VERITAS

InfoScale Availability Technology Track Test Cases Guide for Application Agents and Database Agents

Veritas™ Technology Partner Program (VTPP)

Document version: Rev 2

Table of Contents

About VTPP-enabled agents

The Veritas Technology Partner Program (VTPP) allows technology partners to create custom agents for InfoScale Availability, formerly known as Cluster Server (VCS). The acronym VCS is still used for familiarity. Veritas assists partners to develop custom agents using SDKs provided by the InfoScale Availability Technology Track. These custom agents are tested, validated, certified, and integrated with the InfoScale Availability agents on the Veritas Services and Operations Readiness Tools (SORT) site.

For more information about VTPP, refer to the *InfoScale Availability Technology Track Welcome Kit*.

For more information about developing the agents, refer to the *Cluster Server Agent Developer's Guide*.

Note: After you run the test cases, use the SORT Data Collector to gather the logs that you can then share with Veritas. For more information, refer to the following article:
https://www.veritas.com/support/en_US/article.000081310

About Veritas test cases for VTPP-enabled agents

Partners develop custom InfoScale Availability agents, and Veritas provides generic test cases to test those VTPP-enabled agents. Partners test the agents using self-test tools and Veritas-provided generic test cases. Veritas validates the test results and certifies the custom agent. The test cases contain a set of test inputs, execution conditions, and expected results that aid in verifying the compliance of the agent with Veritas standards. The test cases describe all the conditions that partners must implement to verify that the agents meet the requirements of a High Availability (HA) validation solution.

For more information about testing an agent, refer to the *Cluster Server Agent Developer's Guide*.

Agent installation and configuration test case

This test case is used to verify that the agent binaries are correctly installed and configured and that the resource type is correctly imported.

Note: Before testing an agent, ensure that you have built, installed, and configured the agent.

Table 1: Agent installation and configuration test case

Test case description	Precondition	Action	Expected result

Verification of agent installation and configuration	The agent is built, installed, and configured on the system.	1. Verify that the type definition file is present.
		`#ls -l <path to type definition file>`
		2. Verify that the type definition file is included in the main.cf file.
		`# grep include /etc/VRTSvcs/conf/config/main.cf`
		3. Verify that the agent type is imported on the system.
		`#hatype -display <agentname>`
		4. Verify that all files of the agent are present on the system and are assigned the required permissions.
		`# ls -lR <path to agent directory>`
		5. Verify that the agent is running.
		`#haagent -display <agentname>`
			▪ The type definition file must be present.
			▪ The `hatype -display` command should display the agent attributes.
			▪ All files of the agent present on the system should have the read and execute permissions for the root user.
			▪ The `haagent -display` command should display the status of the agent that is running.

Entry point test cases

The entry point test cases assume that the following items are installed and configured on the target cluster node:

- InfoScale Availability (formerly VCS) or InfoScale Enterprise (formerly SFHA or SFW HA)
- The application that you plan to configure for high availability
- The InfoScale Availability agent package for the application

Online entry point test cases

Table 2: Online entry point test cases

Test case description	Precondition	Action	Expected result
Online operation on a properly configured resource	▪ Each resource attribute is set to a proper value. ▪ The application is cleanly shut down, no traces are left behind, and the state of the resource is offline.	1. Bring the resource online using VCS 2. Verify that the application is fully functional.	▪ The VCS resource should come online. ▪ The VCS logs should not display any error or warning messages. ▪ The application status command, if any, outside VCS should reports the status as running.
Online operation on a resource with an invalid agent attribute value	▪ One of the resource attributes is set to an invalid value. ▪ The application is cleanly shut down, no traces are left behind, and the state of the resource is offline.	1. Bring the resource online using VCS. 2. Repeat the test case with other attributes of the resource.	▪ The VCS resource should move to the UNKNOWN state. ▪ The VCS logs should display appropriate error or warning messages.

Table 2: Online entry point test cases

Test case description	Precondition	Action	Expected result
Online operation on a resource when the required agent attributes for online are missing	▪ One of the required resource attributes is not set to any value. ▪ The application is cleanly shut down, no traces are left behind, and the state of the resource is online.	1. Bring the resource online using VCS. 2. Repeat the test case with other attributes of the resource.	▪ The VCS resource should move to the UNKNOWN state. ▪ The VCS logs should display appropriate error or warning messages.
Online operation on a resource using all supported shells	Each resource attribute is set to a proper value. The application is cleanly shut down, no traces are left behind, and the state of the resource is offline.	1. Set the default shell to one of the supported shells. 2. Bring the resource online using VCS. 3. Verify that the application is fully functional. 4. Repeat the test case with other supported shells.	▪ The VCS resource should come online. ▪ The VCS logs should not display any error or warning messages.
Online operation on a resource with an application instance that is running	Each resource attribute is set to a proper value	1. Start the application outside VCS. 2. Before VCS detects the state as online, bring the resource online using VCS. 3. Verify that the application is fully functional.	▪ The VCS resource should report as online. ▪ The agent should log a message that the application is already running and exit the entry point without starting the application.

Table 2: Online entry point test cases

Test case description	Precondition	Action	Expected result
Online operation on a resource where application instance is partially running	Each resource attribute is set to a proper value.	1. Verify that the partial set of processes for the application is running. 2. Bring the resource online using VCS. 3. Verify that the application resource comes online and is fully functional.	▪ The agent should clean the partial processes that are running before it brings the application online. ▪ The agent should log messages about the partially online processes and messages related to kill such processes.
Online operation when two application instances are configured	▪ The application supports multiple instances of the application on the same system. ▪ Configure two instances of the application on the same system and configure resources for them.	Bring the resource online one after the other.	Each resource should bring the corresponding application instance online.

Offline entry point test cases

Table 3: Offline entry point test cases

Test case description	Precondition	Action	Expected result
Offline operation on a fully configured resource	Each resource attribute is set to a proper value.	1. Bring the resource online. 2. Once VCS detects the resource as online, take the resource offline using VCS. 3. Verify that the resource state is offline. 4. Verify that the application is stopped completely and no traces are left behind. 5. Bring the resource online using VCS and verify whether the application is functioning properly.	▪ The VCS resource should report offline and the application should stop completely. ▪ The VCS logs should not display any error or warning messages. ▪ Once the resource is brought online, the application should be fully functional.
Offline operation on a resource using all supported shells	▪ Each resource attribute is set to a proper value. ▪ The application is cleanly shut down, no traces are left behind, and the state of the resource is offline.	1. Set the default shell to one of the supported shells. 2. Bring the resource online using VCS. 3. Verify that the application is fully functional. 4. Repeat the test with other supported shells.	▪ The VCS resource should report offline and the application should stop completely. ▪ The VCS logs should not display any error or warning messages.

Table 3: Offline entry point test cases

Test case description	Precondition	Action	Expected result
Offline operation on an instance that is already stopped	Each resource attribute is set to a proper value.	1. Bring the resource online. 2. Stop the application outside VCS. 3. Before VCS detects resource as offline, take the application resource offline using VCS. `Text`	▪ The VCS resource should report as offline. ▪ The offline operation should report the application as already offline in the VCS logs.
Offline operation when two application instances are configured	▪ The application supports multiple instances of the application on the same system. ▪ Configure two instances of the application on the same system and configure resources for them. ▪ The instances are brought online.	Take the resource offline one after the other.	Each resource should stop the corresponding application instance.

Monitor entry point test cases

Table 4: Monitor entry point test cases

Test case description	Precondition	Action	Expected result
Monitor operation on an online resource	Each resource attribute is set to a proper value.	1. Bring the resource online using VCS. 2. Probe the resource and verify that the resource state is reported online. 3. Verify that the application is fully functional.	▪ The VCS resource should report online. ▪ The VCS logs should not display any error or warning messages.
Monitor operation on a manually started application	Each resource attribute is set to a proper value.	1. Start the application manually. 2. Probe the resource and verify that the resource state is reported online. 3. Verify that the application is fully functional.	▪ The VCS resource should report online. ▪ The VCS logs should not display any error or warning messages.

Table 4: Monitor entry point test cases

Test case description	Precondition	Action	Expected result
Monitor operation on an offline resource	Each resource attribute is set to a proper value.	1. Bring the resource online using VCS. 2. Verify that the resource state is reported online. 3. Take the resource offline using VCS. 4. Verify that the application is stopped completely and no traces are left behind. 5. Probe the resource and verify that the resource state is reported offline.	▪ The VCS resource should report offline. ▪ The VCS logs should not display any error or warning messages.
Monitor operation for an online VCS resource whose application process is killed outside the VCS control	Each resource attribute is set to a proper value.	1. Bring the resource online using VCS. 2. Verify that the resource state is reported online. 3. Kill the application process manually. 4. Verify that the monitor program detects the application status and resource faults. 5. Repeat the test with all possible ways of failure injection to the application.	▪ The resource should move to the FAULTED state. ▪ The agent should log the unexpected application failure messages in the VCS logs. ▪ If the Critical attribute is set to 1 for the application resource, the resource should come online on the other node.

Table 4: Monitor entry point test cases

Test case description	Precondition	Action	Expected result
Monitor operation on a resource with an invalid agent attribute value	▪ The application is cleanly shut down, no traces are left behind, and the state of the resource is offline. ▪ One of the resource attribute is set to an invalid value.	1. Probe the resource. 2. Repeat the test case with other attributes of the resource.	▪ The resource should report unknown or offline depending on the type of the attribute values. ▪ The agent should log the error or warning messages in the VCS logs.
Monitor operation on a resource when the required agent attributes are missing for an online resource	▪ The application is fully functional and the state of the resource is online. ▪ One of the resource attribute value is missing.	1. Probe the resource. 2. Repeat the test case with other attributes of the resource.	▪ The resource should report as unknown. ▪ The agent should log the error messages in the VCS logs.

Table 4: Monitor entry point test cases

Test case description	Precondition	Action	Expected result
Monitor operation on a resource using all supported shells	▪ Each resource attribute is set to a proper value. ▪ The application is cleanly shut down, no traces are left behind, and the state of the resource is offline.	1. Set the default shell to one of the supported shells. 2. Bring the resource online using VCS. 3. Verify that the application is fully functional and VCS detects the state as online. 4. Take the resource offline using VCS. 5. Verify that VCS detects the state as offline. 6. Repeat the test with other supported shells.	▪ The resource should report as offline. ▪ The VCS logs should not display any error or warning messages.
Monitor operation to test the LevelTwo monitoring, if implemented, when the application is running	▪ Each resource attribute is set to a proper value. ▪ The application is cleanly shut down, no traces are left behind, and the state of the resource is offline.	1. Bring the resource online using VCS. 2. Verify that the application is fully functional and VCS detects the state as online. 3. Set LevelTwoMonitorFreq to some value say 1. 4. Probe the resource repeatedly till the value of the LevelTwoMonitorFreq attribute.	▪ The VCS resource should report online. ▪ The VCS logs should not display any error or warning messages.

Table 4: Monitor entry point test cases

Test case description	Precondition	Action	Expected result
Monitor operation to test the LevelTwo monitoring, if implemented, when the application is not fully functional	▪ Each resource attribute is set to a proper value. ▪ The application is cleanly shut down, no traces are left behind, and the state of the resource is offline.	1. Bring the resource online using VCS. 2. Verify that the application is fully functional and VCS detects the state as online. 3. Inject the fault by issuing a SIGSTOP to the monitored application process. 4. Set LevelTwoMonitorFreq to some value say 1. 5. Probe the resource.	▪ The monitor operation should report the unexpected offline even though the first level monitor reports the state as online. ▪ The VCS logs should display the appropriate error or warning messages.
Monitor operation when two application instances are configured	▪ The application supports multiple instances of the application on the same system. ▪ Configure two instances of the application on the same system and configure resources for them.	1. Bring online the two resources. 2. Take one of the resources offline.	▪ Resources should report the correct state of the application instances. ▪ The application instance corresponding to the resource that is brought online should stop and the resources should report the correct state of the application instances.

Clean entry point test cases

Table 5: Clean entry point test cases

Test case description	Precondition	Action	Expected result
Clean operation on a resource when the application processes are killed	Each resource attribute is set to a proper value.	1. Bring the resource online using VCS. 2. Verify that the resource state is reported online. 3. Kill the application process manually. 4. Repeat the test with all possible ways of failure injection to application.	▪ The monitor program detects the application status, clean program succeeds, and the resource faults. ▪ The clean program completely stops the application.
Clean operation when two application instances are configured	Each resource attribute is set to a proper value.	1. Configure two resources of the application on the same system. 2. Bring the resources online using VCS. 3. Verify that both resources are reported online. 4. Kill one of the application instances manually.	▪ The correct resource faults and the other resource continues to function correctly. ▪ The clean operation gets called for the resource for which the processes are killed.
Clean operation for all possible causes such as online ineffective, offline ineffective, and monitor timeout	Each resource attribute is set to a proper value.	Simulate the online ineffective, offline ineffective, and monitor timeout use cases.	The clean program succeeds and stops the application.

Switchover and failover test cases

Table 6: Switchover and failover test cases

Test case description	Precondition	Action	Expected result
Manual switchover of a service group	Each resource attribute is set to a proper value.	1. Bring the service group online on one node. 2. Perform switch operation of service group to another node using the `hagrp -switch` command.	▪ The service group should come online on the other node in the cluster. ▪ The VCS logs should not display any error or warning messages.
Failover of a service group	▪ Each resource attribute is set to a proper value. ▪ Ensure that the Critical attribute of the resource is set to 1.	1. Bring the service group online on one node. 2. Inject fault on the application and verify that the service group fails over to another node.	The service group should come online on the other node in the cluster.

Intelligent Monitoring Framework test cases

Table 7: Intelligent Monitoring Framework (IMF) test cases

Test case description	Precondition	Action	Expected result
IMF registration for a resource when the application is in a steady online state	▪ Each resource attribute is set to a proper value. ▪ The IMF-related attributes are set to enable the IMF support for the resource. For example, the IMF mode is set to 2.	1. Bring the resource online using VCS. 2. Verify that the resource state is reported online. 3. Probe the resource. 4. Verify the status of the VCS resource using the `amfstat` command.	▪ The `amfstat` command should display the resource name and the corresponding registered process ID (PID). ▪ The MonitorMethod attribute for the resource should be set as IMF. ▪ The VCS logs should display the appropriate messages for the IMF registration.

Table 7: Intelligent Monitoring Framework (IMF) test cases

Test case description	Precondition	Action	Expected result
Instant notification for a resource when the application is stopped or killed outside the VCS control	▪ Each resource attribute is set to a proper value. ▪ The IMF-related attributes are set to enable the IMF support for the resource. For example, the IMF mode is set to 2.	1. Bring the resource online using VCS. 2. Verify that the resource state is reported online. 3. Probe the resource. 4. Verify that the resource is registered with IMF using the `amfstat` command. 5. Before the next monitor starts, outside the VCS control, kill the application process ID (PID) corresponding to the resource that is monitored. This PID should be seen in the `amfstat` output.	▪ The agent should instantly catch the application process kill event. ▪ The VCS should log the corresponding message in the VCS log. ▪ The resource should move to the FAULTED state.If the Critical attribute is set to 1, the resource should failover and come online on the other node.

Table 7: Intelligent Monitoring Framework (IMF) test cases

Test case description	Precondition	Action	Expected result
IMF registration for a resource when the application is in a steady offline state	▪ Each resource attribute is set to a proper value. ▪ The IMF-related attributes are set to enable the IMF support for the resource. For example, the IMF mode is set to 3.	1. Verify that the resource state is reported offline. 2. Probe the resource. 3. Verify that the resource is registered with IMF for offline monitoring using the `amfstat` command.	▪ The `amfstat` command should display the resource name and the corresponding registered process pattern. ▪ The MonitorMethod attribute for the resource should be set as IMF. ▪ The VCS logs should display the appropriate messages for the IMF registration.
Instant notification for a resource when the application is started outside the VCS control	▪ Each resource attribute is set to a proper value. ▪ The IMF-related attributes are set to enable the PROFF IMF support for the resource. For example, the IMF mode is set to 3.	1. Verify that the resource state is reported offline. 2. Probe the resource. 3. Verify that the resource is registered with IMF for offline monitoring using the `amfstat` command. 4. Before the next monitor starts, outside the VCS control, start the application process corresponding to the resource that is monitored.	▪ The agent should instantly catch the application process start event. The VCS should log the corresponding message in the VCS log. ▪ The resource should move to the ONLINE state.

About Veritas Technologies LLC

Veritas Technologies LLC enables organizations to harness the power of their information, with solutions designed to serve the world's largest and most complex heterogeneous environments. Veritas works with 86 percent of Fortune 500 companies today, improving data availability and revealing insights to drive competitive advantage.

For specific country offices and contact numbers, please visit our website.

Veritas World Headquarters
500 East Middlefield Road
Mountain View, CA 94043
+1 (650) 933 1000
www.veritas.com