



用於 Sun Fire™ B1600 的 Sun™ SNMP Management 代理程式指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

文件編號：817-2504-10
2003 年 4 月，修訂版 01

請將您對本文件的意見寄到：<http://www.sun.com/hwdocs/feedback>

Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版權所有。

Sun Microsystems, Inc. 對本文件所述之產品中的相關科技擁有智慧財產權。特別是，且無限制，這些智慧財產權可包含一或多項 <http://www.sun.com/patents> 中列示的美國專利，以及一或多項其他在美國或其他國家的專利或申請中的專利。

本文件及其相關產品受版權保護，且按照限制其使用、複製、分發、和反編譯的授權許可進行分發。未經 Sun 及其授權許可頒發機構的書面授權，不得以任何方式、任何形式複製本產品或本文件的任何部分。

協力廠商軟體，包括字型技術，由 Sun 供應商提供許可和版權。

本產品的某些部分從 Berkeley BSD 系統衍生而來，經 University of California 許可授權。UNIX 是在美國和其他國家註冊的商標，經 X/Open Company, Ltd. 獨家許可授權。

Sun、Sun Microsystems、Sun 標誌、AnswerBook2、docs.sun.com、Sun Fire、Java 和 Solaris 是 Sun Microsystems, Inc. 在美國與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 商標的產品都基於 Sun Microsystems, Inc. 開發的架構。

OPEN LOOK 和 Sun™ 圖形使用者介面是 Sun Microsystems, Inc. 為其用戶與授權許可持有人開發的。Sun 承認 Xerox 在為電腦行業研究和開發可視或圖形使用者介面方面所作出的先行努力。Sun 以非獨佔方式從 Xerox 獲得 Xerox 圖形使用者介面的授權許可，該授權許可亦涵蓋實施 OPEN LOOK GUI 且遵守 Sun 的書面許可協議的授權許可持有人。

本資料按「現有形式」提供，不承擔明確或隱含的條件、陳述和保證，包括對特定目的或非侵害性的商業活動和適用性的任何隱含保證，除非這種不承擔責任的聲明是不合法的。



請回收



Adobe PostScript

目錄

第一部分 . 技術性說明和功能

1. Sun SNMP Management 代理程式補充說明 3
2. 介紹 SNMP 5
 - SNMP 版本 5
 - SNMP 管理員和代理程式 6
 - SNMP 管理資訊資料庫 6
 - MIB 表 7
 - 存取控制 8
 - SNMP 主代理程式 8
 - SNMP 中介程式和 snmpdx 9
3. 主代理程式 11
 - 功能 11
 - 配置概觀 11
4. 平台管理模式 13
 - 顯示 Sun Fire B1600 平台 13
 - 管理物件 14

sunPlat 等級緣由	15
5. Sun Fire B1600 MIB	17
SNMP 呈現模式	17
實體模式	19
等級	21
邏輯模式	22
邏輯和實際階層對映	22
事件與警報模式	22
SUN-PLATFORM-MIB	23
實體模式表擴展	23
邏輯模式表擴展	26
事件 (Event) 和警報 (Alarm) 日誌表	26
事件記錄	27
事件	27
警報	27
6. 實體模式	29
sunPlat 實體等級階層	29
sunPlat 等級定義	31
實際實體	31
sunPlat 設備等級	33
sunPlat 電路封包等級	34
sunPlat 設備支架	35
sunPlat 電源供應器	36
sunPlat 電池	37
sunPlat 監視系統	38
sunPlat 警報	39

sunPlat 風扇	40
sunPlat 感應器	40
sunPlat 二進位感應器	41
sunPlat 數字感應器	42
sunPlat 分離感應器	43
sunPlat 基座	44
7. 邏輯模式	45
sunPlat 邏輯等級階層	45
SunPlat 邏輯等級定義	46
邏輯實體	46
邏輯	47
sunPlat 單一電腦系統	48
sunPlat 管理領域	48
8. sunPlat 通知	49
sunPlat 通知等級階層	49
sunPlat 事件記錄等級	50
sunPlat 等級定義	51
sunPlat 事件記錄	51
sunPlat 事件額外記錄	51
sunPlat 物件建立記錄	52
sunPlat 物件刪除記錄	52
sunPlat 警報記錄	52
sunPlat 未知警報記錄	53
sunPlat 傳送警報記錄	53
sunPlat 環境警報記錄	53
sunPlat 設備警報記錄	54

sunPlat 處理警報記錄	54
sunPlat 服務品質警報記錄	54
sunPlat 屬性值變更記錄	54
sunPlat 狀態變更記錄	55

第二部分 . 安裝和配置

9. 管理軟體組件	59
系統管理選項	59
設備	60
系統需求	61
作業環境	61
磁碟空間需求	61
增補程式	61
Solaris 8	61
Solaris 9	61
Java 環境	61
確認安裝	62
Java SNMP API	63
安裝套裝軟體	63
升級軟體	64
套裝軟體內容	64
安裝領域或目標套裝軟體於 Sun Fire B100s	65
對系統檔案的影響	65
10. 安裝軟體	67
選擇要安裝的軟體配置	67
設備配置	67
管理介面配置	68

安裝 SNMP 軟體	68
安裝領域硬體監視軟體	69
安裝平台硬體監視軟體	70
配置系統控制器	74
介面選項	75
snmpdx 的 SNMP (預設)	75
使用主代理程式和 snmpdx 的 SNMP	76
使用 SNMP 的第三協力廠商主代理程式	78
11. 配置檔案	79
配置檔案	79
一般配置檔	80
spama.conf	80
一般選項	80
主代理程式選項	81
通訊協定中介程式選項	82
存取控制	88
ACL 檔格式	89
acl 群組	89
trap 群組	90
中介程式配置檔	91
spapm.acl 檔	91
spapm_snmpdx.acl 檔	93
主代理程式配置檔	94
spama.acl 檔	95
acl 群組	95
trap 群組	95
spama.uacl 檔	95

	acl 群組	96
	spama.security 檔	96
12.	配置軟體	99
	預設配置	99
	存取控制	99
	啓動和停止中介程式	100
	手動配置直接存取	100
	設定爲第三協力廠商主代理程式之子代理程式的中介程式	100
	中介程式和 SNMPv3 主代理程式	101
	啓動和停止代理程式	101
	轉寄 SNMPv3 陷阱	102
13.	解除軟體安裝	103
	平台代理程式和目標代理程式套裝軟體	103
	領域代理程式套裝軟體	104
14.	疑難排解	105
A.	安裝 J2RE 1.4 並與 J2SE 1.3.1 共存	111
	安裝 J2RE 1.4	111
	編輯啓動 Script	113
	領域硬體監視	113
	平台硬體監視	113
	索引	115

圖表

圖 1-1	領域和平台硬體監視範例	4
圖 4-1	硬體資源階層範例	14
圖 4-2	sunPlat 管理物件等級繼承示意圖	15
圖 5-1	硬體資源階層範例	19
圖 6-1	sunPlat 實體資源繼承等級示意圖	30
圖 7-1	sunPlat 邏輯資源繼承等級示意圖	46
圖 8-1	事件記錄繼承等級示意圖	50
圖 9-1	領域和平台硬體監視範例	60
圖 10-1	SNMP 為 snmpdx 子代理程式時的資料流程	76
圖 10-2	使用主代理程式時的資料流程	77
圖 10-3	使用第三協力廠商主代理程式時的資料流程	78

表格

表 5-1	實際實體表	20
表 5-2	實體對映表	21
表 5-3	實際實體表擴展	25
表 5-4	實際實體表擴展重要資訊 (表 5-3)	26
表 6-1	實際實體母等級之「等級」屬性對映	32
表 6-2	操作狀態屬性值	33
表 6-3	可用性狀態屬性值	34
表 6-4	設備支架種類屬性值	35
表 6-5	設備支架狀態屬性值	36
表 6-6	監視系統動作屬性值	38
表 6-7	警報類型屬性值	39
表 6-8	警報狀態屬性值	39
表 6-9	感應器類型屬性值	41
表 8-1	sunPlat 警報記錄已知嚴重性值	52
表 9-1	SNMP Management 代理程式 套裝軟體說明	63
表 9-2	SNMP Management 代理程式套裝軟體封包	64
表 9-3	啓動 Script	65
表 10-1	圖 10-1 的連接埠摘要	75
表 10-2	圖 10-2 的連接埠摘要	77

表 10-3	圖 10-3 的連接埠摘要	78
表 11-1	spama.conf 的預設值	84
表 11-2	spama.security 中的使用者可配置參數	97

程式範例

程式範例 10-1	設定 SMS IP 位址	74
程式範例 11-1	spama.conf 檔案範例	85
程式範例 11-2	acl 群組範例	90
程式範例 11-3	trap 群組範例	91
程式範例 11-4	spapm.acl 檔範例	92
程式範例 11-5	spapm_snmpdx.acl 檔範例	93
程式範例 11-6	acl 群組範例	95
程式範例 11-7	spama.uacl 檔範例	96
程式範例 11-8	spama-security 檔範例	98

前言

本指南將詳細說明用於 Sun Fire B1600 平台的 Sun SNMP Management 代理程式相關資訊，其支援使用「簡易網路管理協定」(Simple Network Management Protocol)的平台硬體管理。

Sun SNMP Management 代理程式不僅可以*監視*庫存、配置、以及環境和錯誤報告，同時也可以*控制*和*監視*服務指示器、以及處理器 Blade 的電源、待機和重設狀態。

只有經驗豐富的「企業管理員」(Enterprise Administrator)和專業開發者才可讀取此代理程式的內容。

本指南內容分成以下兩個部分：

- 第一部分（從第一章到第八章）介紹 SNMP Management 代理程式並說明其功能。
- 第二部分（從第九章到第十三章）說明如何安裝和配置軟體。

本書編排架構

本指南包含以下章節：

第一部分

第一章將說明 Sun SNMP Management 代理程式軟體的組成元件。

第二章將簡介「簡易網路管理協定」(Simple Network Management Protocol, SNMP)的主要功能。

第三章將說明 SNMPv3 主代理程式的功能和特色。

第四章將概要說明 SNMP 如何呈現 Sun Fire B1600。

第五章將說明 Sun Fire B1600 管理物件與物件間的關係如何呈現在 SNMP 介面上。

第六章將說明 sunPlat 的實體等級階層，以及定義於 sunPlat 模式之管理實體物件等級如何在 SUN-PLATFORM-MIB 中呈現。

第七章將說明 sunPlat 邏輯等級階層，以及定義於 sunPlat 模式之管理物件等級如何在 SUN-PLATFORM-MIB 中呈現。

第八章將說明定義於 SUN-PLATFORM-MIB 中的 SunPlat 通知等級和屬性。

第二部分

第九章將說明 Sun Fire B1600 軟體的組成元件，並列出安裝 SNMP 軟體前必須做的系統檢查。

第十章將說明如何在 Sun Fire B1600 安裝此軟體。

第十一章將提供使用者配置檔的相關資訊。

第十二章將說明安裝完畢後的預設配置，並解釋如何修改配置檔。

第十三章將說明如何解除此軟體安裝。

第十四章將提供此軟體的相關疑難排解。

附錄 A 將說明如何在與 J2SE 並存的情況下安裝 J2RE，以及如何修改啓動 Script 以連至安裝作業的方法。

排印慣例

字體或符號	意義	範例
<i>AaBbCc123</i>	指令、檔案和目錄的名稱；電腦螢幕輸出。	編輯 <code>.login</code> 檔案。 使用 <code>ls -a</code> 列出所有檔案。 <code>% You have mail.</code>
AaBbCc123	輸入的內容，可與電腦螢幕輸出對照	<code>% su</code> Password:
<i>AaBbCc123</i>	書名、新字或專有名詞、要強調的文字。以實際名稱或實際值代替指令行變數。	請閱讀 <i>《使用者指南》</i> 的第六章。 這些即稱為類別選項。 若要刪除檔案，請鍵入 rm 檔案名稱。

Shell 提示符號

Shell	提示符號
C shell	主機名稱 %
C shell 超級使用者	主機名稱 #
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 超級使用者	#

相關文件資料

適用範圍	書名	文件編號
SunMC	《用於 Sun Fire B1600 的 SunMC 3.0 補充資料》	817-2541
發行說明	《SNMP Release Notes for the Sun Fire B1600》	817-1006
Sun Fire B1600 平台	《Sun Fire B1600 Blade System Chassis Hardware Installation Guide》	816-7614
	《Sun Fire B1600 Blade System Chassis Software Setup Guide》	816-3361
	《Sun Fire B1600 Blade System Chassis Administration Guide》	816-4765
	《Sun Fire B1600 Blade System Chassis Switch Administration Guide》	816-3365

存取 Sun 文件資料

若您須檢視與列印多種精選的 Sun 文件資料及各本土化版本，請至：

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

聯絡 Sun 技術支援

若本文件無法解決您對於本產品相關技術上的疑惑，請至以下網址尋求協助：

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

Sun 歡迎您的指教

Sun 一直致力於改善相關的文件資料，因此歡迎您提出批評和建議。您可以將您的意見傳送至：

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

請在您的意見中加入本書名和文件編號：

《用於 Sun Fire B1600 的 Sun SNMP Management 代理程式指南》，文件編號 817-2504-10

第一部分 技術性說明和功能

Sun SNMP Management 代理程式 補充說明

本版次的 Sun™ SNMP Management 代理程式可提供監控 Sun Fire™ B1600 機架 (Shelf) 和 Sun Fire B100s Blade 的功能。

您可依據平台類型選擇適合的代理程式：

- 領域代理程式，執行於 Sun Fire B100s Blade（領域硬體監視）

此軟體會安裝於本端已受到監視的伺服器上，而且只有該伺服器會受到監視。在 Sun Fire B1600 中，每個 Blade 都會受到個別監視。

Sun Fire B100s Blade 的領域硬體監視核心僅會受到 Blade 硬體之限制，而且並不包含其他機架組件，例如服務指示器、PSU、SSC 和機架本身。

- 平台代理程式，透過系統控制器代理（平台硬體監視）

此軟體會安裝於遠端（平台代理程式）伺服器中，其可透過系統控制器存取平台設備，並讓您監視由系統控制器管理的所有硬體。

Sun Fire B1600 的平台硬體監視核心包含機架、機架實體、服務指示器、和其他所有可現場置換單元 (FRU)。除此之外，使用此監視也可取得部分 Sun Fire B100s Blade 硬體資訊（特別是電壓監視）。使用領域硬體監視反而得不到此資訊。

圖 1-1 顯示兩種硬體監視的範例。Sun Fire B1600 機架 A 和 B 透過平台代理程式伺服器（平台硬體監視）連線至網路管理站，而 Sun Fire B1600 機架 C 的範例則是將 Sun Fire B100s Blade 直接連接到網路管理站（領域硬體監視）。

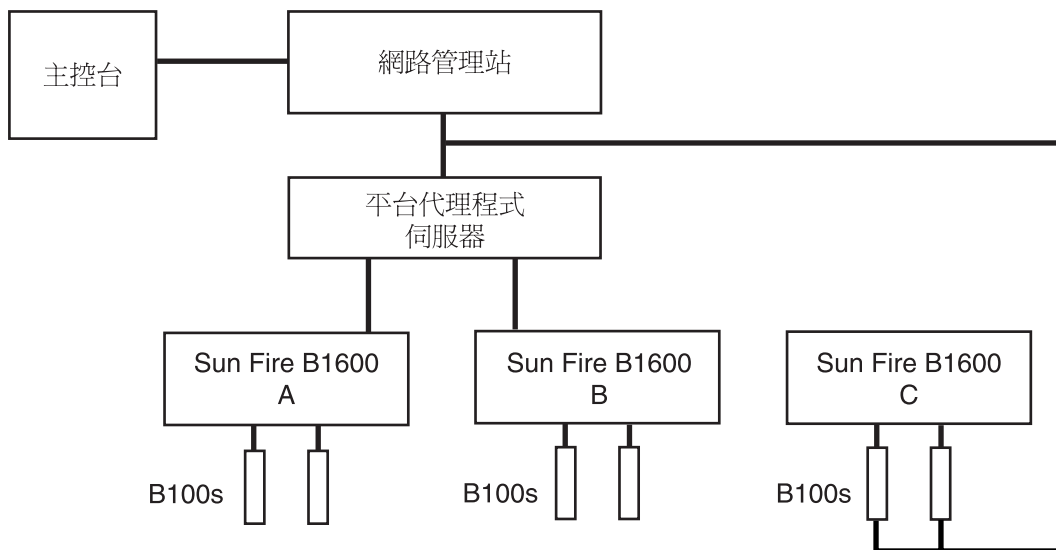


圖 1-1 領域和平台硬體監視範例

此軟體是由數種套裝軟體組成，並提供以下功能：

- SNMP 子代理程式

根據預設值，SNMP 子代理程式會登錄為 Solaris 主代理程式 (snmpdx) 的子代理程式。此子代理程式即為 *SNMP 中介程式*。

- SNMPv3 主代理程式

SNMPv3 主代理程式提供單一且安全的聯結點，讓位於此平台的 SNMP 中介程式可以被存取。主代理程式的功能就有如 Proxy 一般，可轉送要求至 snmpdx。

- Sun Fire B1600 和 Sun Fire B100s 設備。

上述套裝軟體須視使用的硬體監視類型（領域或平台）來選擇安裝。

介紹 SNMP

本章將會簡介「簡易網路管理協定」(Simple Network Management Protocol, SNMP) 的主要功能。本章並不會詳盡說明與 Sun Fire™ B1600 系統特定的相關主題。

本章包含以下內容：

- 第 5 頁的「SNMP 版本」
- 第 6 頁的「SNMP 管理員和代理程式」
- 第 6 頁的「SNMP 管理資訊資料庫」
- 第 8 頁的「SNMP 主代理程式」

SNMP 版本

SNMP 是用來管理連線裝置（系統）的公開網際網路標準。SNMP 與其他網際網路標準相同，都是由「網際網路工程工作小組」(Internet Engineering Task Force, IETF) 制定的數種 RFC (Requests for Comments) 所定義。

用來定義許可標準的 SNMP 版本有三種：

- SNMPv1
- SNMPv2（即為本文件中的 SNMPv2c）
- SNMPv3

SNMPv1 初次定義於 1988 年。SNMPv2 於 1993 年問世，其藉由新增更多通訊協定操作和資料類型，並提供安全性的方式點出 SNMPv1 的缺點。SNMPv1 在安全模式上的限制導致現今 SNMPv2c 標準的出現，並開發出全新的安全性功能。實驗版本 SNMPv2usec 和 SNMPv2* 也在此時出現，但其並未受到廣泛使用，而且始終處於實驗階段。

SNMPv3 於 1999 年問世，其定義出支援可插拔組件（包括安全性）的 SNMP 管理架構。

如需瞭解更多上述標準的相關資訊，請參閱以下 IETF 網站 (<http://www.ietf.org/rfc.html>) 上的 RFC 資訊：

- SNMPv1: RFC1155、RFC1157、RFC1212、RFC1215
- SNMPv2: RFC2578、RFC2579、RFC2580、RFC3416
- SNMPv3: RFC3410、RFC3411、RFC3412、RFC3413、RFC3414、RFC3415
- 標準間的共存性：RFC2576

SNMP 管理員和代理程式

SNMP 是一種網路通訊協定，可讓「網路管理站」(Network Management Station, NMS)，又稱**管理員 (Manager)**，遠端遙控其管理的裝置。

若要管理裝置，此裝置必須要與 **SNMP 代理程式** (即為所謂的 SNMP 中介程式) 相連才行。中介程式 (Mediator) 的功用為：

- 從管理員接收代表裝置狀態的資料要求，並提供適當的回應
- 從管理員接收資料來啟動裝置狀態控制
- 產生 **SNMP 陷阱 (Trap)**，其為自動產生的訊息，並會傳送至一個或多個選定的管理員來通知與此裝置相關的重大事件

SNMP 管理資訊資料庫

若要管理和控制裝置，就必須使用代理程式和管理員都知道的格式來顯示其特徵。這些特徵可顯示實體屬性，例如風扇速度、或諸如路由表的維修。而定義這些特徵的資料架構即稱為**管理資訊資料庫 (MIB)**。此資料模式通常以表格呈現，但也可以包含簡單的數值。前者範例即為路由表，而後者範例即為可顯示何時啟動代理程式的時間戳記。

MIB 會定義可透過 SNMP 存取的虛擬資料儲存區。使用 `get` 和 `set` 操作就可以從管理員存取相關內容：

- 若使用 `get` 操作，中介程式就會提供其管理裝置於本端維護或直接維護的資料。
- 若使用 `set` 操作，代理程式就會執行一些作業來影響其狀態或其管理裝置的狀態。

若要啟動 NMS 透過代理程式來管理裝置，與 MIB 相對的資料 (由代理程式提供) 就必須載入至管理員。此操作機制須視網路管理軟體的使用而有所不同，才可提供管理員正確定址並解讀代理程式顯示的資料模式資訊。

注意 – MIB 之間可相互提供參考定義，所以若要使用現有的 MIB，就必須載入其他 MIB。

若要得知虛擬資料儲存區的內容，就必須為物件辨識碼 (OID) 定義 MIB。一組 OID 包含定義個別名稱空間的整數，其以階層順序排列組成。每個指派的整數都擁有相關的文字名稱。舉例來說，OID 1.3.6.1 代表 iso.org.dod.internet OID 名稱，而 1.3.6.1.4 即代表 iso.org.dod.internet.private OID 名稱。

數字格式可用於 SNMP 通訊協定傳輸中，而文字格式則用於使用者介面以增加可讀性。OID 代表的物件通常以其名稱的最後部分來做為簡寫格式。為了避免使用此慣例會帶來的任何混淆，通常會套用 MIB 特定的字首至所有定義好的物件名稱，例如 *sunPlat*，才不會有相互名稱重複的辨識碼出現。

注意 – MIB 是由 ASN.1 語言所定義。其相關資訊並非本文件的探討重點，但若您有興趣的話，SNMPv2c 的 MIB 架構是由其「管理資訊架構」(Structure of Management Information, SMI) 所定義。請參閱 RFC2578。它還定義了 MIB 的可用語法和基本資料類型。定義於 RFC2579 的文字使用慣例（類型定義）可定義其他資料類型和數值。

MIB 表

MIB 定義的大部分資料內容都是以表格的方式呈現，並由包含其 OID 的物件項目排列而成。舉例來說，風扇特徵表每列皆為一個風扇，並包含對應目前速度、預期速度、與最大可接受速度的欄位。

表格各列之定址：

- 單一數字的索引（表格內的列數，例如 '6'）
- 較複雜、數字較多的實例辨識碼，例如 IP 位址和連接埠編號（例如，127.0.0.1, 1234）

每個 MIB 表格定義都有一個「索引」(INDEX) 說明，指出該使用哪個實例辨識碼選擇現有的項目。在上述兩個情況之下，用來定義特定列之索引的物件必須定義於 MIB 之中。因此，內含單一索引的表格通常都擁有表格「索引」說明之索引欄。而表格內的特定資料項目就會由指定 OID（已套用特定字首）完成定址。

舉例來說，字首為 myFanTable.myFanEntry.myCurrentFanSpeed 的實例辨識碼（例如前例的 127.0.0.1.1234）就會變成 myFanTable.myFanEntry.myCurrentFanSpeed.127.0.0.1.1234。

定義 MIB 語法的 SMI 可增加表格中的欄位，有效擴展表格增加新的項目。此舉可藉由定義擁有「索引」說明的表格達成，即複製已擴展表格的「索引」說明。

若要定義 MIB 表格不需非得經由其內含的索引物件，也可經由其他表格匯入物件來定義（有可能定義於其他 MIB）。此結構可有效地新增欄位至現有表格。

注意 – SUN-PLATFORM-MIB 延申使用此機制以擴展定義於 ENTITY-MIB（請參閱第五章）中的表格。

存取控制

MIB 內所有定義好的定址物件皆擁有相關之最大存取權，例如**唯讀**(read-only) 或**讀寫**(read-write)。上述權利可決定代理程式支援的最大存取，管理員也可藉此用來限制操作者的使用。代理程式可套用較低存取權，即為拒絕寫入擁有讀寫權之物件。拒絕權是以下列為基礎：

- 此操作對於已定址物件會產生多大的效用（例如，MIB 定義物件的位置代表狀態主機，其僅允許特定傳輸）
- 限制特定操作以禁止部分管理員存取之安全性限制

在 SMMPv1 中用來溝通安全性存取權之機制即為**通連字串**(community string)。通連字串為單純的文字字串，例如 *private* 或 *public*，用以通過每個 SNMP 資料要求。因為 SNMPv1 和 SNMPv2 要求並不會加密，所以並沒有任何安全性可言。用來定義代理程式應予以回應之通連字串，以及選取管理員之機制須視代理程式的操作而定，但通常是根據「存取控制清單」(Access Control List, ACL) 內的資料來加以定義，其為說明可用存取限制之檔案。

若要得知如何配置 ACL，請參閱第十一章。

SNMP 主代理程式

管理員藉由傳送 (UDP) 封包至代理程式所在系統上的已知連接埠 (161) 來與代理程式溝通。若現有系統上有數個代理程式正在執行，且每個代理程式分別管理不同的裝置，就有可能造成連接埠資源使用上的衝突。解決方法之一為每個代理程式使用不同、非標準之連接埠編號。另一解決方法則為使用**主代理程式**的觀念，其可代表所有執行於同一系統之代理程式接收 SNMP 要求，並於適當時機轉發出 SNMP 要求。此觀念可讓管理員使用相同方式存取所有 SNMP 代理程式。Sun Fire B1600 可支援以上解決方法其中之一。

如需瞭解主代理程式的相關資訊，請參閱第三章。

SNMP 中介程式和 snmpdx

snmpdx 是標準 Solaris™ SNMP 代理程式，且為一種 SNMPv1 主代理程式。

根據預設值，SNMP 中介程式會登錄為 snmpdx 的子代理程式。在此配置中僅可支援 SNMPv1 get 和 set 要求，即使已發出 SNMPv2c 通知。

snmpdx 與 SNMP 中介程式的關係於第十章和第十一章有更詳盡的說明。

請同時參閱線上說明手冊 snmpdx(1M)。

主代理程式

本章將說明 SNMPv3 主代理程式的功能和特色。

本章包含以下內容：

- 第 11 頁的「功能」
- 第 11 頁的「配置概觀」

功能

SNMPv3 主代理程式提供單一且安全的聯結點以協助存取 SNMP 管理資訊。

SNMPv3 主代理程式會連接至 SNMP 服務連接埠（預設為 161），然後轉寄所有要求至 Solaris 標準主代理程式 (`snmpdx`)，再由主代理程式轉寄各要求至適合的登錄子代理程式。`snmpdx` 為標準 Solaris Distribution 的一部分，但僅支援 SNMPv1，所以無法直接提供 SNMPv3 擁有的安全性。主代理程式會統一將所有要求（SNMPv1、v2c 或 v3）翻譯為 SNMPv1，如此一來 `snmpdx` 才能處理這些要求。

事實上，主代理程式可提供所有現存子代理程式的安全存取，就有如 SNMP 防火牆一般。

配置概觀

本節將介紹如何配置 SNMP 包含主代理程式功能的方法。詳細說明請參閱第十一章，其包含本節待續之完整解說，還有配置檔的範例。

SNMP 中介程式會登錄為使用自動分配連接埠編號之 `snmpdx` 的子代理程式。

注意 – 本指南中的 SNMP 代理程式即為 SNMP 中介程式。

啓動主代理程式後，就會關閉 `snmpdx` 自動啓動，而主代理程式也會登錄至連接埠 161。新的連接埠編號會指定給 `snmpdx`，並在主代理程式啓動檔的控制下啓動。

配置方法如下：

- SNMPv3 安全檔 `spama.uacl` 和 `spama.security`
- SNMPv1/v2 存取控制清單 (ACL) 檔 `spama.acl`
- 配置檔 `spama.conf`
- 啓動 Script `spama`

您無法在主代理程式執行時進行動態配置。

SNMP 中介程式也需要進行配置。詳情請參閱第十一章。

平台管理模式

本章將說明 SNMP 如何使用 Sun 平台 SNMP 模式 (sunPlat) 來顯示 Sun Fire B1600。

本章包含以下內容：

- 第 13 頁的「顯示 Sun Fire B1600 平台」
- 第 14 頁的「管理物件」
- 第 15 頁的「sunPlat 等級緣由」

顯示 Sun Fire B1600 平台

Sun Fire B1600 可顯示基座內的總硬體資源。部分資源可以直接歸類於基座內，例如主機板。但部分資源則會歸類至其他資源 — 舉例來說，主機板內也可以包含處理器。上述關係是由基座內擴展出來，進而形成硬體資源階層，而且每個實體都包含一個封閉的父層 (parent)。此階層是藉由使用代表硬體資源管理物件之間的關係來顯示。

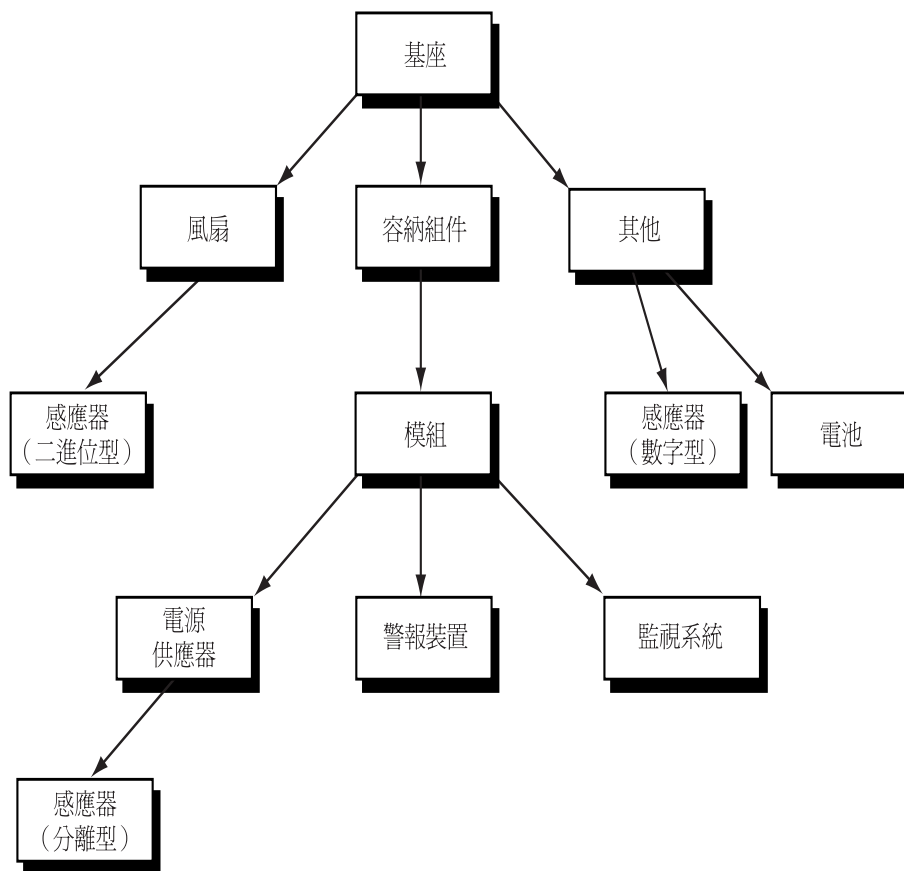


圖 4-1 硬體資源階層範例

管理物件

sunPlat 模式可提供有用的一般平台整體架構組來表示基本硬體資源，而平台整體架構的實例就稱為**管理物件**。若可以監視硬體資源，或硬體資源可提供有用的配置資訊，就會出現於管理物件中。

其他管理物件則可用來顯示管理介面的其他功能。例如，硬體資源可以發出不同狀態報告（**通知**），以回應問題（**警報**）或變更配置（**事件**）。

管理物件是基於管理物件等級所定義，而資源的特徵則是藉由管理物件的**屬性**所呈現。新等級（即**子等級**，subclass）則是基於現有等級所定義。子等級會**繼承**其**母等級**的所有特徵，但也可藉由新增屬性呈現自己的特徵。

圖 4-2 顯示 sunPlat 模式中所定義之硬體整體架構的等級繼承階層。

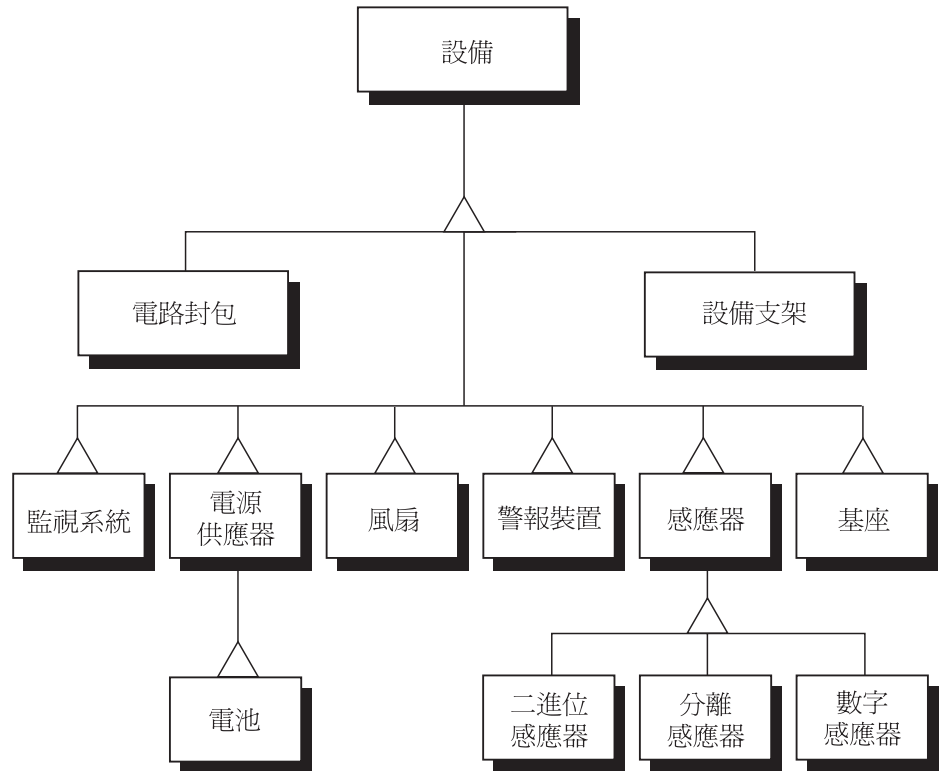


圖 4-2 sunPlat 管理物件等級繼承示意圖

sunPlat 等級緣由

sunPlat 等級是基於企業標準管理觀念而成。Sun Fire B1600 系統使用 ITU-T 通用網路架構 (Generic Network Information Model) 子集，其由硬體架構挑選而出。這可提供強大且可延伸的架構，並支援電信管理網路 (Telecommunications Management Network, TMN) 中的 uniform fault 和配置管理。

Distributed Management Task Force (DMTF) Common Information Model (CIM) Schema 會顯示實際環境和事件定義與相關處理，並提供系統專用之延伸操作至一般模式中。

Sun Fire B1600 MIB

本章將說明 Sun Fire B1600 管理物件與物件間的關係如何藉由 SNMP 介面呈現。

本章包含以下內容：

- 第 17 頁的「SNMP 呈現模式」
- 第 19 頁的「實體模式」
- 第 22 頁的「邏輯模式」
- 第 22 頁的「邏輯和實際階層對映」
- 第 22 頁的「事件與警報模式」
- 第 23 頁的「SUN-PLATFORM-MIB」

SNMP 呈現模式

SNMP 中介程式支援輪詢和事件管理。Sun Fire B1600 系統的實際組件和管理領域的邏輯呈現都是由 ENTITY-MIB 提供（如 RFC 2737 中所定義），並由 SUN-PLATFORM-MIB 擴展。

注意 – 許多 MIB 內定義的物件都有 read-write（讀寫）的 MAX-ACCESS（最大存取），但只在此操作適用於顯示組件時，才可寫入這些物件。

ENTITY-MIB 包含以下幾種群組，其說明了管理系統之實體和邏輯要素：

entityPhysical 群組

`entityPhysical` 群組可說明實際實體 (physical entity) — 由代理程式管理的可辨識實際資源（例如基座、電源供應器、感應器等等）。上述實體於 `entPhysicalTable` 中的各列項目有加以說明。

entityLogical 群組

`entityLogical` 群組可說明由代理程式管理的邏輯實體 (logical entity)。這些實體為高數值 (high value) 邏輯實體的表現，其提供必須由高階管理處理的服務抽象概念，並主要用於平台硬體管理，還包含諸如 OS 重新啟動、硬體重設和電源控制的功能。一般而言，這些實體相當於 Solaris 領域的管理領域或服務控制器。

entityMapping 群組

`entityMapping` 群組可分辨 `entityPhysical` 群組和 `entityLogical` 群組之間的關係。此功能由內部 SNMP 中介程式執行。

entityGeneral 群組

當實際實體表 (Physical Entity Table) 或實體對映表 (Physical Mapping Table) 內的任何實體改變時，`entityGeneral` 群組可提供最後變更的時間戳記。

entityMIBTraps 群組

`entityMIBTraps` 群組可定義用於發送 ENTITY-MIB 內任何物件變更訊號之 `entPhysicalChange` 通知。

第二章對於 SNMP 的一般元件該如何顯示 Sun 平台 SNMP 模式有簡要的說明。

實體模式

sunPlat 實體模式 (physical model) 使用 ENTITY-MIB 提供硬體實體的包含階層。每個實體會顯示為 ENTITY-MIB 之 *entPhysicalTable* 中的單獨各列。

圖 5-1 顯示實際包含階層的範例。右下角的數字為 *entPhysicalTable* 中的相對列索引 (請參閱表 5-1)。

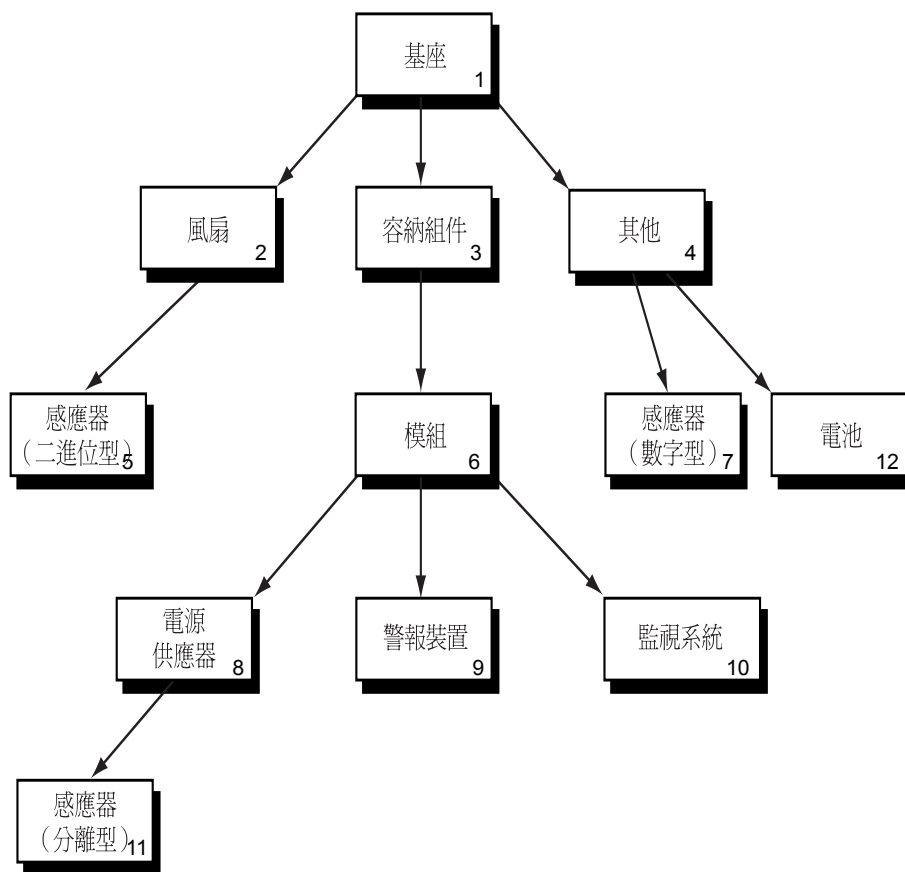


圖 5-1 硬體資源階層範例

使用 SNMP 表將顯示下列資訊：

■ 實際實體表 (*entPhysicalTable*)

此表每列皆為單一硬體實體。這些列就稱為 *實體*，而特定列則稱為 *實例*。每個項目都包含：

- 實體等級 (*entPhysicalClass*)
- 硬體實體的一般特徵
- 一個獨特的索引 (*entPhysicalIndex*)
- 參考值 (*entPhysicalContainedIn*)，可點出資源內做為 *容納組件 (container)* 的硬體實體列。組件的部分則為 0，其通常不會包含在另一個容納組件內，例如基座。

■ 實體對映表 (*entPhysicalContainsTable*)

此表提供了實體對映表中硬體資料的虛擬階層。本表分成二區，先有總內含項目 *entPhysicalIndex* 的索引，然後則是個別內含項目 *entPhysicalIndex* 的索引。

表 5-1 顯示 *entPhysicalTable* 是基於上述何種數值而定，而表 5-2 則顯示了實體對映 (physical mapping)。

表 5-1 實際實體表

entPhysicalIndex	entPhysicalClass	entPhysicalContainedIn
1	基座	0
2	風扇	1
3	容納組件 (例如包含 FRU 的插槽)	1
4	其他	1
5	感應器 (二進位型)	2
6	模組 (例如可插拔的 FRU)	3
7	感應器 (數字型)	4
8	電源供應器	6
9	警報裝置	6
10	監視系統 (watchdog)	6
11	感應器 (分離型)	8
12	電源供應器 (電池供電)	4

表 5-2 實體對映表

entPhysicalIndex	entPhysicalChildIndex
1	2
1	3
1	4
2	5
3	6
4	7
4	12
6	8
6	9
6	10
8	11

等級

entPhysicalClass 為一種數值，可提供特定實際實體的一般硬體類型指示，並呈現在 *entPhysicalTable* 的各列項目中。

以下等級適用於 Sun Fire B1600 平台（請同時參閱圖 5-1 和表 5-1）：

- other(1)

當實際實體無法由以下各項區分時，就適用於 other 等級。

- chassis(3)

chassis 等級代表裝置的整體容納組件。任何實際實體等級都可以包含在基座中。

- container(5)

container 等級適用於可包含一至二個以上可移除 (removable) 實際實體之實際實體，類型相同或異同皆可。舉例來說，基座中的插槽（不論是否為空的）都可被視為容納組件，而可現場置換單元（FRU），如電源供應器或風扇，則被視為容納組件實體內的模組。

- powerSupply(6)

power supply 等級適用於可供應電源的組件。

- fan(7)

fan 等級適用於實際實體為風扇或其他冷卻裝置的設備。

- sensor (8)

sensor 等級適用於可測量部分實體屬性的實際實體。

- module (9)

module 等級適用於可自我容納之子系統，其呈現於其他實際實體之中，例如 chassis 或另一個 module。此實體總是顯示於 container 中。

邏輯模式

sunPlat 邏輯模式 (logical model) 使用 ENTITY-MIB 提供高數值的邏輯實體清單。每個實體會呈現於 ENTITY-MIB *entLogicalTable* 中的單獨各列。注意：邏輯模式與實際模式不同，而且比階層模式 (hierarchical model) 還要平。

ENTITY-MIB 無法分辨不同等級的邏輯物件，與實際物件不同。SUN-PLATFORM-MIB 提供邏輯物件的等級階層。詳情請參閱第七章。

entLogicalTable 中的資料可用來支援使用不同名稱項目之多重範圍 (multi-scoping)，但是本產品並不支援這項功能。特定值資訊為 *entLogicalDescription* 和 *entLogicalTAddress*，且後者有提供可存取邏輯實體之 IP 位址。

邏輯和實際階層對映

ENTITY-MIB 提供邏輯物件和實際物件之對映。此舉可經由使用 *entLPMappingTable* 達成。此表分成二區（與 *entPhysicalContainsTable* 類似），並可分辨瞭解現有邏輯實體之實際實體。實際實體可藉由 *entLPPhysicalIndex* 來分辨，其與 *entPhysicalIndex* 相同。

雖然此表格可顯示所有與現有邏輯實體相關之實際實體，但按照慣例，只能參照內含之實際實體。舉例來說，對於實際模組可辨識之邏輯實體而言，其對映僅可參照模組，而非所有包含在其中的實際實體。

事件與警報模式

ENTITY-MIB 提供單一 SNMP 通知 (*entConfigChange*) 以用於警示 MIB 內任何表格的變更。其已被設定每五秒提供一個陷阱的最大值。

SUN-PLATFORM-MIB 可定義更詳細的通知。詳情請參閱第八章。

SUN-PLATFORM-MIB

SUN-PLATFORM-MIB:

- 擴展實際實體表 (Physical Entity Table) 以顯示新組件的等級
- 擴展邏輯實體表 (Logical Entity Table) 以顯示高數值平台與伺服器物件

注意 – SUN-PLATFORM-MIB 中的所有物件都包含字首 *sunPlat* 以示區別並不與其他物件混淆。

實體模式表擴展

SUN-PLATFORM-MIB 可提供未顯示於實際實體表中的等級額外屬性。其新增下述稀少之移入表擴展來延伸實際實體表：

- 設備表 (Equipment Table) 擴展
此表可擴展實際實體表以提供「設備」(Equipment) 等級管理物件的詳細資訊。此等級適用於所有 Sun Fire B1600 硬體資源。「設備」等級的子等級會顯示於其他表格擴展中。
 - 設備支架表 (Equipment Holder) 擴展
此舉可延申設備表擴展，並提供 `container(5) entPhysical` 等級之管理物件的額外相關資訊。
 - 電路封包表 (Circuit Pack Table) 擴展
此舉可延申設備表擴展，並提供 `module(9) entPhysical` 等級之管理物件的額外相關資訊。
- 實體表 (Physical Table) 擴展
此舉可延申實際實體表，並可用於補充說明實際實體表中的 `entPhysicalClass` 一欄。如果某資源擁有 `entPhysicalClass other(1)`，但其是由 `sunPlat` 所顯示的等級，即監視系統 (Watchdog) 或警報裝置 (AlarmDevice) 等級，則此表可辨識其 `sunPlatPhysicalClass`。
- 感應器表 (Sensor Table) 擴展
此舉可延申設備表擴展，並提供 `entPhysicalClass sensor(8)` 管理物件的額外相關資訊。「感應器」(Sensor) 等級之子等級可顯示於其他表格擴展，並可藉由使用 `sunPlatSensorClass` 之表格來加以辨識。
 - 二進位感應器表 (Binary Sensor Table) 擴展
此舉可延申感應器表擴展，並提供 `entPhysicalClass sensor(8)` 和 `sunPlatSensorClass binary(1)` 管理物件的額外相關資訊。

- 數字感應器表 (Numeric Sensor Table) 擴展
此舉可延申感應器表擴展，並提供 `entPhysicalClass sensor(8)` 和 `sunPlatSensorClass numeric(2)` 管理物件的額外相關資訊。
- 分離感應器表 (Discrete Sensor Table) 擴展
此舉可延申感應器表擴展，並提供 `entPhysicalClass sensor(8)` 和 `sunPlatSensorClass discrete(3)` 管理物件的額外相關資訊。
- 風扇表 (Fan Table) 擴展
此舉可延申設備表擴展，並提供 `entPhysicalClass fan(7)` 管理物件的額外相關資訊。
- 警報表 (Alarm Table) 擴展
此舉可延申設備表擴展，並提供 `entPhysicalClass other(1)` 和 `sunPlatPhysicalClass alarm(8)` 管理物件的額外相關資訊。
- 監視系統表 (Watchdog Table) 擴展
此舉可延申設備表擴展，並提供 `entPhysicalClass other(1)` 和 `sunPlatPhysicalClass watchdog(3)` 管理物件的額外相關資訊。
- 電源供應器表 (Power Supply Table) 擴展
此舉可延申設備表擴展，並提供 `entPhysicalClass powerSupply(6)` 管理物件的額外相關資訊。

表 5-3 顯示表格擴展對實際實體表的影響範例。entPhysicalIndex（表格中的欄 1）值是基於圖 5-1 的硬體資源階層範例而設。

ENTITY-MIB			SUN-PLATFORM-MIB												
entPhysicalIndex	entPhysicalClass				sunPlatPhysicalClass		sunPlatFanClass		sunPlatSensorClass					sunPlatPowerSupplyClass	
1	基座														
3	容納組件														
8	電源供應器													G	
12	電源供應器													H	
2	風扇						A								
	風扇						B								
	風扇						C								
5	感應器								D						
7	感應器								E						
11	感應器								F						
6	模組														
9	其他					警報									
10	其他					監視系統									
4	其他					其他									
entPhysicalTable			sunPlatEquipmentHolderTable		sunPlatCircuitPackTable	sunPlatPhysicalTable	sunPlatWatchdogTable	sunPlatFanTable	sunPlatAlarmTable	sunPlatSensorTable	sunPlatBinarySensorTable	sunPlatNumericSensorTable	sunPlatDiscreteSensorTable	sunPlatDiscreteSensorStatusTable	sunPlatPowerSupplyTable

表 5-3 實際實體表擴展

表 5-4 實際實體表擴展重要資訊 (表 5-3)

參考值	說明
A	風扇
B	冷卻裝置
C	散熱片
D	二進位感應器
E	數字感應器
F	分離感應器
G	電源供應器
H	電池

邏輯模式表擴展

SUN-PLATFORM-MIB 可提供未支援於邏輯實體表內之等級的額外屬性。其新增下述稀少之移入擴展來延伸邏輯實體表：

■ 邏輯等級擴展表 (Logical Class Extension Table)

此表可擴展 `entLogicalTable` 來定義邏輯實體的等級 `SunPlatLogicalClass` 與其狀態 `sunPlatLogicalStatus`。`sunPlatLogicalTable` 適用於 `entLogicalTable` 內的所有項目。「邏輯」(Logical) 等級之子等級「電腦系統」(Computer System) 可顯示於其他表格擴展中：

■ 電腦系統表 (Computer System Table) 擴展

本表格可擴展 `entLogicalTable` 以提供一般電腦系統實例屬性。`sunPlatUnitaryComputerSystemTable` 適用於包含 `sunPlatLogicalClass` `computerSystem(2)` 等級之 `entLogicalTable` 的各列項目。

載入資訊表 (`sunPlatInitialLoadInfoTable`) 中的項目組與每個「電腦系統」邏輯項目都有關連性。此項目組包含用來控制電腦系統啟動設定的參數。

事件 (Event) 和警報 (Alarm) 日誌表

SNMP 陷阱並不保證一定會傳送。因此，為了支援管理應用程式能力以確實追蹤平台警報的正確狀態，MIB 會保留每個管理物件的當前問題清單。此表可列出每個物件未解決之警報。當警報狀態解除時，表中的相關警報記錄也將自動清除。

SUN-PLATFORM-MIB 可定義用於記錄事件或警報（以事件或警報類型，或由受影響的實體來分類）的日誌。Sun Fire B1600 的操作使用這些日誌來保留每個受監視實體之未解決的警報清單，原因已於之前段落陳述。

每個可產生警報之 MIB 中的實體（即為所有實際和邏輯實體）在日誌表 (Log Table) 中都有一個項目 (sunPlatLogTable)。此表可提供當前問題清單中的管理狀態和控制。此日誌將永久使用，而且沒有大小限制。

在日誌記錄表 (Log Record Table) 中可以有 0 個項目或多個項目，以日誌表中的項目為準。上述項目在第八章中有詳細說明。

事件記錄

事件記錄會形成部分的 sunPlat 陷阱通知。模式中的變更會傳送到使用二種 SNMP 陷阱類型的管理應用程式 — 事件和警報。

事件

- 物件建立記錄 (Object Creation Record)
此記錄會顯示新增至物件模式的資源。
- 物件刪除記錄 (Object Deletion Record)
此記錄會顯示從物件模式刪除的資源。
- 狀態變更記錄 (State Change Record)
此記錄會顯示變更的資源狀態。
- INTEGER 屬性值變更記錄 (Integer Attribute Value Change Record)
此記錄會顯示由 INTEGER 類型屬性所顯示的資源特徵變更。INTEGER 可以顯示或隱藏，須視受到影響的物件而定。
- STRING 屬性值變更記錄 (String Attribute Value Change Record)
此記錄會顯示由 OCTET STRING 類型屬性顯示的資源特徵變更。
- OID 屬性值變更記錄 (OID Attribute Value Change Record)
此記錄會顯示 OBJECT IDENTIFIER 類型之物件辨識屬性中的變更。

警報

- 傳送警報記錄 (Communications Alarm Record)
此記錄會顯示資源支援之傳送服務失敗。
- 環境警報記錄 (Environmental Alarm Record)
此記錄會顯示與資源相關之環境狀態。
- 設備警報記錄 (Equipment Alarm Record)
此記錄會顯示錯誤的資源。

- **處理錯誤警報記錄 (Processing Error Alarm Record)**
此記錄會顯示資源的相關軟體錯誤或執行錯誤。
- **服務品質警報記錄 (Quality of Service Alarm Record)**
此記錄會顯示服務品質警報已產生。
- **未知警報記錄 (Indeterminate Alarm Record)**
此記錄會顯示未知類型的警報已產生。

實體模式

本章將說明 sunPlat 的實體等級階層，以及定義於 sunPlat 模式之管理實體物件等級如何在 SUN-PLATFORM-MIB 中呈現。

本章包含以下內容：

- 第 29 頁的「sunPlat 實體等級階層」
- 第 31 頁的「sunPlat 等級定義」

sunPlat 實體等級階層

圖 6-1 顯示用來呈現 Sun Fire B1600 硬體資源的 sunPlat 等級繼承階層。

實際實體母等級 (superclass) 可提供定義管理物件之間關係的屬性，還可提供標準 SNMP 屬性，與「設備」(Equipment) 等級中的屬性相對應。

sunPlat 設備等級是從實際實體母等級衍生而出，用以提供定義於相對等級中適用於錯誤監視的額外屬性。

sunPlat 設備支架 (Equipment Holder) 和 sunPlat 電路封包 (Circuit Pack) 等級是由 sunPlat 設備母等級衍生而出，可分別顯示其插入的容納組件和組件。

sunPlat 設備等級是專門用來提供由 DMTF 衍生而出的等級。

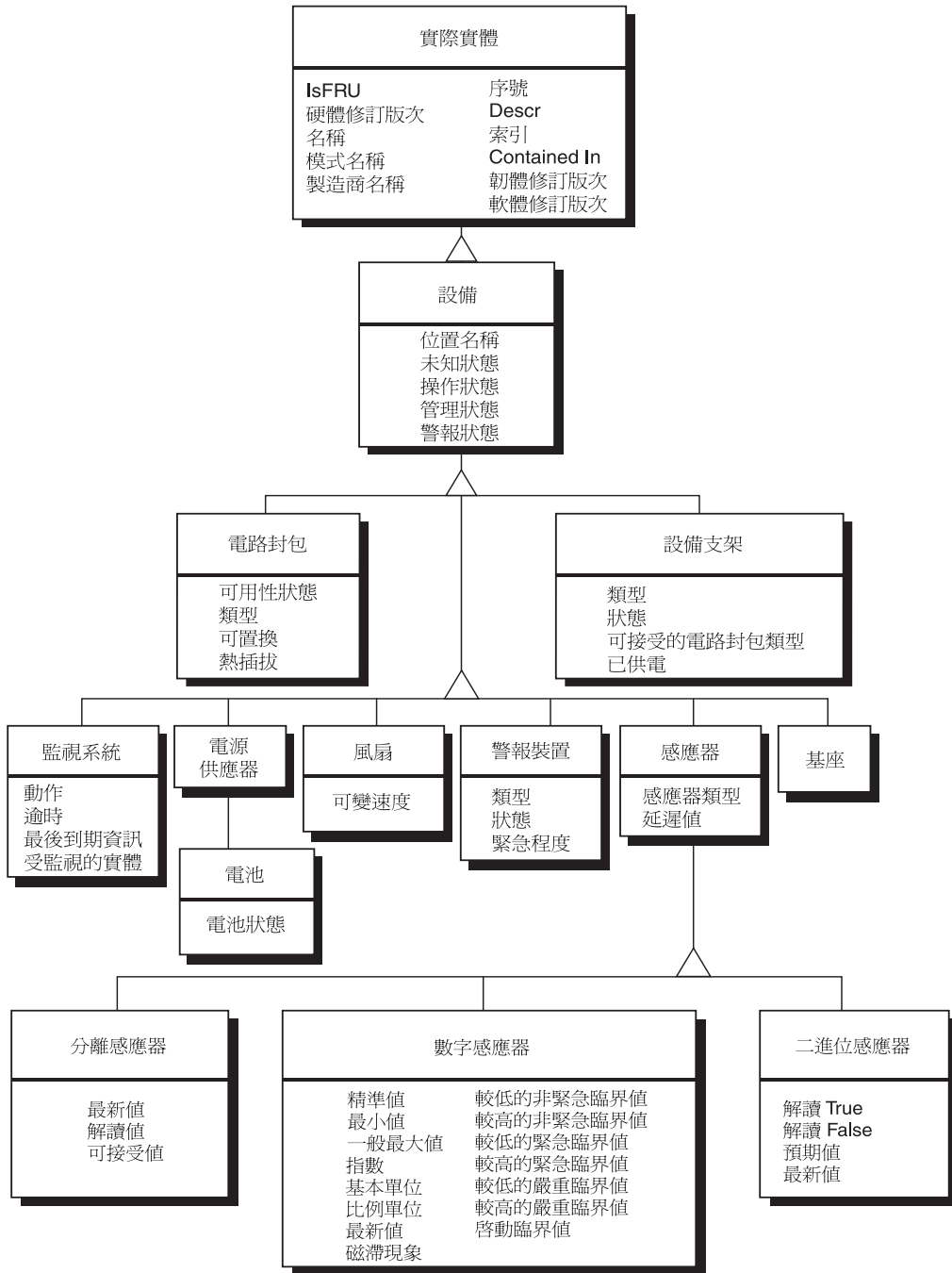


圖 6-1 sunPlat 實體資源繼承等級示意圖

sunPlat 等級定義

sunPlat 等級的屬性是用來呈現硬體資源的特徵。管理員可由管理物件的 *狀態* 來查詢資源的可用性與操作性。不同的 sunPlat 等級各自擁有不同的屬性來顯示管理物件狀態的現況。

實際實體

實際實體 (Physical Entity) 母等級可顯示所有資源的一般特徵。

注意 – 字首 *entPhysical* 已自下述屬性名稱省略以求簡潔明白。

- **Descr**
此為包含未知資源名稱的文字字串。此名稱通常是用以說明產品文件資料、產品圖例、或儲存於韌體之名稱中的資源。
- **Is FRU**
此為一種布林值 (boolean)，顯示此資源是否為可現場置換單元 (FRU)。只有 *sunPlatCircuitPack* 等級的硬體資源才會被視為 FRU。
- **硬體修訂版次**
此為包含資源製造商之硬體修改資訊的文字字串。並非所有的硬體資源皆會附上硬體修改資訊。
- **名稱**
此為包含作業系統和相關公用程式已知資源之邏輯名稱的文字字串。此名稱可以是裝置節點，或者系統公用程式使用的已定義名稱（視情況而定）。並非所有的資源都有裝置名稱。
- **模式名稱**
此為包含製造商顧客可見零件編號或零件定義之文字字串。並非所有硬體資源都有其相關之零件編號或定義。
- **序號**
此為包含資源製造商之序號的文字字串。並非所有硬體資源都有其相關的序號。
- **製造商名稱**
此為包含資源製造商名稱的文字字串。並非所有硬體資源都有其相關的製造商名稱。

實際實體母等級同時也包含可用於說明硬體資源階層的屬性：

■ 等級

以下類型包含特定實際資源的一般硬體類型指示。此等級支援的各項值皆由 ENTITY-MIB 所定義。此屬性可當作管理物件之相關表格擴展的指示。ENTITY-MIB 等級和 sunPlat 等級之間的對映顯示在表 6-1 中：

表 6-1 實際實體母等級之「等級」屬性對映

entPhysicalClass	sunPlat 等級
chassis(3)	sunPlat 基座 (Chassis)
backplane(4)	未提供
container(5)	sunPlat 設備支架 (Equipment Holder)
powerSupply(6)	sunPlat 電源供應器 (Power Supply)
fan(7)	sunPlat 風扇 (Fan)
sensor(8)	sunPlat 感應器 (Sensor) 與其子等級
module(9)	sunPlat 電路封包 (Circuit Pack)
port(10)	未提供
stack(11)	未提供
other(1)	sunPlat 設備 (Equipment) 與其子等級
unknown(2)	未提供

■ 索引

此整數值可單獨辨識實際實體表（其可辨識管理物件）中的項目。此值並非預設值，而且會依每個代理程式的呼叫而有所改變。

■ Contained In

此整數值可顯示包含在管理物件內之管理物件的索引。所以，此屬性可呈現管理物件之間的關係。

注意 – 位於實際包含階層（通常為基座）root 之物件並非實際包含於此表的另一個實體中。若要表示上述情況，*entPhysicalContainedIn* 會設定為 0。

■ 韌體修訂版次

此為包含資源製造商之韌體修改資訊的文字字串。並非所有硬體資源都會有其相關的韌體修改資訊。

■ 軟體修訂版次

此為包含資源製造商之軟體修改資訊的文字字串。並非所有硬體資源都會有其相關的軟體修改資訊。

sunPlat 設備等級

sunPlat 設備 (Equipment) 等級是用來呈現所有硬體資源的一般特徵。此等級包含代表配置和一般正常狀態資訊的屬性，並設有子等級來提供更多特定資源類型的詳細配置資訊和監視資料。

entPhysicalClass 是根據呈現的子等級而定。

sunPlat 設備等級包含以下屬性：

注意 – 字首 *sunPlatEquipment* 已自下述屬性的名稱省略以求簡潔明白。

■ 管理狀態

此可讀寫屬性使用以下代表目前資源管理狀態屬性值的其中之一：

- locked(1)
- unlocked(2)
- shuttingDown(3)

■ 操作狀態

此唯讀屬性可顯示資源是否已實際安裝並能否提供服務。此屬性是用於管理物件狀態，還能使用表 6-2 中的值。

表 6-2 操作狀態屬性值

屬性值	說明
disabled(1)	無法使用此資源，且無法提供服務給使用者。
enabled(2)	可使用部分或全部資源，並已準備好可供使用。

■ 警報狀態

此唯讀屬性使用以下各值表示目前的資源警報狀態，並顯示管理物件上最嚴重的未解決警報。此屬性可使用以下屬性值：

- critical(1)、
- major(2)、
- minor(3)、
- indeterminate(4)、
- warning(5)、
- pending(6)、
- cleared(7)

- **未知狀態**
此唯讀屬性會顯示其他狀態屬性是否會影響資源的 `true` 狀態。此屬性可使用布林值來顯示管理物件是否可正確回報資源的錯誤資訊。如果資源無法正確回報其狀態，就會將此屬性設定為 `true`。
- **位置名稱**
此唯讀屬性包含一個資源定址器 (`locator`)。若資源是直接包含在基座中，此屬性就會聯繫插槽和產品文件資料中的圖例，並提供此資源在基座中的實際位置指標。其他硬體資源通常會有與此資源內的管理物件名稱相對應之位置。

sunPlat 電路封包等級

sunPlat 電路封包 (Circuit Pack) 等級是用於表示可置換資源或 FRU 的一般特徵。可置換資源被定義為硬體模組，其目的為封包內部硬體組件為可辨識之體積大小。一般而言，FRU 已預設體積大小與實際外觀。FRU 為可插拔之移動單元（可連接插頭），可以更永久地固定在插座、或插入機盒、機櫃或機架。

此等級使用 `entPhysicalClass module(9)`。

sunPlat 電路封包等級包含以下屬性：

注意 – 字首 `sunPlatCircuitPack` 已自下述屬性名稱省略以求簡潔明白。

- **類型**
此唯讀屬性為用於存取資源的相容性與其容納組件之文字字串。此屬性可辨識資源的功能與體積特徵。
- **可用性狀態**
此唯讀屬性可加以說明管理物件的**操作狀態**。此物件使用位元 (BIT) 語法，並且可能為 0 或其他數值組，如表 6-3 所示。並非所有數值皆適用於每個管理物件等級。此屬性值是用於管理物件的**狀態**。

表 6-3 可用性狀態屬性值

屬性值	位元編號	Hex.	說明
<code>inTest(0)</code>	0	80	此資源已進入測試程序。
<code>failed(1)</code>	1	40	此資源出現內部錯誤導致無法操作。 操作狀態 為 <code>disabled(1)</code> 。
<code>powerOff(2)</code>	2	20	此資源需要電源才可操作，且目前沒有供應任何電源。
<code>offLine(3)</code>	3	10	此資源需要執行例行性操作以維持其連線狀態，並確保其可用性。 操作狀態 為 <code>disabled(1)</code> 。

表 6-3 可用性狀態屬性值 (接上頁)

屬性值	位元編號	Hex.	說明
offDuty(4)	4	08	此資源已因內部控制操作而無法進行作業。
dependency(5)	5	04	此資源無法操作，因為與其相關之其他資源也無法操作。操作狀態為 disabled(1)。
degraded(6)	6	02	資源的可用服務就某方面來看已減少，例如速度或操作可行性，但是此資源仍然可以使用。操作狀態為 enabled(2)。
notInstalled(7)	7	01	沒有出現管理物件代表的資源，或者此資源不完整。操作狀態為 disabled(1)。

- **可置換**
此唯讀屬性使用布林值，顯示特定資源是否為可置換單元。
- **熱插拔**
此唯讀屬性使用布林值，顯示特定可置換資源是否支援熱插拔。

sunPlat 設備支架

sunPlat 設備支架 (Equipment Holder) 等級是用於表示硬體資源的支架特徵。

此等級使用 `entPhysicalClass container(5)`。

sunPlat 設備支架等級包含以下屬性：

注意 – 字首 `sunPlatEquipmentHolder` 已自下述屬性名稱省略以求簡潔明白。

- **類型**
此唯讀屬性表示資源的支架類型，如表 6-4 所示：

表 6-4 設備支架種類屬性值

屬性值	說明
bay(1)	插座通常為垂直的單元，位於內含機架或機盒之機櫃中，用以撐托電信設備。SunPlat 將其於基座內的作用解讀為實際容納組件，並需要纜線以連接訊號。
shelf(2)	一種平行支援或用於機櫃內撐托電信設備之子機櫃

表 6-4 設備支架種類屬性值 (接上頁)

drawer(3)	一種於機櫃內用以撐托電信設備之平行裝置
slot(4)	一種實際容納組件和完整接頭，用來連接移動裝置之訊號。
rack(5)	機櫃是一種支托建設，用來支撐其包含裝置之電信設備、支架、以及纜線管理系統。

■ **可接受類型**

此唯讀屬性為文字字串清單，顯示支架撐托之移動資源（電路封裝）的類型。此類型已通過其對移動資源之**類型**屬性的相容性測試。

■ **狀態**

此唯讀屬性可顯示任何支架內的可置換硬體資源（電路封包）的狀態，如表 6-5 所示。

表 6-5 設備支架狀態屬性值

屬性值	說明
holderEmpty(1)	支架內沒有可移動資源。
inTheAcceptableList(2)	支架包含了可移動資源，其為 可接受的電路封包類型 清單中的類型之一。
notInTheAcceptableList(3)	支架包含了網路元件可辨識的可移動資源，但並非為 可接受的電路封包類型 清單中的類型之一。
unknownType(4)	支架包含了無法辨識的可移動資源

■ **已供電**

此可讀寫屬性可表示資源的電源狀態。可使用的屬性值包含：

- other(1)
- unknown(2)
- powerOff(3)
- powerOn(4)

sunPlat 電源供應器

sunPlat 電源供應器 (Power Supply) 等級代表電源供應器，而且不會擴展 sunPlat 設備等級的特徵。電源供應器通常包含了可代表監視屬性的感應器，例如電壓、電流和溫度感應器，同時也包含其他硬體資源，例如風扇。其是藉用管理物件之間的關係來呈現。

如果電源供應器為可移動資源，就可使用 sunPlat 電路封包等級的管理物件來呈現。

此等級使用 entPhysicalClass powerSupply(6)。

sunPlat 電源供應器等級包含以下屬性：

注意 – 字首 `sunPlatPowerSupply` 已自下述屬性名稱省略以求簡潔明白。

■ **等級**

此唯讀屬性代表電源供應器的等級，並可使用以下屬性值：

- `other(1)`
- `powerSupply(2)`
- `battery(3)`

sunPlat 電池

sunPlat 電池 (Battery) 等級是用來代表使用電池供電的電源供應器。

此等級可使用 `entPhysicalClass powerSupply(6)` 和 `SunPlatPowerSupplyClass battery(3)`。

sunPlat 電池等級包含以下屬性：

注意 – 字首 `sunPlatBattery` 已自下述屬性名稱省略以求簡潔明白。

■ **狀態**

此唯讀屬性代表電池的狀態，並可使用以下屬性值：

- `other(1)`
- `unknown(2)`
- `fullyCharged(3)`
- `low(4)`
- `critical(5)`
- `charging(6)`
- `chargingAndHigh(7)`
- `chargingAndLow(8)`
- `chargingAndCritical(9)`
- `undefined(10)`
- `partiallyCharged(11)`

sunPlat 監視系統

sunPlat 監視系統 (Watchdog) 等級是用來表示計時器硬體資源的特徵，其允許硬體來監視作業系統或應用程式的狀態。

此等級使用 `entPhysicalClass other (1)` 和 `sunPlatPhysicalClass watchdog (3)`。

sunPlat 監視系統等級包含以下屬性：

注意 – 字首 `sunPlatWatchdog` 已自下述屬性名稱省略以求簡潔明白。

- **逾時**
此唯讀屬性為一整數值，並可在監視系統即將逾時且未重設的時候，用來表示以毫秒計算的間隔時間。
- **動作**
若監視系統在**逾時**指定的時間內沒有重設，此唯讀屬性便可用來表示監視系統執行的動作。可能使用的屬性值列於表 6-6。

表 6-6 監視系統動作屬性值

動作	說明
<code>statusOnly(1)</code>	監視系統可由軟體讀取，但無法執行任何動作
<code>systemInterrupt(2)</code>	監視系統會在受監視的系統中產生硬體中斷
<code>systemReset(3)</code>	監視系統重設受監視的系統
<code>systemPowerOff(4)</code>	監視系統中斷受監視的系統電源
<code>systemPowerCycle(5)</code>	監視系統中斷、然後開啓受監視的系統電源

- **最後到期資訊**
此唯讀屬性可表示監視系統最後到期的日期和時間。
- **受監視的實體**
此唯讀屬性可表示監視系統可監視的實體。可使用的屬性值包含：
 - `unknown(1)`
 - `other(2)`
 - `operatingSystem(3)`
 - `operatingSystemBootProcess(4)`
 - `operatingSystemShutdownProcess(5)`
 - `firmwareBootProcess(6)`
 - `biosBootProcess(7)`
 - `application(8)`

- `serviceProcessors(9)`

sunPlat 警報

sunPlat 警報等級是用來表示可發出相關問題警示的硬體資源的特徵，例如蜂鳴器、LED、繼電器、振動器、以及軟體警報裝置。

此等級使用 `entPhysicalClass other(1)` 和 `sunPlatPhysicalClass alarm(2)`。

sunPlat 警報等級包含以下屬性：

注意 – 字首 *sunPlatAlarm* 已自下述屬性名稱省略以求簡潔明白。

■ 類型

此唯讀屬性可用來表示發生警報時的通知方式。可能使用的屬性值列於表 6-7。

表 6-7 警報類型屬性值

屬性值	說明
<code>other(1)</code>	警報裝置類型不屬於以下列出的類型之一
<code>audible(2)</code>	裝置的警報裝置聲音警示有變更
<code>visible(3)</code>	裝置警報造成可見變更
<code>motion(4)</code>	警報造成裝置運轉
<code>switch(5)</code>	警報導致電波訊號改變

■ 狀態

此唯讀屬性可表示警報的狀態。可能使用的屬性值列於表 6-8。

表 6-8 警報狀態屬性值

屬性值	說明
<code>unknown(1)</code>	警報狀態尚未定義或無法觀測
<code>off(2)</code>	尚未啟動警報
<code>steady(3)</code>	已啟動警報
<code>alternating(4)</code>	警報處於未開啓與開啓狀態之間

■ 緊急程度

此唯讀屬性代表警報閃動、振動、以及 / 或發出聲音的相對頻率。可使用的屬性值包含：

- other(1)
- unknown(2)
- notSupported(3)
- informational(4)
- nonCritical(5)
- critical(6)
- unrecoverable(7)

sunPlat 風扇

sunPlat 風扇等級是用來表示已啓動冷卻裝置的特徵。風扇通常會包含一個可顯示運轉速度的感應器，其利用 sunPlat 風扇管理物件與 sunPlat 感應器等級轉速器管理物件之間的實體包含關係來呈現。

此等級使用 entPhysicalClass fan(7)。

sunPlat 風扇等級包含以下屬性：

注意 – 字首 *sunPlatFan* 已自下述屬性名稱省略已求簡潔明白。

- **等級**
此唯讀屬性代表冷卻裝置的等級，並可使用以下屬性值：
 - other(1)
 - fan(2)
 - refrigeration(3)
 - heatPipe(4)

sunPlat 感應器

sunPlat 感應器母等級是用來表示可測量其他硬體資源屬性之硬體資源的一般特徵。

此等級使用 entPhysicalClass sensor(8)。

sunPlat 感應器等級包含以下屬性：

注意 – 字首 *sunPlatSensor* 已自下述屬性名稱省略以求簡潔明白。

- **等級**
此唯讀屬性代表感應器的等級，並可使用以下屬性值：

- `binary(1)`
- `numeric(2)`
- `discrete(3)`
- **類型**
此唯讀屬性可辨識感應器測量的屬性。部分類型的可用數值已列於表 6-9。

表 6-9 感應器類型屬性值

類型	說明
<code>temperature(3)</code>	測量環境溫度的感應器
<code>voltage(4)</code>	測量電壓的感應器
<code>current(5)</code>	測量電流的感應器
<code>tachometer(6)</code>	測量裝置速度 / 運轉的感應器
<code>counter(7)</code>	測量已定義事件的一般感應器

- **延遲值**
此唯讀屬性可表示以下事項：
 - 在感應器輪詢的地方，此整數值代表更新的間隔時間（以毫秒計算）。
 - 在感應器事件驅動的地方，此值代表進行事件的最大期待延遲值。

sunPlat 二進位感應器

`sunPlat Binary Sensor` 等級可用來表示回報二進位輸出的感應器特徵，其擴展 `sunPlatSensor` 表以提供特定二進位感應器的屬性。

此等級使用 `entPhysicalClass sensor(8)` 和 `sunPlatSensorClass binary(1)`。

`sunPlat` 二進位感應器等級包含以下屬性：

注意 – 字首 `sunPlatBinarySensor` 已自下述屬性名稱省略以求簡潔明白。

- **最新值**
此唯讀屬性使用布林值來表示感應器的最新數值。
- **預期值**
此唯讀屬性使用布林值來表示感應器的預期值。
- **解讀 True**
此唯讀屬性為表示感應器 `true` 值解讀的文字字串。
- **解讀 False**
此唯讀屬性為表示感應器 `false` 解讀的文字字串。

sunPlat 數字感應器

sunPlat 數字感應器等級是用來表示可回報數字讀取值的感應器特徵。此數字感應器值可藉由下述的測量單位來加以說明：

測量單位 = 基準單位 * 10ⁿ

上述說明可做為毫安培或毫伏特的測量單位。若已定義比例單位，就會再額外定義測量單位：

測量單位 = 基本單位 * 每比例單位的 10ⁿ

上述說明可做為 rmp 或 km/hr 的測量單位。

此等級使用 `entPhysicalClass sensor(8)` 和 `sunPlatSensorClass numeric(2)`。

sunPlat 數字感應器等級包含以下屬性：

注意 – 字首 `sunPlatNumericSensor` 已自下述屬性名稱省略以求簡潔明白。

■ 基本單位

此唯讀屬性是用來表示定義下述說明之前的測量單位。此類型的屬性值範例為：

- `degC(3)`
- `volts(6)`
- `amps(7)`

■ 指數

此唯讀屬性為一整數值，其使用 10 的次方來計算基本單位。例如，若將 `sunPlatNumericSensorBaseUnits` 設為 `volts` 而 `sunPlatNumericSensorExponent` 設為 `-6`，則出現的數值單位便會是 `microVolts`。

■ 比例單位

此唯讀屬性可用來表示感應器測量的值是否為一絕對值（當數值為 `none`）或一個比例值。在後者中，指定於 `sunPlatNumericSensorBaseUnits` 的單位會顯示為「個別時間單位」（per unit of time）。舉例來說，如果將 `sunPlatNumericSensorBaseUnits` 設定為 `degC`，而 `sunPlatNumericSensorRateUnits` 設定為 `perSecond`，顯示出來的單位值就會是 `degC/second`。

此類型數值的範例如下：

- `perMicrosecond(2)`
- `perMillisecond(3)`
- `perSecond(4)`
- `perMinute(5)`
- `perHour(6)`
- `none(1)`

- **最新值**
此唯讀屬性使用整數來表示感應器的最新數值。
- **一般最小值**
此唯讀屬性使用整數來表示低於預設最低臨界值的感應器讀取值。此值會依據以下定義的測量單位來呈現。此屬性並非適用於所有的感應器。
- **一般最大值**
此唯讀屬性使用整數來表示高於預設最大臨界值的感應器讀取值。此值會依據以下定義的測量單位來呈現。此屬性並非適用於所有的感應器。
- **精準值**
此唯讀屬性為整數，用來表示感應器所測量屬性的錯誤程度。此值以百分比或以小數點後二位來呈現，但也可能視感應器讀取值是否以直線讀取其動態範圍而有所改變。
- **較低的非緊急臨界值**
此唯讀屬性為整數，可表示 nonCritical 情況發生時的較低臨界值。
- **較高的非緊急臨界值**
此唯讀屬性為整數，可表示 nonCritical 情況發生時的較高臨界值。
- **較低的緊急臨界值**
此唯讀屬性為整數，可表示 critical 情況發生時的較低臨界值。
- **較高的緊急臨界值**
此唯讀屬性為整數，可表示 critical 情況發生時的較高臨界值。
- **較低的嚴重臨界值**
此唯讀屬性為整數，可表示 fatal 情況發生時的較低臨界值。
- **較高的嚴重臨界值**
此唯讀屬性為整數，可表示 fatal 情況發生時的較高臨界值。
- **磁滯現象**
此唯讀屬性可說明臨界值周遭的磁滯現象。
- **啟動臨界值**
此唯讀屬性可重設感應器回其預設值（當寫入時）。

sunPlat 分離感應器

當 sunPlat 數字感應器或 sunPlat 二進位感應器等級無法呈現感應器時，就可以使用 sunPlat 分離感應器 (Discrete Sensor) 等級。

此等級使用 `entPhysicalClass sensor (8)` 和 `sunPlatSensorClass discrete(3)`。

此等級由二個表格組成。sunPlatDiscreteSensor 表的 `sunPlatDiscreteSensorCurrent` 屬性可表示感應器的最新狀態，其以 sunPlatDiscreteSensorStates 表中的索引來呈現。

sunPlat 分離感應器等級包含以下屬性：

注意 – 字首 *sunPlatDiscreteSensorState* 已自下述屬性名稱省略以求簡潔明白。

- **索引**
此唯讀屬性使用 `sunPlatDiscreteSensorStates` 表中代表各列索引值的數值，用來辨識此感應器狀態。
- **解讀值**
此唯讀屬性為文字字串，用來說明 `sunPlatDiscreteSensorStatesTable` 中相對列的狀態。
- **可接受值**
此唯讀屬性使用布林值來表示表格列中的狀態是否為可接受值。

sunPlat 基座

`sunPlat` 基座 (Chassis) 等級是用來呈現主要的內含裝置，而且不會擴展 `sunPlat` 設備等級的特徵。此基座包含所有出現過的硬體資源，而且它並不包含在其他資源中。

此等級使用 `entPhysicalClass chassis(3)`。

邏輯模式

本章將說明 sunPlat 邏輯等級階層，以及定義於 sunPlat 模式之管理物件等級如何在 SUN-PLATFORM-MIB 中呈現。

本章包含以下內容：

- 第 45 頁的「sunPlat 邏輯等級階層」
- 第 46 頁的「SunPlat 邏輯等級定義」

sunPlat 邏輯等級階層

圖 7-1 顯示 sunPlat 邏輯等級的繼承階層。

邏輯實體 (Logical Entity) 等級可提供所有邏輯物件的一般資源。

單一電腦系統 (Unitary Computer System) 等級可新增與回報的電腦系統電源狀態相關之屬性（例如，Sun Fire B1600 基座中的 Sun Fire B100s Blade），同時也可用來影響強迫重設。

管理領域 (Administrative Domain) 等級無法新增額外的屬性，但可用來表示與展示系統有管理接觸的邏輯物件。在 Sun Fire B1600 平台中，此等級是用來表示系統控制器。

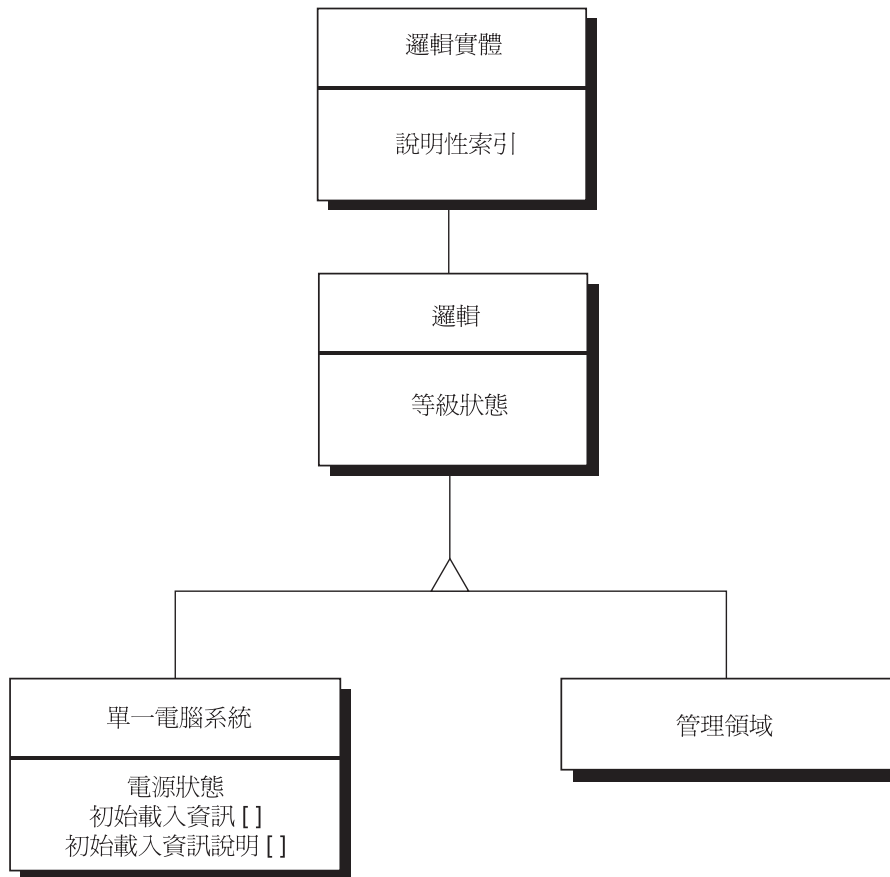


圖 7-1 sunPlat 邏輯資源繼承等級示意圖

SunPlat 邏輯等級定義

邏輯 sunPlat 等級的屬性是用來呈現邏輯資源的特徵。此資源表示高數值的物件，例如多領域系統中的領域。管理員可由管理物件的狀態來查詢資源的可用性與操作性。不同的 sunPlat 等級各自擁有不同的屬性來表現管理物件狀態的現況。

邏輯實體

邏輯實體 (Logical Entity) 等級表示可提供辨識資訊的邏輯實體。重要物件包含：

注意 – 字首 *entLogical* 已自下述屬性名稱省略以求簡潔明白。

■ **說明**

此物件可辨識管理物件的類型。

■ **T 位址**

在可以直接管理實體之處提供 IP 位址和 UDP 連接埠編號。在 Sun Fire B1600 系統的 Sun Fire B100s Blade 中，此值可在 Blade 的標準 Solaris SNMP 代理程式接觸之處提供 Blade 的 IP 位址和連接埠 161。

邏輯

邏輯 (Logical) 等級代表邏輯實體呈現的資源狀態類型。此等級包含以下物件：

注意 – 字首 *sunPlatLogical* 已自下述屬性名稱省略以求簡潔明白。

■ **等級**

此屬性代表邏輯等級的類型，並使用以下屬性值：

- other(1)
- computerSystem(2)
- adminDomain(3)

■ **狀態**

此屬性可表示邏輯等級的狀態，並使用以下屬性值：

- ok(1)
- error(2)
- degraded(3)
- unknown(4)
- predFail(5)
- starting(6)
- stopping(7)
- service(8)
- stressed(9)
- nonRecover(10)
- noContact(11)
- lostComm(12)
- stopped(13)

sunPlat 單一電腦系統

單一電腦系統 (Unitary Computer System) 等級的特定屬性會呈現在 `sunPlatUnitaryComputerSystemTable`，並使用 `computerSystem(2)` `sunPlat` 邏輯等級。

此等級包含以下物件：

注意 – 字首 `sunPlatUnitaryComputerSystem` 已自下述屬性名稱省略以求簡潔明白。

■ 電源狀態

此屬性可表示讀取時的最新電源狀態，還可啟動電源狀態的遠端控制，例如啟動或關閉一個 `Blade`，或者使用循環（開啓關閉）來影響強制重設。

此屬性可使用以下屬性值：

- `unknown(1)`
- `fullPower(2)`
- `psLowPower(3)`
- `psStandby(4)`
- `psOther(5)`
- `powerCycle(6)`
- `powerOff(7)`
- `psWarning(8)`
- `hibernate(9)`
- `softOff(10)`
- `reset(11)`

■ 套用設定

寫入此屬性會啟動預設值，或者套用自定的啟動參數組。

與 `sunPlatUnitaryComputerSystemTable` 相關的每個項目皆為 `sunPlatInitialLoadInfoTable` 中的項目組，其中定義了最新的啟動參數設定和可藉由寫入 `sunPlatUnitaryComputerSystemApplySettings` 物件來套用的替代組。

sunPlat 管理領域

管理領域 (Administrative Domain) 等級無法新增邏輯實體等級的屬性，因此也就沒有任何與 MIB 相關之物件。此等級使用 `adminDomain(3)` `sunPlat` 邏輯等級。

sunPlat 通知

本章將說明定義於 SUN-PLATFORM-MIB 中的 SunPlat 通知等級和屬性。

sunPlat 通知等級是一種非同步訊息，由代理程式傳送至已登錄的網路管理員。此等級可更有效率地傳達事件資訊，優於透過輪詢管理物件來傳達。

本章包含以下內容：

- 第 49 頁的「sunPlat 通知等級階層」
- 第 51 頁的「sunPlat 等級定義」

sunPlat 通知等級階層

圖 8-1 顯示 sunPlat 通知等級的繼承階層。

通知 (Notification) 等級組使用抽象和實際的等級階層，並利用此二種等級的一般屬性來呈現。

sunPlat 事件記錄等級

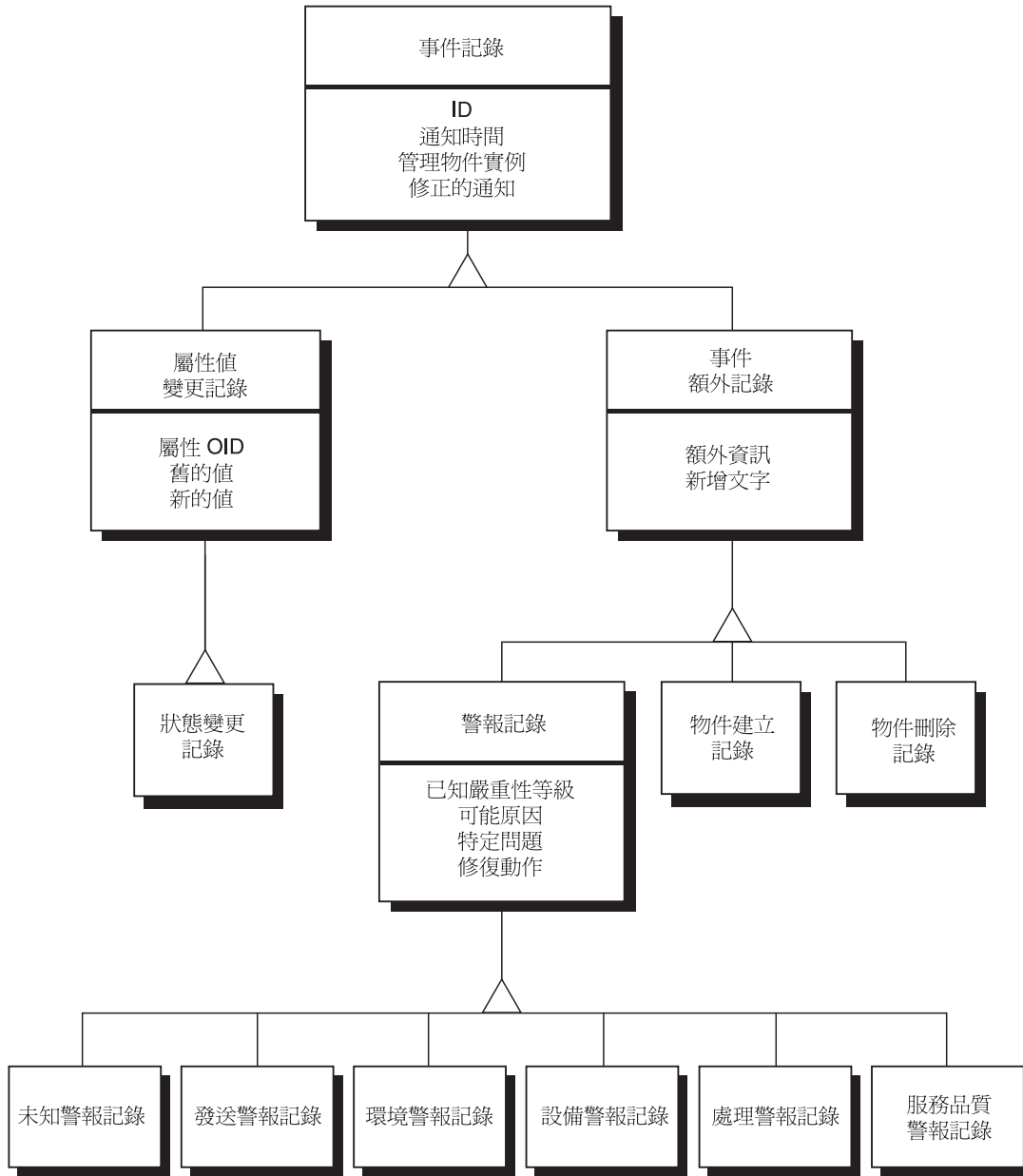


圖 8-1 事件記錄繼承等級示意圖

sunPlat 等級定義

sunPlat 事件記錄

sunPlat 事件記錄 (Event Record) 母等級可呈現所有通知的一般屬性，而其子等級則會提供與其記錄的特定事件相關的額外資訊。

sunPlat 事件記錄等級包含以下屬性：

注意 – 字首 *sunPlatLogRecord* 已自下述屬性名稱省略以求簡潔明白。

- **ID**
此為獨特的整數值，可用以辨識通知，並指示由代理程式所發出的通知順序。請注意：代理程式無法保證通知順序即為特定事件的發生順序。
- **通知時間**
此唯讀屬性為時間戳記，可辨識通知產生的最新時間。
- **管理物件實例**
此唯讀屬性為 OID，可於代表此資源與其相關事件的 MIB 中提供各項目的直接參考值。
- **相關通知**
此唯讀屬性為一個以逗點分隔之 ID 值清單，其可辨識與此事件相關的其他事件。

sunPlat 事件額外記錄

sunPlat 事件額外記錄 (Event Additional Record) 母等級可表示當以下事件發生時所產生的通知之一般屬性：

- 建立物件
- 刪除物件
- 警報

其子等級會提供適用於其記錄的特定事件之額外資訊。

sunPlat 事件額外等級包含以下屬性：

注意 – 字首 *sunPlatLogRecord* 已自下述屬性名稱省略以求簡潔明白。

- **額外資訊**
此唯讀屬性為某物件之選用 OID，其可提供與此通知相關的額外資訊。

- **額外補充文字**

此為唯讀屬性之選用文字字串，提供可辨識受影響組件通知的相關資訊（由標示和 `entPhysical` 名稱得知）

sunPlat 物件建立記錄

sunPlat 物件建立記錄 (Object Creation Record) 等級可顯示已於下述相關資源新增資源至階層的事件，例如熱插拔事件。*額外資訊* 屬性包含可顯示新增資源之**實際實體表**項目的 OID。

邏輯物件建立於 0.0 管理物件實例之下。

sunPlat 物件刪除記錄

sunPlat 物件刪除記錄 (Object Deletion Record) 是用來顯示已於相關資源底下從階層移除的資源。*額外資訊* 屬性包含可顯示移除資源之**實際實體表**項目的 OID。

注意 – 此 OID 已不能再使用，但仍然可以為接收管理員所用。

sunPlat 警報記錄

sunPlat 警報記錄 (Alarm Record) 母等級可顯示所有警報通知的一般額外屬性。

其子等級則會辨識產生警報的等級。

sunPlat 警報記錄等級包含以下屬性：

注意 – 字首 *sunPlatAlarmRecord* 已自下述屬性名稱省略以求簡潔明白。

- **已知嚴重性等級**

此唯讀屬性可定義六種嚴重性等級，並顯示資源的服務將受到何種影響。使用的屬性值列於表 8-1。

表 8-1 sunPlat 警報記錄已知嚴重性值

已知嚴重性等級	說明
<code>indeterminate(1)</code>	此警報嚴重性等級尚無法確認。
<code>critical(2)</code>	影響資源運作的情況已發生，需要立刻修正此錯誤。
<code>major(3)</code>	影響資源運作的情況已發生，需要緊急修正此錯誤。

表 8-1 sunPlat 警報記錄已知嚴重性值 (接上頁)

minor(4)	此情況不會影響資源運作，並應執行修復動作以避免更嚴重的情況發生。
warning(5)	已偵測可能影響資源運作的情況，且應執行修復動作以避免更嚴重的情況發生。
cleared(6)	此等級將清除資源內的所有警報，其位於相同的警報等級，並擁有相同的 <i>可能原因</i> 以及 <i>特定問題</i> （若有提供的話）。

■ *可能原因*

此唯讀屬性非必須屬性，其可提供已產生警報之狀態類型的深入說明。此類型數值的範例如下：

- coolingSystemFailure(134)
- IODeviceError(75)
- powerProblem(58)
- softwareProgramError(283)

■ *特定問題*

此唯讀屬性非必須文字字串，其可辨識警報*可能原因*的精確說明。

■ *修復動作*

此唯讀屬性為可列出建議修復動作的字串。

sunPlat 未知警報記錄

sunPlat 未知警報記錄 (Indeterminate Alarm Record) 等級不會擴展由 sunPlat 警報記錄等級提供的資訊。此等級可用來記錄不屬於下述等級的警報：

sunPlat 傳送警報記錄

sunPlat 傳送警報記錄 (Communications Alarm Record) 等級不會擴展由 sunPlat 警報記錄等級提供的資訊。此等級可用來記錄相關資源已偵測之傳送錯誤。

sunPlat 環境警報記錄

sunPlat 環境警報記錄 (Environmental Alarm Record) 等級不會擴展由 sunPlat 警報記錄等級提供的資訊。此等級可用來記錄相關資源已偵測之環境問題。

sunPlat 設備警報記錄

sunPlat 設備警報記錄 (Equipment Alarm Record) 等級不會擴展由 sunPlat 警報記錄等級提供的資訊。此等級可用來記錄相關資源已偵測之錯誤。

sunPlat 處理警報記錄

sunPlat 處理警報記錄 (Processing Alarm Record) 等級不會擴展由 sunPlat 警報記錄等級提供的資訊。此等級可用來記錄相關資源已偵測之軟體或操作失敗。

sunPlat 服務品質警報記錄

sunPlat 服務品質警報記錄 (Quality of Service Alarm Record) 等級不會擴展由 sunPlat 警報記錄等級提供的資訊。此等級可用來記錄相關資源已偵測之服務品質的變更。

sunPlat 屬性值變更記錄

sunPlat 屬性值變更記錄 (Attribute Value Change Record) 母等級可顯示與相關資源中屬性變更通知有關的一般額外屬性。

其子等級則會顯示可能的屬性類型。

sunPlat 屬性值變更記錄等級包含以下屬性：

注意 – 字首 *sunPlatLogRecordChange* 已自下述屬性名稱省略以求簡潔明白。

■ *OID*

此唯讀屬性為 *OID*，可於代表已變更之管理物件屬性值的實際實體表或邏輯實體表中的物件提供直接參考值。

根據受影響屬性的語法，會使用以下物件配對組來表示新的和舊的整數值：

■ *新 INTEGER 值*

此唯讀屬性可辨識管理物件已變更屬性的新 *INTEGER* 值。不論此類型是顯示或隱藏，皆會對應至變更的屬性。

■ *舊 INTEGER 值*

此唯讀屬性可辨識管理物件已變更屬性的舊 *INTEGER* 值。不論此類型是顯示或隱藏，皆會對應至變更的屬性。

■ *新 STRING*

此唯讀屬性可辨識屬性變更通知中的新 *OCTET STRING* 值。

■ *舊 STRING*

此唯讀屬性可辨識屬性變更通知中的舊 *OCTET STRING* 值。

- *新OID*
此唯讀屬性可辨識屬性變更通知中的新 OBJECT IDENTIFIER 值。
- *舊OID*
此唯讀屬性可辨識屬性變更通知中的舊 OBJECT IDENTIFIER 值。

sunPlat 狀態變更記錄

sunPlat 狀態變更記錄 (State Change Record) 等級不會擴展由 sunPlat 屬性值變更記錄 (Attribute Value Change Record) 等級提供的資訊。此等級是用來顯示管理物件屬性中的變更，其反應出資源的*狀態*。

第二部分 安裝和配置

管理軟體組件

本章將說明 Sun Fire B1600 管理軟體的組件，並列出安裝 SNMP 軟體的要求。

本章包含以下內容：

- 第 59 頁的「系統管理選項」
- 第 61 頁的「系統需求」
- 第 63 頁的「安裝套裝軟體」
- 第 64 頁的「套裝軟體內容」
- 第 65 頁的「對系統檔案的影響」

系統管理選項

以下系統管理選項是專為 Sun Fire B1600 所提供：

- 使用 SNMP 監視並控制系統
- SNMPv3 功能支援安全性管理
- 使用 Sun Management Center 3.0 做系統監視¹

1. 如需安裝和配置的詳細說明，請參閱《用於 Sun Fire B1600 的 Sun Management Center 3.0 補充資料》（文件編號：817-2541-10）

設備

您可依據平台類型選擇適合的代理程式：

- 領域代理程式，執行於 Sun Fire B100s Blade（領域硬體監視）

此軟體會安裝於本端已受到監視的伺服器上，而且只有該伺服器會受到監視。在 Sun Fire B1600 中，每個 Blade 會個別受到監視，而且每個代理程式實例一次只可檢視一個 Blade。

- 平台代理程式，透過系統控制器代理（平台硬體監視）

此軟體會安裝於遠端伺服器中，其可透過系統控制器存取平台設備，並讓您監視由系統控制器管理的所有硬體。在 Sun Fire B1600 中可以監視一個完整的 Blade 機架，包括所有類型的 Blade、電源供應器和系統控制器。

在圖 9-1 中，平台硬體監視是用於 Sun Fire B1600 機架 A 和 B，而領域硬體監視是用於 Sun Fire B1600 機架 C。

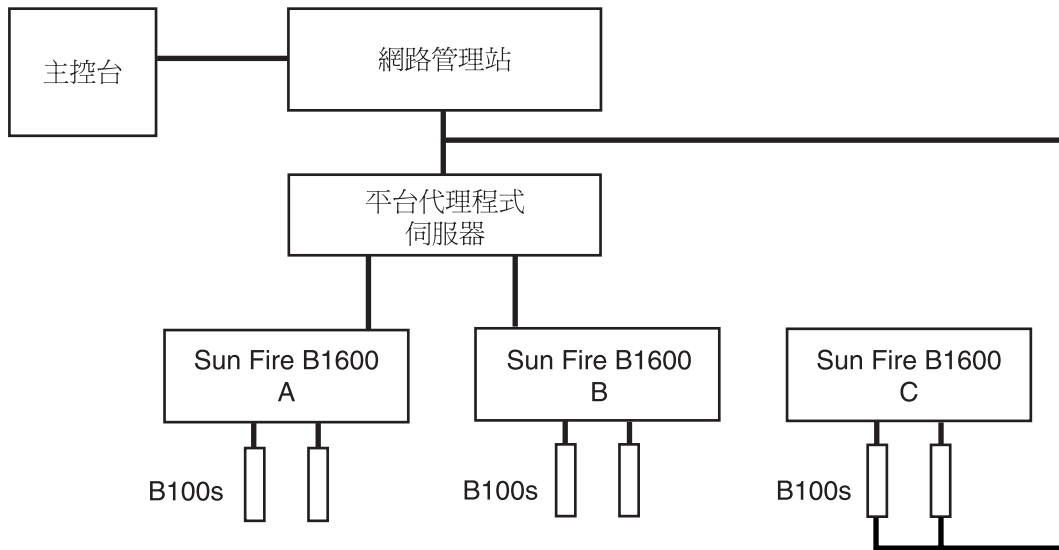


圖 9-1 領域和平台硬體監視範例

系統需求

安裝 SNMP Management 代理程式時，請確認您的系統符合本節所討論的必要條件和要求。

作業環境

SNMP Management 代理程式軟體須在 Solaris 8 Update 3 或更新的版本上才能執行。

磁碟空間需求

平台代理程式伺服器上最少應有 512MB 的可用空間（建議容量為 1.0GB）。

增補程式

除了安裝 Solaris 作業環境軟體之外，還必須安裝以下增補程式：

Solaris 8

Sun Fire B100s Blade 不需安裝任何增補程式。

在安裝 SNMP Management 代理程式軟體之前，必須先在平台代理程式伺服器和 Blade 上安裝 Java1.4 以執行平台和領域硬體監視（請參閱第 61 頁的「Java 環境」）。

Solaris 9

無需安裝任何增補程式。

Java 環境

若要完整監視 Sun Fire B100s Blade，您必須先在每個監視的 Sun Fire B100s Blade 和平台代理程式伺服器安裝 Java J2SE 1.4 組件。

注意 – 此安裝作業可升級任何現有的 J2SE 軟體。若您不要升級此軟體（因為您有部分應用程式不相容於預設 J2SE 版本 1.3.1），您可以安裝 J2RE 與預設系統 J2SE 共存。這需要另外配置 Sun SNMP Management 代理程式軟體。詳情請參閱附錄 A。

若您只是要監視 Sun Fire B1600 機架而沒有任何特定監視設備，您必須先安裝 Java J2SE 1.4 組件於平台代理程式伺服器上。在此情況下，硬碟設備、CPU 資訊和乙太網路 MAC 位址將無法使用。

若要確保 Java 1.4 檔案會安裝至正確的位址 (/usr/j2se)，請使用 j2sdk-1_4_0_03-solaris-sparc.tar.z 套裝軟體執行安裝。

該檔案可從以下網站取得：

<http://java.sun.com/j2se/1.4/download.html>

選擇 SDK 來下載 Solaris SPARC 32-bit tar.z

請遵照上述網址的指示來下載檔案。

注意 – 此檔案名稱在寫入時是正確的。確認您的檔案是否為最新的版本。此檔案名稱的格式為 j2sdk-1_4_0_<ver>-solaris-sparc.tar.z，而 <ver> 即為軟體的版本。

當此安裝取代 J2SE 系統時，為了確保其他現有的 Java 應用程式仍然可以正確執行，您必須同時安裝 64-bit J2SE 1.4 套裝軟體，其包含了 j2sdk-1_4_0_<ver>-solaris-sparcv9.tar.z 檔案。



警告 – J2SE 1.4 會取代 Solaris 8 上的 J2SE 1.3.1，所以您在安裝 J2SE 1.4 之前，您必須先解安裝 J2SE 1.3.1。若您安裝的是 Solaris 8 下季更新版，部分 J2SE 1.4 套裝軟體將會被 J2SE 1.3.1 套裝軟體所覆蓋。若要確保 J2SE 1.4 安裝的位址正確，請使用 pkgadd 進行安裝。

確認安裝

若要確認安裝正確，請使用以下指令：

```
# /usr/j2se/bin/java -version
java version "1.4.1_03"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1_03-
b04)
Java HotSpot(TM) Client VM (build 1.4.1_03-b04, mixed mode)
```

此指令將回報系統上的安裝版本。

Java SNMP API

此安裝套裝軟體包含在 Java SNMP API 的更新版，SUNWjsnmp。在安裝 Management 代理程式軟體之前，請使用 pkgrm 移除此現有版本的套裝軟體。

安裝套裝軟體

此套裝軟體由管理軟體組成，可分成以下幾個組件：

- 領域硬體監視需要的套裝軟體
- 在平台代理程式伺服器執行平台硬體監視必要的套裝軟體
- 在目標設備上執行平台硬體監視必要的套裝軟體

請見表 9-1 得知上述套裝軟體。

表 9-1 SNMP Management 代理程式 套裝軟體說明

套裝軟體	套裝軟體名稱	功能
SUNWbgpc	SPA 個人模組架構	支援個人模組的架構
SUNWbgptk	SPA 個人模組工具箱	可再利用的組件模組資料庫和資料存取庫
SUNWbgpr	SPA 個人模組 (root)	RDP 啟動 Script
SUNWbgcm	SPA HW 平台物件管理員	平台物件管理員
SUNWbgcmr	SPA HW 平台物件管理員 (root)	平台物件管理員啟動 Script
SUNWbgpm	SPA SNMP 通訊協定中介程式 / 主代理程式	SNMP 通訊協定支援和主代理程式
SUNWbgpmr	SPA SNMP 通訊協定中介程式 / 主代理程式 (root)	SNMP 組件啟動 Script
SUNWbgidr	SPA 領域搜尋 (root)	領域代理程式搜尋啟動 Script
SUNWbgod	SPA 平台搜尋	平台代理程式搜尋常駐程式
SUNWbgodr	SPA 平台搜尋 (root)	平台代理程式搜尋啟動 Script
SUNWbgpji	SPA Sun Fire B100s 領域個人模組	B100s 領域設備
SUNWbgpjo	SPA Sun Fire B1600 平台個人模組	B1600 平台設備

注意 – 套裝軟體間有其內部安裝要求，所以必須依照特定順序來安裝（請參閱第 68 頁的「安裝 SNMP 軟體」）。

升級軟體

若要升級軟體，您必須在重新安裝新版本前移除有軟體（請參閱第十三章）。

套裝軟體內容

套裝軟體隨附於 tar 的檔案封包中。表 9-2 顯示 SUNWspa.1.0.tar.z 檔案封包的內容，其內含支援 Sun Fire B1600 平台管理和 Sun Fire B100s 領域管理之 SNMP 的套裝軟體。

打開封包後，各套裝軟體就會存放至表中所顯示的目錄底下。如需安裝軟體的詳細指示說明，請參閱第十章。

注意 – 確認您的檔案是否為最新的版本。

表 9-2 SNMP Management 代理程式套裝軟體封包

封包	說明	內容
SUNspa.1.0.22.tar.z	SNMP 平台代理程式套裝軟體，安裝於平台代理程式伺服器	platform/proxy/SUNWbgpc platform/proxy/SUNWbgpk platform/proxy/SUNWbgcm platform/proxy/SUNWbgcmr platform/proxy/SUNWbgod platform/proxy/SUNWbgodr platform/proxy/SUNWbgpjo platform/proxy/SUNWbgpm platform/proxy/SUNWbgpmr platform/proxy/SUNWjdr platform/proxy/SUNWjsnmp
	SNMP 設備套裝軟體，安裝於 Sun Fire B100s Blade	platform/target/SUNWbgpc platform/target/SUNWbgptk platform/target/SUNWbgpr platform/target/SUNWbgcm

表 9-2 SNMP Management 代理程式套裝軟體封包 (接上頁)

封包	說明	內容
	SNMP 領域代理程式套裝軟體，安裝於 Sun Fire B100s Blade	domain/SUNWbgpc domain/SUNWbgptk domain/SUNWbgcm domain/SUNWbgcmr domain/SUNWbgidr domain/SUNWbgpji domain/SUNWbgpm domain/SUNWbgpmr domain/SUNWjdr domain/SUNWjsnmp

若要打開 tar 檔案，請鍵入：

```
$ zcat SUNWspa.1.0.tar.Z | tar xf -
```

安裝領域或目標套裝軟體於 Sun Fire B100s

領域套裝軟體（領域硬體監視）或目標套裝軟體（平台硬體監視）必須安裝於 Sun Fire B100s Blade 中。打開封包的方法有很多，包含以下幾種：

- 在個別 Sun Fire B100s Blade 上打開領域或目標套裝軟體
- 在要安裝軟體的個別 Sun Fire B100s Blade 上，於 root 使用者可見之共享目錄打開領域或目標套裝軟體
- 設定網路安裝的領域或目標套裝軟體

對系統檔案的影響

部分新的啟動檔會建立於 /etc/init.d，如表 9-3 所示，並可連結至 /etc/rc<n>.d。

表 9-3 啟動 Script

組件	啟動 Script	套裝軟體名稱	套裝軟體說明
平台物件管理員	spapom	SUNWbgcmr	平台物件管理員 (root)

表 9-3 啓動 Script (接上頁)

組件	啓動 Script	套裝軟體名稱	套裝軟體說明
領域硬體搜尋模組	spaibdm*	SUNWbgidr	領域硬體搜尋模組 (root)
遠端資料外掛	spardp†	SUNWbgpr	個人模組 (root)
SNMP 通訊協定中介程式 / 主代理程式	spama	SUNWbgpmr	SNMP 通訊協定中介程式 SNMP 主代理程式 (root)

* 在 Sun Fire B100s Blade 只有領域套裝軟體。

† 在 Sun Fire B100s Blade 只有平台 / 目標套裝軟體。

搜尋模組會自動由 inetd 啓動，然後於 /etc/inetd.conf 檔案中建立新的項目。

平台物件管理員 (POM) 會監視其客戶端 IP 連接埠上所要求的活動。連接埠會登錄於 /etc/services 檔案中。

安裝軟體

本章將說明如何在 Sun Fire B1600 中安裝管理軟體。

本章包含以下內容：

- 第 67 頁的「選擇要安裝的軟體配置」
- 第 68 頁的「安裝 SNMP 軟體」
- 第 75 頁的「介面選項」

選擇要安裝的軟體配置

決定安裝何種軟體配置時，有以下二個主要考量：

1. 設備配置
2. 管理介面配置

設備配置

您可依據平台類型選擇適合的代理程式：

- 領域代理程式，執行於監視系統上

本端安裝軟體於受監視的 Sun Fire B100s Blade 上（領域）。每個 Blade 都受到個別監視，而且一次只能檢視一個 Blade。此即為所謂的**領域硬體監視**。

- 平台代理程式，由系統控制器代理

將軟體安裝於遠端（平台代理程式）伺服器上，其可透過 Sun Fire B1600 系統控制器和受監視的 Sun Fire B100s（目標）Blade 存取平台指示。如此一來，您就可以監視由系統控制器管理的所有硬體，包含電源供應器、系統控制器和所有 Blade。此即為所謂的**平台硬體監視**。

若您使用的是平台硬體監視，想要得知有關硬體、CPU 和乙太網路 Mac 位址的資訊，您就必須在每個受監視的 Sun Fire B100s Blade 上安裝目標套裝軟體。

注意 – Sun Fire B100s Blade 即為領域硬體監視中的 *領域*，以及平台硬體監視中的 *目標*。

請同時參閱第 60 頁的「設備」。

管理介面配置

此管理軟體是專為執行數種安裝方式所設計，且目前支援以下幾種安裝方法：

- snmpdx 的 SNMP，不需主代理程式（此為預設安裝）
- 使用主代理程式和 snmpdx 的 SNMP
在此情況下，您必須手動配置安裝。請參閱第十一章和第十二章。

安裝 SNMP 軟體

本節將大致說明安裝監視軟體的程序，而 每種安裝類型的詳細流程將於本章的各小節中說明。

安裝本軟體之前，請確認：

- 已在領域或目標（Sun Fire B100s），和平台代理程式伺服器安裝必要等級的 Solaris（請參閱第 61 頁的「作業環境」）。
- 已安裝所有必要的增補程式（請參閱第 61 頁的「增補程式」）和未提供於 SNMP 軟體的其他必要套裝軟體（請參閱第 61 頁的「Java 環境」）。
- 已安裝 Java 1.4（不論是透過升級現有的 J2SE 或個別安裝 J2RE）。請參閱第 61 頁的「Java 環境」以瞭解詳細資訊。

若您的系統已符合上述要求，就可以繼續執行 SNMP 軟體。

您現在必須決定要使用領域硬體監視或平台硬體監視。

- 若您選擇使用領域硬體監視，請遵照第 69 頁的「安裝領域硬體監視軟體」中的指示進行安裝。
- 若您選擇使用平台硬體監視，請遵照第 70 頁的「安裝平台硬體監視軟體」中的指示進行安裝。

安裝領域硬體監視軟體

在每個受到監視的 Blade 上安裝此套裝軟體（請參閱第 65 頁的「安裝領域或目標套裝軟體於 Sun Fire B100s」）。

▼ 安裝軟體



1. 確認您已安裝 Java 1.4。
請參閱第 61 頁的「Java 環境」。



2. 確認已移除任何現有的 SUNWjsnmp 版本。
請參閱第 63 頁的「Java SNMP API」。



3. 請依照以下順序在 Sun Fire B100s Blade 上安裝領域代理程式套裝軟體以避免回報任何不符合要求之事件。

```
# pkgadd -d . SUNWbgptk SUNWbgpc SUNWbgcm SUNWbgcmr SUNWbgidr \  
SUNWbgpji SUNWjsnmp SUNWjdrtr SUNWbgpm SUNWbgpmr
```



4. 配置 Java 環境。
 - a. 若您是依照第 61 頁的「Java 環境」所述來安裝 J2SE 1.4，請跳過此步驟。
 - b. 若您是依照第 111 頁的「安裝 J2RE 1.4」所述來安裝 J2RE 1.4，請編輯領域硬體監視啟動 Script（請參閱第 113 頁的「領域硬體監視」）。



5. 配置軟體。
請參閱第十一章。



6. 重新啓動 Sun Fire B100s Blade。
您現在已完成領域硬體監視軟體的安裝。請依照第 75 頁的「介面選項」的指示繼續進行安裝。

7. 鍵入以下指令確認作業已正確啓動：

```
# ps -ef | grep spa.snmp
root 15789      1  1 13:44:01 pts/2    0:00 /usr/j2se/bin/java
-Dcom.sun.spa.snmp.LOG_LEVEL=INFO -Djdk.security.file=//etc
#
# ps -ef | grep spa.wbem

root    278      1  0   Feb 24 ?          44:19 /usr/j2se/bin/java
-Dcom.sun.spa.wbem.pomi.port=3333 -Xms64m -Xmx768m -Dcom.sun
#
```

如果輸出與上述範例類似，就表示已開始執行作業。

安裝平台硬體監視軟體

- 決定是否要在 Sun Fire B100s Blade 上安裝 Java 1.4 (請參閱第 61 頁的「Java 環境」的相關討論)。若要支援目標設備，就必須安裝 Java 1.4。
 - 若要使用目標設備安裝軟體，請執行第 70 頁的「使用目標設備安裝軟體」中所述的安裝作業。
 - 若不要使用目標設備安裝軟體，請執行第 72 頁的「不使用目標設備安裝軟體」中所述的安裝作業。
- 在平台代理程式伺服器上安裝平台代理程式套裝軟體。
- 在每個受監視的 Blade 上安裝目標平台代理程式套裝軟體 (如為必要選項)。
- 設定系統控制器 SMS IP 位址。

▼ 使用目標設備安裝軟體

1. 確認已在做為平台代理程式的伺服器上安裝 Java 1.4。
請參閱以下的第 61 頁的「Java 環境」和步驟 4。
2. 確認已從平台代理程式伺服器移除任何現有的 SUNWjsnmp 版本。
請參閱第 63 頁的「Java SNMP API」。
3. 請依照以下順序在平台代理程式伺服器安裝平台代理程式套裝軟體以避免不符合要求的情況發生：

```
# pkgadd -d . SUNWbgptk SUNWbgpc SUNWbgcm SUNWbgcmr SUNWbgod \
SUNWbgodr SUNWbgpjo SUNWjsnmp SUNWjdrtr SUNWbgpm SUNWbgpmr
```

4. 配置 Java 環境。

- a. 若您是依照第 61 頁的「Java 環境」所述來安裝 J2SE 1.4，請跳過此步驟。
- b. 若您是依照第 111 頁的「安裝 J2RE 1.4」所述來安裝 J2RE 1.4，請編輯平台硬體監視啓動 Script（請參閱第 113 頁的「平台硬體監視」）。



5. 配置軟體。
請參閱第十一章。



6. 鍵入以下指令手動啓動平台代理程式：

```
# /etc/init.d/spapom start
# /etc/init.d/init.snmpdx stop
# /etc/init.d/spama stop
# /etc/init.snmpdx start
# pkill -l inetd
```

或重新啓動平台代理程式。



7. 鍵入以下指令確認作業已正確啓動：

```
# ps -ef | grep spa.snmp
root 15789 1 1 13:44:01 pts/2 0:00 /usr/j2se/bin/java
-Dcom.sun.spa.snmp.LOG_LEVEL=INFO -Djdk.security.file=//etc
#
# ps -ef | grep spa.wbem

root 278 1 0 Feb 24 ? 44:19 /usr/j2se/bin/java
-Dcom.sun.spa.wbem.pomi.port=3333 -Xms64m -Xmx768m -Dcom.sun
#
# netstat -a | grep mism
*.mismi *.* 0 0 24576 0 LISTEN
*.mismi *.* 0
0 24576 0 LISTEN
#
```

如果輸出與上述範例類似，就表示已開始執行作業。

8. 確認已在受監視的目標 Sun Fire B100s Blade 上安裝 Java 1.4。
請參閱以下的第 61 頁的「Java 環境」和步驟 11。



9. 確認已從目標 Blade 移除任何現有的 SUNWjsnmp 版本。
請參閱第 63 頁的「Java SNMP API」。



10. 在目標 Blade 上安裝目標平台代理程式套裝軟體。
這些套裝軟體可幫助存取使用受監視設備之 Solaris 介面的設備資料。

請依照以下順序來安裝套裝軟體以避免不符合要求的情況發生：

```
# pkgadd -d . SUNWbgpstk SUNWbgpc SUNWbgcm SUNWbgpr
```

11. 配置 Java 環境。

- a. 若您是依照第 61 頁的「Java 環境」所述來安裝 J2SE 1.4，請跳過此步驟。
- b. 若您是依照第 111 頁的「安裝 J2RE 1.4」所述來安裝 J2RE 1.4，請編輯目標硬體監視啟動 Script（請參閱第 113 頁的「平台硬體監視」）。

12. 鍵入以下指令手動啟動目標設備：

```
# /etc/init.d/spardp start
```

或重新啟動系統。

13. 鍵入以下指令確認作業已正確啟動：

```
# netstat -an | grep 1099
*.1099          *.*              0                0 24576           0 LISTEN
```

如果輸出與上述範例類似，就表示已開始執行作業。

14. 使用 `setupsc` 設定 SMS IP 位址。

請依照第 74 頁的「配置系統控制器」的指示繼續安裝。

▼ 不使用目標設備安裝軟體

1. 確認已在做為平台代理程式的伺服器上安裝 Java 1.4。

請參閱第 61 頁的「Java 環境」。

2. 確認已從平台代理程式伺服器移除任何現有的 SUNWjsnmp 版本。

請參閱第 63 頁的「Java SNMP API」。

3. 請依照以下順序在平台代理程式伺服器安裝平台代理程式套裝軟體以避免不符合要求的情況發生：

```
# pkgadd -d . SUNWbgpstk SUNWbgpc SUNWbgcm SUNWbgcmr SUNWbgod \
SUNWbgodr SUNWbgpjo SUNWjsnmp SUNWjdrdt SUNWbgpm SUNWbgpmr
```

4. 配置 Java 環境。

- a. 若您是依照第 61 頁的「Java 環境」所述來安裝 J2SE 1.4，請跳過此步驟。
- b. 若您是依照第 111 頁的「安裝 J2RE 1.4」所述來安裝 J2RE 1.4，請編輯平台硬體監視啓動 Script（請參閱第 113 頁的「平台硬體監視」）。



5. 配置軟體。
請參閱第十一章。



6. 鍵入以下指令手動啓動平台代理程式：

```
# /etc/init.d/spapom start
# /etc/init.d/init.snmpdx stop
# /etc/init.d/spama stop
# /etc/init.snmpdx start
# pkill -l inetd
```

或重新啓動平台代理程式。



7. 鍵入以下指令確認作業已正確啓動：

```
# ps -ef | grep spa.snmp
root 15789 1 1 13:44:01 pts/2 0:00 /usr/j2se/bin/java
-Dcom.sun.spa.snmp.LOG_LEVEL=INFO -Djdk.security.file=/etc
#
# ps -ef | grep spa.wbem

root 278 1 0 Feb 24 ? 44:19 /usr/j2se/bin/java
-Dcom.sun.spa.wbem.pomi.port=3333 -Xms64m -Xmx768m -Dcom.sun
#
# netstat -a | grep mism
*.mismi *.* 0 0 24576 0 LISTEN
*.mismi *.* 0
0 24576 0 LISTEN
#
```

如果輸出與上述範例類似，就表示已開始執行作業。



8. 使用 `setupsc` 設定 SMS IP 位址。
請依照下列指示繼續進行。

配置系統控制器

在安裝 SNMP 軟體之後，您必須在系統控制器設定平台代理程式伺服器的 SMS IP 位址。若要進行設定，請登入系統控制器主控台、執行 `setupsc`，然後新增平台代理程式伺服器的 IP 位址。

在以下範例中，IP 位址被設定為 10.5.1.1。

在每次出現問題後按 [ENTER] 接受目前的值，直到出現以下文字行：

```
Enter the SMS IP address
```

輸入 IP 位址，然後按 [ENTER]。繼續按 [ENTER] 來回答之後的問題。

注意 - `setupsc` 指令於 《*Sun Fire B1600 Blade System Chassis Software Setup Guide*》中有詳細說明。

程式範例 10-1 設定 SMS IP 位址

```
hornet-sc>setupsc
Entering Interactive setup mode.
Use Ctrl-z to exit & save. Use Ctrl-c to abort.
Do you want to configure the enabled interfaces [y]?
Should the SC network interface be enabled [y]?
Should the SC telnet interface be enabled for new connections [y]?
Do you want to configure the network interface [y]?
Should the SC use DHCP to obtain its network configuration [n]?
Enter the SC IP address [129.156.174.140]:
Enter the SC IP netmask [255.255.255.0]:
Enter the SC IP gateway [129.156.174.1]:
Do you want to configure the SC private addresses [y]?
Enter the SSC0/SC IP private address [129.156.174.118]:
Enter the SSC1/SC IP private address [129.156.174.128]:
Do you want to enable a VLAN for the SC [n]?
Enter the SMS IP address [0.0.0.0]: 10.5.1.1
<truncated>

hornet-sc>
```

介面選項

預設安裝可透過 SNMP 做為 `snmpdx` 的子代理程式來提供管理。安裝期間不需輸入任何文字。您可以在配置結束後自定配置。

您可以編輯配置檔來新增 SNMP 和 `snmpdx` 的主代理程式功能。

注意 – 在所有情況之下，SNMP 的「存取控制清單」(ACL) 使用預設配置以避免他人存取。您必須進行配置才可進行存取（請參閱第十一章）。

`snmpdx` 的 SNMP（預設）

此選項會登錄 SNMP 中介程式為 `snmpdx` 的子代理程式，並使用自動分配的 UDP 連接埠編號至發送要求給中介程式之處。發出的要求可以透過 `snmpdx`，或藉由圖 10-1 所示之虛線來引導至中介程式（若其是由中介程式 ACL 檔案來啟動。請參閱第十二章）。

表 10-1 圖 10-1 的連接埠摘要

編號	功能	<code>spama.conf</code> 中的參數	預設
1	<code>snmpdx</code> 轉寄要求至 SNMP 中介程式的連接埠	<code>SPAPM_REQ_PORT</code>	
2	中介程式傳送陷阱至 SNMP 管理員的連接埠	<code>SPAPM_TRAP_PORT</code>	162

注意 – 雖然預設配置會自動啟動，您仍然必須配置 `snmpdx` 的 ACL 檔和 / 或中介程式的 ACL 檔以支援您的管理員配置。

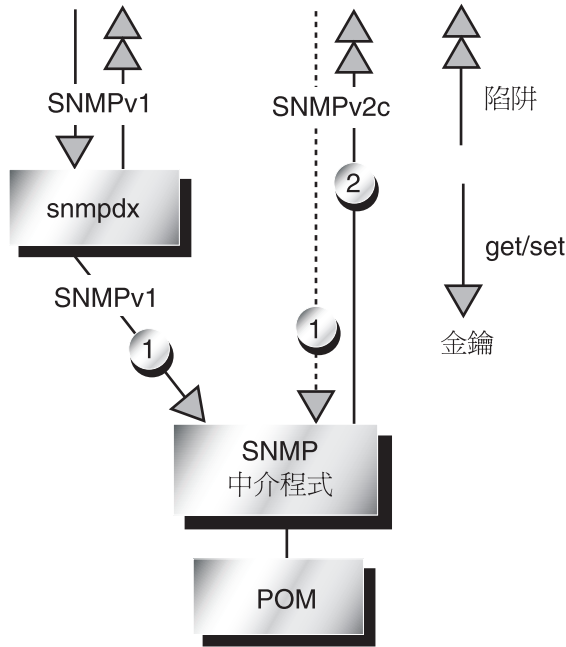


圖 10-1 SNMP 為 snmpdx 子代理程式時的資料流程

使用主代理程式和 snmpdx 的 SNMP

此選項可增加 SNMP 和 snmpdx 主代理程式 SNMPv3 之安全性功能。snmpdx 自動啟動會關閉，而且主代理程式會登錄至連接埠 161。新的連接埠編號會自動指定給 snmpdx。

中介程式的陷阱會由主代理程式選擇翻譯至 SNMPv3，或直接傳送出去。

snmpdx 的陷阱只會直接轉寄至 SNMP 管理員，而且無法由主代理程式翻譯。

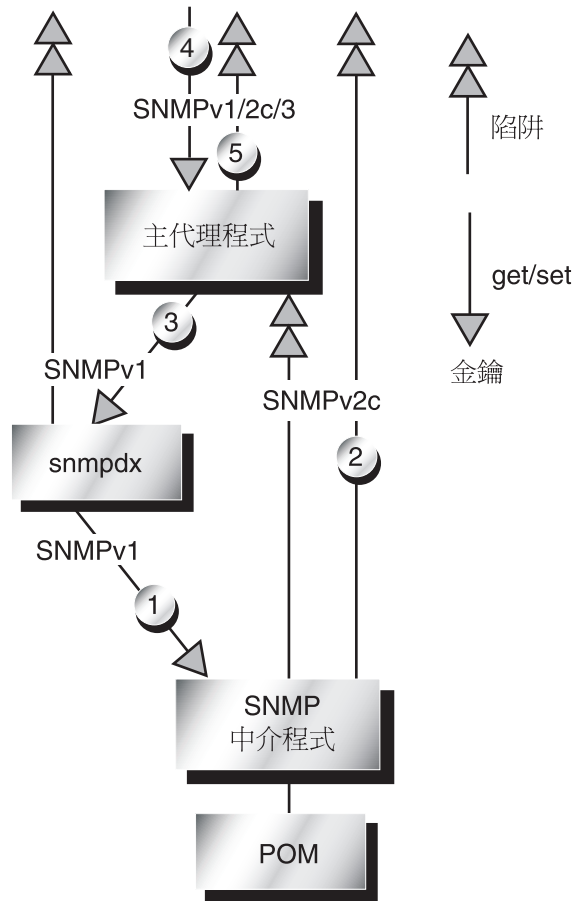


圖 10-2 使用主代理程式時的資料流程

表 10-2 圖 10-2 的連接埠摘要

編號	功能	spama.conf 中的參數	預設
1	snmpd 轉寄要求至 SNMP 中介程式子代理程式的連接埠	SPAPM_REQ_PORT	
2	中介程式直接傳送陷阱至 SNMP 管理員的連接埠	SPAPM_TRAP_PORT	162
3	主代理程式轉寄要求至 snmpd 的連接埠	SNMPDX_REQ_FORWARD_PORT	
4	由主代理程式監視要求的連接埠	MASTER_AGENT_REQ_PORT	161
5	主代理程式傳送陷阱至 SNMP 管理員的連接埠	SPAPM_TRAP_PORT	162

使用 SNMP 的第三協力廠商主代理程式

此選項會登錄 SNMP 中介程式為子代理程式，其使用手動分配或由第三協力廠商主代理程式分配之連接埠編號。若要啟動直接存取，您必須手動配置中介程式 ACL 檔。請參閱第十二章的說明。

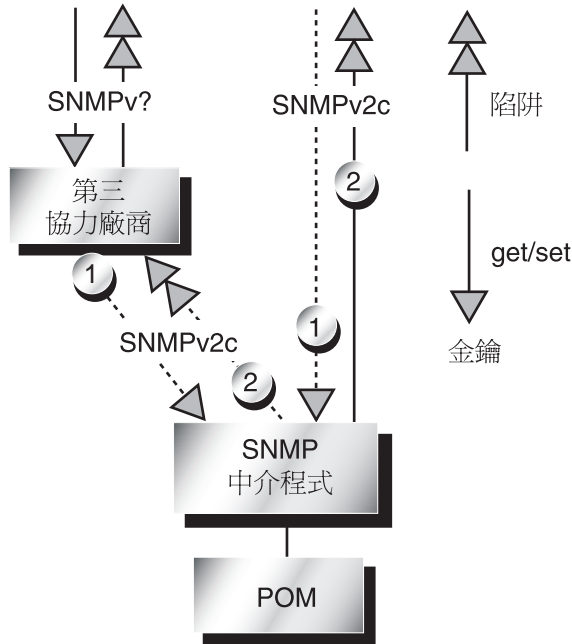


圖 10-3 使用第三協力廠商主代理程式時的資料流程

表 10-3 圖 10-3 的連接埠摘要

編號	功能	spama.conf 中的參數	預設
1	SNMP 中介程式用來啟動直接存取，或第三協力廠商主代理程式用來轉寄要求至 SNMP 中介程式的連接埠	SPAPM_REQ_PORT	
2	SNM 中介程式用來傳送 SNMPv2c 陷阱至第三協力廠商代理程式，或直接用來傳送至 SNMP 管理員的連接埠	SPAPM_TRAP_PORT	

配置檔案

本章將概略說明您可編輯用來配置軟體的檔案。它會列出可配置參數，並介紹存取控制 (Access Control) 的觀念。

在閱讀本章前請先參閱第十二章，以得知如何使用本章說明的檔案來配置基本 SNMP 選項。

本章包含以下內容：

- 第 79 頁的「配置檔案」
- 第 80 頁的「一般配置檔」
- 第 88 頁的「存取控制」
- 第 89 頁的「ACL 檔格式」
- 第 91 頁的「中介程式配置檔」
- 第 94 頁的「主代理程式配置檔」

注意 – 若想瞭解 SNMP 套裝軟體與安裝的方法，請參閱第九章和第十章。

配置檔案

位於 `/etc/opt/SUNWspa/` 內的下述檔案可決定 SNMP 的配置。

- 一般配置檔
 - `spama.conf` — 可定義如何配置主代理程式和中介程式
- 中介程式配置檔
 - `spapm.acl` — 可定義中介程式的存取控制
 - `spapm_snmpdx.acl` — 可定義做為 `snmpdx` 子代理程式的中介程式之存取控制
- 主代理程式配置檔

若您不是使用主代理程式，就無需配置以下檔案。

- `spama.acl` — 可定義主代理程式的存取控制
- `spama.uacl` — 可定義主代理程式的 SNMPv3 使用者和使用內容存取控制
- `spama.security` — 可定義 `spama.uacl` 所提及的 SNMPv3 使用者

上述檔案的詳細說明請參閱以下各節。

一般配置檔

`spama.conf`

`spama.conf` 檔案包含配置參數。請參閱下節說明和表 11-1。`spama.conf` 檔案的範例顯示於程式範例 11-1 中。

一般選項

START_MEDIATOR

設定此參數為 `yes` 以執行中介程式，或設定為 `no`（請同時參閱圖 10-1）。

預設值為：

```
START_MEDIATOR=yes
```

START_MASTER_AGENT

若您想使用 SNMPv3 的安全性功能，但您使用的第三協力廠商主代理程式沒有此功能，請將此參數設定為 `yes` 來啓動主代理程式（請參閱圖 10-2 和隨附的圖表）。

若您無需使用 SNMPv3 安全性功能，而且您正在使用其他主代理程式（包括 `snmpdx`），或者您不想使用任何代理程式，請將此參數設定為 `no`（請同時參閱圖 10-1 和圖 10-3，以及隨附的圖表）。

預設值為：

```
START_MASTER_AGENT=no
```

AGENT_INTERFACE_NAME

若已啓動主代理程式（請見上述說明），在此設定的值便會指定主代理程式應附掛的網路介面，和附掛於 localhost 的通訊協定中介程式。

若沒有啓動主代理程式，在此設定的值便會定義中介程式附掛的網路介面主機名稱。若您沒有指定值，預設存取值將設定預設介面。

若您將中介程式當成 snmpdx 的子代理程式在使用，請將此值設定為 localhost。

預設值為：

```
AGENT_INTERFACE_NAME=localhost
```

主代理程式選項

如此一來，主代理程式才可搭配 snmpdx 運作。主代理程式啓動 Script 會停止 snmpdx、接掌其 SNMP 連接埠，然後在另一個連接埠上重新啓動 snmpdx 為子代理程式。

如果 SNMPDX_REQ_FORWARD_PORT 參數包含 null 值，主代理程式的啓動 Script 就會在其他未知範圍 32768 至 65535 搜尋空白連接埠，然後在該空白連接埠上重新啓動 snmpdx 為子代理程式。此啓動 Script 同時也會搜尋 /etc/services，而且不會用到此處列出的任何連接埠。

但若您指定 SNMPDX_REQ_FORWARD_PORT 值，主代理程式就會使用此連接埠來轉寄要求至 snmpdx。而在此情況下，主代理程式不會檢查該連接埠是否正在使用中。

MASTER_AGENT_REQ_PORT

此為主代理程式接收管理員要求的連接埠。至於其他的配置就無需變更此值（請同時參閱圖 10-2 與其隨附的圖表）。

若沒有指定的話，其預設值為 161。

ENABLE_SNMPV2C_SETS

此參數可控制主代理程式是否可設定使用 SNMPv1 或 SNMPv2c 來控制操作。將此值設定為 yes 會降低其實質安全性，因為 SNMPv1 和 SNMPv2C 通訊協定實際上是不安全的。

預設值為：

```
ENABLE_SNMPV2C_SETS=no
```

SNMPDX_REQ_FORWARD_PORT

此參數可控制主代理程式轉寄要求至 `snmpdx` 的連接埠。若您沒有指定值，主代理程式就會執行自動配置（請參閱本節前言以及圖 10-2 與其隨附的圖表）。

若您有指定值，您就必須手動配置 `snmpdx` 以於此連接埠上執行聽取作業。

預設值為

```
SNMPDX_REQ_FORWARD_PORT=
```

SNMPV3_USER

此參數可決定發出 SNMPv3 陷阱的 SNMPv3 使用者。

預設值為：

```
SNMPV3_USER=defaultUser
```

注意 – 若要傳送 SNMPv3 陷阱，您就必須設定 `SPAPM_TRAPS_ARE_V3=yes`。

通訊協定中介程式選項

SUB_AGENT

此選項可決定中介程式或主代理程式是否要由主代理程式 `snmpdx` 啟動，而非自動啟動。

預設值為：

```
SUB_AGENT=yes
```

若您指定為 `yes`，您就必須通過 `<port>` 參數來啟動中介程式。如下所示：

```
# /etc/init.d/spama start <port>
```

`<port>` 處為中介程式聽取 SNMP 要求的 UDP 連接埠。例如，當使用 `snmpdx` 時，中介程式的呼叫就會由 `/etc/snmp/conf/spapm.rsrc` 檔案中的以下文字行所控制：

```
command = "etc/init.d/spama start $PORT"
```

注意 – 若 SUB-AGENT=YES，則請忽略 START_MASTER_AGENT 值。若您啓動主代理程式功能，您就必須將 SUB_AGENT 設定爲 no。

若您指定爲 no，中介程式就會在啓動階段啓動（使用啓動 Script /etc/rc3.d/S80spama），而其聽取 SNMP 要求的 UDP 連接埠也將由 SPAPM_REQ_PORT 設定所定義。請參閱以下說明。

SPAPM_REQ_PORT

當 SUB_AGENT 設定爲 no 時，此參數可決定中介程式接收要求的連接埠。請參閱圖 10-1、圖 10-2 和圖 10-3 與其隨附的圖表。

預設值爲：

SPAPM_REQ_PORT=

在預設設定值中，若 START_MASTER_AGENT=yes，就會自動分配連接埠編號。若 START_MASTER_AGENT=no 就會使用預設連接埠編號 33000。

若 START_MASTER_AGENT=no，請將 SPAPM_REQ_PORT 設定爲指定的值。如此一來，使用固定連接埠的本端主代理程式或遠端 SNMP 管理員才可直接存取中介程式。

SPAPM_TRAPS_ARE_V3

此參數可決定中介程式的陷阱爲 SNMPv3 或 SNMPv2c。

設定陷阱爲 SNMPv2c 的預設值爲：

SPAPM_TRAPS_ARE_V3=no

注意 – 若您啓動 SNMPv3 陷阱 (SPAPM_TRAPS_ARE_V3=yes)，您就必須同時設定 START_MASTER_AGENT=yes 和 START_MEDIATOR=yes。

SPAPM_TRAP_PORT

此參數可決定中介程式陷阱傳送的連接埠編號。請參閱圖 10-1、圖 10-2 和圖 10-3 與其隨附的圖表。

預設值爲：

SPAPM_TRAP_PORT=162

SPAPM_TRAP_INTERFACE

此參數可決定中介程式傳送 SNMP 陷阱的介面。若沒有定義此參數，陷阱就會從主機的預設介面發出。

注意 – SNMPv2c 陷阱可直接傳送出去，而非透過 snmpdx 發出。

尚未定義此預設值：

SPAPM_TRAP_INTERFACE=

SPAPM_OPTIONS

此參數可讓您藉由指定以下選項來修改中介程式的運作：

- -a 傳送屬性變更通知
- -s 傳送狀態變更通知
- -c 傳送物件建立通知
- -C 於初始期間啟動物件建立通知
- -d 傳送物件刪除通知
- -l 預設啟動目前問題清單日誌

此參數的格式和預設值為：

SPAPM_OPTIONS=" -ascCd1 "

表 11-1 spama.conf 的預設值

參數	預設安裝後的值
START_MEDIATOR	START_MEDIATOR=yes
START_MASTER_AGENT	START_MASTER_AGENT=no
AGENT_INTERFACE_NAME	AGENT_INTERFACE_NAME=localhost
MASTER_AGENT_REQ_PORT	MASTER_AGENT_REQ_PORT=161 (若未指定值的話，此值即為預設值)
ENABLE_SNMPV2C_SETS	ENABLE_SNMPV2C_SETS=no
SNMPDX_REQ_FORWARD_PORT	SNMPDX_REQ_FORWARD_PORT=
SNMPV3_USER	SNMPV3_USER=defaultUser
SUB_AGENT	SUB_AGENT=yes
SPAPM_REQ_PORT	SPAPM_REQ_PORT=

表 11-1 spama.conf 的預設值 (接上頁)

參數	預設安裝後的值
SPAPM_TRAPS_ARE_V3	SPAPM_TRAPS_ARE_V3=no
SPAPM_TRAP_PORT	SPAPM_TRAP_PORT=162
SPAPM_TRAP_INTERFACE	SPAPM_TRAP_INTERFACE=
SPAPM_OPTIONS	SPAPM_OPTIONS="-ascCdl"

程式範例 11-1 spama.conf 檔案範例

```
#!/sbin/sh
#
#ident  "@(#)spama.conf1.17 01/29/03 SMI"
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.

#
# This file is used to control the configuration of the Master Agent and
# Protocol Mediator
#

#
# Master Agent / Mediator configuration
#

#####
# General options
#####

#
# Set to "yes" if the mediator component should be started
#
START_MEDIATOR=yes

#
# Set to "yes" to enable the master agent
#
START_MASTER_AGENT=no

#
# Hostname of the network interface for the agent to bind to. If this
# is not specified the agent will be accessible via the default
# interface.
#
# If the mediator is being used as a sub-agent this should be
# set to localhost.
```

```

#
# If the master agent is enabled, this setting applies to its interface,
# the protocol mediator being bound to localhost.
AGENT_INTERFACE_NAME=localhost

#####
# Master Agent options
#####

#
# SNMP port for master agent to receive SNMP get/set requests.
#
# This port number will be used to listen for SNMP get/set
# requests.
#
# If this value is blank, default will be 161 if START_MASTER_AGENT=yes
#
MASTER_AGENT_REQ_PORT=

#
# set to "yes" to enable SNMPv1/SNMPv2c SET operations via the master agent.
#
ENABLE_SNMPV2C_SETS=no

#
# SNMP sub-agent port to which non-SNMP Protocol Mediator (snmpdx) requests
# will be sent.
#
# If this port setting is blank (default), automatic configuration will be
# performed. The port number for snmpdx will be dynamically determined
# if snmpdx is already using the UDP port where the Master Agent listens
# for SNMP get/set requests (by default UDP port 161) at Master Agent startup.
#
# If this port setting is blank but snmpdx is not using the same port
# as the Master Agent will listen for SNMP get/set requests, the Master
# Agent will use the port number which is being used by snmpdx to forward
# the SNMP set/get requests to snmpdx.
#
# If this port is set, it is expected that the user should perform
# the "listening" port configuration for the sub-agents and the Master
# Agent will use this port number to forward non-SNMP Protocol Mediator
# requests.
#
SNMPDX_REQ_FORWARD_PORT=

#
# SNMPv3 user

```


程式範例 11-1 spama.conf 檔案範例 (接上頁)

```
#
# The Mediator will use this user to send the V3 traps (if enabled with
# SPAPM_TRAPS_ARE_V3).
#
# If this value is blank, default will be 'defaultUser'.
SNMPV3_USER=

#####
# Protocol Mediator options
#####

#
#
# Sub-agent configuration
#
# If the master agent is not being used (i.e. START_MASTER_AGENT=no), then
# setting SUB_AGENT=yes indicates that the mediator should be started with a
# port number argument by snmpdx or a third party master agent. Otherwise, set
# to no if the mediator is to be started with a manually configured port
# number.
#
# If START_MASTER_AGENT=yes then this setting is ignored.
#
SUB_AGENT=yes

#
# Mediator request port. If the START_MASTER_AGENT="no" and SUB_AGENT="no", the
# default is 33000, otherwise it is dynamically allocated.
#
SPAPM_REQ_PORT=

#
# set to yes to enable v3 mediator traps (requires START_MASTER_AGENT=yes and
# START_MEDIATOR=yes)
#
SPAPM_TRAPS_ARE_V3=no

#
# Default port for traps
#
SPAPM_TRAP_PORT=162

# This is also used to define the interface to which SNMP traps will be
# sent by the protocol mediator independently of the setting of
# AGENT_INTERFACE_NAME. If not defined, traps will be issued from the
# host's default interface.
#
```

```
SPAPM_TRAP_INTERFACE=
#
# Agent option flags
#
# -a    Send attribute change notifications
# -s    Send state change notifications
# -c    Send object creation notifications
# -C    Enable object creation notifications during initialization
# -d    Send object deletion notifications
# -l    Enable current problem list logs by default
#
SPAPM_OPTIONS="-ascCdl"
```

存取控制

存取控制是基於管理主機電腦的 IP 位址和社群，以及每個子代理程式所指定的社群而定。社群的存取權和主機電腦皆定義於 ACL 檔中。

ACL 檔同時也定義了代理程式傳送陷阱的主機。陷阱傳送出去後，代理程式會將此陷阱傳送至所有列於 ACL 檔 <陷阱利益主機清單> 中的主機。

ACL 檔總共有三個：

- spapm.acl 不僅可控制中介程式的存取（直接由管理應用程式或一般 snmpdx 來控制），還可定義 SNMP 陷阱的必要接收者。
- 當中介程式被配置為 snmpdx 子代理程式時，spapm_snmpdx.acl 就可以控制中介程式的存取。
- spama.acl 可透過 SNMPv3 主代理程式來控制存取。

SNMPv3 的存取控制、社群和陷阱轉寄參數皆定義於 spama.uacl 和 spama.security 中。

ACL 檔格式

ACL 檔包含一個定義社群和管理員存取權的 `acl` 群組，以及定義傳送陷阱之社群和主機之 `trap` 群組。ACL 檔同時也包含註解行，其以井字號 (#) 做為起首字元。

注意 - `spapm_snmpdx.acl` 檔中的語法有部分差異。請參閱 `Script` 中的註釋以得知詳細資訊。

acl 群組

`acl` 群組包含一個或多個使用以下語法的社群配置清單：

```
acl = {  
    < 清單 1 >  
    < 清單 2 >  
    ...  
    < 清單 N >  
}
```

檔案中的 `acl` 群組可指定特定社群和管理員的存取權。此群組由社群配置清單所組成，並使用以下格式：

```
{  
    communities = < 社群清單 >  
    access = < 存取權 >  
    managers = < 主機清單 >  
}
```

< 社群清單 > 處為此存取控制套用的 SNMP 社群名稱清單。清單中的社群名稱以逗點分隔。

< 存取權 > 處可指定授與存取權的管理員，其為管理員物件中所有執行於指定電腦上的管理員。可使用的值有二個：

- `read-write`
- `read-only`

< 主機清單 > 物件處可指定被授與存取權的管理員之主機電腦。< 主機清單 > 處為一以逗點分隔之主機清單，其代表意義如下：

- 主機名稱（例如 hubble）
- IP 位址（例如 123.456.789.12）。
- 子網路遮罩（例如 123!255!255!255）

注意 – 若要在 ACL 檔中分辨 IP 位址和子網路遮罩，請注意子網路遮罩中的每個整數值皆以驚嘆號 (!) 分開，而非點 (.)。

程式範例 11-2 acl 群組範例

```
acl = {
  {
    communities = public, private
    access = read-only
    managers = rag, tag, bobtail
  }
  {
    communities = tigger
    access = read-write
    managers = brittas
  }
}
```

trap 群組

trap 群組可指定代理程式傳送陷阱的主機。當需要中介程式傳送 SNMPv2 陷阱時才需要進行配置。當使用 SNMP 主代理程式傳送 SNMPv3 陷阱時，就不需配置中介程式。

此群組包含一個或多個陷阱社群定義，並使用以下語法：

```
trap = {
  < 社群 1 >
  < 社群 2 >
  ...
  < 社群 N >
}
```

每行皆定義了主機組與即將傳送至主機組的陷阱內之 SNMP 社群字串的關係。每個 trap-community 定義皆使用以下格式：

```
{
    trap-community = < 陷阱社群字串 >
    hosts = < 陷阱利益主機清單 >
}
```

< 陷阱社群字串 > 物件處可指定 SNMP 社群字串，並包含於 hosts 物件內指定要傳送至主機的陷阱中。

< 陷阱利益主機清單 > 物件處可指定以逗點分隔之主機清單。每個主機必須使用其名稱或完整的 IP 位址以供辨識，如以下範例所示：

程式範例 11-3 trap 群組範例

```
trap = {
    {
        trap-community = tigger
        hosts = gandalf, frodo
    }
}
```

中介程式配置檔

本節將說明以下配置檔的格式：

第 91 頁的「spapm.acl 檔」

第 93 頁的「spapm_snmpdx.acl 檔」

spapm.acl 檔

spapm.acl 檔可定義通訊協定中介程式 (Protocol Mediator) 的存取控制。若要瞭解此檔案的一般格式，請參閱第 89 頁的「ACL 檔格式」。本節包含 spapm.acl 檔的特定相關資訊，以及關於此檔案的範例（請見程式範例 11-4）。

根據預設值，此檔案位於 /etc/opt/SUNWspa 中。

若有 ACL 檔，其定義的存取權將適用於透過其 SNMP 配接卡存取代理程式的所有管理員或平台代理程式伺服器。若啟動代理程式時沒有出現 ACL 檔，所有的管理員將被授與可透過 SNMP 配接卡完整存取代理程式的權限，而且不會發出任何陷阱。

若要啟動 SNMP 配接卡的存取控制和陷阱，當代理程式啟動時須確認有 ACL 檔。因為 ACL 檔包含安全性的相關資訊，所以其指定的限制存取權只能由 root 讀取。

acl 和 trap 群組的格式說明請分別參閱第 89 頁的「acl 群組」和第 90 頁的「trap 群組」中的說明。

若中介程式已登錄為主代理程式的子代理程式（例如 snmpdx），您就必須在 spapm.acl 檔中指定 localhost 為管理員，因為其為主代理程式轉寄之 SNMP 封包的來源處。當使用 snmpdx 時，其會原封不動地轉寄社群字串。因此，您也必須指定 spapm_snmpdx.acl 檔中本檔所列出的社群（請參閱程式範例 11-4）。

程式範例 11-4 spapm.acl 檔範例

```
#
#  @(#)spapm.acl      1.6 03/01/29 SMI
#
#  Copyright 2003 Sun Microsystems, Inc. All rights reserved.
#
#  Template ACL file for Sun SNMP Management Agent for Sun Fire B1600

acl = {
  {
    communities = public, private
    access = read-only
    managers = rag, tag, bobtail
  }
  {
    communities = tigger
    access = read-write
    managers = brittas
  }
}

trap = {
  {
    trap-community = tigger
    hosts = brittas
  }
}
```

此陷阱群組可定義 SNMPv2c 通知發送的目的地。

spapm_snmpdx.acl 檔

在預設配置中，中介程式會以 snmpdx 的子代理程式之身份執行。您可以修改本檔案以根據來源主機名稱啟動存取。社群和存取權必須符合 spama.acl 檔中的社群和存取權。

檔案中的 acl 群組可指定特定社群和管理員的存取權。此群組由社群配置清單所組成，並使用以下格式：

```
# {
#     communities = < 社群清單 >
#     access = < 存取權 >
#     managers = < 主機清單 >
# }
```

- *社群清單* 處為此存取控制套用的社群名稱清單，其以逗點分隔。
- *存取權* 處為指定授與 *主機清單* 中之管理員的權限。
- *主機清單* 處為被授與特定 *存取權* 之主機名稱清單，其以逗點分隔。

在程式範例 11-5 的第一個範例中，rag、tag 和 bobtail 系統是為 public 和 private 社群上的 read-write 存取所配置。brittas 系統是為 tigger 社群的 read-write 存取所配置。

第二個範例而適用於使用 SNMPv3 主代理程式（spama.conf 中的 START_MASTER_AGENT=yes）的配置，可看見 localhost 上 snmpdx 接收 SNMP 封包之處。唯讀存取是為 public 和 private 社群所配置，而讀寫存取則是為 tigger 社群所配置。以上社群皆由主代理程式的 SNMPv3 內容所對映。因此，此存取必要的 SNMPv3 內容就必須於本檔中指定一個與其相對應之社群。

程式範例 11-5 spapm_snmpdx.acl 檔範例

```
# @(#)spapm_snmpdx.acl1.8 03/01/29 SMI
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# Template snmpdx Access Control file for Sun SNMP Management Agent for Sun
# Fire B1600
#
# Example 1:
#
# acl = {
#     {
#         communities = public, private
#         access = read-only
#         managers = rag, tag, bobtail
#     }
# }
```

```
# {
#     communities = tigger
#     access = read-write
#     managers = rag, tag, bobtail
# }
# }
#
# Example 2:
#
acl = {
    {
        communities = public, private
        access = read-only
        managers = localhost
    }
    {
        communities = tigger
        access = read-write
        managers = localhost
    }
}
#
# Trap destinations are defined in spapm.acl and spama.acl.
# This entry does not need to be edited.
trap = {
}
```

主代理程式配置檔

當您啟動主代理程式功能（在 `spama.conf` 的 `START_MASTER_AGENT=yes` 中）時才可配置這些檔案。

本節將說明以下配置檔的格式：

- 第 95 頁的「`spama.acl` 檔」
- 第 95 頁的「`spama.uacl` 檔」
- 第 96 頁的「`spama.security` 檔」

spama.acl 檔

spama.acl 檔可定義主代理程式的存取控制。若要瞭解此檔案的一般格式，請參閱第 89 頁的「ACL 檔格式」。本節包含 spama.acl 檔的特定相關資訊。

根據預設值，此檔案位於 /etc/opt/SUNWspa 中。

若要啟動 SNMP 配接卡的存取控制和陷阱，當代理程式啟動時須確認有 ACL 檔。因為 ACL 檔包含安全性的相關資訊，所以其指定的限制存取權只能由 root 讀取。

此檔案可定義 SNMPv1 和 SNMPv2c 的存取權限，以及 SNMP 通知的接收者。若 spama.conf 中為 SPAPM_TRAPS_ARE_V3=yes，陷阱就會以 SNMPv3 陷阱發送出去，不然就是以 SNMPv2c 陷阱發送出去。

acl 群組

若您正在使用 SNMPv3，您可能會想要限制 SNMPv1 和 SNMPv2c 的寫入存取。因此，acl 只允許唯讀存取。

程式範例 11-6 acl 群組範例

```
acl = {
    {
        communities = public, private
        access = read-only
        manager = localhost
    }
}
```

trap 群組

此檔案的 trap 群組格式說明請參閱第 90 頁的「trap 群組」。

spama.uacl 檔

當主代理程式啟動時，此檔案會與 spama.security 檔搭配使用以啟動 SNMPv3 安全性功能。若要瞭解此檔案的一般格式，請參閱第 89 頁的「ACL 檔格式」。本節還將說明其他的配置參數。程式範例 11-7 為本檔案的範例，其已移除大部分的註解以求簡潔明白。

根據預設值，此檔案位於 /etc/opt/SUNWspa 中。

acl 群組

acl 群組包含以下參數：

- *內容名稱* — 此為以逗點分隔之內容名稱清單。
- *存取* — 可能使用的值為：
 - read-only
 - read-write
- *安全等級* — 可能使用的值為：
 - noAuthNoPrivacy
 - authNoPrivacy
 - authPrivacy
- *使用者* — 此為以逗點分隔之使用者名稱清單。

在以下範例中，已授與存取權給 defaultUser，並同時授與 public 和 null 中來自 defaultUser 的任何要求以及 authNoPrivacy 的最小安全性。其他所有 SNMP 要求都會被退回。

本檔案中沒有 trap 群組。

程式範例 11-7 spama.uacl 檔範例

```
#ident "@(#)spama.uacl 1.4 01/29/03 SMI"
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# This software is the proprietary information of Sun Microsystems, Inc.
# Use is subject to license terms.
#
# Template ACL file
#
acl = {
  {
    context-names = public,null
    access = read-write
    security-level=authNoPriv
    users = defaultUser
  }
}
```

spama.security 檔

spama.security 檔可指定允許存取至主代理程式、SNMPv3 加密與授權金鑰的使用者。

本檔案是由 userEntry 列集所組成，並使用以下格式：

```
userEntry=< 引擎 ID>,< 使用者名稱>,< 安全性名稱>,< 授權演算法>,< 授權金鑰>,< 保密演算法>,< 保密金鑰>,< 儲存類型>,< 範本>
```



警告 - 請勿編輯本檔案 userEntry 列以外的參數。

請見表 11-2 得知這些欄位的說明。

表 11-2 spama.security 中的使用者可配置參數

參數	說明
引擎 ID	使用的 SNMP 引擎 ID。此 ID 可能為： <ul style="list-style-type: none">■ 十六進位字串■ 使用 <address>:<port>:<IANA number> 格式的 engineID 文字字串■ localEngineID 字串適用於大部份的配置
使用者名稱	此項目套用的使用者名稱
保密性名稱	對映至使用者名稱的安全性名稱。一般而言，這兩個名稱應該是相同的。
授權演算法	使用的授權演算法。可能會使用以下演算法其中之一： <ul style="list-style-type: none">• usmHMACMD5AuthProtocol• usmHMACSHHAuthProtocol• usmNoAuthProtocol
授權金鑰	與授權演算法搭配使用的金鑰。可能會使用以下其中一把金鑰： <ul style="list-style-type: none">• 文字密碼（最少八個字元）• 已本土化的十六進位金鑰，例如： 0x0098768905AB67EFAA855A453B665B12
保密演算法	使用的保密演算法。可能會使用以下演算法其中之一： <ul style="list-style-type: none">• usmDESPrivProtocol• usmNoPrivProtocol（若沒指定時的預設值）
保密金鑰	與保密演算法搭配使用的金鑰。可能會使用以下其中一把金鑰： <ul style="list-style-type: none">• 文字密碼（最少八個字元）• 已本土化的十六進位金鑰，例如： 0x0098768905AB67EFAA855A453B665B12
儲存類型	唯一可接受的數值為 3（若沒指定時的預設值）
範本	預設值為 false（無需變更此值）

程式範例 11-8 為本檔案的範例，其已移除大部分的註解以求簡潔明白。

根據預設值，此檔案位於 `/etc/opt/SUNWspa` 中。

預設的 `spama.security` 檔案包含兩個範例使用者，您可以修改並取消註解來定義您自己的使用者。首先指定的使用者名稱為 `defaultUser`，再加上：

- 僅使用 MD5 的授權演算法
- 無保密性
- 授權密碼 "mypassword"

之後指定的使用者名稱為 `defaultUser`，再加上：

- 使用 MD5 的授權演算法和授權密碼 "mypassword"
- 使用 DES 的保密演算法和保密密碼 "mypassword"
- 使用 DES 的保密演算法

程式範例 11-8 `spama-security` 檔範例

```
#ident "@(#)spama.security 1.7 01/29/03 SMI"
#
# Copyright 2002 Sun Microsystems, Inc. All rights reserved.
# This software is the proprietary information of Sun Microsystems, Inc.
# Use is subject to license terms.
#
# Template security file

# localEngineBoots=0

# defaultUser configuration. Authentication only.
# userEntry=localEngineID,defaultUser,,usmHMACMD5AuthProtocol,mypasswd

# defaultUser configuration. Authentication and encryption.
# userEntry=
localEngineID,defaultUser,null,usmHMACMD5AuthProtocol,mypasswd,usmDESPrivProt
ocol,mypasswd,3,
```

第十二章

配置軟體

本章將說明安裝完畢後的預設配置，並於第十一章說明如何修改檔案。

本章包含以下內容：

- 第 99 頁的「預設配置」
 - 第 100 頁的「手動配置直接存取」
 - 第 101 頁的「中介程式和 SNMPv3 主代理程式」
-

預設配置

本軟體會安裝以下預設配置：

- 關閉主代理程式 (`START_MASTER_AGENT=no`)。
- 啟動中介程式 (`START_MEDIATOR_AGENT=yes`)。
- 中介程式會配置為 `snmpdx` 子代理程式。

注意 - 基於安全考量，您應該配置 `snmpdx ACL` 檔來限制除了監視代理程式以外的其他系統存取。

存取控制

若要啟動中介程式的存取控制，請配置中介程式 `ACL` 檔。請參閱第 91 頁的「`spapm.acl` 檔」內的說明。

若使用 `snmpdx`（預設配置），請修改 `spapm_snmpdx.acl` 來設定存取權限，和修改 `spapm.acl` 來設定陷阱接收者。

啓動和停止中介程式

使用一般 `snmpdx` 啓動 Script 來啓動中介程式：

```
# /etc/init.d/init.snmpdx start
```

使用中介程式 Script 停止中介程式：

```
# /etc/init.d/spama stop
```

手動配置直接存取

既然 `snmpdx` 只支援 SNMPv1，若您想要使用 SNMPv2c 特定 `get-bulk` 操作且不想使用主代理程式，您可以配置中介程式使用的連接埠來啓動直接 SNMPv2c 存取。

若要手動配置中介程式，請在 `spama.conf` 中進行以下變更：

1. 設定 `SUB_AGENT=no`。
2. 將 `SPAPM_REQ_PORT` 設定為指定的連接埠編號。
傳送至中介程式的 SNMPv2c 要求必須傳送至該連接埠。

設定為第三協力廠商主代理程式之子代理程式的中介程式

若要將中介程式配置為第三協力廠商之子代理程式，其可使用指令行參數支援連接埠編號規格：

1. 配置中介程式 ACL 檔以允許從 `localhost` 進行存取（請參閱第 91 頁的「`spapm.acl` 檔」）。
2. 配置主代理程式，再搭配以下呼叫和適當的連接埠編號來啓動中介程式：

```
/etc/init.d/spama start <連接埠>
```

3. 配置主代理程式來轉寄要求至以下 OID 子樹狀圖：
 - `.iso.org.dod.internet.mgmt.mib-2.entityMIB`

- .iso.org.dod.internet.private.enterprises.sun.products.sunFire.sunPlatMIB

或使用其相對數字轉寄：

- .1.3.6.1.2.1.47
- .1.3.6.1.4.1.42.2.70.101

中介程式和 SNMPv3 主代理程式

若要啟動中介程式和主代理程式，您最起碼必須進行以下變更：

1. 在 `spama.conf` 檔案中：
 - a. 設定 `START_MASTER_AGENT=yes`。
 - b. 設定 `SUB_AGENT=no`。
2. 配置中介程式 ACL 檔以從 `localhost` 進行存取（請參閱第 91 頁的「`spapm.acl` 檔」中的說明）。
3. 配置 `snmpdx` 以從 `localhost` 進行存取（請參閱第 93 頁的「`spapm_snmpdx.acl` 檔」中的說明）。
4. 配置主代理程式 ACL 檔以從想要的管理員進行存取（請參閱第 95 頁的「`spama.acl` 檔」中的說明）。
5. 配置安全檔來定義 SNMPv3 使用者、使用內容、授權以及加密等級（請參閱第 95 頁的「`spama.acl` 檔」和第 96 頁的「`spama.security` 檔」中的說明）。

啟動和停止代理程式

使用中介程式 Script 來啟動中介程式和主代理程式：

```
# /etc/init.d/spama start
```

使用中介程式 Script 來停止中介程式和主代理程式：

```
# /etc/init.d/spama stop
```

轉寄 SNMPv3 陷阱

若要配置主代理程式從中介程式來轉寄 SNMPv3 陷阱，在 `spama.conf` 檔案（請參閱第 80 頁的「`spama.conf`」）中：

1. 設定 `SPAPM_TRAPS_ARE_V3=yes`。
2. 設定 `SNMPV3_USER`（選擇使用）。

注意 - 陷阱使用者必須被配置為 `spama.uacl` 和 `spama.security` 中的 SNMPV3 使用者。

解除軟體安裝

本章將說明如何解除此軟體安裝。

一般而言，若要解安裝 SNMP，就必須使用 `pkgrm` 指令來移除安裝的套裝軟體。此程序將移除所有相關檔案和連結，並重新啟動 `snmpdx`。

由 SNMP 軟體自動進行的配置變更也將恢復回其原始狀態。但若您有修改任何外部檔案設定，例如 `snmpdx` ACL 檔，則在移除 SNMP 軟體後，您就必須手動復原其狀態。

注意 - 以下列出的程序不會解除 Java SNMP API 套裝軟體 `SUNWjsnmp` 的安裝。若您想要解除此套裝軟體的 Solaris 版本安裝，您必須先移除 Java SNMP API。

平台代理程式和目標代理程式套裝軟體

若要從平台代理程式伺服器移除平台代理程式套裝軟體，請鍵入：

```
# pkgrm SUNWbgpnr SUNWbgpm SUNWjdrdt SUNWjsnmp SUNWbgpjo \  
SUNWbgodr SUNWbgod SUNWbgcmr SUNWbgcm SUNWbgpc SUNWbgptk
```



警告 - 請小心移除 `SUNWjdrdt` 和 `SUNWjsnmp` 套裝軟體，因為這二個套裝軟體皆屬於系統套裝軟體，而且別的產品也可能有使用到這二個套裝軟體。

若要從 Sun Fire B100s Blade 移除目標平台套裝軟體，請鍵入：

```
# pkgrm SUNWbgpr SUNWbgcm SUNWbgpc SUNWbgptk
```

領域代理程式套裝軟體

若要從 Sun Fire B100s Blade 移除領域代理程式套裝軟體，請鍵入：

```
# pkgrm SUNWbgpnr SUNWbgpm SUNWjdrdt SUNWjsnmp SUNWbgpji \  
SUNWbgidr SUNWbgcmr SUNWngcm SUNWbgpc SUNWbgptk
```



警告 - 請小心移除 SUNWjdrdt 和 SUNWjsnmp 套裝軟體，因為這二個套裝軟體皆屬於系統套裝軟體，而且別的產品也可能有使用到這二個套裝軟體。

疑難排解

本章將提供可幫助您解決與系統相關問題的資訊。

問題

- 使用預設配置 (snmpdx) 時 SNMP 代理程式沒有回應。

1. 鍵入以下指令確認是否已啟動中介程式：

```
# ps -ef | grep spa.snmp
root 15789      1  1 13:44:01 pts/2      0:00 /usr/j2se/bin/java
-Dcom.sun.spa.snmp.LOG_LEVEL=INFO -Djdk.security.file=//etc
```

如果輸出回應與上述範例類似，就表示中介程式已啟動。

鍵入以下指令來停止和重新啟動中介程式：

```
# /etc/init.d/spama stop
# /etc/init.d/init.snmpdx start
```

2. 鍵入以下指令確認是否已安裝正確的 Java 版本：

```
# /usr/j2se/bin/java -version
java version "1.4.1_03"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1_03-b04)
Java HotSpot(TM) Client VM (build 1.4.1_03-b04, mixed mode)
```

回報版本應為 1.4 或更新的版本。如果不是上述版本，請安裝 Java 1.4 JDK（請參閱第 61 頁的「Java 環境」）。

3. 鍵入以下指令確認是否已安裝正確的 SUNWjsnmp 版本：

```
# pkginfo -l SUNWjsnmp | grep VERSION
VERSION: 5.0
```

若出現的版本為 1.0，請移除 SUNWjsnmp 套裝軟體並重新安裝此版本於 SUNWspa.*.tar.Z 檔案中（請參閱第 63 頁的「Java SNMP API」）。

4. 確認 spama.conf 檔案包含以下項目：

```
START_MASTER_AGENT=no
START_MEDIATOR=yes
SUB_AGENT=yes
```

5. 鍵入以下指令確認中介程式已使用 snmpdx 正確登錄：

```
# cat /var/snmp/snmpdx.st
spapm spapm 2516 34050
snmpd snmpd 2567 34053
```

上述 spapm 項目顯示中介程式已登錄為 snmpdx 子代理程式。

6. 確認 /etc/snmp/conf/spapm.reg 和 /etc/snmp/conf/spapm.rsrc 沒有損壞。

鍵入以下指令來停止和重新啟動中介程式：

```
# /etc/init.d/spama stop
# /etc/init.d/init.snmpdx start
```

7. 確認 ACL 檔的權限設定正確。

- spapm_snmpdx.acl 可定義使用的 SNMP 管理員存取。
- spapm.acl 可定義 localhost 的存取。

若要瞭解詳細資訊，請參閱第十一章。

問題

- 使用平台代理程式時，沒有硬碟 (HDD) 或乙太網路 MAC 位址設備。
只有作業環境執行於目標 Sun Fire B100s Blade 時才會出現此資訊。
1. 確認 Sun Fire B100s Blade 已啟動。

2. 請鍵入以下指令確認目標設備執行於 Sun Fire B100s Blade 中：

```
# netstat -an | grep 1099
*.1099          *.*            0            0 24576        0 LISTEN
```

如果連接埠沒有聽取，就表示目標設備沒有在執行。

鍵入以下指令在 Sun Fire B100s Blade 中啟動設備：

```
# /etc/init.d/spardp start
```

3. 鍵入以下指令確認是否已安裝正確的 Java 版本：

```
# /usr/j2se/bin/java -version
java version "1.4.1_03"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1_03-b04)
Java HotSpot(TM) Client VM (build 1.4.1_03-b04, mixed mode)
```

回報版本應為 1.4 或更新的版本。版本不正確將導致設備啟動後，不久即宣告失敗。

如果不是上述版本，請安裝 Java 1.4 JDK（請參閱第 61 頁的「Java 環境」）。

問題

- 可以存取代理程式，但沒有受監視平台使用的設備。

1. 鍵入以下指令確認是否已啟動搜尋常駐程式：

```
# netstat -a | grep mism
*.mismi          *.*            0            0 24576        0 LISTEN
*.mismi          *.*            0
0 24576          0 LISTEN
```

以上輸出表示搜尋常駐程式正從管理平台聽取要求。

a. 確認 `/etc/services` 包含以下項目：

```
mismi            8265/tcp        # MISMI Discovery
```

b. 確認 `/etc/inetd.conf` 包含以下項目：

```
# MISMIDISCOVERY - mismiDiscovery daemon
mismi stream tcp6 nowait root /opt/SUNWspa/bin/mismiDiscovery
mismiDiscovery
```

c. 鍵入以下指令確認 `/etc/inetd.conf` 為至 `/etc/inet/inetd.conf` 的符號連結：

```
# ls -l /etc/inetd.conf
lrwxrwxrwx 1 root root 17 Jan 7 17:08 /etc/inetd.conf ->
./inet/inetd.conf
```

若此連結不存在，在安裝 `SUNWbgodr` 套裝軟體的過程中將無法更新檔案。

鍵入以下指令修正 `inetd` 配置並重新啓動：

```
# pkill -l inetd
```

2. 鍵入以下指令確認已搜尋到平台：

```
# netstat -a | grep mismi
*.mismi *.* 0 0 24576 0 LISTEN
blade-174-119.36780 hornet-sc.mismi 8192 0 24820 0 ESTABLISHED
*.mismi *.* 0
0 24576 0 LISTEN
```

以上輸入顯示搜尋常駐程式正在進行聽取作業，而且也已建立與平台系統控制器（即為 `<hornet-sc>`）的連線。

如果沒有建立任何連線，請檢查系統控制器設定（請參閱第 74 頁的「配置系統控制器」中的說明）。

問題

- SNMPv3 `get` 和 `set` 要求逾時。
 - 可能原因

`spama.security` 檔案內的 `localEngineId` 或 `localEngineBoots` 編號已遭編輯或刪除。

- 檢查

您無法直接得知檔案是否已遭編輯。

- 修改

如下所示來重新啓動代理程式，然後重新啓動管理應用程式進行同步處理更新。

```
# /etc/init.d/spama stop
# /etc/init.d/spama start
```

問題

- SNMP get 和 set 要求逾時。

- 可能原因

在負載過大的系統中，snmpdx 主代理程式很有可會讓至 SNMP 中介程式的要求逾時。逾時的目前設定為 2s (2000000 s)。

- 檢查

您無法直接得知管理應用程式回報的逾時是否發生於 snmpdx 和 SNMP 中介程式之間，或發生於管理應用程式和 snmpdx 之間。

- 修改

您可以編輯 /etc/snmp/conf/spapm.reg 檔案的逾時屬性來增加逾時。若您編輯此檔案，請鍵入以下指令重新啓動中介程式：

```
# /etc/init.d/spama stop
# /etc/init.d/init.snmpdx start
```


安裝 J2RE 1.4 並與 J2SE 1.3.1 共存

本附錄將說明如何於平台代理程式伺服器 and B100s 領域安裝 Java 2 Runtime Environment (J2RE) Standard Edition 1.4 並與 J2SE 1.3.1 共存，以及如何修改啓動 Script 以連至本安裝的方法。

本附錄包含以下內容：

- 第 111 頁的「安裝 J2RE 1.4」
- 第 113 頁的「編輯啓動 Script」

安裝 J2RE 1.4

若要安裝 J2RE 1.4 並與 J2SE 1.3.1 共存（如第 61 頁的「Java 環境」所述），請遵照以下程序進行安裝：

J2RE 1.4 爲一可自行解壓縮的二進位檔案，並可從以下網站取得：

<http://java.sun.com/j2se/1.4/download.html>

請遵照以下指示來安裝 J2RE。上述網站還包含了下載檔案的相關資訊。

注意 – 本產品僅需要 32 位元支援，所以不需要安裝 J2RE 的 64 位元補充資料。

在以下步驟中，鍵入合適的 <ver> J2RE 更新版本編號。

例如，若您下載的是更新版 1.4.0_01，以下指令：

```
# chmod +x j2re-1_4_<version>-solaris-sparc.sh
```

就會變成：

```
chmod +x j2re-1_4_0_01-solaris-sparc.sh
```

1. 下載並檢查檔案大小

需檢查的檔案為：

```
j2re-1_4_<ver>-solaris-sparc.sh
```

在下載檔案前，記錄下載網頁上提供的檔案大小。下載結束後，檢查下載的檔案大小是否完整，而且沒有任何損壞。

確認您將檔案下載至 root 可以存取的位置（例如 /tmp 中）。

2. 執行 su 變成 root，然後輸入超級使用者密碼。

3. 確認已在自行解壓縮二進制檔案上設定好執行權限。

```
# chmod +x j2re-1_4_<ver>-solaris-sparc.sh
```

4. 變更至檔案安裝的目錄。

```
# cd /usr
```

5. 執行自我解壓縮二進制檔案。

接著便會建立一個名為 /usr/j2re1.4.<ver> 的目錄，目錄內即包含 J2RE。

6. 確認 J2RE 安裝正確。

```
# /usr/j2re1.4.1_01/bin/java -version
java version "1.4.1_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1_01-
b01)
Java HotSpot(TM) Client VM (build 1.4.1_01-b01, mixed mode)
```

回報版本應為 1.4 或更新的版本。此範例顯示安裝版本為 1.4.1_01。

7. 刪除自我解壓縮二進位檔案。

8. 退出 root shell。

編輯啓動 Script

本節將說明如何在安裝 J2RE 1.4 時修改啓動 Script 。

請參閱本節與第十章內的說明。

領域硬體監視

這些步驟與第 69 頁的「安裝領域硬體監視軟體」之步驟 4 有關。

1. 在每個受監視 B100s Blade 上，編輯以下啓動 Script：

- /etc/init.d/spaibdm
- /etc/init.d/spapom

將以下文字行

```
JAVA=/usr/j2se/bin/java
```

取代為

```
JAVA=/usr/j2re1.4.<ver>/bin/java
```

2. 在每個受監視 B100s Blade 上，編輯以下啓動 Script：

- /etc/init.d/spama

將以下文字行

```
JAVA_JAVA=/usr/j2se/bin/java
```

取代為

```
JAVA_JAVA=/usr/j2re1.4.<ver>/bin.java
```

平台硬體監視

步驟 1 和步驟 2 與第 70 頁的「使用目標設備安裝軟體」之步驟 4 和第 72 頁的「不使用目標設備安裝軟體」之步驟 4 有關。

步驟 3 只與第 72 頁的「不使用目標設備安裝軟體」之步驟 11 有關。

1. 在平台代理程式伺服器上，編輯以下啓動 Script：

- /etc/init.d/spapom

將以下文字行

```
JAVA=/usr/j2se/bin/java
```

取代為

```
JAVA=/usr/j2re1.4.<ver>/bin/java
```

2. 在平台代理程式伺服器上，編輯以下啓動 Script：
/etc/init.d/spama

將以下文字行

```
JAVA_JAVA=/usr/j2se/bin/java
```

取代為

```
JAVA_JAVA=/usr/j2re1.4.<ver>/bin.java
```

3. 在每個受監視 B100s Blade（目標）上，編輯以下啓動 Script：
 - /etc/init.d/spardp

將以下文字行

```
JAVA=/usr/j2se/bin/java
```

取代為

```
JAVA=/usr/j2re1.4.<ver>/bin/java
```

索引

A

ACL, 12, 75, 78, 88, 99, 100
 trap 群組, 90
 格式, 89
acl 群組, 89
 社群, 89
 管理員, 89
 範例, 90

C

CIM, 15
Common Information Model。請參閱 CIM

E

entityGeneral 群組, 18
entityLogical 群組, 18
entityMapping 群組, 18
ENTITY-MIB, 8, 17, 19, 22, 32
entityMIBTraps 群組, 18
entityPhysical 群組, 17
entLogicalTable, 22
entLPMappingTable, 22
entLPPhysicalIndex, 22
entPhysicalClass, 20, 21
entPhysicalContainedIn, 20
entPhysicalContainsTable, 20, 22

entPhysicalIndex, 20, 22
entPhysicalTable, 17, 19, 20, 21

G

get 指令, 6, 9

I

inetd.conf 檔案, 66
inetd 指令, 66
INTEGER 屬性值變更記錄 (Integer Attribute Value Change Record) 等級, 27

J

J2RE 1.4
 下載, 111
 安裝, 111
 確認安裝, 112
 編輯啓動 Script, 113
Java
 SNMP API, 63
 下載, 62
 確認安裝, 62
 確認安裝版本, 105, 107
 環境, 61

- L**
LED, 39
localhost, 81, 101
Log 表, 27
- M**
MIB, 6
 表格, 7
- N**
NMS, 6
- O**
OID, 7
OID 屬性值變更記錄 (OID Attribute Value Change Record) 等級, 27
- P**
Physical Entity 表, 18, 23
Physical Mapping 表, 18
- S**
setupsc 指令, 73
set 指令, 6, 9
SNMP, 6
 陷阱, 6
snmpdx, 4, 9, 11, 68
snmpdx(1M), 9
SNMPv1, 5, 11
SNMPv2c, 5
SNMPv3, 5, 11, 59, 76, 80
SNMPv3 主代理程式, 4
SNMP 代理程式
 疑難排解, 105
SNMP 管理軟體, 63
 安裝, 68
 套裝軟體內容, 64
Solaris 主代理程式, 4
spama, 12
spama.acl, 12, 95
spama.conf, 100, 102
 一般選項, 80
 中介程式選項, 82
 主代理程式選項, 81
 預設值, 84
spama.conf 中的一般選項, 80
spama.conf<Default Para Font, 80
spama.security, 96
 配置參數, 97
spama.security1, 12
spama.uacl, 12, 95
spapm.acl, 91, 99
spapm.rsrc, 82
spapm_snmpdx.acl, 93, 99
STRING 屬性值變更記錄 (String Attribute Value Change Record) 等級, 27
SUN-PLATFORM-MIB, 8, 17, 22, 23, 26
sunPlat 模式, 14
- T**
trap 群組
 trap-community, 91
 主機, 91
- 二畫**
二進位感應器 (Binary Sensor) 等級, 41
二進位感應器表 (Binary Sensor Table) 擴展, 23
- 三畫**
下載 Java, 62
子等級, 14

四畫

- 中介程式, 4, 6, 11, 17, 75, 78, 81
 - spama.conf 中的選項, 82
 - 手動配置, 100
 - 配置為子代理程式, 100
 - 停止, 100
 - 啟動, 100
 - 確認登錄, 106
- 介面選項
 - snmpdx 的 SNMP, 75
 - 使用主代理程式和 snmpdx 的 SNMP, 76
- 分離感應器 (Discrete Sensor) 等級, 43
- 分離感應器表 (Discrete Sensor Table) 擴展, 24
- 升級 SNMP 管理軟體, 64
- 手動配置, 100
- 日誌表, 26

五畫

- 主代理程式, 8, 11, 68, 75, 80, 99
 - spama.conf 中的選項, 81
 - 第三協力廠商, 78
- 代理程式, 6
 - 平台, 3
 - 領域, 3
- 可插拔之移動單元, 34
- 可置換之硬體資源, 34
- 平台代理程式, 3, 60, 67
- 平台代理程式套裝軟體
 - 移除, 103
- 平台物件管理員, 66
- 平台硬體監視, 60, 67
 - 安裝
 - 不使用目標監視, 72
 - 使用目標設備, 70
 - 編輯啟動 Script, 113
- 平台模式, 13
- 未知警報記錄 (Indeterminate Alarm Record) 等級, 28, 53
- 母等級, 14
- 目標, 68
- 目標平台代理程式套裝軟體

移除, 103

六畫

- 存取控制, 88, 99
- 存取控制清單 請參閱 ACL
- 存取權, 8
- 安全檔, 12
- 安裝 J2RE 1.4, 111
- 安裝套裝軟體, 63
- 安裝管理軟體, 68

七畫

- 序號, 31
- 系統控制器, 3, 60
 - 配置, 74
- 系統管理選項, 59
- 防火牆, 11

八畫

- 事件, 14, 26
- 事件記錄 (Event Record) 母等級, 51
- 事件記錄 (Event Record) 等級, 50
- 事件額外記錄 (Event Additional Record) 母等級, 51
- 定址物件, 8
- 服務品質警報記錄 (Quality of Service Alarm Record) 等級, 28, 54
- 物件刪除記錄 (Object Deletion Record) 等級, 27, 52
- 物件建立記錄 (Object Creation Record) 等級, 27, 52
- 物件辨識碼。請參閱 OID
- 狀態, 31
- 狀態變更記錄 (State Change Record) 等級, 27, 55
- 表
 - 擴展, 23
- 表格, 6
 - 定義, 7
 - 擴展, 7

九畫

風扇

特徵, 7

速度, 6

風扇 (Fan) 等級, 40

風扇表 (Fan Table) 擴展, 24

十畫

索引

說明, 7

欄, 7

配置

SNMP, 11

系統控制器, 74

設備, 67

預設, 99

管理介面, 67

檔, 12

十一畫

停止中介程式, 100

停止中介程式和主代理程式, 101

基座 (Chassis) 等級, 44

啓動 Script, 12, 83

啓動中介程式, 100

啓動中介程式和主代理程式, 101

處理錯誤警報記錄 (Processing Error Alarm Record) 等級, 28

處理警報記錄 (Processing Alarm Record) 等級, 54

設備, 3

配置, 67

設備 (Equipment) 等級, 33

設備支架 (Equipment Holder) 等級, 35

設備支架表 (Equipment Holder Table) 擴展, 23

設備表 (Equipment Table) 擴展, 23

設備警報記錄 (Equipment Alarm Record) 等級, 27, 54

軟體

修改, 32

錯誤, 54

警報, 39

軟體安裝

對系統檔案的影響, 65

通知, 9, 14, 18, 22, 27

通知 (Notification), 49

通知 (Notification) 等級, 49

通連字串, 8

連接埠, 8, 11, 12, 75, 76, 81, 82, 83, 100

陷阱, 6, 22, 26, 76, 82, 83, 88, 99

預設連接埠, 83

轉寄, 102

陷阱通知, 27

十二畫

單一電腦系統 (Unitary Computer System) 等級, 45, 48

測量單位, 42

硬體

資源, 13

類型, 32

硬體修改, 31

硬體資源

位置, 34

階層, 31

錯誤回報, 34

等級, 14

定義, 31

繼承, 14

韌體修改, 32

十三畫

傳送警報記錄 (Communications Alarm Record) 等級, 27, 53

感應器 (Sensor) 母等級, 40

感應器表 (Sensor Table) 擴展, 23

搜尋模組, 66

群組, 17

路由表, 6

逾時, 38

電池 (Battery) 等級, 37

- 電源供應器 (Power Supply) 等級, 36
- 電源供應器表 (Power Supply Table) 擴展, 24
- 電腦系統表 (Computer System Table) 擴展, 26
- 電路封包 (Circuit Pack) 等級, 34
- 電路封包表 (Circuit Pack Table) 擴展, 23
- 零件編號, 31

十四畫

- 實例辨識碼, 7
- 實際實體 (Physical Entity) 母等級, 31
- 實際實體 (Physical Entity) 表, 23
- 實際實體表 (Physical Entity Table), 20
- 實際模式, 19
- 實體表 (Physical Table) 擴展, 23
- 實體等級 (Physical Class), 20
- 實體對映表 (Physical Mapping Table), 20
- 疑難排解
 - ACL 權限, 106
 - get 和 set 要求, 108
 - Java 版本, 105, 107
 - SNMP 代理程式, 105
 - 中介程式登錄, 106
 - 平台搜尋, 108
 - 目標設備, 107
 - 搜尋常駐程式, 107
- 監視系統 (Watchdog) 等級, 38
- 監視系統表 (Watchdog Table) 擴展, 24
- 監視資料, 33
- 管理介面, 13
- 管理介面配置, 67
- 管理物件, 13, 14
- 管理員, 6
- 管理資訊資料庫。請參閱 MIB
- 管理領域 (Administrative Domain) 等級, 45, 48
- 網路通訊協定, 6
- 網路管理站。請參閱 NMS
- 網際網路標準, 5
- 製造商名稱, 31
- 語法

- acl 群組, 89
- trap 群組, 90
- 需求
 - Java 環境, 61
 - 作業環境, 61
 - 作業環境增補程式, 61
 - 磁碟空間, 61
- 領域, 68
- 領域代理程式, 3, 60, 67
- 領域代理程式套裝軟體
 - 移除, 104
- 領域硬體監視, 3, 60, 67
 - 安裝, 69
 - 編輯啓動 Script, 113

十五畫

- 數字感應器 (Numeric Sensor) 等級, 42
- 數字感應器表 (Numeric Sensor Table) 擴展, 24
- 數字感應器讀取值, 42
- 熱插拔事件, 52
- 編輯啓動 Script
 - 平台硬體監視, 113
 - 領域硬體監視, 113

十六畫

- 操作失敗, 54
- 機架, 34

十七畫

- 環境警報記錄 (Environmental Alarm Record) 等級, 27, 53

十八畫

- 簡易網路管理協定。請參閱 SNMP

十九畫

關係, 13

二十畫

繼承階層, 29, 30, 46

警報, 14, 26

警報 (Alarm) 等級, 39

警報表 (Alarm Table) 擴展, 24

警報記錄 (Alarm Record) 子等級, 52

警報嚴重性等級, 52

二十一畫

屬性, 14

屬性值變更記錄 (Attribute Value Change Record) 母等級, 54

二十三畫

邏輯 (Logical) 等級, 47

邏輯名稱, 31

邏輯等級擴展表 (Logical Class Extension Table), 26

邏輯實體 (Logical Entity) 等級, 45, 46

邏輯模式, 22