

System Configuration Parameters

This chapter describes the NVRAM configuration variables and OpenBoot PROM (OBP) commands available for configuring the following aspects of Ultra 450 system behavior:

- UPA probing
- PCI probing
- Memory interleaving
- Environmental monitoring and control
- Automatic System Recovery (ASR)

NVRAM configuration variables covered in this chapter include:

- `upa-port-skip-list`
- `pci0-probe-list`
- `pci-slot-skip-list`
- `memory-interleave`
- `env-monitor`
- `asr-disable-list`
- `auto-boot-on-error?`
- `diag-trigger`

OBP commands covered in this chapter include:

- `asr-enable`
- `asr-disable`
- `.asr`

UPA Probing

Ultra 450 systems, like all UltraSPARC™-based systems, are based on the high-speed Ultra Port Architecture (UPA) bus, a switched system bus that provides up to 32 port ID addresses (or slots) for high-speed motherboard devices like CPUs, I/O bridges, and frame buffers. While most Ultra systems employ only three or four active UPA ports, an Ultra 450 system provides up to nine active ports spread among the following subsystems.

Device Type	UPA Slot	Physical Implementation
CPU	0-3	Four plug-in slots
UPA-PCI bridge	4,6,1f	Soldered on motherboard
UPA graphics frame buffer	1d, 1e	Two plug-in slots

The order of probing these nine port IDs is *not* subject to user control, however a list of ports *can be excluded* from probing via the `upa-port-skip-list` NVRAM variable. In the following example, the `upa-port-skip-list` variable is used to exclude one of the UPA-PCI bridges and the primary UPA graphics card from the UPA probe list.

```
ok setenv upa-port-skip-list 4,1d
```

This capability lets you exclude a given device from probing (and subsequent use) by the system without physically removing the plug-in card. This can be useful in helping to isolate a failing card in a system experiencing transient failures.

PCI Probing

Of the Ultra 450 system's six PCI buses, Bus 0 (`/pci@1f,4000` in the device tree) is unique in that it is the only PCI bus that contains motherboard (non plug-in) devices such as standard Ethernet and SCSI controllers. By definition, these devices cannot be unplugged and swapped to change the order in which they are probed. To control the probing order of these devices, the system provides the NVRAM variable

`pci0-probe-list`. This variable controls both the probing order and exclusion of devices on PCI Bus 0. The values in the `pci0-probe-list` are defined in the following table.

PCI Device Number	Function
0	UPA-PCI bus bridge (not probed)
1	EBus/Ethernet interface (always probed, never included in probe list)
2	On-board SCSI controller for removable media devices and external SCSI port
3	On-board SCSI controller for 4-slot UltraSCSI backplane
4	Back panel PCI slot 10

Note – The values in this list are based on the PCI device number and do *not* refer to the back panel slot numbering scheme of 1–10.

In the following example, the `pci0-probe-list` variable is used to define a probing order of 3-4, while excluding from the probe list the on-board SCSI controller for removable media devices and the external SCSI port.

```
ok setenv pci0-probe-list 3,4
```

The probing order of the other five PCI buses (PCI slots 1 through 9) is *not* subject to user control. These slots are always probed in the following order: 5-3-2-1-4-9-8-7-6. However, a list of PCI slots *can be excluded* from probing via the NVRAM variable `pci-slot-skip-list`. In the following example, the `pci-slot-skip-list` variable is used to exclude back panel slots 3 and 8 from the PCI probe list.

```
ok setenv pci-slot-skip-list 3,8
```

Note – The values in the `pci-slot-skip-list` correspond to the back panel numbering scheme of 1–10. If slot 10 is in this list, then it will be excluded from probing even if `pci0-probe-list` includes device number 4 (back panel slot 10).

Memory Interleaving

Memory interleaving in an Ultra 450 system is controlled by the NVRAM variable `memory-interleave`. The following table shows the various settings for this variable and the effect each setting has on memory configuration. For a related discussion about memory interleaving and memory configuration guidelines, see “About Memory” in the owner’s guide provided with your Ultra 450 system.

Setting	Effect on Memory Configuration
<code>auto</code> (<i>default</i>)	Enables four-way interleaving if all four memory banks contain identical capacity DIMMs. Enables two-way interleaving if only Banks A and B are used and both banks contain identical capacity DIMMs. Otherwise, interleaving is disabled.
<code>max-size</code>	Same as <code>auto</code> setting for Ultra 450 systems.
<code>max-interleave</code>	Enables the maximum level of interleaving possible for a given memory configuration, but some memory capacity remains unused if DIMMs of different capacities are installed. Within each DIMM, uses an amount of memory equal to the smallest capacity DIMM installed.
1	Disables interleaving; uses all of the available memory capacity.
2	Forces two-way interleaving between Banks A and B. Some memory capacity remains unused if DIMMs of different capacities are installed. The smallest capacity DIMM must be installed in Bank B. Banks C and D, if populated, remain unused.
4	Forces four-way interleaving between all four banks. Some memory capacity remains unused if DIMMs of different capacities are installed. The smallest capacity DIMM must be installed in Bank D.

The following example shows how to configure the system for maximum memory interleaving.

```
ok setenv memory-interleave max-interleave
```

Environmental Monitoring and Control

Environmental monitoring and control capabilities for an Ultra 450 system reside at both the operating system level and the OBP firmware level. This ensures that monitoring capabilities are operational even if the system has halted or is unable to boot. The way in which OBP monitors and reacts to environmental over temperature conditions is controlled by the NVRAM variable `env-monitor`. The following table shows the various settings for this variable and the effect each setting has on OBP behavior. For additional information about the system's environmental monitoring capabilities, see "About Reliability, Availability, and Serviceability Features" in the owner's guide provided with your Ultra 450 system.

Setting	Monitor Active?	Action Taken
<code>enabled</code> (<i>default</i>)	Yes	In response to an over temperature condition or a fan failure in either the CPU or disk fan tray, OBP issues a warning and automatically shuts down the system after 30 seconds.
<code>advise</code>	Yes	OBP issues a warning only, without shutting down the system.
<code>disabled</code>	No	OBP takes no action at all; environmental monitoring at the OBP level is disabled.

In the following example, the `env-monitor` variable is used to disable environmental monitoring at the OBP level.

```
ok setenv env-monitor disabled
```

Note – This NVRAM variable does not affect the system's environmental monitoring and control capabilities while the operating system is running.

Automatic System Recovery

The automatic system recovery (ASR) feature allows an Ultra 450 system to resume operation after experiencing certain hardware faults or failures. Power-on self-test (POST) and OpenBoot diagnostics (OBDDiag) can automatically detect failed hardware components, while an auto-configuring capability designed into the OBP

firmware allows the system to deconfigure failed components and restore system operation. As long as the system is capable of operating without the failed component, the ASR features will enable the system to reboot automatically, without operator intervention. Such a “degraded boot” allows the system to continue operating while a service call is generated to replace the faulty part.

If a faulty component is detected during the power-on sequence, the component is deconfigured and, if the system remains capable of functioning without it, the boot sequence continues. In a running system, certain types of failures (such as a processor failure) can cause an automatic system reset. If this happens, the ASR functionality allows the system to reboot immediately, provided that the system can function without the failed component. This prevents a faulty hardware component from keeping the entire system down or causing the system to crash again.

“Soft” Deconfiguration via Status Property

To support a degraded boot capability, the OBP uses the 1275 Client Interface (via the device tree) to “mark” devices as either *failed* or *disabled*, by creating an appropriate “status” property in the corresponding device tree node. By convention, UNIX will not activate a driver for any subsystem so marked.

Thus, as long as the failed component is electrically dormant (that is, it will not cause random bus errors or signal noise, etc.), the system can be rebooted automatically and resume operation while a service call is made.

“Hard” Deconfiguration

In two special cases of deconfiguring a subsystem (CPUs and memory), the OBP actually takes action beyond just creating an appropriate “status” property in the device tree. At the first moments after reset, the OBP must initialize and functionally configure (or bypass) these functions in order for the rest of the system to work correctly. These actions are taken based on the status of two NVRAM configuration variables, `post-status` and `asr-status`, which hold the override information supplied either from POST or via a manual user override (see “ASR User Override Capability” on page 11).

CPU Deconfiguration

If any CPU is marked as having failed POST, or if a user chooses to disable a CPU, then the OBP will set the Master Disable bit of the affected CPU, which essentially turns it off as an active UPA device until the next power-on system reset.

Memory Deconfiguration

Detecting and isolating system memory problems is one of the more difficult diagnostic tasks. This problem is further complicated by the system's various modes of memory interleaving as well as the possibility of mismatching memory DIMMs within the same bank.

Given a failed memory component, the firmware will deconfigure the entire bank associated with the failure. This policy also means that the degraded configuration may mean a lower interleave factor, a less than 100 percent utilization of remaining banks, or both depending on the interleave factor.

ASR User Override Capability

While the default settings will properly configure or deconfigure an Ultra 450 system in most cases, it is useful to provide advanced users with a manual override capability. Because of the nature of “soft” versus “hard” deconfiguration, it is necessary to provide two related but different override mechanisms.

“Soft” Deconfigure Override

For any subsystem represented by a distinct device tree node, users may disable that function via the NVRAM variable `asr-disable-list`, which is simply a list of device tree paths separated by spaces.

```
ok setenv asr-disable-list /pci/ebus/ecpp /pci@1f,4000/scsi@3
```

The Ultra 450 OBP will use this information to create disabled status properties for each node listed in the variable `asr-disable-list`.

“Hard” Deconfigure Override

For overriding those subsystems that require “hard” deconfiguration (CPU and memory), the OBP commands `asr-enable` and `asr-disable` are used to selectively enable or disable each subsystem.

Note – There are duplications between the soft and hard overrides. If possible, the hard override commands `asr-enable` and `asr-disable` should be used.

To keep track of the status of all manual overrides, a new user command, `.asr`, is provided to summarize the current settings.

```
ok asr-disable cpu1 bank3
ok .asr
CPU0:                Enabled
CPU1:                Disabled
CPU2:                Enabled
CPU3:                Enabled
SC-Marvin:           Enabled
Psycho@1f:           Enabled
Psycho@4:            Enabled
Psycho@6:            Enabled
Cheerio:             Enabled
SCSI:                Enabled
Mem Bank0:           Enabled
Mem Bank1:           Enabled
Mem Bank2:           Enabled
Mem Bank3:           Disabled
PROM:                Enabled
NVRAM:               Enabled
TTY:                 Enabled
Audio:               Enabled
SuperIO:             Enabled
PCI Slots:           Enabled
```

Auto-Boot Options

OpenBoot provides for an NVRAM controlled switch called `auto-boot?`, which controls whether OBP will automatically boot the operating system after each reset. The default for Sun platforms is `true`.

If a system fails power-on diagnostics, then `auto-boot?` is ignored and the system does not boot unless the user does it manually. This behavior is obviously not acceptable for a degraded boot scenario, so the Ultra 450 OBP provides a second NVRAM-controlled switch called `auto-boot-on-error?`. This switch controls whether the system will attempt a degraded boot when a subsystem failure is detected. Both the `auto-boot?` and `auto-boot-on-error?` switches must be set to `true` to enable a degraded boot.

```
ok setenv auto-boot-on-error? true
```

Note – The default setting for `auto-boot-on-error?` is `false`. Therefore, the system will not attempt a degraded boot unless you change this setting to `true`. In addition, the system will not attempt a degraded boot in response to any fatal unrecoverable error, even if degraded booting is enabled. An example of a fatal unrecoverable error is when all of the system’s CPUs have been disabled, either by failing POST or as a result of a manual user override.

Reset Scenarios

The standard system reset protocol bypasses firmware diagnostics completely unless the NVRAM variable `diag-switch?` is set to `true`. The default setting for this variable is `false`.

To support ASR in an Ultra 450 system, it is desirable to be able to run firmware diagnostics (POST/OBDiag) on any or all reset events. Rather than simply changing the default setting of `diag-switch?` to `true`, which carries with it other side effects (see the *OpenBoot 3.x Command Reference Manual*), the Ultra 450 OBP provides a new NVRAM variable called `diag-trigger` that lets you choose which reset events, if any, will automatically engage POST/OBDiag. The `diag-trigger` variable, and its various settings are described in the following table.

Note – `diag-trigger` has no effect unless `diag-switch?` is set to `true`.

Setting	Function
<code>power-reset</code> <i>(default)</i>	Runs diagnostics only on power-on resets.
<code>error-reset</code>	Runs diagnostics only on power-on resets, fatal hardware errors, and watchdog reset events.
<code>soft-reset</code>	Runs diagnostics on all resets (except XIR), including resets triggered by UNIX <code>init 6</code> or <code>reboot</code> commands.
<code>none</code>	Disables the automatic triggering of diagnostics by any reset event. Users can still invoke diagnostics manually by holding down the Stop and d keys when powering on the system, or by turning the front panel keyswitch to the Diagnostics position when powering on the system.

In the following example, the `diag-trigger` variable is used to trigger POST and OpenBoot diagnostics on all resets except XIR resets.

```
ok setenv diag-switch? true
ok setenv diag-trigger soft-reset
```