:

```
----- sage_dump_db -----
-all:                no
-config:             no
-detail:            no
-detail_all:        no
-restore_history:   no
-save_history:      no
-sequence:          no
-sage_id:           no
-match:
-after
-before
```

<i>Option</i>	<i>Filename</i>	<i>Contents</i>
-config	sage_config.table	SAGE configuration information, used by all programs.
-detail	detail.sage	Name of every file saved. Used by <code>sage_find_file</code> to locate a specific file or directory to restore. <i>Note that this command dumps the file header only.</i>
-detail_all	detail.sage	Name of every file saved. Used by <code>sage_find_file</code> to locate a specific file or directory to restore. <i>Note that this command dumps every record in the file, which can be quite large.</i>
-restore_history	restorehistory.sage	Status information for completed <code>sage_restore</code> operations.
-sage_id	sageid.sage	Next available SAGE ID.
-save_history	savehistory.sage	Status information for completed <code>sage_save</code> operations.
-sequence	sequence.sage	Schedule of future save and restore operations.

**Figure 5. sage\_dump\_db Commands**

The `-all` switch causes all of the files to be dumped (This option dumps only the detail header, not the entire file).

The `-match` option takes an integer SAGE ID for an argument and dumps only those records matching the given value.

The commands `-after` and `-before` take date-time strings, and allow the user to limit the dump to data from a specific time period. They can be used alone or together, just like `-saved_after` and `-saved_before` in `sage_find_file`.

### **sage\_find\_file**

The `sage_find_file` command is used to identify a specific SAGE save containing a file or directory to be restored.

```
----- sage_find_file -----
file_or_path:
-saved_after:
-saved_before:
```

The `-saved_after` and `-saved_before` options allow the user to establish a time period for the search. If the options are omitted, every SAGE save containing the file or directory will be displayed. If only `-saved_after` is given, all saves after the specified time are displayed. If only `-saved_before` is given, all saves before that time will be displayed. If both are given, the displayed saves will fall between the two times.

Typical output from the command follows. Note that every occurrence of a specific file name is shown.

```
sage_find_file sage_save.c
```

```
File saved in Sage ID 1 on 06-07-05 22:44:56 edt:  
%system#d02>Users>user_name>sage>src>sage_save.c
```

```
File saved in Sage ID 1 on 06-07-05 22:44:56 edt:  
%system#d02>Users>user_name>sage_1.2f>src>sage_save.c
```

```
File saved in Sage ID 6 on 06-07-06 10:57:03 edt:  
%system#d02>Users>user_name>sage>src>sage_save.c
```

```
File saved in Sage ID 6 on 06-07-06 10:57:03 edt:  
%system#d02>Users>user_name>sage_1.2f>src>sage_save.c
```

## sage\_monitor

The `sage_monitor` process is started by `sage_control` at SAGE startup. It sets up and initializes the shared memory area used by SAGE and starts `sage_notify` and `sage_schedule`. It then monitors the processes it started, logging any failures and attempting to restart either process if it fails. The program attempts the restart five times, and then exits to prevent a permanent error from unnecessarily consuming system resources.

```
----- sage_monitor -----  
No arguments required. Press ENTER to continue.
```

<p><b>Note:</b> The <code>sage_monitor</code> command is for internal use by the SAGE system and may not be used directly by a SAGE user.</p>
---

## sage\_notify

The `sage_notify` process establishes a TCP/IP listen port, and waits for connection requests from the `sage_request_server` program running on the nearline storage device. The program handles the completion of Netbackup saves and restores for the SAGE system.

```
----- sage_notify -----  
No arguments required. Press ENTER to continue.
```

<p><b>Note:</b> The <code>sage_notify</code> command is for internal use by the SAGE system and may not be used directly by a SAGE user.</p>
--

## sage\_restore

The `sage_restore` command is used to restore files from one of the saves managed by the SAGE system. `sage_restore` can be used to schedule a restore in the future. For example, the actual restore could be run at midnight to minimize system impact.

The `-restore_date_time` option is used to schedule the restore. It defaults to the current date and time so that a restore without this option specified will start as soon as possible.

```
----- sage_restore -----
-restore_date_time: 06-07-11 13:00:38 edt
-source_sage_id:
-use_config:
-use_sd:
-sage_name
object_names:
-source:
-destination:
-dynamic_extents: as_saved
-restore:         yes           -replace:         no
-same_place:      no           -first:           no
-pack:            no           -trim:            no
-restore_acls:    no           -backup:          no
-extent_new_size:
-dynamic_index_size:
-static_index_size:
-sleep_ms:        0           -sleep_modulus:  0
```

SAGE saves are identified by the integer SAGE ID, which is reported at the time the save is scheduled, and can also be found using the `sage_find_file` command. This value must be specified as `-source_sage_id` to the `sage_restore` command.

The `-use_config` and `-use_sd` options allow the user to specify a specific restore configuration and/or a specific starting disk in the configuration. For most users, the default values chosen by the command (the first disk in the first restore configuration) are acceptable, and these options can be ignored. For specific needs or unusual configurations, they provide control that might be required.

The `-sage_name` option allows the user to specify a name for the restore to assist in identification. The name may contain any text string (for example, “Restore of John’s home directory.”) and can be used to identify a specific restore. If omitted, the save is named “Restore from Sage ID n”.

The remaining options on the command are passed to the VOS restore command. The VOS save and VOS restore commands are documented in Stratus Document R285 -- *VOS System Administration: -- Backing Up and Restoring Data*, which can be viewed online at the StrataDOC website, <http://stratadoc.stratus.com/>. The VOS restore options presented by `sage_restore` on your system may differ from what is shown here. Specific releases of VOS support different sets of restore options. SAGE checks the release installed on the system and presents only the supported options in the form display.

## sage\_run\_command\_macro

The `sage_run_command_macro` command is used by `sage_schedule` to manage the VOS save or VOS restore process. The program starts the save or restore using the command macro passed to it, and waits for the started process to complete. Upon completion, the program checks the output log file and updates the sequence record appropriately.

```
----- sage_run_command_macro -----
cm_file:
-sage_id:
-command_type:          save | restore
-log_file:
```

**Note:** The `sage_run_command_macro` command is for internal use by the SAGE system and may not be used directly by a SAGE user.

## sage\_save

The `sage_save` command is used to schedule a save through the SAGE system.

```
----- sage_save -----
-save_date_time:        06-07-11 12:42:16 edt
-repeat_every:         never
-repeat_count:
-expire_after:         1 month
-sage_name
-use_disk:
-use_config:
-use_filename:
-before_macro:
-after_macro:
-enterprise_backup:    yes
pathnames:
-date:
-modified:
-modified_since_saved: no
-backup:              no
-do_not_save:
-control_file:
-preread_lead:        32
-preread_size:        8
-sleep_ms:            0
-sleep_modulus:       0
-old_format:          no
-save_open_options:   yes
```

The first three parameters specify the scheduling: when to perform the save, whether and when it repeats, and the number of times it repeats. The first argument, `-save_date_time`, is the date and time the program will start the save. This argument defaults to “right now”, but can be changed to allow for saves in the future.

The `-repeat_every` argument allows the user to specify the frequency with which the save will be repeated. The argument may contain a text string consisting of `never` or an optional number followed by the keyword `day`, `week`, `month` or `year`, optionally followed by an “s.” For example, these are valid repeat strings:

<code>never</code>	<code>week</code>
<code>3 days</code>	<code>6 months</code>
<code>1 year</code>	<code>2 weeks</code>

The default value for this option is `never`.

The `-repeat_count` argument specifies the number of times the save is to be performed. A save that runs daily except for the weekend could be specified to repeat every day for five days by using the argument string

```
-repeat_every "1 day" -repeat_count 5.
```

The next option, `-expire_after` allows the user to specify an expiration time for the save.

The content of the string is the same as that for the `-repeat_every` option above. The default value for this option comes from the configuration file.

The `-save_name` option allows the user to specify a name for the save to assist in identification. It can contain any text string (for example, “Special save of John’s home directory.”), and can be used to identify a specific save. If omitted, the program generates a default string from the options provided.

The next two arguments, `-use_config` and `-use_disk`, allow saves to be scripted if desired. If the user does not define these arguments SAGE presents a list of available configurations and asks the user to choose one. Then, SAGE displays a list of disks for the configuration and asks the user to choose a disk. If these options are provided with values, the given value is checked against the list of available configurations and the list of available disks to make sure the choices are valid.

If the `-use_config` and `-use_disk` arguments are not given, `sage_save` shows the possible values so the user can select the configuration and disk to be used. Following is an example showing the choices given interactively:

```
sage_save (home_dir)
```

```
These configurations are available for save:
```

- `1. backup_daily`
- `2. backup_weekly1`
- `3. backup_weekly2 (current)`
- `4. backup_weekly3`
- `5. backup_weekly4`

```
Please enter your choice, or 0 to exit: 1
```

You can obtain the same results with this command:

```
sage_save (home_dir) -use_config backup_daily -use_disk ss1_daily
```

These disks are available for use as a save destination:

1. ss1\_daily
2. ss2\_daily
3. ss3\_daily
4. ss4\_daily
5. ss5\_daily

Please enter your choice, or 0 to exit: 1

The `-use_filename` option allows the user to specify a name for the save file created by the VOS save. If a name is not given the program provides one.

The options `-before_macro` and `-after_macro` allow users to add customization to SAGE. The command macro specified in the `-before_macro` option is executed immediately before SAGE starts the VOS save. The `-after_macro` is executed upon the completion of the VOS save.

Neither is required, and the specific processing in the command macros will vary from site to site. An example is using the `-before_macro` to build a custom control file for the save, based on decisions made in a program they've written.

The `-enterprise_backup` switch allows the user to specify whether or not the enterprise backup is to be performed immediately after the completion of the save. This switch is useful in some cases to lighten the load on the enterprise backup system: if a series of saves is to be done to one nearline storage device disk, only a single enterprise backup is required after the last save is completed. By default, the value is `yes`. The enterprise backup will be performed as soon as possible.

The remaining options are passed to the VOS save command that actually runs the save, and work as documented in Stratus Document R285 -- *VOS System Administration: -- Backing Up and Restoring Data*, which can be viewed online at the StrataDOC website, <http://stratadoc.stratus.com/>. The VOS save options presented by `sage_save` on your system may differ from what is shown here. Specific releases of VOS support different sets of save options. SAGE checks the release installed on the system and presents only the supported options in the form display.

## **sage\_schedule**

The `sage_schedule` process is the main driving process for the SAGE system. It periodically wakes up and examines the sequence file for save or restore records that need service. This may include tasks such as a scheduled save or restore, starting a Netbackup following a VOS save, running a VOS restore after a Netbackup restore, etc.. The program requests configuration changes, starts processes, and manages the database files as appropriate.

The configuration parameter `check_interval` controls the frequency with which `sage_schedule` checks the configuration file.

```
----- sage_schedule -----
No arguments required. Press ENTER to continue.
```

<p><b>Note:</b> The <code>sage_schedule</code> command is for internal use by the SAGE system and may not be used directly by a SAGE user.</p>
--

## sage\_show\_log

```
----- sage_show_log -----
-sage_id:
-match:
```

This command displays the save or restore log written by a VOS save or restore command. The command requires the `-sage_id` argument, and accepts a `-match` string. If the match string is specified, only the records that contain the text are displayed. Otherwise, the entire file is displayed. An example follows:

```
sage_show_log -sage_id 23 -match start_up
06-09-05_14:47:28      file
%system#d02>Users>user_name>start_up.ostcp.cm
06-09-05_14:47:29      file
%system#d02>Users>user_name>start_up.normal.cm
06-09-05_14:47:29      file
%system#d02>Users>user_name>start_up.cm
06-09-05_14:47:54      file
%system#d02>Users>user_name>sage>xx>start_up.cm
```

## sage\_show\_schedule

The `sage_show_schedule` command is used to display saves and restores performed (or to be performed) by the Sage system.

```
----- sage_show_schedule -----
-start:          2006-10-19 10:41:18 edt
-period:         1 month
-show_saves:     yes
-show_restores:  no
-show_future:    yes
-show_history:   no
-show_command:   no
-output_file:    sage_schedule.ics
```

The `-start` and `-period` options specify when the displayed schedule is to start and the length of time to be displayed. The period specification is in the same format as the `repeat` and `expire` fields in `sage_save`.

The next five switches control the output data. By default, the command displays future saves only, but you can choose to display future restores and save and restore history, as desired. The `-show_command` switch places the actual save (or restore) command into the calendar entry, so it can be viewed if desired.

The final option specified the output file name.

Output from the program is a file in Internet Calendaring and Scheduling (.ics) format. This format is also called iCalendar, iCal, and RFC-2445 format, and is supported by most calendar programs on PCs, Macs, Linux systems, and handheld devices. Once the file is moved from the VOS system, it can be imported into a calendar program, as seen here in Figure 11:

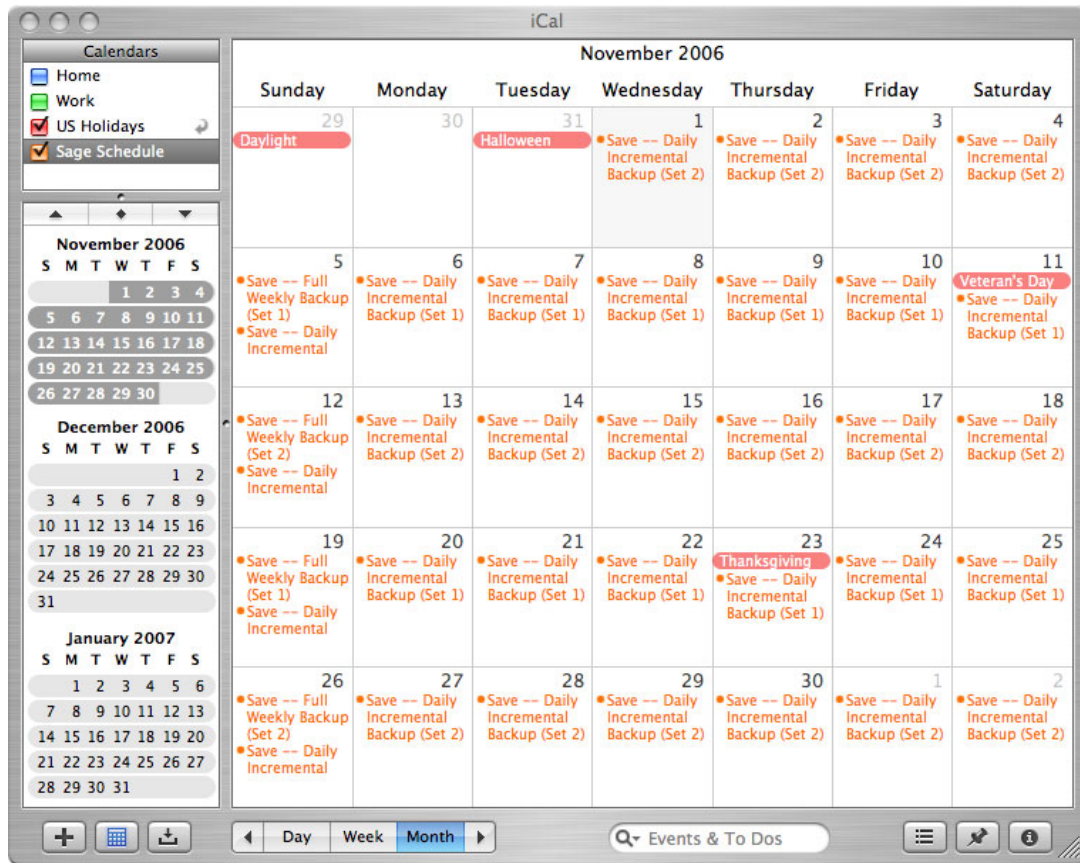


Figure 6. `sage_show_schedule` Calendar

## `sage_supersync_admin`

The `sage_supersync_admin` command allows a VOS administrator to query and change the configuration of the nearline storage device attached to the VOS system. If the `-new_config` option is specified and valid, the configuration will be changed to the specified configuration, as

long as changing the configuration will not interrupt current SAGE operations in progress.

If the configuration is not specified on the command line, the user will be presented with the list of available configuration files so one can be chosen. The current configuration is indicated in the list. Following is an example of `sage_supersync_admin`:

```
sage_supersync_admin
```

```
These configurations are available:
```

1. backup\_daily (current)
2. backup\_weekly1
3. backup\_weekly2
4. backup\_weekly3
5. backup\_weekly4
6. restore

```
Please enter your choice, or 0 to exit: 0
```