

Hard & Soft Logic

Introduction

The current state of monitored services and hosts is determined by two components:

- The status of the check or host (i.e. **OK**, **WARNING**, **UP**, **DOWN**, etc.)
- The type of state the check or host is in.

There are two state types in Shinken Enterprise - **SOFT** states and **HARD** states. These state types are a crucial part of the monitoring logic, as they are used to determine when event handlers are executed and when notifications are initially sent out.

This document describes the difference between **SOFT** and **HARD** states, how they occur, and what happens when they occur.

Service and Host Check Retries

In order to prevent false alarms from transient problems, Shinken Enterprise allows you to define how many times a service or host should be (re)checked before it is considered to have a "real" problem. This is controlled by the `max_check_attempts` option in the host and service definitions. Understanding how hosts and services are (re)checked in order to determine if a real problem exists is important in understanding how state types work.

Soft States

Soft states occurs in the following situations...

- When a service or host check results in a non-OK or non-UP state and the service check has not yet been (re)checked the number of times specified by the `max_check_attempts` directive in the service or host definition. This is called a soft error.
- When a service or host recovers from a soft error. This is considered as a soft recovery.

The following things occur when hosts or services experience SOFT state changes:

- The SOFT state is logged.
- Event handlers are executed to handle the SOFT state.

SOFT states are only logged if you enabled the `log_service_retries` or `log_host_retries` options in your main configuration file.

The only important thing that really happens during a soft state is the execution of event handlers. Using event handlers can be particularly useful if you want to try and proactively fix a problem before it turns into a HARD state. The `$HOSTSTATETYPE$` or `$SERVICESTATETYPE$` macros will have a value of "**SOFT**" when event handlers are executed, which allows your event handler scripts to know when they should take corrective action. More information on event handlers can be found [:ref:`here <advanced/eventhandlers>`](#).

Hard States

Hard states occur for hosts and services in the following situations:

- When a host or service check results in a non-UP or non-OK state and it has been (re)checked the number of times specified by the `max_check_attempts` option in the host or service definition. This is a hard error state.
- When a host or service transitions from one hard error state to another error state (e.g. WARNING to CRITICAL).
- When a service check results in a non-OK state and its corresponding host is either DOWN or UNREACHABLE.
- When a host or service recovers from a hard error state. This is considered to be a hard recovery.
- When a passive host check is received. Passive host checks are treated as HARD.

The following things occurs when hosts or services experience HARD state changes:

- The HARD state is logged.
- Event handlers are executed to handle the HARD state.
- Contacts are notified about the host or check problem or recovery.

The `$HOSTSTATETYPE$` or `$SERVICESTATETYPE$` data will have a value of **"HARD"** when event handlers are executed, which allows your event handler scripts to know when they should take corrective action.

Example

Here's an example of how state types are determined, when state changes occur, and when event handlers and notifications are sent out. The table below shows consecutive checks of a check over time. The check has a `max_check_attempts` value of **3**.

Time	Check Number	State Type	Type State	Changes	Notes
0	1	OK	HARD	No	Initial state
1	1	CRITICAL	SOFT	Yes	First detection of a non-OK state. Event handlers execute.
2	2	WARNING	SOFT	Yes	Check continues to be in a non-OK state. Event handlers execute.
3	3	CRITICAL	HARD	Yes	Max check attempts has been reached, so check goes into a HARD state. Event handlers execute and a problem notification is sent out. Check number is reset to 1 immediately after this happens.
4	1	WARNING	HARD	Yes	Check changes to a HARD WARNING state. Event handlers execute and a problem notification is sent out.
5	1	WARNING	HARD	No	Check stabilizes in a HARD problem state. Depending on what the notification interval for the service is, another notification might be sent out.
6	1	OK	HARD	Yes	Check experiences a HARD recovery. Event handlers execute and a recovery notification is sent out.
7	1	OK	HARD	No	Check is still OK.
8	1	UNKNOWN	SOFT	Yes	Check is detected as changing to a SOFT non-OK state. Event handlers execute
9	2	OK	SOFT	Yes	Check experiences a SOFT recovery. Event handlers execute, but notification are not sent, as this wasn't a "real" problem. State type is set HARD and check number is reset to 1 immediately after this happens.
10	1	OK	HARD	No	Check stabilizes in an OK state.