VERITAS™

Veritas High Availability Agent 4.1 for Apache HTTP

Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

N18562C

March 2006

Disclaimer

The information contained in this publication is subject to change without notice. Veritas Software Corporation makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Veritas Software Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

Veritas Legal Notice

Veritas Software Corporation
350 Ellis Street
Mountain View, CA 94043
USA
Phone 650–527–8000 Fax 650–527–2908
www.veritas.com

Third-Party Legal Notices

Certain third-party software may be distributed, embedded, or bundled with this Veritas product, or recommended for use in conjunction with Veritas product installation and operation. Such third-party software is separately licensed by its copyright holder. See the Third-Party Legal Notice appendix in this document for the licenses that govern the use of the third-party software and for proprietary notices of the copyright holders.

Contents

Preface

This document provides instructions for installing and configuring the Veritas Cluster Server (VCS) Agent for Apache HTTP. For information about Veritas Cluster Server, refer to the *Veritas Cluster Server User's Guide*.

How This Guide is Organized

"Introducing the Agent for Apache" on page 1 introduces the Apache HTTP agent that can be used with Veritas Cluster Server.

"Installing the Agent for Apache" on page 3 covers steps to install the Apache HTTP agent on Solaris, HP-UX, AIX, and Linux platforms.

"Configuring the Agent for Apache" on page 7 covers the attributes that you can set for configuring the Apache HTTP agent.

"Clustering Apache Servers" on page 15 covers steps that you must perform before clustering Apache Servers.

"Uninstalling the Agent for Apache" on page 17 covers steps to uninstall the Apache HTTP agent.

"Troubleshooting the Agent for Apache" on page 19 covers tips and pointers that you can use while working with the Apache agent.

Conventions

Typographic Conventions

Typeface	Usage	Examples
`monospace`	Computer output, files, directories, account names, roles, permissions, events, event types, workspaces, software elements such as command options, function names, and parameters	Read tunables from the `/etc/vx/tunefstab` file. You must have the `Full Control` permission.
`monospace` (**`bold`**)	User input	**`# mount - vxfs /h/filesys`**
italic	New terms, book titles, emphasis, variables	See the *User's Guide* for details. The variable *vxfs_ninode* determines the value of...
Palatino (Bold)	GUI objects (buttons, labels, categories, field names, selections)	Click **OK**. Enter the IP address in the **IP Address** field.
*	Indicates required field	**IP Address** *
$	Bourne/Korn/Bash shell prompt	
#	Superuser prompt (all shells)	
[ ]	In a command synopsis, brackets indicates an optional argument	`ls [ -a ]`
|	In a command synopsis, a vertical bar separates mutually exclusive arguments	`mount [ suid | nosuid ]`
blue text	Indicates an active hypertext link. In PDF and HTML files, click on links to move to the specified location	http://www.veritas.com

Getting Help

For technical assistance, visit http://support.veritas.com and select phone or email support. This site also provides access to resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service. Use the Knowledge Base Search feature to access additional product information, including current and past releases of Veritas documentation.

Diagnostic tools are also available to assist in troubleshooting problems associated with the product. These tools are available on disc or can be downloaded from the Veritas FTP site. See the `README.VRTSspt` file in the `/support` directory for details.

Additional Resources

For license information, software updates and sales contacts, visit https://my.veritas.com/productcenter/ContactVeritas.jsp. For information on purchasing product documentation, visit http://webstore.veritas.com.

Documentation Feedback

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions to clusteringdocs@veritas.com. Include the title and part number of the document (located in the lower left corner of the title page), and chapter and section titles of the text on which you are reporting. Our goal is to ensure customer satisfaction by providing effective, quality documentation. For assistance with topics other than documentation, visit http://support.veritas.com.

Introducing the Agent for Apache 1

Welcome to the Veritas Cluster Server (VCS) Agent for Apache HTTP. This guide describes the agent, its modes of operation, and its attributes. It also describes how to install and configure the agent. The guide assumes the reader understands the primary components and basic functionality of VCS. It also assumes a basic understanding of the Apache HTTP Server architecture and configuration process.

What's New in this Agent

The enhancements in this release of Apache Agent are:

- Added support for Apache HTTP version 2.2.
- Integrated with the enhanced version of ACC library, that includes numerous fixes for improved functionality..
- Added support to determine the Apache HTTP version during run time.
- Removed the ApacheVersion attribute.

Supported Software

The VCS Agent for Apache HTTP is supported in the following environments:

Environment	Supported Versions
Veritas Cluster Server	4.0 and 4.1
ACC library	4.1.04.0
Operating Systems	AIX 5.1, 5.2, 5.3 on pSeries HP-UX 11i V2 on PA-RISC Red Hat Enterprise Linux 3.0, 4.0 on Intel Solaris 8, 9, 10 on SPARC

Environment	Supported Versions
Apache HTTP Server	1.3, 2.0, 2.2
IBM HTTP Server	1.3, 2.0

About the Agent

The VCS agent for Apache HTTP consists of resource type declarations and agent scripts. The agent brings an Apache Server online and offline, and monitors the processes. The operations are described below:

- *Online*—Starts an Apache HTTP Server by executing the program httpdDir/httpd with the appropriate arguments. If a file is specified by attribute EnvFile, it will be sourced before executing httpd.
- *Offline*—Stops the Apache HTTP Server by either executing the program httpdDir/httpd with the appropriate arguments (Apache v2.0) or by sending a TERM signal to the HTTP Server parent process (Apache v1.3). If a file is specified by attribute EnvFile, it will be sourced before executing httpd.
- *Monitor*—Monitors the state of an Apache HTTP Server by first checking if the processes exist and then performing an optional, secondary state check. Refer to "Detecting Application Failure" on page 2 for more information.
- *Clean*—Removes Apache HTTP Server system resources that may remain after a server fault or after an unsuccessful attempt to online or offline an Apache HTTP Server. These resources include the parent httpd daemon and all its child daemons.

Detecting Application Failure

The agent provides two methods to evaluate the state of an Apache HTTP Server instance. The first state check is mandatory and the second is optional.

The first check determines the state of the Apache HTTP Server by searching for the existence of the parent httpd daemon and for at least one child httpd daemon. If the parent process and at least one child do not exist, VCS reports the resource as offline. If they do exist, and if the agent attribute SecondLevelMonitor is set to true, then a socket connection is established with the Apache HTTP Server using the values specified by agent attributes Host and Port. Once connected, the agent issues an HTTP request to the server to test its ability to respond. If the HTTP Server responds with a return code between `0` and `408`, the agent considers the server online. If the server fails to respond or returns any other code, the agent considers the server offline.

Installing the Agent for Apache 2

This chapter describes how to install VCS Agent for Apache HTTP Server in a VCS cluster. You must install the Apache agent on all the systems that will host an Apache service group.

Prerequisites

Perform the following steps before installing the Apache HTTP agent.

- Install and configure Veritas Cluster Server 4.0 or later.
- Install the ACC Library 4.1.04.0 (`VRTSacclib`) if it is not already installed.

  If the ACC Library needs to be installed or updated, the library and its documentation can be obtained from the agent software media. For more information about the ACC library, refer to "About the ACC Library" on page 4.
- Remove any prior version of this agent.

About the ACC Library

The entry point programs for VCS Agent for Apache HTTP Server depend on a set of Perl modules known as the ACC Library. The library must be installed on each system in the cluster that will run the Apache HTTP Server agent. The ACC Library contains common, reusable functions that perform tasks such as process identification, logging, and system calls.

The library is included with your purchase of the agent, but the library package is distinct from the agent package and must be installed separately. The ACC Library package is included within the agent's software distribution media (tar file or CD). Installation instructions for the library are provided in the ACC Library package (`VRTSacclib`) and are not included in this document.

Upgrading the Agent Software

If an older version of the agent software is already installed on the target system, first follow the instructions to remove the older version. Refer to "Uninstalling the Agent for Apache" on page 17 for the unistallation procedure. Then follow the instructions below to install the new agent software.

Installing the Agent Software in a VCS environment

For each platform, perform the following steps on each system in the cluster.

▼ To install the agent on AIX systems

1. Log in as `root`.
2. Go to the `/aix/application/apache_agent/`*`Version_Number`*`/pkgs` directory.
3. Install the package:

   ```
   # installp -ac -d VRTSvcsap.Version_Number.rte.bff VRTSvcsap.rte
   ```

▼ To install the agent on HP-UX systems

1. Log in as `root`.
2. Go to the `/hpux/application/apache_agent/`*`Version_Number`*`/pkgs` directory.

3. Install the package:

```
# swinstall -s /Temp_Directory/pkgs VRTSvcsap
```

▼ To install the agent on Linux systems

1. Log in as `root`.

2. Go to the `/linux/linux/application/apache_agent/Version_Number/pkgs` directory.

3. Install the package:

```
# rpm -ihv VRTSvcsap.Version_Number.rpm
```

▼ To install the agent on Solaris systems

1. Log in as `root`.

2. Go to the `/solaris/sparc/application/apache_agent/Version_Number/pkgs` directory.

3. Install the package:

```
# pkgadd -d . VRTSvcsap
```


Configuring the Agent for Apache 3

After installing the agent package, you must import the agent configuration file. Then you can create and configure an Apache Server resource using VCS Cluster Manager. Before you configure a resource, review the attributes table that describes the Apache HTTP resource type and its attributes. The resource type definition file and a sample `main.cf` configuration are also shown for reference.

Importing the Agent Types Files

To use the Apache HTTP agent without stopping and restarting Veritas Cluster Server, import the `ApacheTypes.cf` file into the VCS engine by performing the following steps in VCS Cluster Manager (the VCS graphical user interface).

▼ To import the agent types files

1. Start VCS Cluster Manager.
2. Click **File** > **Import Types**.
3. In the **Import Types** dialog box, select the following file:

   `/etc/VRTSvcs/conf/sample_Apache/ApacheTypes.cf`
4. Click **Import**.
5. Save the VCS Cluster Server configuration.

At this point, the Apache types have been imported into the Veritas Cluster Server engine. You can now create Apache resources. For detailed information about using the VCS Cluster Manager, refer to the *Veritas Cluster Server User's Guide*.

Type Definition

```
type Apache (
   static str ArgList[] = { ResLogLevel,  State, IState, httpdDir,
   SharedObjDir, EnvFile, HostName, Port, User,
   SecondLevelMonitor, SecondLevelTimeout, ConfigFile, EnableSSL,
   DirectiveAfter, DirectiveBefore}
   str ResLogLevel = "INFO"
   str httpdDir
   str SharedObjDir
   str EnvFile
   str HostName
   int Port = 80
   str User
   boolean SecondLevelMonitor
   int SecondLevelTimeout = 30
   str ConfigFile
   boolean EnableSSL
   str DirectiveAfter{}
   str DirectiveBefore{}
)
```

Apache Agent Attributes

The Apache agent attributes are as described below.

Required Attributes

Attribute	Description
ConfigFile	Full path and file name of the main configuration file for the Apache HTTP Server. Type: String-scalar Example: **`/apache/server1/conf/httpd.conf`** Default: “”
httpdDir	Type: String-scalar Full path of the directory in which the `httpd` binary file is located. Example: **`/apache/server1/bin`** Default: “”

Attribute	Description
HostName	Virtual host name assigned to the Apache HTTP Server instance. The host name is used in second-level monitoring to establish a socket connection with the Apache HTTP Server. Specifying this attribute is required only if SecondLevelMonitor is set to **1** (true). Type: String-scalar Example: **web1.veritas.com** Default: ""
Port	Port number on which the Apache HTTP Server instance is listening. The port number is used in second-level monitoring to establish a socket connection with the server. Specifying this attribute is required only if SecondLevelMonitor is set to **1** (true). Type: Integer Example: **80** Default: 80
ResLogLevel	Controls the logging detail performed by the agent for a specific instance of a resource. Valid values are: ◆ **ERROR**: Only error level messages are logged. ◆ **WARN**: Error and warning level messages are logged. ◆ **INFO**: Error, warning, and informational level messages are logged. ◆ **TRACE**: Error, warning, informational, and trace level messages are logged. Trace logging is very verbose and should only be used during initial configuration or to facilitate troubleshooting. Type: String-scalar Example: **TRACE** Default: INFO
User	Account name under which the agent will execute the httpd program. If unspecified, httpd is executed as user root. Type: String-scalar Example: **apache1** Default: ""

Optional Attributes

Attribute	Description
DirectiveAfter	A list of directives that httpd processes after reading the configuration file. In VCS, the directives are specified as an array of name=value combinations, separated by commas. Specifying this attribute is optional. Type: String-association Example: **`DirectiveAfter{} = { KeepAlive=On }`** Default: “”
DirectiveBefore	A list of directives that httpd processes before reading the configuration file. In VCS, the directives are specified as an array of name=value combinations, separated by commas. Specifying this attribute is optional. Type: String-association Example: **`DirectiveBefore{} = { User=nobody, Group=nobody }`** Default: “”
EnableSSL	If set to **`1`** (true), the online entry point will add support for SSL by including the option `-DSSL` in the start command. For example: `/usr/sbin/httpd -k start -DSSL` If set to **`0`** (false), the `-DSSL` option is excluded from the command. Type: Boolean Example: **`1`** Default: `0`
EnvFile	Full path and file name of the file that is sourced prior to executing `httpdDir/httpd`. With Apache 2.0, the file `<ServerRoot>/bin/envvars`, which is supplied in most Apache 2.0 distributions, is commonly used to set the environment prior to executing httpd. Specifying this attribute is optional. If EnvFile is specified, the login shell for user root must be Bourne, Korn, or C shell. Type: String-scalar Example: **`/apache/server1/bin/envvars`** Default: “”

Attribute	Description
SecondLevelMonitor	Enables second-level monitoring for the resource. Second-level monitoring is a deeper, more thorough state check of the Apache HTTP Server performed by issuing an HTTP GET request on the web server's root directory. Valid attribute values are **1** (true) and **0** (false). Specifying this attribute is required. Type: Boolean Example: **1** Default: 0
SharedObjDir	Full path of the directory in which the Apache HTTP shared object files are located. Specifying this attribute is optional. It is used when the HTTP Server is compiled using the SHARED_CORE rule. If specified, the directory is passed to the -R option when executing the httpd program. Refer to the httpd man pages for more information about the -R option. Type: String-scalar Example: **/apache/server1/libexec** Default: ""
SecondLevelTimeOut	Number of seconds monitor entry point will wait on the execution of second-level monitor. If the second-level monitor program does not return to the calling monitor entry point before the SecondLevelTimeout window expires, the monitor entry point will no longer block on the program sub-process but will report that the resource is offline. The value should be sufficiently high to allow second level monitor enough time to complete, but the value should also be less than the value specified by the agent's MonitorTimeout. Type: Integer Example: **30** Default: 30

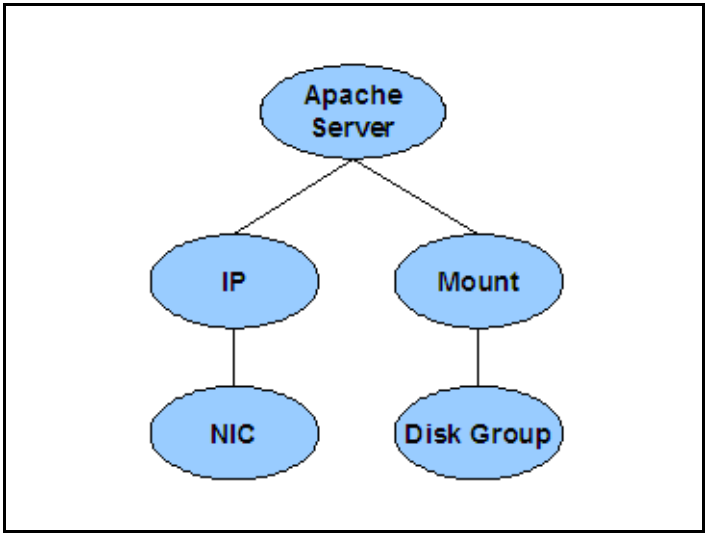
Sample Configuration

The following is an excerpt from a main.cf file that included an Apache HTTP Server resource.

```
Apache Web1_http (
   HostName = web1
   Port = 80
   SecondLevelMonitor = 1
   SecondLevelTimeout = 25
   httpdDir = "/apache/server1/bin"
   EnvFile = "/apache/server1/bin/envvars"
   ConfigFile = "/apache/server1/conf/httpd.conf"
)
```

Service Group Configuration

The following figure depicts a service group that supports a single Apache Server. At a minimum, the service group will require disk and network resources to support the Apache Server resource.



The disk group is the Veritas Volume Manager disk group that contains the volume and the file system (mount resource) that will store the Apache Server installation files (i.e. the files in ServerRoot). The disk group is created from shared disk so that it can be imported into any system in the cluster. In non-clustered configurations, an Apache Server is commonly installed on internal system disks, which *couples* the instance to one system

and prevents the server from being started on another system. Installing the Apache Server on shared disk decouples the instance and allows it to run on any system in the cluster. This flexibility is important when clustering an application server.

The network interface resource is the network card through which the Apache Server communicates. The virtual IP is the IP address dedicated to the server. External services, programs, and clients use this address to communicate with this server instance; thus, the server must be configured to bind to this virtual IP address.

Clustering Apache Servers 4

This chapter covers the procedures that you must follow while clustering Apache Servers within VCS service groups. Before using the Apache Servers in a cluster, Apache must be installed on shared disk. Also, each Apache Server must have a unique host name and port.

Clustering Prerequisites

Before clustering Apache Servers:

- Install Apache on shared disk.

  The Apache agent is designed to cluster Apache Servers installed on either shared disks or on non-shared disks, such as internal system drives. However, Veritas recommends installing all Apache program files, configuration files, and data files on shared disks.

  Installing all files on shared disks and assigning a virtual host name to the Apache Server *decouples* the Apache Server from any system in the cluster completely. This decoupling provides the flexibility of running the Apache Server on any system in the cluster.

  If you have a Veritas Volume Manager, you can place the Apache Server files on the Volume Manager volume. In case of failure, VCS can replicate the Apache Server to another volume at an alternate data center. You can then start the server in its last known state using the alternate data center.

  To install Apache on shared disk, refer to the Apache HTTP installation documentation.

- Assign virtual host name and port to Apache Server.

  If multiple Apache Server instances are running on the same system, configure each Apache Server instance to bind to its dedicated virtual IP host name and port number. Binding a unique host name and port prevent communication conflicts.

  If you configure the `Listen` directive in the Apache Server configuration file, you can listen on a unique IP host name and port. The ConfigFile attribute specifies the `Listen` directive.

Syntax:

```
Listen VirtualIPHostname:Port
```

Example:

```
Listen web1.veritas.com:80
```

Ensure that you set the ServerName attribute in the configuration file to the virtual IP host name assigned to the Apache HTTP Server.

Syntax:

```
ServerName VirtualIPHostname
```

Example:

```
ServerName web1.veritas.com
```

For more information about these directives, refer to the Apache HTTP documentation.

Uninstalling the Agent for Apache 5

Follow the steps below to remove the agent for Apache from the cluster. These steps must be performed while the cluster is active.

▼ To uninstall the agent

1. Log in as `root`.
2. Remove all resources from the cluster that use the agent for Apache.
3. Set the VCS configuration mode to read/write by typing the following command from any system in the cluster:

   ```
   # haconf -makerw
   ```

4. Remove the agent from the VCS configuration by typing the following command from any system in the cluster. This will remove the include statement for the agent from `main.cf`, but the agent's type file will not be removed.

   ```
   # hatype -delete Apache
   ```

5. Set the VCS configuration mode to read-only by typing the following command from any system in the cluster:

   ```
   # haconf -dump -makero
   ```

6. On each node in the cluster, use the package removal program to remove the agent software for Apache HTTP.

   Execute the following command to unistall the agent:

Platform	Command
AIX	`# installp -u VRTSvcsap.rte`
HP-UX	`# swremove VRTSvcsap`

Platform	Command
Linux	`# rpm -e VRTSvcsap`
Solaris	`# pkgrm VRTSvcsap`

Troubleshooting the Agent for Apache 6

This chapter covers tips and pointers on using the agent for Apache HTTP with Veritas high availability products. To resolve issues effectively, follow the steps in the order presented below. You may come across unique issues, but make sure that you follow these steps in the presented order to avoid unnecessary issues.

Starting the Apache Server Outside the VCS Framework

If you face problems while working with a resource, you must disable the resource within the VCS framework. A disabled resource is not under the control of the cluster framework, and so you can test the Apache Server independent of the cluster framework. Refer to VCS documentation for information about disabling a resource.

You can then restart the Apache Server outside the VCS framework.

Note Use the same parameters that the resource attributes define within the VCS framework while restarting the resource outside the framework.

▼ To restart the Apache Server outside the VCS framework

1. Using the user name specified in the "User" attribute, log into the host on which the Apache Server application is to run.

2. Use the values defined in the agent attributes to initiate the Apache Server start program.

   For example, assume that the following values are assigned to the agent attributes:

Attribute	Value
User	`root`
ConfigFile	`/apache/conf/httpd.conf`

EnvFile	**/apache/bin/envvars**
httpdDir	**/apache/bin**
HostName	**vcssun68**
Port	**8080**

3. Set the Apache Server environment by sourcing the file specified in the EnvFile attribute.

4. Start the Apache Server:

Apache Version	Command
1.3	`/apache/bin/httpd -f /apache/conf/httpd.conf`
2.0	`/apache/bin/httpd -f /apache/conf/httpd.conf -k start`

Ensure that the Apache Server starts successfully. If the server works properly outside the VCS framework, you can then attempt to implement the server within the VCS framework.

Using Correct Software and OS Versions

Be sure that no issues arise because of incorrect software and operating system versions. Refer to "Supported Software" on page 1 for correct versions of the software and operating systems.

Meeting Installation Prerequisites

Before installing the VCS agent for Apache HTTP, double check that you have met the prerequisite requirements. Refer to "Prerequisites" on page 3 for the list of prerequisites.

Configuring Apache Servers

You must configure the agent attributes properly before attempting to use the agent for Apache. Refer to "Apache Agent Attributes" on page 8 for the required and optional attributes table.

Meeting Clustering Prerequisites

Before attempting to cluster Apache Servers, double check that you have met the clustering prerequisites. Refer to "Clustering Prerequisites" on page 15 for more information about clustering prerequisites.

Using Error Log Files

If you face problems while using the Apache Server or the agent for Apache, use the error log files described in this section to invetigate the problems. Contact Veritas help for more information.

Reviewing Apache Server Log Files

If the Apache Server is facing problems, access and review the error log files that the Apache Server generates. The ErrorLog directive defines the name and location of the error log files. If you have not defined an absolute location, ErrorLog refers to the ServerRoot directive and stores the files in the `<ServerRoot>/logs/error_log` directory.

The log entries are in the following format:

```
[date and time when error occured] [severity of error] [IP address of
server] [error message]
```

For example:

```
[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied by
server configuration: /export/home/live/ap/htdocs/test
```

You can set the severity of the error messages that are logged in the review file by setting the LogLevel directive. Refer to the Apache HTTP documentation for more information about setting these directives.

Reviewing Agent Log Files

If an Apache Server instance is facing issues while using the agent for Apache, review the VCS engine log file to further investigate the problem. The VCS engine log file is `/var/VRTSvcs/log/engine_A.log`.

You can set the "ResLogLevel" attribute to control the severity of the log entries. Depending upon the value of ResLogLevel, the error information is logged into the VCS engine file.

Using Trace Level Logging

If you set ResLogLevel to **`TRACE`**, a very high volume of messages is produced. Veritas recommends that you must localize the ResLogLevel attribute for particular resource:

▼ To localize ResLogLevel attribute for a resource

1. Identify the resource for which you want to enable detailed logging.
2. Localize the ResLogLevel attribute for the identified resource:

   ```
   hares -local Resource_Name ResLogLevel
   ```

3. Set the ResLogLevel attribute to **`TRACE`** for the identified resource:

   ```
   hares -modify Resource_Name ResLogLevel TRACE -sys SysA
   ```

4. Test the identified resource. The operation reproduces the problem that you are attempting to diagnose.
5. Set the ResLogLevel attribute back to **`INFO`** for the identified resource:

   ```
   hares -modify Resource_Name ResLogLevel INFO -sys SysA
   ```

6. Review the contents of the VCS engine output log file in order to diagnose the problem.

You may also contact Veritas Support for more help.

Index