



Netra™ 240 伺服器系統管理指南

Sun Microsystems, Inc.
www.sun.com

文件號碼：817-5017-11
2004 年 7 月，修訂版 A

將您對此文件的意見傳送到：<http://www.sun.com/hwdocs/feedback>

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版權所有。

Sun Microsystems, Inc. 對於本文件提及的技術擁有其相關智慧財產權。這些智慧財產權包括 <http://www.sun.com/patens> 所列之一項或多項美國專利，以及在美國與其他國家已經申請到或正在申請的一項或多項專利，但並不以此為限。

本文件及相關產品在限制其使用、複製、發行及反編譯的授權下發行。未經 Sun 及其授權人 (如果適用) 事先的書面許可，不得使用任何方法以任何形式來複製本產品或文件的任何部分。

協力廠商軟體，包含字型技術，其著作權歸 Sun 供應商所有，經授權後使用。

本產品中的某些部分可能衍生自加州大學授權的 Berkeley BSD 系統的開發成果。UNIX 為在美國與其他國家的註冊商標，並已向 X/Open Company, Ltd. 取得獨家授權。

Sun、Sun Microsystems、Sun 標誌、AnswerBook2、docs.sun.com、OpenBoot、Netra、SunVTS、Sun Enterprise Authentication Mechanism 及 Solaris 是 Sun Microsystems, Inc. 在美國和其他國家的商標或註冊商標。

所有的 SPARC 商標都按授權許可使用，是 SPARC International, Inc. 在美國和其他國家的商標或註冊商標。具有 SPARC 商標的產品都是採用 Sun Microsystems, Inc. 開發的架構。

OPEN LOOK 與 Sun™ Graphical User Interface 是由 Sun Microsystems, Inc. 為其使用者與下游申請授權者所開發的。Sun 公司感謝 Xerox 公司在研究和開發視覺化或圖形化使用者介面概念方面，為電腦工業所作的先驅性努力。Sun 擁有經 Xerox 授權的 Xerox 圖形使用者介面非專屬授權，該授權亦涵蓋使用 OPEN LOOK GUI 並遵守 Sun 書面授權合約的 Sun 公司授權者。

美國政府權限 — 商業用途。政府人員應遵守 Sun Microsystems, Inc. 的標準授權合約，而且適用於 FAR 及其補充文件的條款。

此處提供之物品僅以本文件為限，任何明確或隱含之條件、陳述與擔保，包括任何隱含之銷售保固、適合某特定目的或無傷害保證，均屬無效，但若法律裁定此無效聲明違法，則不在此限。



Adobe PostScript

目錄

前言 xi

1. 疑難排解工具 1
 - 診斷工具概觀 2
 - 系統提示 2
 - 進階無電管理軟體 3
 - 伺服器狀態指示燈 4
 - ▼ 顯示位置指示 LED 狀態 4
 - ▼ 開啓位置指示 LED 5
 - ▼ 關閉位置指示 LED 5
 - 警報狀態指示燈 6
 - 開機自我測試 7
 - 控制 POST 診斷 8
 - ▼ 啓動 POST 診斷 9
 - OpenBoot 指令 10
 - probe-scsi 與 probe-scsi-all 指令 10
 - probe-ide 指令 11
 - show-devs 指令 12
 - ▼ 執行 OpenBoot 指令 13

OpenBoot 診斷	13
▼ 啓動 OpenBoot 診斷	13
控制 OpenBoot 診斷測試	14
test 與 test-all 指令	15
OpenBoot 診斷錯誤訊息	16
作業環境診斷工具	17
錯誤與系統訊息記錄檔	17
Solaris 軟體系統資訊指令	17
prtconf 指令	17
prtdiag 指令	19
prtfru 指令	21
psrinfo 指令	22
showrev 指令	23
▼ 執行 Solaris 平台系統資訊指令	24
最新診斷測試結果	24
▼ 檢視最新測試結果	24
OpenBoot 組態變數	25
▼ 檢視與設定 OpenBoot 組態變數	25
使用 watch-net 與 watch-net-all 指令檢查網路連線	26
自動系統回復	27
Auto-Boot 選項	27
錯誤處理摘要	28
重新開機程序	28
▼ 啓用 ASR	29
▼ 停用 ASR	29
2. SunVTS 軟體	31
SunVTS 軟體概觀	31
SunVTS 測試	32

SunVTS 軟體與安全性	33
▼ 確定 SunVTS 軟體是否已安裝	33
安裝 SunVTS 軟體	33
檢視 SunVTS 軟體文件	34
3. 進階無電管理軟體	35
進階無電管理軟體概觀	35
ALOM 連接埠	36
設定 admin 密碼	36
基本的 ALOM 功能	37
▼ 切換至 ALOM 提示	37
▼ 切換至伺服器主控台提示	37
▼ 取消其他使用者的主控台寫入能力	38
自動伺服器重新啓動	38
環境監視與控制	39
A. 警報繼電器輸出應用程式介面	41
索引	47



-
- 圖 1-1 系統提示流程圖 3
 - 圖 1-2 前面板指示燈位置 4

表

表 1-1	疑難排解工具摘要	2
表 1-2	伺服器狀態指示燈 (前後面板)	4
表 1-3	警報指示燈及乾接點警報狀態	6
表 1-4	OpenBoot 組態變數	8
表 1-5	OpenBoot 組態變數 <code>test-args</code> 關鍵字	15
表 1-6	Solaris 平台資訊顯示指令	24
表 2-1	SunVTS 軟體測試	32
表 3-1	由 ALOM 監控的元件	36
表 3-2	Netra 240 伺服器外殼溫度臨界值	39

前言

《Netra 240 伺服器系統管理指南》是要提供給有經驗的系統管理員使用。並包含 Netra™ 240 伺服器診斷工具及各種伺服器管理工作的一般說明。

若要使用本指南中的資訊，您必須具有電腦網路概念與術語的實際應用知識，以及 Solaris™ 作業環境 (Solaris OS) 的進階知識。

閱讀本書之前

本書並未涵蓋伺服器安裝與機架安裝，如需關於這些主題的詳細資訊，請參閱 《Netra 240 伺服器安裝指南》(文件號碼 817-5000-11)。

執行本書所述的任何程序之前，請確認您已閱讀 《Important Safety Information for Sun Hardware Systems》(文件號碼 816-7190-10)。

使用 UNIX 指令

使用本節警示讀者此文件未提供所有 UNIX 指令。

例如：

本文件並未包含基本 UNIX[®] 指令和程序的相關資訊，例如系統關機、系統開機及設定裝置。如虛此資訊，請參閱下列文件：

- 系統隨附的軟體文件
- Solaris[™] 作業環境文件，位於：

<http://docs.sun.com>

Shell 提示

Shell	提示
C shell	機器名稱 %
C shell 超級使用者	機器名稱 #
Bourne shell 與 Korn shell	\$
Bourne shell 與 Korn shell 超級使用者	#

排版慣例

字體 ⁱ	含義	範例
AaBbCc123	指令、檔案和目錄的名稱；電腦的螢幕輸出	請編輯您的 <code>.login</code> 檔。 請使用 <code>ls -a</code> 列出所有的檔案。 % You have mail.
AaBbCc123	您所鍵入的內容（相對於電腦的螢幕輸出）	% su Password:
<i>AaBbCc123</i>	書名、新的字彙或術語、要強調的字彙。將用實際的名稱或數值取代指令行變數。	請閱讀《 <i>使用者指南</i> 》的第六章。 它們稱為 <i>class</i> 選項。 您必須是超級使用者才能執行此操作。 若要刪除檔案，請鍵入 <code>rm filename</code> 。

i 您瀏覽器上的設定可能與這些設定不同。

相關文件

應用	書名	文件號碼
安裝概觀	<i>Netra 240 Server Quick Start Guide</i>	817-3904-xx
最新產品更新	<i>Netra 240 Server Release Notes</i>	817-3142-xx
符合性與安全	<i>Important Safety Information for Sun Hardware Systems</i>	816-7190-10
	<i>Netra 240 Server Safety and Compliance Manual</i>	817-5018-11
文件網站位置	<i>Sun Netra 240 Server Product Documentation</i>	817-2697-10
安裝	<i>Netra 240 伺服器安裝指南</i>	817-5000-11
進階無電管理	<i>Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南</i>	817-5009-11
維修	<i>Netra 240 Server Service Manual</i>	817-2699-xx

如何取得 Sun 的說明文件

您可以在下列網站檢視、列印及購買各種 Sun 文件 (包括本土化版本)：

<http://www.sun.com/documentation>

協力廠商網站

Sun 對於本文件所述之協力廠商網站的可用性，概不負責。Sun 對於這種網站或資源所提供的任何內容、廣告、產品或其他資料，不會為其背書，也不會為其擔負任何責任。對於使用這種網站或資源所提供的任何內容、貨品或服務所造成的任何實際或辯稱的損壞或損失，Sun 概不負責。

與 Sun 技術支援部門聯繫

如果您有任何關於此產品的技術問題，但此文件中未解答，請前往：

<http://www.sun.com/service/contacting>

Sun 歡迎您提出寶貴意見

Sun 非常樂於提高文件品質，誠心歡迎您的建議與意見。您可以到以下網頁，傳送您的意見：

<http://www.sun.com/hwdocs/feedback>

請在您的意見上註明文件的標題及號碼：

Netra 240 伺服器系統管理指南，文件號碼 817-5017-11

疑難排解工具

本章說明 Netra 240 伺服器可用的診斷工具。本章包含下列幾節：

- 第 2 頁的「診斷工具概觀」
- 第 2 頁的「系統提示」
- 第 3 頁的「進階無電管理軟體」
- 第 7 頁的「開機自我測試」
- 第 10 頁的「OpenBoot 指令」
- 第 13 頁的「OpenBoot 診斷」
- 第 17 頁的「作業環境診斷工具」
- 第 24 頁的「最新診斷測試結果」
- 第 25 頁的「OpenBoot 組態變數」
- 第 27 頁的「自動系統回復」

診斷工具概觀

Sun 提供各種 Netra 240 伺服器使用的診斷工具，請見下表中的摘要說明。

表 1-1 疑難排解工具摘要

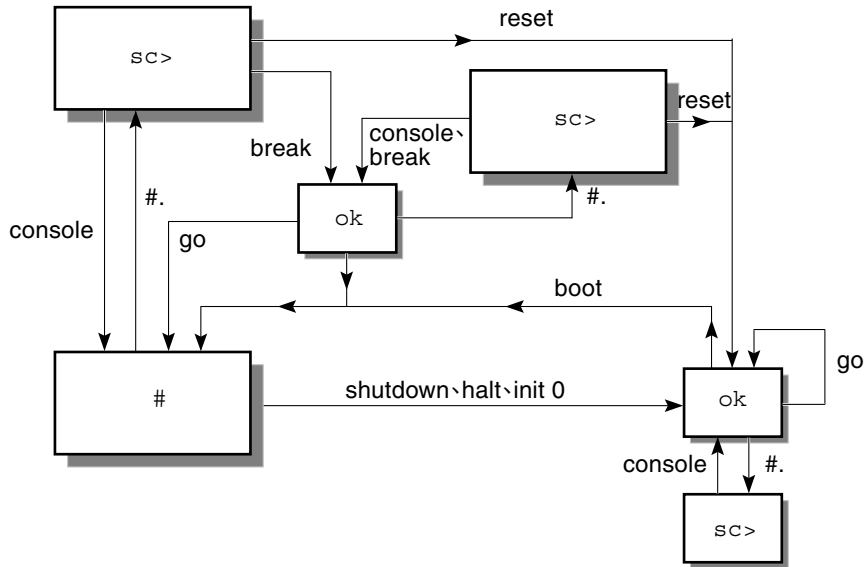
診斷工具	類型	說明	可存取性與可用性	遠端功能
ALOM	硬體與軟體	監控環境條件、執行基本的故障隔離及提供遠端主控台存取。	可以在待命電源模式與沒有作業系統時運作。	專為遠端存取設計。
LED	硬體	指示整個系統與特定元件的狀態。	可從系統機殼進行維修與更換。可在任何有電源的時候使用。	本機，但可經由 ALOM 檢視。
開機自我測試 (POST)	韌體	測試系統的核心元件。	在啟動時自動執行。可在作業系統未執行時使用。	本機，但可經由 ALOM 檢視。
OpenBoot 指令	韌體	顯示各種系統資訊。	可在作業系統未執行時使用。	本機，但可經由 ALOM 存取。
OpenBoot 診斷	韌體	測試系統元件，主要針對週邊設備與 I/O 裝置。	自動或互動執行。可在作業系統未執行時使用。	本機，但可經由 ALOM 檢視。
Solaris 軟體指令	軟體	顯示各種系統資訊。	需要作業系統。	本機，但可經由 ALOM 存取。
SunVTS™ 軟體	軟體	使系統運作，在系統負載大增的狀態下，同時執行測試。	需要作業系統。選用套件。	可透過網路進行檢視與控制。

系統提示

Netra 240 伺服器使用下列預設的伺服器提示：

- `ok` — OpenBoot PROM 提示
- `sc>` — 進階無電管理軟體 (ALOM) 提示
- `#` — Solaris 軟體超級使用者 (Bourne 與 Korn shell) 提示

圖 1-1 顯示三種提示之間的關係，並說明如何在三種提示之間變更。



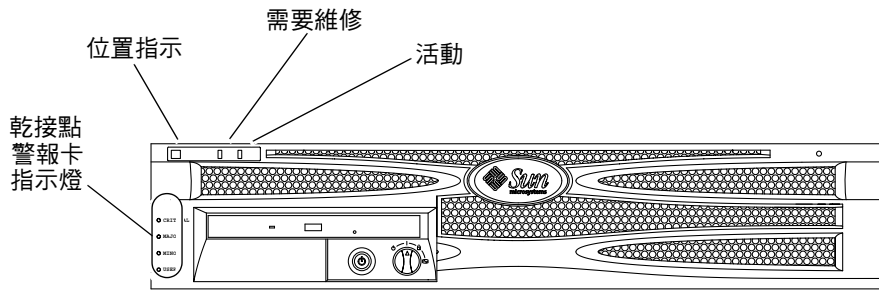
■ 1-1 系統提示流程圖

圖 1-1 所示的流程圖中包含下列指令：

- ALOM 指令：console、reset、break
- 逸出序列：#.
- Solaris 軟體指令：shutdown、halt、init 0
- OpenBoot 指令：go、boot

進階無電管理軟體

Netra 240 伺服器的 Sun™ 進階無電管理軟體 (ALOM) 提供一系列 LED 狀態指示燈，本節詳細說明 LED 狀態的意義，以及如何開啓和關閉 LED。如需有關 ALOM 的詳細資訊，請參閱第 3 章。



■ 1-2 前面板指示燈位置

伺服器狀態指示燈

伺服器上有三種 LED 狀態指示燈，除了位於伺服器的前面板上 (圖 1-2)，後面板上也有一組完全相同的指示燈。表 1-2 中提供指示燈的摘要說明。

表 1-2 伺服器狀態指示燈 (前後面板)

指示燈	LED 顏色	LED 狀態	含義
活動	綠色	亮起	伺服器電源已開啓，且正在執行 Solaris OS。
		熄滅	電源未開啓或 Solaris OS 並未執行。
需要維修	黃色	亮起	伺服器偵測到問題，需要維修人員多加注意。
		熄滅	伺服器未偵測到故障。
位置指示	白色	亮起	使用 <code>setlocator</code> 指令時，LED 燈會持續亮起，以從機架上的所有伺服器中區分出該伺服器。

從系統主控台或 ALOM 指令行介面 (CLI)，都可以開啓及關閉位置指示 LED。

▼ 顯示位置指示 LED 狀態

- 執行下列任一項操作：
 - 以超級使用者身分，鍵入：

```
# /usr/sbin/locator
```

- 在 ALOM 指令行介面鍵入：

```
sc> showlocator
```

▼ 開啓位置指示 LED

- 執行下列任一項操作：
 - 以超級使用者身分，鍵入：

```
# /usr/sbin/locator -n
```

- 在 ALOM 指令行介面鍵入：

```
sc> setlocator on
```

▼ 關閉位置指示 LED

- 執行下列任一項操作：
 - 以超級使用者身分，鍵入：

```
# /usr/sbin/locator -f
```

- 在 ALOM 指令行介面鍵入：

```
sc> setlocator off
```

警報狀態指示燈

乾接點警報卡有四個 ALOM 支援的 LED 狀態指示燈，這些指示燈垂直排列在前面板上 (圖 1-2)。有關警報指示燈及乾接點警報狀態的資訊在表 1-3 中提供。如需關於警報指示燈的詳細資訊，請參閱《Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南》(文件號碼 817-5009-11)。如需關於 API 的詳細資訊以控制警報指示燈，請參閱附錄 A。

表 1-3 警報指示燈及乾接點警報狀態

指示燈 與繼電器 標籤	指示燈 顏色	應用程式 或伺服器狀態	狀況或動作	系統指示 燈狀態	警報指示 燈狀態	繼電器 NC ^{iv} 狀態	繼電器 NO ^v 狀態	註解
嚴重 (Critical) (Alarm0)	紅色	伺服器狀態 (電源開啓/ 關閉，以及 Solaris OS 運作正常/ 不正常)	沒有電源輸入。	熄滅	熄滅	關閉	開啓	預設狀態。
			系統電源關閉。	熄滅	熄滅 ⁱⁱⁱ	關閉	開啓	已連接輸入 電源。
			系統電源開啓； Solaris OS 未完全 載入。	熄滅	熄滅 ⁱⁱⁱ	關閉	開啓	過渡性狀 態。
			Solaris OS 已成功 載入。	亮起	熄滅	開啓	關閉	正常操作狀 態。
			監視程式逾時。	熄滅	亮起	關閉	開啓	過渡性狀 態，重新啓 動 Solaris OS。
			由使用者進行 ⁱ Solaris OS 關機。	熄滅	熄滅 ⁱⁱⁱ	關閉	開啓	過渡性狀 態。
			失去輸入電源。	熄滅	熄滅	關閉	開啓	預設狀態。
		由使用者關閉系 統電源。	熄滅	熄滅 ⁱⁱⁱ	關閉	開啓	過渡性狀 態。	
		應用程式 狀態	使用者啓動嚴重 (Critical) 警報 ⁱⁱ 。	—	亮起	關閉	開啓	偵測到嚴重 故障。
		使用者關閉嚴重 (Critical) 警報 ⁱⁱ 。	—	熄滅	開啓	關閉	嚴重故障已 清除。	
主要 (Major) (Alarm1)	紅色	應用程式 狀態	使用者啓動主要 (Major) 警報 ⁱⁱ 。	—	亮起	開啓	關閉	偵測到主要 故障。
			使用者關閉主要 (Major) 警報 ⁱⁱ 。	—	熄滅	關閉	開啓	主要故障已 清除。

表 1-3 警報指示燈及乾接點警報狀態 (續)

指示燈與繼電器標籤	指示燈顏色	應用程式或伺服器狀態	狀況或動作	系統指示燈狀態	警報指示燈狀態	繼電器 NC ^{iv} 狀態	繼電器 NO ^v 狀態	註解
次要 (Minor) (Alarm2)	黃色	應用程式狀態	使用者啟動次要 (Minor) 警報 ⁱⁱ 。	—	亮起	開啓	關閉	偵測到次要故障。
			使用者關閉次要 (Minor) 警報 ⁱⁱ 。	—	熄滅	關閉	開啓	次要故障已清除。
使用者 (User) (Alarm3)	黃色	應用程式狀態	使用者啟動使用者 (User) 警報 ⁱⁱ 。	—	亮起	開啓	關閉	偵測到使用者故障。
			使用者關閉使用者 (User) 警報 ⁱⁱ 。	—	熄滅	關閉	開啓	使用者故障已清除。

i 使用者可以使用諸如 `init0` 和 `init6` 的指令來關閉系統，這並不包括關閉系統電源。

ii 如果確定發生故障狀況，使用者可以利用 Solaris 平台警報 API 或 ALOM 來啟動警報。如需關於警報 API 的詳細資訊，請參閱附錄 A，如需關於 ALOM CLI 的詳細資訊，則請參閱《Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南》(文件號碼 817-5009-11)。

iii 此警報指示燈狀態的實作方式有可能變更。

iv NC 狀態是常閉狀態，此狀態代表常閉狀態中的繼電器接點預設模式。

v NO 狀態是常開狀態，此狀態代表常開狀態中的繼電器接點預設模式。

在所有情況下，當使用者設定警報時，主控台上都會顯示一個訊息。例如，已設定嚴重警報時，下列訊息便會顯示在主控台上：

```
SC Alert: CRITICAL ALARM is set
```

請注意在某些情況下，即使已設定嚴重警報，對應的警報指示燈也不會亮起。此實作方式可能會在未來版本中變更 (請參閱表 1-3 的註腳 ⁱⁱⁱ)。

開機自我測試

開機自我測試 (POST) 是韌體程式，可協助確定系統的某個部分是否故障。POST 會確認系統的核心功能，包括 CPU 模組、主機板、記憶體和一些內建 I/O 裝置，然後產生有助於確定硬體故障原因的訊息。即使系統無法啟動，您也可以執行 POST。

POST 可偵測大多數系統故障，並位於主機板 OpenBoot PROM 中。可以透過設定兩個環境變數，命令 OpenBoot 軟體在開機時執行 `POST:diag-switch?` 及 `diag-level` 旗標。這兩個變數儲存在系統組態卡上。

如果下列所有條件都符合，POST 便會在系統電源接通時或是系統自動重設後自動執行：

- `diag-switch?` 設定為 `true` (預設值為 `false`)。
- `diag-level` 設定為 `min`、`max` 或 `menus` (預設值為 `min`)。
- `post-trigger` 符合重設類別 (預設值為 `power-on-reset`)。

如果 `diag-level` 設定為 `min` 或 `max`，POST 將分別執行小型或擴充測試。

如果 `diag-level` 設定為 `menus`，將顯示開啓電源時所執行的所有測試的功能表。

POST 診斷與錯誤訊息報告將顯示在主控台中。

控制 POST 診斷

設定 OpenBoot 組態變數後，即可控制 POST 診斷 (與開機程序的其他方面)。OpenBoot 組態變數的變更，必須在系統重新啓動後才會生效。[表 1-4](#) 列出其中最重要也最有用變數。關於更改 OpenBoot 組態變數的方法，請參閱[第 25 頁](#)的「[檢視與設定 OpenBoot 組態變數](#)」。

表 1-4 OpenBoot 組態變數

OpenBoot 組態變數	說明與關鍵字
<code>auto-boot</code>	確定系統是否自動啓動。預設值為 <code>true</code> 。 <ul style="list-style-type: none">• <code>true</code> — 作業系統在韌體測試完成後自動啓動。• <code>false</code> — 系統保持在 <code>ok</code> 提示下，直到您鍵入 <code>boot</code> 為止。
<code>diag-level</code>	確定所有執行診斷的層級或類型。預設值為 <code>min</code> 。 <ul style="list-style-type: none">• <code>off</code> — 沒有測試。• <code>min</code> — 僅執行基本測試。• <code>max</code> — 視裝置而定，可執行更廣泛的測試。• <code>menus</code> — 可以個別執行 POST 層級的功能表驅動程式。
<code>diag-script</code>	決定哪些裝置由 OpenBoot 診斷來測試。預設值為 <code>none</code> 。 <ul style="list-style-type: none">• <code>none</code> — 不會測試任何裝置。• <code>normal</code> — 將測試具有自我測試功能的內建 (以中央翼為主) 裝置。• <code>all</code> — 將測試具有自我測試功能的所有裝置。
<code>diag-switch?</code>	將系統切換為診斷模式開啓或關閉。預設值為 <code>false</code> 。 <ul style="list-style-type: none">• <code>true</code> — 診斷模式：執行 POST 診斷與 OpenBoot 診斷測試。• <code>false</code> — 預設模式：不執行 POST 或 OpenBoot 診斷測試。

表 1-4 OpenBoot 組態變數 (續)

OpenBoot 組態變數	說明與關鍵字
post-trigger obdiag-trigger	<p>這兩個變數指定導致執行開機自我測試 (或 OpenBoot 診斷測試) 的重設事件類別。這些變數可以接受單個關鍵字, 及由空格隔開的前三個關鍵字。相關詳細資訊, 請參閱第 25 頁的「檢視與設定 OpenBoot 組態變數」。</p> <ul style="list-style-type: none"> • <code>error-reset</code> — 由某些無法復原之硬體錯誤條件導致的重設。一般而言, 當硬體問題導致系統狀態資料毀損時, 便會發生錯誤重設, 例如, CPU 與系統監視裝置重設、嚴重錯誤以及特定的 CPU 重設事件 (預設) 等。 • <code>power-on-reset</code> — 按下開啓 / 待命按鈕所導致的重設 (預設)。 • <code>user-reset</code> — 由使用者或作業系統啓始的重設。 • <code>all-resets</code> — 任何系統重設。 • <code>none</code> — 不會執行開機自我測試 (或 OpenBoot 診斷測試)。
input-device	<p>選擇主控台輸入的位置。預設值為 <code>ttya</code>。</p> <ul style="list-style-type: none"> • <code>ttya</code> — 來自內建的 SERIAL MGT 連接埠。 • <code>ttyb</code> — 來自內建的一般用途序列埠 (10101)。 • <code>keyboard</code> — 來自屬於繪圖終端機一部分的已安裝鍵盤。
output-device	<p>選擇診斷與其他主控台輸出顯示的位置。預設值為 <code>ttya</code>。</p> <ul style="list-style-type: none"> • <code>ttya</code> — 輸出到內建的 SERIAL MGT 連接埠。 • <code>ttyb</code> — 輸出到內建的一般用途序列埠 (10101)。 • <code>screen</code> — 輸出到屬於繪圖終端機一部分的已安裝螢幕。ⁱ

i POST 訊息無法顯示在圖形終端機上; 即使 `output-device` 設定成 `screen`, 訊息還是會傳送至 `ttya`。

備註 — 這些變數會影響 OpenBoot 診斷測試與 POST 診斷。

完成執行 POST 診斷後, POST 將向 OpenBoot 韌體報告所執行的每個測試之狀態, 然後控制將交回給 OpenBoot 韌體碼。

如果 POST 診斷沒有找到故障, 且您的伺服器仍沒有啓動, 請執行 OpenBoot 診斷測試。

▼ 啓動 POST 診斷

1. 移至 `ok` 提示。
2. 請鍵入下列指令：

```
ok setenv diag-switch? true
```

3. 請鍵入下列指令：

```
ok setenv diag-level value
```

此處 *value* 為 min、max 或 menus，這是取決於您希望看到的診斷資訊數量。

4. 請鍵入下列指令：

```
ok reset-all
```

如果 post-trigger 設定為 user-reset，系統便會執行 POST 診斷。狀態與錯誤訊息會顯示在主控台視窗中。如果 POST 偵測到錯誤，將會顯示說明故障的錯誤訊息。

5. 完成執行 POST 後，請鍵入下列指令將 diag-switch? 的值恢復為 false：

```
ok setenv diag-switch? false
```

將 diag-switch? 重設為 false 可最小化開機時間。

OpenBoot 指令

OpenBoot 指令是您在 ok 提示後鍵入的指令。可提供有用診斷資訊的 OpenBoot 指令如下：

- probe-scsi 與 probe-scsi-all
- probe-ide
- show-devs

probe-scsi 與 probe-scsi-all 指令

probe-scsi 與 probe-scsi-all 指令可診斷 SCSI 裝置的問題。



警告 – 若您已經使用 halt 指令或 Stop-A 按鍵序列跳到 ok 提示符號，則執行 probe-scsi 或 probe-scsi-all 指令會導致系統當機。

probe-scsi 指令會與主機板內建 SCSI 控制器所連接的所有 SCSI 裝置通訊。probe-scsi-all 指令則還會進一步存取 PCI 插槽上所安裝之任何主機配接卡連接的裝置。

對於已經連上且正在作用的任何 SCSI 裝置，`probe-scsi` 與 `probe-scsi-all` 指令都會顯示其迴圈 ID、主機配接卡、邏輯單元編號、唯一的 world-wide name (WWN) 以及涵蓋其類型與製造廠商的裝置說明。

以下為 `probe-scsi` 指令的輸出範例。

代碼示例 1-1 `probe-scsi` 指令輸出

```
{1} ok probe-scsi
Target 0
  Unit 0   Disk      SEAGATE ST373307LSUN72G 0207
Target 1
  Unit 0   Disk      SEAGATE ST336607LSUN36G 0207
{1} ok
```

以下則為 `probe-scsi-all` 指令的輸出範例。

代碼示例 1-2 `probe-scsi-all` 指令輸出

```
{1} ok probe-scsi-all
/pci@1c,600000/scsi@2,1

/pci@1c,600000/scsi@2
Target 0
  Unit 0   Disk      SEAGATE ST373307LSUN72G 0207
Target 1
  Unit 0   Disk      SEAGATE ST336607LSUN36G 0207

{1} ok
```

probe-ide 指令

`probe-ide` 指令會與連接 IDE 匯流排的所有「整合驅動電子介面」(IDE) 裝置進行通訊，這是 DVD-ROM 光碟機之類的媒體裝置專用的內部系統匯流排。



警告 – 若您已經使用 `halt` 指令或 `Stop-A` 按鍵序列跳到 `ok` 提示符號，則執行 `probe-ide` 指令會導致系統當機。

以下為 `probe-ide` 指令的輸出範例。

代碼示例 1-3 probe-ide 指令輸出

```
{1} ok probe-ide
Device 0 ( Primary Master )
      Not Present

Device 1 ( Primary Slave )
      Not Present

Device 2 ( Secondary Master )
      Not Present

Device 3 ( Secondary Slave )
      Not Present

{1} ok
```

show-devs 指令

show-devs 指令會列出韌體裝置樹中每個裝置的硬體裝置路徑，[代碼示例 1-4](#) 顯示部份範例輸出。

代碼示例 1-4 show-devs 指令輸出

```
/pci@1d,700000
/pci@1c,600000
/pci@1e,600000
/pci@1f,700000
/memory-controller@1,0
/SUNW,UltraSPARC-IIIi@1,0
/memory-controller@0,0
/SUNW,UltraSPARC-IIIi@0,0
/virtual-memory
/memory@m0,0
/aliases
/options
/openprom
/chosen
/packages
/pci@1d,700000/network@2,1
/pci@1d,700000/network@2
/pci@1c,600000/scsi@2,1
/pci@1c,600000/scsi@2
/pci@1c,600000/scsi@2,1/tape
/pci@1c,600000/scsi@2,1/disk
```

代碼示例 1-4 show-devs 指令輸出 (續)

```
/pci@1c,600000/scsi@2/tape
/pci@1c,600000/scsi@2/disk
/pci@1e,600000/ide@d
/pci@1e,600000/usb@a
/pci@1e,600000/pmu@6
/pci@1e,600000/isa@7
/pci@1e,600000/ide@d/cdrom
/pci@1e,600000/ide@d/disk.....
```

▼ 執行 OpenBoot 指令

1. 中止系統以進入 ok 提示。
在關閉系統之前警告使用者。
2. 在主控制台提示下鍵入適當指令。

OpenBoot 診斷

與 POST 診斷一樣，OpenBoot 診斷碼以韌體為基礎，且位於 Boot PROM 中。

▼ 啓動 OpenBoot 診斷

1. 類型：

```
ok setenv diag-switch? true
ok setenv auto-boot? false
ok reset-all
```

2. 類型：

```
ok obdiag
```

此指令將顯示 OpenBoot 診斷功能表。

```
ok obdiag

o b d i a g

1 i2c@0,320      2 ide@d          3 network@2
4 network@2,1    5 rtc@0,70       6 scsi@2
7 scsi@2,1      8 serial@0,2e8   9 serial@0,3f8
10 usb@a        11 usb@b         12 flashprom@2,0

Commands:test test-all except help what printenvs setenv versions exit
```

備註 – 如果您的伺服器已安裝 PCI 卡，則其他測試將出現在 obdiag 功能表中。

3. 請鍵入下列指令：

```
obdiag> test n
```

其中 *n* 代表與您要執行的測試相對應的數字。

測試摘要將會出現。在 obdiag> 提示下，鍵入：

```
obdiag> help
```

控制 OpenBoot 診斷測試

您用來控制 POST (請參閱表 1-4) 的 OpenBoot 組態變數，大部分也會影響 OpenBoot 診斷測試。

- 使用 diag-level 變數控制 OpenBoot 診斷測試等級。
- 使用 test-args 自訂測試的執行方式。

test-args 的預設值為空字串，您可以使用表 1-5 中所示的一個或多個預留關鍵字來修改 test-args。

表 1-5 OpenBoot 組態變數 test-args 關鍵字

關鍵字	說明
bist	啓動外部與週邊裝置上的內建自我測試 (BIST)。
debug	顯示所有除錯訊息。
iopath	確認匯流排及互連完整性。
loopback	執行裝置的外部回送路徑。
media	檢查外部與週邊裝置媒體可存取性。
restore	如果之前執行的測試失敗，將嘗試恢復裝置的原始狀態。
_silent	僅顯示錯誤，而非每個測試的狀態。
subtests	顯示主測試與叫用的每個子測試。
verbose	顯示所有測試的詳細狀態訊息。
callers= <i>n</i>	發生錯誤時，回溯顯示前 <i>N</i> 個呼叫者。 callers=0 一 回溯顯示發生錯誤之前的所有呼叫者。
errors= <i>n</i>	繼續執行測試，直到遇到 <i>N</i> 個錯誤為止。 errors=0 一 在不終止測試下顯示所有錯誤報告。

若要自訂 OpenBoot 診斷測試，可將 test-args 設定成以逗號分隔的關鍵字清單，如下範例所示：

```
ok setenv test-args debug,loopback,media
```

test 與 test-all 指令

您也可以從 ok 提示下直接執行 OpenBoot 診斷測試，只要輸入 test 指令，後面加上欲測試裝置 (或一組裝置) 的完整硬體路徑即可。例如：

```
ok test /pci@x,y/SUNW,qlc@2
```

若要自訂個別的測試，您可依下例所示使用 test-args：

```
ok test /usb@1,3:test-args={verbose,debug}
```

此語法只會影響目前的測試，而不會變更以下 test-args OpenBoot 組態變數的值。

利用 `test-all` 指令可以測試裝置樹中的所有裝置：

```
ok test-all
```

如果您在 `test-all` 指令中指定路徑參數，則只會測試指定的裝置及其下屬子裝置。以下範例顯示的指令，可測試 USB 匯流排以及與 USB 匯流排連接且具有自我測試功能的所有裝置：

```
ok test-all /pci@9,700000/usb@1,3
```

OpenBoot 診斷錯誤訊息

OpenBoot 診斷的錯誤結果以表格形式顯示，摘要列出所遇到的問題、受影響的硬體裝置、失敗的子測試以及其他診斷資訊。代碼示例 1-5 顯示 OpenBoot 診斷的錯誤訊息範例。

代碼示例 1-5 OpenBoot 診斷錯誤訊息

```
Testing /pci@1e,600000/isa@7/flashprom@2,0

ERROR   : FLASHPROM CRC-32 is incorrect
SUMMARY : Obs=0x729f6392 Exp=0x3d6cdf53 XOR=0x4ff3bcc1 Addr=0xfeebbffc
DEVICE  : /pci@1e,600000/isa@7/flashprom@2,0
SUBTEST : selftest:crc-subtest
MACHINE : Netra 240
SERIAL# : 52965531
DATE    : 03/05/2003 01:33:59 GMT
CONTROLS: diag-level=max test-args=

Error: /pci@1e,600000/isa@7/flashprom@2,0 selftest failed, return code = 1
Selftest at /pci@1e,600000/isa@7/flashprom@2,0 (errors=1) .....
failed
Pass:1 (of 1) Errors:1 (of 1) Tests Failed:1 Elapsed Time: 0:0:0:27
```

作業環境診斷工具

當系統通過 OpenBoot 診斷測試時，便會嘗試啟動 Solaris OS。伺服器在多位使用者模式中執行時，您可以存取軟體型的診斷工具及 SunVTS 軟體。這些工具可讓您監控伺服器、測試伺服器並隔離故障。

備註 – 如果您將 `auto-boot?OpenBoot` 組態變數設定成 `false`，則韌體測試執行完畢後，作業環境並不會啟動。

除上述工具外，您可以參閱錯誤與系統訊息記錄檔及 Solaris 軟體資訊指令。

錯誤與系統訊息記錄檔

錯誤訊息與其他系統訊息都儲存在 `/var/adm/messages` 檔案中。這個檔案內記錄的訊息有許多來源，包括作業系統、環境控制子系統以及各式各樣的軟體應用程式。

Solaris 軟體系統資訊指令

下列 Solaris 軟體系統資訊指令會顯示可在評估 Netra 240 伺服器狀況時使用的資料：

- `prtconf`
- `prtdiag`
- `prtfru`
- `psrinfo`
- `showrev`

本章節說明了這些指令給您的資訊。關於使用這些指令的詳細資訊，請參閱適當的說明頁。

prtconf 指令

`prtconf` 指令可顯示 Solaris 軟體裝置樹，此裝置樹包括由 OpenBoot 韌體探查到的所有裝置，以及其他只有作業環境軟體可辨識的裝置，如個別磁碟。`prtconf` 的輸出中也包含系統記憶體總大小。[代碼示例 1-6](#) 為 `prtconf` 輸出的摘錄內容。

代碼示例 1-6 prtconf 指令輸出

```
# prtconf

System Configuration: Sun Microsystems sun4u
Memory size: 5120 Megabytes
System Peripherals (Software Nodes):

SUNW,Netra-240
  packages (driver not attached)
    SUNW,builtin-drivers (driver not attached)
    deblocker (driver not attached)
    disk-label (driver not attached)
    terminal-emulator (driver not attached)
    dropins (driver not attached)
    kbd-translator (driver not attached)
    obp-tftp (driver not attached)
    SUNW,i2c-ram-device (driver not attached)
    SUNW,fru-device (driver not attached)
    ufs-file-system (driver not attached)
  chosen (driver not attached)
  openprom (driver not attached)
    client-services (driver not attached)
  options, instance #0
  aliases (driver not attached)
  memory (driver not attached)
  virtual-memory (driver not attached)
  SUNW,UltraSPARC-IIIi (driver not attached)
  memory-controller, instance #0
  SUNW,UltraSPARC-IIIi (driver not attached)
  memory-controller, instance #1
  pci, instance #0.....
```

prtconf 指令的 -p 選項可產生與 OpenBoot show-devs 指令相似的輸出。以下輸出內容僅列出系統韌體所彙整的裝置。

prtdiag 指令

prtdiag 指令會顯示診斷資訊表，摘要列出系統元件的狀態。prtdiag 指令使用的顯示格式，會依系統所執行的 Solaris OS 版本而有所不同。以下的程式碼範例，顯示在正常執行 Solaris 軟體的 Netra 240 伺服器上使用 prtdiag 指令所產生的輸出摘錄內容。

代碼示例 1-7 prtdiag 指令輸出

```
# prtdiag
System Configuration: Sun Microsystems sun4u Netra 240
System clock frequency: 160 MHz
Memory size: 2GB
===== CPUs =====
      CPU  Freq      E$      CPU      CPU      Temperature      Fan
      CPU  Freq      Size      Impl.  Mask      Die      Ambient      Speed  Unit
-----
      MB/P0 1280 MHz 1MB      US-IIIi  2.3      -      -
      MB/P1 1280 MHz 1MB      US-IIIi  2.3      -      -
===== IO Devices =====
      Bus  Freq
Brd  Type  MHz  Slot      Name      Model
-----
0    pci   66      2  network-pci14e4,1648.108e.16+
0    pci   66      2  network-pci14e4,1648.108e.16+
0    pci   66      2  scsi-pci1000,21.1000.1000.1 +
0    pci   66      2  scsi-pci1000,21.1000.1000.1 +
0    pci   66      2  network-pci14e4,1648.108e.16+
0    pci   66      2  network-pci14e4,1648.108e.16+
0    pci   33      7  isa/serial-su16550 (serial)
0    pci   33      7  isa/serial-su16550 (serial)
0    pci   33      7  isa/rmc-comm-rmc_comm (seria+
0    pci   33      13  ide-pci10b9,5229.c4 (ide)
===== Memory Configuration =====
Segment Table:
-----
Base Address      Size      Interleave Factor  Contains
-----
0x0               1GB      1                  GroupID 0
0x1000000000     1GB      1                  GroupID 0
```

代碼示例 1-7 prtdiag 指令輸出 (續)

```
Memory Module Groups:
-----
ControllerID   GroupID   Labels
-----
0               0         MB/P0/B0/D0,MB/P0/B0/D1
Memory Module Groups:
-----
ControllerID   GroupID   Labels
-----
1               0         MB/P1/B0/D0,MB/P1/B0/D1
```

除了代碼示例 1-7 中所顯示的資訊外，加上詳細選項 (-v) 的 prtdiag 指令，還會顯示出前面板狀態、磁碟狀態、風扇狀態、電源供應器、硬體修訂版本以及系統溫度等 (請參閱代碼示例 1-8)。

代碼示例 1-8 prtdiag 詳細輸出

```
-----
Location      Sensor      Temperature  Lo LoWarn HiWarn  Hi Status
-----
MB            T_ENC       22C         -7C   -5C    55C   58C okay
MB/P0        T_CORE      57C         -     -     110C  115C okay
MB/P1        T_CORE      54C         -     -     110C  115C okay
PS0          FF_OT       -           -     -     -     -   okay
PS1          FF_OT       -           -     -     -     -   okay
```

如果發生過熱狀況時，prtdiag 會在狀態欄中報告錯誤訊息 (代碼示例 1-9)。

代碼示例 1-9 prtdiag 過熱指示輸出

```
-----
Location      Sensor      Temperature  Lo LoWarn HiWarn  Hi Status
-----
MB            T_ENC       22C         -7C   -5C    55C   58C okay
MB/P0        T_CORE      118C        -     -     110C  115C failed
MB/P1        T_CORE      112C        -     -     110C  115C warning
PS0          FF_OT       -           -     -     -     -   okay
PS1          FF_OT       -           -     -     -     -   okay
```

同樣地，特定元件發生故障時，prtdiag 也會在適當的狀態欄中報告故障狀況 (代碼示例 1-10)。

代碼示例 1-10 prtdiag 故障指示輸出

```
Fan Speeds:
-----
Location      Sensor      Status      Speed
-----
MB/P0/F0      RS          failed      0 rpm
MB/P0/F1      RS          okay        3994 rpm
F2            RS          okay        2896 rpm
PS0           FF_FAN      okay
F3            RS          okay        2576 rpm
PS1           FF_FAN      okay
-----
```

prtf fru 指令

Netra 240 伺服器會維護一份列出系統中所有現場可更換裝置 (FRU) 的階層式清單，並保有關於不同 FRU 的特定資訊。

prtf fru 指令不僅顯示此階層式清單，也可以顯示許多 FRU 上之序列電子可擦拭可程式唯讀記憶體 (EEPROM) 裝置內含的資料。代碼示例 1-11 顯示 prtf fru 指令加上 -l 選項所產生之 FRU 階層式清單的部分摘錄內容。

代碼示例 1-11 prtf fru -l 指令輸出

```
# prtf fru -l
/frutree
/frutree/chassis (fru)
/frutree/chassis/MB?Label=MB
/frutree/chassis/MB?Label=MB/system-board (container)
/frutree/chassis/MB?Label=MB/system-board/SC?Label=SC
/frutree/chassis/MB?Label=MB/system-board/SC?Label=SC/sc (fru)
/frutree/chassis/MB?Label=MB/system-board/BAT?Label=BAT
/frutree/chassis/MB?Label=MB/system-board/BAT?Label=BAT/battery (fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu (fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F0?Label=F0
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F0?Label=F0/fan-unit
(fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F1?Label=F1
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F1?Label=F1/fan-unit
(fru).....
```

代碼示例 1-12 顯示 prtfrfu 指令加上 -c 選項所產生的 SEEPROM 資料的摘錄內容。此輸出只顯示容器及其中的資料，並未列出 FRU 樹狀階層。

代碼示例 1-12 prtfrfu -c 指令輸出

```
# prtfrfu -c
/frutree/chassis/MB?Label=MB/system-board (container)
  SEGMENT: SD
    /ManR
      /ManR/UNIX_Stamp32: Mon Dec  2 19:47:38 PST 2002
      /ManR/Fru_Description: FRUID, INSTR, M'BD, 2X1.28GHZ, CPU
      /ManR/Manufacture_Loc: Hsinchu, Taiwan
      /ManR/Sun_Part_No: 3753120
      /ManR/Sun_Serial_No: 000615
      /ManR/Vendor_Name: Mitac International
      /ManR/Initial_HW_Dash_Level: 02
      /ManR/Initial_HW_Rev_Level: 0E
      /ManR/Fru_Shortname: MOTHERBOARD
      /SpecPartNo: 885-0076-11
    /frutree/chassis/MB?Label=MB/system-board/P0?Label=
    P0/cpu/B0?Label=B0/bank/D0?Label=D0/mem-module (container)
    /frutree/chassis/MB?Label=MB/system-board/P0?Label=
    P0/cpu/B0?Label=B0/bank/D1?Label=D1/mem-module (container).....
```

prtfrfu 指令顯示的資料依 FRU 的類型而有所不同，這些資料通常包含：

- FRU 說明
- 製造廠商名稱與位置
- 零件編號與序號
- 硬體修訂版本

psrinfo 指令

psrinfo 指令顯示每個 CPU 開始上線日期與時間；利用詳細 (-v) 選項，該指令會顯示關於 CPU 的額外資訊，包括其時脈速度。代碼示例 1-13 顯示 psrinfo 指令與 -v 選項的輸出範例。

代碼示例 1-13 psrinfo -v 指令輸出

```
# psrinfo -v
Status of processor 0 as of: 07/28/2003 14:43:29
  Processor has been on-line since 07/21/2003 18:43:37.
  The sparcv9 processor operates at 1280 MHz,
    and has a sparcv9 floating point processor.
Status of processor 1 as of: 07/28/2003 14:43:29
  Processor has been on-line since 07/21/2003 18:43:36.
  The sparcv9 processor operates at 1280 MHz,
    and has a sparcv9 floating point processor
```

showrev 指令

showrev 指令會顯示目前硬體與軟體的修訂版本資訊；代碼示例 1-14 即為 showrev 指令的輸出範例。

代碼示例 1-14 showrev 指令輸出

```
# showrev
Hostname: vsp78-36
Hostid: 8328c87b
Release: 5.8
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: vsplab.SFBay.Sun.COM
Kernel version: SunOS 5.8 Generic 108528-18 November 2002
```

加上 -p 選項時，showrev 指令會顯示已安裝的修補程式。代碼示例 1-15 為 showrev 指令加上 -p 選項的輸出內容摘錄範例。

代碼示例 1-15 showrev -p 指令輸出

```
Patch: 109729-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109783-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109807-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109809-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110905-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110910-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110914-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 108964-04 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr
```

▼ 執行 Solaris 平台系統資訊指令

- 在指令提示下，輸入您要顯示之系統資訊類型的指令。

如需更多資訊，請參閱第 17 頁的「Solaris 軟體系統資訊指令」。請參閱表 1-6 以取得指令摘要。

表 1-6 Solaris 平台資訊顯示指令

指令	顯示的資訊	鍵入的指令	注意事項
prtconf	系統組態資訊	/usr/sbin/prtconf	—
prtdiag	診斷與組態資訊	/usr/platform/sun4u/sbin/prtdiag	使用 -v 選項可列出其他詳細資訊。
prtfru	FRU 階層與 SEEPROM 記憶體內容	/usr/sbin/prtfru	使用 -l 選項可顯示階層結構；-c 選項則可顯示 SEEPROM 資料。
psrinfo	每個 CPU 連線的日期與時間；處理器時脈速度	/usr/sbin/psrinfo	使用 -v 選項可列出時脈速度與其他資料。
showrev	硬體與軟體修訂版資訊	/usr/bin/showrev	使用 -p 選項可顯示軟體修補程式。

最新診斷測試結果

開機自我測試 (POST) 與 OpenBoot 診斷測試最新的測試結果摘要，將先關機、然後再開機的過程中儲存。

▼ 檢視最新測試結果

1. 移至 ok 提示。
2. 執行下列任一項操作：
 - 要查看最新的 POST 結果摘要，請鍵入下列指令：

```
ok show-post-results
```


- 要查看最新的 OpenBoot 診斷測試結果摘要，請鍵入下列指令：

```
ok show-obdiag-results
```

此指令隨後便會產生與系統相關的硬體元件清單，並且會指示哪些元件通過 POST 或 OpenBoot 診斷測試，哪些元件沒有通過測試。

OpenBoot 組態變數

儲存在 IDPROM 中的開關與診斷組態變數，將決定 POST 與 OpenBoot 診斷測試的執行方式與時間。本章節說明如何存取與修改 OpenBoot 組態變數。如需重要 OpenBoot 組態變數清單，請參閱表 1-4。

對 OpenBoot 組態變數所作的變更會在下次重新啟動系統時生效。

▼ 檢視與設定 OpenBoot 組態變數

- 中斷伺服器，顯示 ok 提示。
 - 若要顯示所有 OpenBoot 組態變數目前的值，請使用 `printenv` 指令。
下列範例顯示了此指令執行結果的簡要摘錄。

```
ok printenv
Variable Name      Value                Default Value
diag-level         min                  min
diag-switch?      false                false
```

- 若要設定或變更 OpenBoot 組態變數的值，請使用 `setenv` 指令。

```
ok setenv diag-level max
diag-level =          max
```

- 若要設定能接受多個關鍵字的 OpenBoot 組態變數，請以空格分隔各個關鍵字。

使用 watch-net 與 watch-net-all 指令檢查網路連線

watch-net 診斷測試可監控主網路介面上的乙太網路封包。watch-net-all 診斷測試可監控主網路介面，以及與主機板連接的任何其他網路介面上的乙太網路封包。系統收到的正確封包將以句號 (.) 表示。錯誤封包，例如框架錯誤與循環冗餘檢查 (CRC) 錯誤，則以 X 表示，並帶有相關錯誤說明。

- 若要啟動 watch-net 診斷測試，在 ok 提示下鍵入 watch-net 指令 (代碼示例 1-16)。

代碼示例 1-16 watch-net 診斷輸出訊息

```
{0} ok watch-net
Internal loopback test -- succeeded.
Link is -- up
Looking for Ethernet Packets.
`.` is a Good Packet.`X` is a Bad Packet.
Type any key to stop.....
```

- 若要啟動 watch-net-all 診斷測試，請在 ok 提示下鍵入 watch-net-all (代碼示例 1-17)。

代碼示例 1-17 watch-net-all 診斷輸出訊息

```
{0} ok watch-net-all
/pci@1f,0/pci@1,1/network@c,1
Internal loopback test -- succeeded.
Link is -- up
Looking for Ethernet Packets.
`.` is a Good Packet.`X` is a Bad Packet.
Type any key to stop.
```

自動系統回復

備註 – 自動系統回復 (ASR) 與自動伺服器重新啓動不同，Netra 240 伺服器也支援後者。如需自動伺服器重新啓動的資訊，請參閱第 3 章。

自動系統回復 (ASR) 功能包含自我測試功能與自動設定功能，可偵測發生故障的硬體元件並取消其設定。若啓用 ASR，當某些硬體發生非嚴重性故障時，伺服器將可以恢復作業。

如果某個元件在 ASR 的監控之下，且伺服器在沒有該元件時仍可繼續作業，則在該元件發生故障或失敗時，伺服器會自動重新啓動。這個功能可以防止發生故障的硬體元件使整個系統當機，或導致系統重複失敗。

如果系統在執行開機程序時偵測到故障，則會停用發生故障的元件。如果系統仍可繼續正常運作，則會繼續執行啓動程序。

為支援此種降級啓動能力，OpenBoot 韌體使用 1275 用戶端介面（透過裝置樹），藉由在對應的裝置樹狀節點上建立適當的「狀態」屬性，將裝置標記為故障或停用。Solaris OS 將不會啓動以此方式標記的任何子系統之驅動程式。

只要發生故障的元件被切斷電源（例如：不會導致隨機匯流排錯誤或訊號噪音），系統將在收到服務呼叫時自動重新啓動並恢復作業。

一旦故障或停用的裝置更換為新的裝置後，OpenBoot 韌體會自動在重新啓動時修改裝置狀態。

備註 – ASR 必須等到您將其啓動後才會啓用（請參閱第 29 頁的「啓用 ASR」）。

Auto-Boot 選項

auto-boot? 設定可控制韌體在每次重設後是否自動啓動作業系統。預設值為 true。

auto-boot-on-error? 設定可控制當偵測到子系統故障時，系統是否要嘗試進行降級啓動。auto-boot? 與 auto-boot-on-error? 設定都必須設定為 true，系統才會自動進行降級啓動。

- 要設定這兩個參數，請鍵入下列指令：

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

備註 — auto-boot-on-error? 的預設設定為 false。因此，除非您將此設定變更為 true，否則系統不會嘗試執行降級啟動。此外，即使已啟用降級啟動，系統也不會針對任何無法回復的嚴重錯誤嘗試進行降級啟動。有關無法回復之嚴重錯誤的範例，請參閱第 28 頁的「錯誤處理摘要」。

錯誤處理摘要

在執行開機程序時的錯誤處理，將分為下列三種方式：

- 如果 POST 或 OpenBoot 診斷未偵測到任何錯誤，且 auto-boot? 為 true 的話，系統就會嘗試啟動。
- 如果 POST 或 OpenBoot 診斷只偵測出非嚴重錯誤，而 auto-boot? 為 true 且 auto-boot-on-error? 為 true，系統便會嘗試啟動。

備註 — 如果 POST 或 OpenBoot 診斷偵測到與正常啟動裝置相關的非嚴重錯誤，OpenBoot 韌體就會自動將故障裝置解除組態，並嘗試使用下一個啟動裝置，如 boot-device 組態參數所指定的一般。

- 如果 POST 或 OpenBoot 診斷偵測到嚴重錯誤，不論 auto-boot? 或 auto-boot-on-error? 的設定為何，系統都不會啟動。無法復原的嚴重錯誤包括：
 - 所有 CPU 損壞
 - 所有邏輯記憶體庫損壞
 - 快閃 RAM 循環冗餘檢查 (CRC) 失敗
 - 重要的現場可更換裝置 (FRU) PROM 組態資料損壞
 - 重要的特定應用積體電路 (ASIC) 發生故障

重新開機程序

OpenBoot 組態變數有三個 — diag-switch?、obdiag-trigger 與 post-trigger — 可用來控制系統執行韌體診斷的方式，以回應系統重新開機事件。

標準系統重設通訊協定會略過 POST 與 OpenBoot 診斷測試，除非將 diag-switch? 設定為 true。此變數的預設值為 false。由於 ASR 仰賴韌體診斷來偵測發生故障的裝置，因此 diag-switch? 必須設定為 true，ASR 才能執行。相關操作說明，請參閱第 29 頁的「啟用 ASR」。

要控制哪個重設事件（如果有的話）自動初始化韌體診斷，請使用 `obdiag-trigger` 與 `post-trigger`。關於這些變數與其使用方法的詳細說明，請參閱第 8 頁的「控制 POST 診斷」與第 14 頁的「控制 OpenBoot 診斷測試」。

▼ 啓用 ASR

1. 在系統 `ok` 提示符號後輸入：

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

2. 將 `obdiag-trigger` 變數設定成 `power-on-reset`、`error-reset` 或 `user-reset`。
例如，鍵入下列指令：

```
ok setenv obdiag-trigger user-reset
```

3. 請鍵入下列指令：

```
ok reset-all
```

如果 OpenBoot 變數 `auto-boot?` 設定為 `true`（其預設值），系統便會永久儲存參數變更並自動啓動。

備註 – 若要儲存參數變更，您也可以使用前面板上的開啓/待命按鈕，關閉系統電源後再開啓。

▼ 停用 ASR

1. 在系統 `ok` 提示符號後輸入：

```
ok setenv diag-switch? false
```

2. 請鍵入下列指令：

```
ok reset-all
```

系統將永久儲存參數變更。

備註 – 若要儲存參數變更，您也可以使用前面板上的開啓/待命按鈕，關閉系統電源後再開啓。

SunVTS 軟體

本章說明 SunVTS。本章包含下列主題：

- 第 31 頁的「SunVTS 軟體概觀」
- 第 32 頁的「SunVTS 測試」
- 第 33 頁的「SunVTS 軟體與安全性」
- 第 33 頁的「安裝 SunVTS 軟體」
- 第 34 頁的「檢視 SunVTS 軟體文件」

SunVTS 軟體概觀

Netra 240 伺服器支援 SunVTS 5.1 修補程式集 5 (PS5) 軟體及相容的版本。

SunVTS 軟體是線上診斷工具 Sun Validation Test Suite (驗證測試套件) 的縮寫，可用來檢驗各種硬體控制器、裝置與平台的組態和功能是否正常。SunVTS 是在 Solaris 作業環境中執行，並具有下列介面：

- 指令行介面 (CLI)
- 序列 (tty) 介面

SunVTS 軟體套件可執行系統與週邊設備壓力測試。您可透過網路檢視與控制 SunVTS 軟體階段作業。您可使用遠端機器來檢視測試階段作業的程序、變更測試選項，以及控制網路中其他機器的所有測試功能。

SunVTS 軟體可在三種測試模式下執行：

- *連線模式* 檢查裝置控制器是否存在，這個確認動作通常只需幾分鐘的時間，是對系統連線進行「健全檢查」的好方法。
- *功能模式* 只會測試您選擇的特定子系統，此為預設模式。
- *自動設定模式* 自動偵測所有子系統，並使用下列兩種方法之一來進行測試：
 - *信心測試* — 對所有子系統執行一遍測試，然後停止。對於一般系統組態，執行此測試需要一或兩小時。

- **全面測試** — 重複測試所有子系統長達 24 小時。

由於 SunVTS 軟體能夠同時執行多個測試，這樣會消耗大量系統資源，所以在生產系統中使用此軟體時請務必小心。如果使用 SunVTS 軟體的「全面測試」模式對系統執行壓力測試，請勿同時在該系統中執行任何其他程序。

伺服器必須執行 Solaris OS，SunVTS 軟體才能對其進行測試。由於 SunVTS 軟體套件是選用的，因此您的系統可能尚未安裝。相關操作說明，請參閱第 33 頁的「[確定 SunVTS 軟體是否已安裝](#)」。

SunVTS 測試

您可以使用 SunVTS 軟體來檢視及控制遠端連線伺服器上的測試階段作業。表 2-1 中列出一些可用的測試。

表 2-1 SunVTS 軟體測試

SunVTS 軟體測試	說明
cputest	測試 CPU。
disktest	測試本機磁碟機。
dvdtest	測試 DVD-ROM 光碟機。
n240atest	測試警報卡的警報繼電器、LED 和 FRU ID。
fputest	測試浮點裝置。
nettest	測試主機板上的乙太網路硬體與選用 PCI 卡上的網路硬體。
netlbttest	執行回送測試以檢查乙太網路轉接器是否能傳送與接收封包。
pmem	測試實體記憶體 (唯讀)。
sutest	測試伺服器的內建序列埠。
vmem	測試虛擬記憶體 (置換分割區與實體記憶體組合)。
env6test	測試環境裝置。
ssptest	測試 ALOM 硬體裝置。
i2c2test	測試 I ² C 裝置是否正常操作。

SunVTS 軟體與安全性

安裝 SunVTS 軟體時，您必須選擇基本安全性或 Sun 企業認證機制 (Sun Enterprise Authentication Mechanism™, SEAM) 安全性。基本安全性使用 SunVTS 軟體安裝目錄中的本機安全性檔案，以限制允許使用 SunVTS 軟體的使用者、群組及主機。SEAM 安全性依據標準網路驗證通訊協定 Kerberos，對透過網路傳輸的資料提供安全使用者驗證、資料完整性及隱私權。

如果網站使用 SEAM 安全性，則必須具有 SEAM 用戶端，而且必須在網路環境中安裝伺服器軟體，並在 Solaris 與 SunVTS 軟體中正確設定。如果網站不使用 SEAM 安全性，則在安裝 SunVTS 軟體過程中，不要選擇 SEAM 選項。

如果在安裝過程中啓用了錯誤的安全性層級，或錯誤設定了所選的安全性層級，則可能無法執行 SunVTS 軟體測試。如需詳細資訊，請參閱《SunVTS 使用者指南》，以及 SEAM 軟體所附的操作說明。

▼ 確定 SunVTS 軟體是否已安裝

- 請鍵入下列指令：

```
# pkginfo -l SUNWvts
```

- 如果 SunVTS 軟體已經載入，便會顯示此套件的相關資訊。
- 如果未載入 SunVTS 軟體，下列錯誤訊息將會出現：

```
ERROR: information for "SUNWvts" was not found
```

安裝 SunVTS 軟體

據預設值，SunVTS 軟體並未安裝在 Netra 240 伺服器中。但在 Solaris OS 軟體補充 (Software Supplement) CD 中已提供該軟體，而且最新的修訂版可從下列網站下載：

<http://www.sun.com/oem/products/vts/>

備註 – Netra 240 伺服器支援 SunVTS 5.1 修補程式集 5 (PS5) 軟體及相容的版本。

若要瞭解更多有關使用 SunVTS 軟體的資訊，請參閱符合您所執行的 Solaris 軟體版本之 SunVTS 文件。您也可以在上面網站找到關於 SunVTS 軟體的其他資訊，以及安裝說明。

檢視 SunVTS 軟體文件

SunVTS 軟體文件隨附在屬於所有 Solaris Media Kit 版本一部分的 Software Supplement CD 中。這些文件也可以從 <http://docs.sun.com> 取得。

要取得進一步資訊，您也可以參閱下列 SunVTS 軟體文件：

- 《SunVTS 使用者指南》說明如何安裝、設定及執行 SunVTS 診斷軟體。
- 《SunVTS 快速參考卡片》概述 SunVTS 介面的使用方法。
- 《SunVTS 測試參考手冊》提供個別 SunVTS 測試的相關詳細資訊。

進階無電管理軟體

本章提供 Sun™ 進階無電管理軟體 (ALOM) 的概觀。本章包含下列主題：

- 第 35 頁的「進階無電管理軟體概觀」
 - 第 36 頁的「ALOM 連接埠」
 - 第 36 頁的「設定 admin 密碼」
 - 第 37 頁的「基本的 ALOM 功能」
 - 第 38 頁的「自動伺服器重新啟動」
 - 第 39 頁的「環境監視與控制」
-

進階無電管理軟體概觀

Netra 240 伺服器在出貨時已安裝好 Sun 進階無電管理軟體，系統主控台依預設指向 ALOM，並設定為在啟動時顯示伺服器主控台資訊。

ALOM 可讓您透過序列連接 (使用 SERIAL MGT 連接埠) 或乙太網路連線 (使用 NET MGT 連接埠)，以監控與控制您的伺服器。如需設定乙太網路連線的詳細資訊，請參閱《Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南》(817-5009-11)。

備註 – 標有 SERIAL MGT 的 ALOM 序列埠僅用於伺服器管理。如果您需要一般用途的序列埠，請使用標有 10101 的序列埠。

您可以設定 ALOM，以便傳送與伺服器或 ALOM 相關的硬體故障與其他事件的電子郵件通知。

ALOM 電路使用伺服器的待命電源，以便：

- 一旦連接伺服器電源，ALOM 就處於啟用狀態，直到拔下電源線為止。
- 伺服器作業系統離線時，ALOM 韌體與軟體仍然有效。

表 3-1 列出由 ALOM 監控的元件，以及該軟體提供給每個元件的資訊。

表 3-1 由 ALOM 監控的元件

元件	提供的資訊
硬碟	運作狀況與狀態
系統與 CPU 風扇	速度與狀態
CPU	存在、溫度與任何熱警告或故障狀況
電源供應器	運作狀況與狀態
系統溫度	環境溫度與任何熱警告或故障狀況
伺服器前面板	旋轉式開關位置與 LED 狀態
電壓	狀態與門檻值
SCSI 與 USB 斷路器	狀態
乾接點繼電器警報	狀態

ALOM 連接埠

預設的管理連接埠標有 SERIAL MGT，此連接埠使用 RJ-45 接頭，且僅適用於伺服器管理；只支援與外部主控台連接的 ASCII。第一次操作伺服器時，請使用此連接埠。

其他序列埠 — 標有 10101 — 適用於一般用途的序列資訊傳輸。此連接埠使用 DB-9 接頭。如需接腳的詳細資訊，請參閱《Netra 240 伺服器安裝指南》(文件號碼 817-5000-11)。

此外，伺服器具有一個 10BASE-T 乙太網路管理網域介面，標有 NET MGT。要使用此連接埠，需要 ALOM 組態。如需詳細資訊，請參閱《Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南》(文件號碼 817-5009-11)。

設定 admin 密碼

當您在初次開機後切換至 ALOM 軟體時，會看到 `sc>` 提示。此時，您可以執行不需使用者權限的指令 (請參閱《Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南》(文件號碼 817-5009-11)，以取得指令清單)。如嘗試執行任何需要使用者權限的指令，系統會提示您設定 admin 使用者密碼。

- 如果系統提示要設定密碼，請設定 `admin` 使用者密碼。

密碼必須包含：

- 至少兩個字母字元
- 至少一個數字或一個特殊字元
- 介於六到八個字元之間

設定密碼後，`admin` 使用者將具有所有權限，並可以執行所有 `ALOM CLI` 指令。當使用者隨後切換至 `ALOM` 時，系統會提示他們使用 `admin` 密碼登入。

基本的 ALOM 功能

本節包含一些基本的 `ALOM` 功能。如需完整的文件，請參閱《*Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南*》(文件號碼 817-5009-11)，以及《*Netra 240 Server Release Notes*》(817-3142-xx)。

▼ 切換至 ALOM 提示

- 在指令提示下，鍵入以下的 `#.` 按鍵序列：

```
# #.
```

備註 – 切換到 `ALOM` 提示時，您將以 `admin` 的使用者 ID 登入。請參閱第 36 頁的「設定 `admin` 密碼」。

▼ 切換至伺服器主控台提示

- 請鍵入下列指令：

```
sc> console
```

多個 `ALOM` 使用者可以同時連線到伺服器主控台，但只允許一個使用者鍵入主控台輸入字元。

如果其他使用者已登入且可以寫入，您在輸入 `console` 指令後，將看到下列訊息：

```
sc> Console session already in use.[view mode]
```

▼ 取消其他使用者的主控台寫入能力

- 請鍵入下列指令：

```
sc> console -f
```

自動伺服器重新啓動

備註 – 自動系統回復 (ASR) 與自動伺服器重新啓動不同，Netra 240 伺服器也支援前者。

自動伺服器重新啓動是 ALOM 的一個元件，會在 Solaris 作業環境執行時對其進行監控，還會根據預設值，在伺服器當機時同步檔案系統並重新啓動伺服器。

ALOM 使用監控器程序，並且只監控核心。如果監控器程序當機，但核心仍在執行，ALOM 不會重新啓動伺服器。使用者無法設定監控器取樣間隔與監控器逾時的 ALOM 監控器參數。

如果核心當機且監控器逾時，ALOM 會報告並記錄此事件，然後執行下列三個使用者可設定的動作之一：

- **xir** — 此預設動作會導致伺服器同步檔案系統並重新啓動。如果系統當機，ALOM 會在 15 分鐘後回到硬重設程序。
- **重設** — 此為硬重設，會導致系統快速回復，但不會儲存與當機相關的診斷資料，並且可能會嚴重損壞檔案系統。
- **無** — 在系統報告監控器逾時後，此動作會導致系統無限期處於當機狀態。

如需詳細資訊，請參閱《Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南》(文件號碼 817-5009-11) 中的〈`sys_autorestart`〉一節。

關於使用自動系統回復 (ASR) 的說明，請參閱第 1 章。

環境監視與控制

Netra 240 伺服器配備環境監視子系統，可協助伺服器及其元件防範：

- 高溫
- 系統流通空氣不足
- 使用遺失或組態錯誤的元件運作
- 電源供應器故障
- 內部硬體故障

監視與控制功能由 ALOM 韌體處理，這可確保即使在系統中止運作或無法開機的狀況下，監視功能也能維持正常運作。同時，從 ALOM 韌體監視系統不需要系統指派 CPU 與記憶體資源給作業系統和軟體應用程式。

環境監視子系統使用符合業界標準的 I²C 匯流排。I²C 匯流排是簡單的雙線序列匯流排，遍佈於整個系統內，以便監視及控制溫度感應器、風扇、電源供應器、狀態 LED 及前面板系統控制旋轉式開關。

伺服器包含三個溫度感應器，用於監控系統的周圍溫度及兩個 CPU 的極限溫度。監視子系統會讀取每個感應器偵測出的溫度，只要遇到比指定溫度範圍高或低的狀況便會回報，並採取適當反應。其他 I²C 裝置可偵測元件是否存在與故障。

硬體及軟體同時確保機殼內的溫度不會超出預先設定的「安全運作」範圍。如果感應器偵測的溫度低於低溫警告臨界值或超出高溫警告臨界值，監視子系統軟體便會點亮前後面板上的需要維修 LED。若溫度異常狀況持續，並達到高溫或低溫軟關機溫度臨界值，系統便會進行正常關機程序。若溫度達到高溫或低溫硬關機溫度臨界值，系統便會進行強制關機程序。

錯誤與警告訊息會傳送至系統主控台，並記錄在 `/var/adm/messages` 檔案中，在執行自動系統關機後，需要維修 LED 仍會保持亮起，以協助診斷問題。

傳送至系統主控台並記錄在 `/var/adm/messages` 檔案中的訊息類型，取決於您設定 `sc_clieventlevel` 及 `sys_eventlevel` ALOM 使用者變數的方式。關於設定這些變數的資訊，請參閱《Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南》(817-5009-11)。

表 3-2 Netra 240 伺服器外殼溫度臨界值

溫度臨界值	溫度	伺服器動作
低溫，硬關機	-11°C	伺服器會進行強制系統關機程序。
低溫，軟關機	-9°C	伺服器會執行正常系統關機程序。
低溫警告	-7°C	伺服器會點亮前後面板上的系統需要維修 LED 指示燈。

表 3-2 Netra 240 伺服器外殼溫度臨界值 (續)

溫度臨界值	溫度	伺服器動作
高溫警告	57°C	伺服器會點亮前後面板上的系統需要維修 LED 指示燈。
高溫，軟關機	60°C	伺服器會執行正常系統關機程序。
高溫，硬關機	63°C	伺服器會進行強制系統關機程序。

監視子系統也能偵測四個系統風箱是否故障，如果有任何風箱故障，監視子系統會偵測故障情形，並將錯誤訊息傳送至系統主控台，且記錄在 `/var/adm/messages` 檔案中，並點亮需要維修 LED。

電源子系統也是以類似的方式受到監視。監視子系統會偶爾輪詢電源供應器的狀態，以指出每個供應器的輸出、輸入及存在狀態。

如果偵測出電源供應器故障，錯誤訊息便會傳送到系統主控台，並記錄在 `/var/adm/messages` 檔案中。此外，每個電源供應器上的 LED 也會亮起，表示發生故障。系統需要維修 LED 會亮起，表示系統故障。ALOM 主控台警示會記錄電源供應器故障。

使用 `showenvironment ALOM` 指令，可檢視電源子系統與風扇速度的警告臨界值。如需使用此指令的說明，請參閱《Netra 240 伺服器的 Sun 進階無電管理軟體使用者指南》(文件號碼 817-5009-11)。

警報繼電器輸出應用程式介面

本範例提供一個簡單的程式 (代碼示例 A-1)，說明如何 get/set 警報狀態。應用程式可以使用 LOMIOCALSTATE ioctl 函數來取得每個警報的狀態，並使用 LOMIOCALCTL ioctl 函數個別設定警報狀態。如需警報指示燈的詳細資訊，請參閱《Netra 240 Server Service Manual》(817-2699-xx)。

代碼示例 A-1 get/set 警報狀態的範例程式

```
#include <sys/types.h>
#include <string.h>
#include <stdlib.h>
#include <sys/unistd.h>
#include <fcntl.h>
#include "lom_io.h"

#define ALARM_INVALID    -1
#define LOM_DEVICE      "/dev/lom"

static void usage();
static void get_alarm(const char *alarm);
static int set_alarm(const char *alarm, const char *alarmval);
static int parse_alarm(const char *alarm);
static int lom_ioctl(int ioc, char *buf);
static char *get_alarmval(int state);
static void get_alarmvals();

main(int argc, char *argv[])
{
    if (argc < 3) {
        usage();
        if (argc == 1)
            get_alarmvals();
        exit (1);
    }
}
```

代碼示例 A-1 get/set 警報狀態的範例程式 (續)

```
#include <sys/types.h>

    if (strcmp(argv[1], "get") == 0) {
        if (argc != 3) {
            usage();
            exit (1);
        }
        get_alarm(argv[2]);
    }
    else
    if (strcmp(argv[1], "set") == 0) {
        if (argc != 4) {
            usage();
            exit (1);
        }
        set_alarm(argv[2], argv[3]);
    } else {
        usage();
        exit (1);
    }
}

static void
usage()
{
    printf("usage:alarm [get|set] [crit|major|minor|user] [on|off]\n");
}

static void
get_alarm(const char *alarm)
{
    ts_aldata_t    ald;
    int altype = parse_alarm(alarm);
    char *val;

    if (altype == ALARM_INVALID) {
        usage();
        exit (1);
    }

    ald.alarm_no = altype;
    ald.alarm_state = ALARM_OFF;

    lom_ioctl(LOMIOCALSTATE, (char *)&ald);

    if ((ald.alarm_state != ALARM_OFF) &&
        (ald.alarm_state != ALARM_ON)) {
```

代碼示例 A-1 get/set 警報狀態的範例程式 (續)

```
#include <sys/types.h>
    printf("Invalid value returned:%d\n", ald.alarm_state);
    exit (1);
}

printf("ALARM.%s = %s\n", alarm, get_alarmval(ald.alarm_state));
}

static int
set_alarm(const char *alarm, const char *alarmstate)
{
    ts_aldata_t    ald;
    int alarmval = ALARM_OFF, altype = parse_alarm(alarm);

    if (altype == ALARM_INVALID) {
        usage();
        exit (1);
    }

    if (strcmp(alarmstate, "on") == 0)
        alarmval = ALARM_ON;
    else
        if (strcmp(alarmstate, "off") == 0)
            alarmval = ALARM_OFF;
        else {
            usage();
            exit (1);
        }

    ald.alarm_no = altype;
    ald.alarm_state = alarmval;

    if (lom_ioctl(LOMIOCALCTL, (char *)&ald) != 0) {
        printf("Setting ALARM.%s to %s failed\n", alarm, alarmstate);
        return (1);
    } else {
        printf("Setting ALARM.%s successfully set to %s\n", alarm,
alarmstate);
        return (1);
    }
}

static int
parse_alarm(const char *alarm)
{
    int altype;
```

代碼示例 A-1 get/set 警報狀態的範例程式 (續)

```
#include <sys/types.h>
if (strcmp(alarm, "crit") == 0)
    altype = ALARM_CRITICAL;
else
if (strcmp(alarm, "major") == 0)
    altype = ALARM_MAJOR;
else
if (strcmp(alarm, "minor") == 0)
    altype = ALARM_MINOR;
else
if (strcmp(alarm, "user") == 0)
    altype = ALARM_USER;
else {
    printf("invalid alarm value:%s\n", alarm);
    altype = ALARM_INVALID;
}

return (altype);
}

static int
lom_ioctl(int ioc, char *buf)
{
    int fd, ret;

    fd = open(LOM_DEVICE, O_RDWR);

    if (fd == -1) {
        printf("Error opening device:%s\n", LOM_DEVICE);
        exit (1);
    }

    ret = ioctl(fd, ioc, (void *)buf);

    close (fd);

    return (ret);
}

static char *
get_alarmval(int state)
{
    if (state == ALARM_OFF)
        return ("off");
    else
        if (state == ALARM_ON)
```

代碼示例 A-1 get/set 警報狀態的範例程式 (續)

```
#include <sys/types.h>
        return ("on");
    else
        return (NULL);
}
static void
get_alarmvals()
{
    get_alarm("crit");
    get_alarm("major");
    get_alarm("minor");
    get_alarm("user");
}
```


索引

字母

ALOM

- LED 狀態指示器， 3
- 自動伺服器重新啓動， 38
- 基本功能， 37
- 設定密碼， 36
- 連接埠， 36
- 診斷工具， 2
- 概觀， 35
- 環境監視子系統， 39

ASR， 27

auto-boot? 變數， 8

BIST， *請參閱* 內建自我測試

CPU

- 時脈速度， 22
- 顯示資訊， 22

diag-level 變數， 8， 14

diag-script 變數， 8

diag-switch? 變數， 7， 8

FRU， 21 至 22

I²C 匯流排， 39

I²C 感應器， 39

IDE 匯流排， 11

input-device 變數， 9

LED， 診斷工具， 2

obdiag-trigger 變數， 9

OpenBoot PROM 參數， diag-level 變數， 7

OpenBoot 指令

probe-ide， 11

probe-scsi 與 probe-scsi-all， 10

show-devs， 12

執行， 13

診斷工具， 2

OpenBoot 組態變數

說明， 8

關鍵字， 8

OpenBoot 診斷， 13

控制測試， 14

啓動， 13

診斷工具， 2

OpenBoot 診斷測試

ok 提示處， 15

test 指令， 15

test-all 指令， 16

硬體裝置路徑， 15

錯誤訊息， 解譯， 16

output-device 變數， 9

POST

啓動診斷， 9

診斷， 控制， 8

診斷工具， 2

錯誤訊息， 8

post-trigger 變數， 9

probe-ide 指令 (OpenBoot)， 11

probe-scsi 與 probe-scsi-all 指令 (OpenBoot)， 10

prtconf 指令， Solaris， 17

- prtdiag 指令, Solaris, 19
- prtfru 指令, Solaris, 21
- psrinfo 指令, Solaris, 22
- SCSI 裝置, 診斷問題, 10
- SEAM, 33
- show-devs 指令, OpenBoot, 12
- showrev 指令, Solaris, 23
- Solaris OS
 - SunVTS, 32
 - 裝置樹, 17
- Solaris 指令
 - prtconf, 17
 - prtdiag, 19
 - prtfru, 21
 - psrinfo, 22
 - showrev, 23
 - 診斷工具, 2
- Sun Validation Test Suite (驗證測試套件)
 - 請參閱 SunVTS
- Sun 企業認證機制, 請參閱 SEAM
- SunVTS, 31 至 34
 - SEAM 安全性, 33
 - Solaris OS, 32
 - 介面, 31
 - 文件, 34
 - 可用測試, 32
 - 安裝, 33
 - 相容版本, 31, 33
 - 基本安全性, 33
 - 軟體, 測試模式, 31
 - 診斷工具, 2
 - 概觀, 31
 - 確定安裝, 33
 - 選用的軟體套件, 32
- test 指令 (OpenBoot 診斷測試), 15
- test-all 指令 (OpenBoot 診斷測試), 16
- test-args 變數, 15
 - 自訂測試, 14
 - 關鍵字, 15
- USB 裝置, OpenBoot 診斷自我測試, 16
- watch-net, 26
- World Wide Name (probe-scsi), 11

四畫

- 中央處理單元, 請參閱 CPU
- 內建自我測試, test-args 變數, 15

五畫

- 主要 (Major), 警報指示燈, 6
- 主機配接卡 (probe-scsi), 11

六畫

- 次要 (Minor), 警報指示燈, 7
- 自動伺服器重新啟動, 38
- 自動系統回復 請參閱 ASR, 27

七畫

- 位置指示 LED, 4
 - 狀態, 4
 - 亮起, 5
 - 熄滅, 5
- 伺服器狀態指示燈, 前後面板, 4
- 伺服器提示
 - OpenBoot 提示, 2
 - Solaris 軟體超級使用者提示, 2
 - 進階無電管理軟體提示, 2
- 系統狀態 LED
 - 請參閱 LED
 - 環境故障指示燈, 40
- 系統記憶體, 確定大小, 17
- 系統組態卡, 7

八畫

- 使用者 (User), 警報指示燈, 7

九畫

- 重設事件, 類型, 9

十畫

- 修訂版, 硬體與軟體, 顯示 showrev, 23
- 修補程式, 已安裝, showrev, 23
- 時脈速度, CPU, 22
- 記錄檔, 17
 - 系統訊息, 17
 - 錯誤訊息, 17

訊息

- POST, 錯誤, 8
- 解譯錯誤, 16

迴圈 ID (probe-scsi), 11

十一畫

- 常閉 (NC), 繼電器狀態, 7
- 常開 (NO), 繼電器狀態, 7
- 處理器速度, 顯示, 22
- 軟體修訂版, 顯示 showrev, 23

十二畫

- 硬體修訂版, 顯示 showrev, 23
- 硬體裝置路徑, 12, 15

診斷

- OpenBoot, 13
- POST, 8
- Solaris OS, 17
- SunVTS, 31

診斷工具

- ALOM, 2
- LED, 2
- OpenBoot 指令, 2
- OpenBoot 診斷, 2
- Solaris 軟體指令, 2
- SunVTS, 2
- 開機自我測試, 2

診斷測試, 略過, 9

進階無電管理軟體
請參閱 ALOM

開機自我測試
請參閱 POST

十三畫

- 溫度感應器, 39
- 溫度過低, 監視子系統, 39
- 溫度過高, 監視子系統, 39
- 裝置路徑, 硬體, 12, 15
- 裝置樹, Solaris 軟體, 顯示, 17
- 過熱狀況, 20
- 電源供應器, 故障監控, 40

十四畫

- 疑難排解工具, 2
- 監視子系統
 - 溫度過低, 39
 - 溫度過高, 39
- 需要維修 LED, 39

十六畫

- 整合驅動電子介面 (IDE), *請參閱* IDE 匯流排
- 錯誤訊息
 - OpenBoot 診斷, 解譯, 16
 - OpenBoot 診斷測試, 16
 - 記錄檔, 39
 - 電源相關, 40

十七畫

環境監視子系統, 39

二十畫以上

- 嚴重 (Critical), 警報指示燈, 6
- 繼電器狀態
 - 常閉 (NC), 7
 - 常開 (NO), 7
- 警報
 - get 狀態, 41 至 45
 - set 狀態, 41 至 45
 - 繼電器輸出 API, 41 至 45
- 警報卡
 - 警報狀態, 6
 - 警報指示燈, 6
- 警報狀態, 乾接點, 6
- 警報指示燈, 6
 - 主要 (Major), 6
 - 次要 (Minor), 7
 - 使用者 (User), 7
 - 嚴重 (Critical), 6
- 邏輯單元編號 (probe-scsi), 11

