



# CHAPTER 51

## Configuring Online Diagnostics

---

This chapter describes how to configure the online diagnostics on the Catalyst 6500 series switches.



### Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

<http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html>

---

This chapter consists of these sections:

- [Understanding How Online Diagnostics Work, page 51-1](#)
- [Configuring Online Diagnostics, page 51-2](#)
- [Running Online Diagnostic Tests, page 51-6](#)
- [Performing Memory Tests, page 51-10](#)

For descriptions of the online diagnostics tests, refer to [Appendix A, “Online Diagnostic Tests.”](#)

## Understanding How Online Diagnostics Work

With online diagnostics, you can test and verify the hardware functionality of the supervisor engine, modules, and switch while the switch is connected to a live network.

The online diagnostics contain packet switching tests that check different hardware components and verify the data path and control signals. Disruptive online diagnostic tests, such as the built-in self-test (BIST) and the disruptive loopback test, and nondisruptive online diagnostic tests, such as packet switching, run during bootup, line card online insertion and removal (OIR), and system reset. The nondisruptive online diagnostic tests run as part of background health monitoring or at the user’s request (on-demand).

The online diagnostics detect problems in the following areas:

- Hardware components
- Interfaces (GBICs, Ethernet ports, and so forth)
- Connectors (loose connectors, bent pins, and so forth)
- Solder joints
- Memory (failure over time)

Online diagnostics is one of the requirements for the high availability feature. High availability is a set of quality standards that seek to limit the impact of equipment failures on the network. A key part of high availability is detecting hardware failures and taking corrective action while the switch runs in a live network. Online diagnostics in high availability detect hardware failures and provide feedback to high availability software components to make switchover decisions.

Online diagnostics are categorized as bootup, on-demand, schedule, or health-monitoring diagnostics. Bootup diagnostics run during bootup, module OIR, or switchover to a backup supervisor engine; on-demand diagnostics run from the CLI; schedule diagnostics run at user-designated intervals or specified times when the switch is connected to a live network; and health-monitoring runs in the background.

## Configuring Online Diagnostics

These sections describe how to configure online diagnostics:

- [Setting Bootup Online Diagnostics Level, page 51-2](#)
- [Configuring On-Demand Online Diagnostics, page 51-3](#)
- [Scheduling Online Diagnostics, page 51-4](#)

### Setting Bootup Online Diagnostics Level

You can set the bootup diagnostics level as minimal or complete or you can bypass the bootup diagnostics entirely. Enter the **complete** keyword to run all diagnostic tests; enter the **minimal** keyword to run only EARL tests for the supervisor engine and loopback tests for all ports in the switch. Enter the **no** form of the command to bypass all diagnostic tests. The default bootup diagnostics level is minimal.



**Note**

The diagnostic level applies to the entire switch and cannot be configured on a per-module basis.

To set the bootup diagnostic level, perform this task:

Command	Purpose
Router(config)# <b>diagnostic bootup level</b> { <b>minimal</b>   <b>complete</b> }	Sets the bootup diagnostic level.

This example shows how to set the bootup online diagnostic level:

```
Router(config)# diagnostic bootup level complete
Router(config)#
```

This example shows how to display the bootup online diagnostic level:

```
Router(config)# do show diagnostic bootup level
Router(config)#
```

## Configuring On-Demand Online Diagnostics

You can run the on-demand online diagnostic tests from the CLI. You can set the execution action to either stop or continue the test when a failure is detected or to stop the test after a specific number of failures occur by using the failure count setting. You can configure a test to run multiple times using the iteration setting.

You should run packet-switching tests before memory tests. Run the memory tests on the other modules before running them on the supervisor engine.

**Note**

Do not use the **diagnostic start all** command until all of the following steps are completed.

Because some on-demand online diagnostic tests can affect the outcome of other tests, you should perform the tests in the following order:

1. Run the nondisruptive tests.
2. Run all tests in the relevant functional area.
3. Run the TestTrafficStress test.
4. Run the TestEobcStressPing test.
5. Run the exhaustive-memory tests.

To run on-demand online diagnostic tests, perform this task:

**Step 1** Run the nondisruptive tests.

To display the available tests and their attributes, and determine which commands are in the nondisruptive category, enter the **show diagnostic content** command.

**Step 2** Run all tests in the relevant functional area.

Packet-switching tests fall into specific functional areas. When a problem is suspected in a particular functional area, run all tests in that functional area. Not all functional areas are present on each module. If you are unsure about which functional area you need to test, or if you want to run all available tests, enter the **complete** keyword.

**Step 3** Run the TestTrafficStress test.

This is a disruptive packet-switching test that is only available on the supervisor engine. This test switches packets between pairs of ports at line rate for the purpose of stress testing. During this test all of the ports are shut down, and you may see link flaps. The link flaps will not recover after the test is complete. The test takes several minutes to complete.

Disable all health-monitoring tests for the module being tested before running this test by using the **no diagnostic monitor module *module* test all** command.

**Step 4** Run the TestEobcStressPing test.

This is a disruptive test and tests the Ethernet over backplane channel (EOBC) connection for the module. The test takes several minutes to complete. You cannot run any of the packet-switching tests described in previous steps after running this test. However, you can run tests described in subsequent steps after running this test.

Disable all health-monitoring tests for the module being tested before running this test by using the **no diagnostic monitor module *module* test all** command. The EOBC connection is disrupted during this test and will cause the health-monitoring tests to fail and take recovery action.

**Step 5** Run the exhaustive-memory tests.

All modules have exhaustive-memory tests available on them. Because the supervisor engine goes into an unusable state and must be rebooted after the exhaustive-memory tests, run the tests on all other modules first. Some of the exhaustive-memory tests can take several hours to complete because of the large memory size of the modules.

Before running the exhaustive-memory tests, all health-monitoring tests should be disabled on the module that will run the exhaustive-memory tests because the tests will fail with health monitoring enabled and the switch will take recovery action. Disable the health-monitoring diagnostic tests by using the **no diagnostic monitor module *module* test all** command.

Perform the exhaustive-memory tests in the following order (you can skip any tests not available for a particular module):

1. TestFibTcamSSRAM
2. TestAclQosTcam
3. TestNetFlowTcam
4. TestAsicMemory
5. TestAsicMemory

You must reboot the supervisor engine after running the exhaustive-memory tests before it is operational again. You cannot run any other tests on the supervisor engine or other modules after running the exhaustive-memory tests. Do not save the configuration when rebooting as it will have changed during the tests. You will need to power cycle the modules before they can be operational. After a module comes back on line, reenable the health-monitoring tests using the **diagnostic monitor module *module* test all** command

To set the bootup diagnostic level, perform this task:

Command	Purpose
Router# <b>diagnostic ondemand</b> {iteration iteration_count}   {action-on-error {continue   stop} [error_count]}	Configures on-demand diagnostic tests to run, how many times to run (iterations), and what action to take when errors are found.

This example shows how to set the on-demand testing iteration count:

```
Router# diagnostic ondemand iteration 3
Router#
```

This example shows how to set the execution action when an error is detected:

```
Router# diagnostic ondemand action-on-error continue 2
Router#
```

## Scheduling Online Diagnostics

You can schedule online diagnostics to run at a designated time of day or on a daily, weekly, or monthly basis for a specific module. You can schedule tests to run only once or to repeat at an interval. Use the **no** form of this command to remove the scheduling.

To schedule online diagnostics, perform this task:

Command	Purpose
Router(config)# <b>diagnostic schedule</b> { <b>module num</b> } <b>test</b> { <b>test_id</b>   <b>test_id_range</b>   <b>all</b> } [ <b>port</b> { <b>num</b>   <b>num_range</b>   <b>all</b> }] { <b>on mm dd yyyy hh:mm</b> }   { <b>daily hh:mm</b> }   { <b>weekly day_of_week hh:mm</b> }	Schedules on-demand diagnostic tests for a specific date and time, how many times to run (iterations), and what action to take when errors are found.

This example shows how to schedule diagnostic testing on a specific date and time for a specific module and port:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 on january 3 2003 23:32
Router(config)#
```

This example shows how to schedule diagnostic testing to occur daily at a certain time for a specific port and module:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 daily 12:34
Router(config)#
```

This example shows how to schedule diagnostic testing to occur weekly on a certain day for a specific port and module:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 weekly friday 09:23
Router(config)#
```

## Configuring Health-Monitoring Diagnostics

You can configure health-monitoring diagnostic testing on specified modules while the switch is connected to a live network. You can configure the execution interval for each health-monitoring test, whether or not to generate a system message upon test failure, or to enable or disable an individual test. Use the **no** form of this command to disable testing.

To configure health-monitoring diagnostic testing, perform this task:

	Command	Purpose
<b>Step 1</b>	Router(config)# <b>diagnostic monitor interval</b> { <b>module num</b> } <b>test</b> { <b>test_id</b>   <b>test_id_range</b>   <b>all</b> } [ <b>hour hh</b> ] [ <b>min mm</b> ] [ <b>second ss</b> ] [ <b>millisec ms</b> ] [ <b>day day</b> ]	Configures the health-monitoring interval of the specified tests for the specified module. The <b>no</b> form of this command will change the interval to the default interval, or zero.
<b>Step 2</b>	Router(config)# [ <b>no</b> ] <b>diagnostic monitor</b> { <b>module num</b> } <b>test</b> { <b>test_id</b>   <b>test_id_range</b>   <b>all</b> }	Enables or disables health-monitoring diagnostic tests.

This example shows how to configure the specified test to run every two minutes:

```
Router(config)# diagnostic monitor interval module 1 test 1 min 2
Router(config)#
```

This example shows how to run the test on the specified module if health monitoring has not previously been enabled:

```
Router(config)# diagnostic monitor module 1 test 1
```

This example shows how to enable the generation of a syslog message when any health-monitoring test fails:

```
Router(config)# diagnostic monitor syslog
Router(config)#
```

## Running Online Diagnostic Tests

After you configure online diagnostics, you can start or stop diagnostic tests or display the test results. You can also see which tests are configured for each module and what diagnostic tests have already run.

These sections describe how to run online diagnostic tests after they have been configured:

- [Starting and Stopping Online Diagnostic Tests, page 51-6](#)
- [Displaying Online Diagnostic Tests and Test Results, page 51-6](#)

## Starting and Stopping Online Diagnostic Tests

After you configure diagnostic tests to run on the switch or individual modules, you can use the **start** and **stop** to begin or end a diagnostic test.

To start or stop an online diagnostic command, perform one of these tasks:

Command	Purpose
<code>diagnostic start {module num} test {test_id   test_id_range   minimal   complete   basic   per-port   non-disruptive   all} [port {num   port#_range   all}]</code>	Starts a diagnostic test on a specific module and port or range of ports.
<code>diagnostic stop {module num}</code>	Stops a diagnostic test on a specific module.

This example shows how to start a diagnostic test on a specific module:

```
Router# diagnostic start module 1 test 5
Module 1:Running test(s) 5 may disrupt normal system operation
Do you want to run disruptive tests? [no]yes
00:48:14:Running OnDemand Diagnostics [Iteration #1] ...
00:48:14:%DIAG-SP-6-TEST_RUNNING:Module 1:Running TestNewLearn{ID=5} ...
00:48:14:%DIAG-SP-6-TEST_OK:Module 1:TestNewLearn{ID=5} has completed successfully
00:48:14:Running OnDemand Diagnostics [Iteration #2] ...
00:48:14:%DIAG-SP-6-TEST_RUNNING:Module 1:Running TestNewLearn{ID=5} ...
00:48:14:%DIAG-SP-6-TEST_OK:Module 1:TestNewLearn{ID=5} has completed successfully
Router#
```

This example shows how to stop a diagnostic test on a specific module:

```
Router# diagnostic stop module 3
Router#
```

## Displaying Online Diagnostic Tests and Test Results

You can display the online diagnostic tests that are configured for specific modules and check the results of the tests using the **show** commands.

To display the diagnostic tests that are configured for a module, perform this task:

Command	Purpose
<code>show diagnostic content [module num]</code>	Displays the online diagnostics configured for a module.

This example shows how to display the online diagnostics that are configured on a module:

```
Router# show diagnostic content module 7
```

```
Module 7:
```

```
Diagnostics test suite attributes:
```

```
M/C/* - Minimal bootup level test / Complete bootup level test / NA
B/* - Basic ondemand test / NA
P/V/* - Per port test / Per device test / NA
D/N/* - Disruptive test / Non-disruptive test / NA
S/* - Only applicable to standby unit / NA
X/* - Not a health monitoring test / NA
F/* - Fixed monitoring interval test / NA
E/* - Always enabled monitoring test / NA
A/I - Monitoring is active / Monitoring is inactive
R/* - Power-down line cards and need reset supervisor / NA
K/* - Require resetting the line card after the test has completed / NA
```

ID	Test Name	Attributes	Testing Interval (day hh:mm:ss.ms)
1)	TestScratchRegister	***N***A**	000 00:00:30.00
2)	TestSPRPInbandPing	***N***A**	000 00:00:15.00
3)	TestTransceiverIntegrity	**PD***I**	not configured
4)	TestActiveToStandbyLoopback	M*PD***I**	not configured
5)	TestLoopback	M*PD***I**	not configured
6)	TestNewLearn	M**N***I**	not configured
7)	TestIndexLearn	M**N***I**	not configured
8)	TestDontLearn	M**N***I**	not configured
9)	TestConditionalLearn	M**N***I**	not configured
10)	TestBadBpdu	M**D***I**	not configured
11)	TestTrap	M**D***I**	not configured
12)	TestMatch	M**D***I**	not configured
13)	TestCapture	M**D***I**	not configured
14)	TestProtocolMatch	M**D***I**	not configured
15)	TestChannel	M**D***I**	not configured
16)	TestFibDevices	M**N***I**	not configured
17)	TestIPv4FibShortcut	M**N***I**	not configured
18)	TestL3Capture2	M**N***I**	not configured
19)	TestIPv6FibShortcut	M**N***I**	not configured
20)	TestMPLSFibShortcut	M**N***I**	not configured
21)	TestNATFibShortcut	M**N***I**	not configured
22)	TestAclPermit	M**N***I**	not configured
23)	TestAclDeny	M**D***I**	not configured
24)	TestQoS Tcam	M**D***I**	not configured
25)	TestL3VlanMet	M**N***I**	not configured
26)	TestIngressSpan	M**N***I**	not configured
27)	TestEgressSpan	M**N***I**	not configured
28)	TestNetflowInlineRewrite	C*PD***I**	not configured
29)	TestFabricSnakeForward	M**N***I**	not configured
30)	TestFabricSnakeBackward	M**N***I**	not configured
31)	TestFibTcamSSRAM	***D***IR*	not configured
32)	ScheduleSwitchover	***D***I**	not configured

```
Router#
```

This example shows how to display the online diagnostic results for a module:

```
Router# show diagnostic result module 5
Current bootup diagnostic level:minimal
```

```
Module 5:
```

```
Overall Diagnostic Result for Module 5 :PASS
Diagnostic level at card bootup:minimal
```

```
Test results:(. = Pass, F = Fail, U = Untested)
```

```
1) TestScratchRegister -----> .
2) TestSPRPInbandPing -----> .
3) TestGBICIntegrity:
```

```
Port 1 2
-----
      U U
```

```
4) TestActiveToStandbyLoopback:
```

```
Port 1 2
-----
      U U
```

```
5) TestLoopback:
```

```
Port 1 2
-----
      . .
```

```
6) TestNewLearn -----> .
7) TestIndexLearn -----> .
8) TestDontLearn -----> .
9) TestConditionalLearn -----> .
10) TestBadBpdu -----> .
11) TestTrap -----> .
12) TestMatch -----> .
13) TestCapture -----> .
14) TestProtocolMatch -----> .
15) TestChannel -----> .
16) TestIPv4FibShortcut -----> .
17) TestL3Capture2 -----> .
18) TestL3VlanMet -----> .
19) TestIngressSpan -----> .
20) TestEgressSpan -----> .
21) TestIPv6FibShortcut -----> .
22) TestMPLSFibShortcut -----> .
23) TestNATFibShortcut -----> .
24) TestAclPermit -----> .
25) TestAclDeny -----> .
26) TestQoSSTcam -----> .
27) TestNetflowInlineRewrite:
```

```
Port 1 2
-----
      U U
```



```

28) TestFabricSnakeForward -----> .
29) TestFabricSnakeBackward -----> .
30) TestFibTcam - RESET -----> U
Router#

```

This example shows how to display the detailed online diagnostic results for a module:

```

Router# show diagnostic result module 5 detail
Current bootup diagnostic level:minimal

```

Module 5:

```

Overall Diagnostic Result for Module 5 :PASS
Diagnostic level at card bootup:minimal

```

Test results:(. = Pass, F = Fail, U = Untested)

---

```

1) TestScratchRegister -----> .

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 330
Last test execution time ----> May 12 2003 14:49:36
First test failure time ----> n/a
Last test failure time ----> n/a
Last test pass time -----> May 12 2003 14:49:36
Total failure count -----> 0
Consecutive failure count ---> 0

```

---

```

2) TestSPRPInbandPing -----> .

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 660
Last test execution time ----> May 12 2003 14:49:38
First test failure time ----> n/a
Last test failure time ----> n/a
Last test pass time -----> May 12 2003 14:49:38
Total failure count -----> 0
Consecutive failure count ---> 0

```

---

3) TestGBICIntegrity:

```

Port 1 2
-----
      U U

```

```

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 0
Last test execution time ----> n/a
First test failure time ----> n/a
Last test failure time ----> n/a
Last test pass time -----> n/a
Total failure count -----> 0
Consecutive failure count ---> 0

```

---

Router#

# Schedule Switchover

The schedule switchover test is used to check the readiness of the standby supervisor engine to take over in case the active supervisor engine fails or is taken out of service. You can run this test once or schedule it to run on a regular (daily, weekly, or monthly) basis.



## Note

When setting the time for a schedule switchover on both supervisor engines, the switchover for the active and standby supervisor engines should be scheduled at least 10 minutes apart to reduce system downtime if the switchover fails.

To configure a schedule switchover, perform this task:

	Command	Purpose
Step 1	<code>show diagnostic content [module num]</code>	Displays the online diagnostics configured for a module. Use this command to obtain the test ID for the schedule switchover.
Step 2	<code>Router(config)# diagnostic schedule module {num   active-sup-slot} test {test-id} {on mm dd yyyy hh:mm}   {daily hh:mm }   {weekly day-of-week hh:mm}</code>	Sets up the schedule switchover test for a specific date and time for the supervisor engine.

This example shows how to schedule a switchover for the active supervisor engine every Friday at 10:00 PM, and switch the standby supervisor engine back to the active supervisor engine 10 minutes after the scheduled switchover from the active supervisor engine occurs.

```
Router(config)# diagnostic schedule module 5 test 32 weekly Friday 22:00
Router(config)# diagnostic schedule module 6 test 32 weekly Friday 22:10
Router(config)#
```

## Performing Memory Tests

Most online diagnostic tests do not need any special setup or configuration. However, the memory tests, which include the TestFibTcamSSRAM and TestLinecardMemory tests, have some required tasks and some recommended tasks that you should complete before running them.

Before you run any of the online diagnostic memory tests, perform the following tasks:

- Required tasks
  - Isolate network traffic by disabling all connected ports.
  - Do not send test packets during a memory test.
  - Remove all switching modules for testing FIB TCAM and SSRAM on the policy feature card (PFC3B) of the supervisor engine.
  - Reset the system or the module you are testing before returning the system to normal operating mode.



## Note

Turn off all background health-monitoring tests on the supervisor engine and switching modules using the `no diagnostic monitor module num test all` command.