



Sun™ One Grid Engine 5.3 管理和使用者指南

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

文件編號：816-7467-10
2002年9月，修訂版A

請將您對本文件的意見寄到：docfeedback@sun.com

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 版權所有。

本產品或文件按照限制其使用、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 和 Solaris 是 Sun Microsystems, Inc. 在美國和其他國家的商標、註冊商標或服務標記。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。「能源之星」標誌是 EPA 的註冊商標。OPEN LOOK 和 Sun™ 圖形使用者介面是 Sun Microsystems, Inc. 為其使用者和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

前言 xv

本書編排架構 xv

使用 UNIX 指令 xvi

排印慣例 xvi

Shell 提示符號 xvii

相關說明文件 xvii

線上存取 Sun 出版文件 xvii

Sun 歡迎您的指教 xviii

部分 I. 背景和定義

1. Sun Grid Engine 5.3 簡介 1

什麼是格線運算？ 1

藉由管理資源和策略來管理工作量 2

系統作業的方式 2

 使資源與需求相符 2

 作業和佇列：Sun Grid Engine 世界 3

Sun Grid Engine 5.3 組件 3

 主機 3

 主要主機 4

執行主機	4
管理主機	4
提交主機	4
常駐程式	4
sge_qmaster – 主要常駐程式	4
sge_schedd – 排程器常駐程式	5
sge_execd – 執行常駐程式	5
sge_commd – 通訊常駐程式	5
佇列	5
客戶端指令	5
QMON, Sun Grid Engine 圖形使用者介面	7
自訂 QMON	8
Sun Grid Engine 專有名詞的詞彙	9
部分 II. 入門	
2. 安裝	13
快速啟動安裝簡介	13
快速啟動安裝的先決條件	14
安裝帳號	14
安裝目錄	14
▼ 如何建立安裝目錄	14
通訊埠編號	15
讀取分發媒體 (Distribution Media)	15
▼ 如何解開 Sun Grid Engine 5.3 分發	15
為您的叢機安裝預設的 Sun Grid Engine 系統	16
▼ 如何安裝主要主機	16
▼ 如何安裝執行主機	17

預設系統配置	17
完整安裝簡介	19
階段 1 – 計劃	19
階段 2 – 安裝軟體	20
階段 3 – 驗證安裝	20
計劃安裝	20
先決條件作業	20
安裝目錄 <code><sgc_root></code>	21
root 目錄下的多工緩衝處理目錄	21
目錄組織	21
磁碟空間需求	22
安裝帳號	23
檔案存取權限	23
網路服務	23
主要主機	23
備用主要主機	24
執行主機	24
管理主機	24
提交主機	24
儲存格	24
使用者名稱	24
佇列	25
▼ 如何計劃安裝	25
▼ 如何讀取分發媒體	26
▼ 如何安裝主要主機	27
▼ 如何安裝執行主機	28
▼ 如何安裝管理和提交主機	28

- 用加強的安全性安裝 29
 - 所需的額外設定 29
- ▼ 如何安裝和設定 CSP 安全化系統 (CSP-Secured System) 30
- ▼ 如何產生使用者證明文件和私密金鑰 39
- ▼ 如何檢查證明文件 40
 - 顯示證明文件 41
 - 檢查發出者 41
 - 檢查主題 41
 - 顯示證明文件的電子郵件 41
 - 顯示有效性 42
 - 顯示特點 42
- ▼ 如何驗證安裝 42

部分 III. 使用 Sun Grid Engine 5.3 軟體

- 3. 導覽 Sun Grid Engine 5.3 程式 49
 - Sun Grid Engine 使用者種類和操作 49
 - 佇列和佇列屬性 50
 - QMON 瀏覽器 50
 - ▼ 如何啓動 QMON 瀏覽器 51
 - 「佇列控制」QMON 對話方塊 51
 - ▼ 如何顯示佇列清單 51
 - ▼ 如何顯示佇列屬性 52
 - 使用 QMON 瀏覽器 52
 - 從指令行 54
 - 解譯佇列屬性資訊 54
 - 主機功能 54
 - ▼ 如何找出主要主機名稱 55

- ▼ 如何顯示執行主機清單 55
- ▼ 如何顯示管理主機清單 55
- ▼ 如何顯示提交主機清單 55
- 可要求的屬性 56
 - ▼ 如何顯示可要求屬性清單 57
 - 使用者存取權限 59
 - 管理員、操作員和所有人 60
- 4. 提交作業 61
 - 執行簡單作業 61
 - ▼ 如何從指令行執行簡單作業 61
 - ▼ 如何提交作業，從圖形使用者介面，QMON 63
 - 提交批次作業 67
 - 關於 Shell Script 67
 - Script 檔的範例 67
 - 使用 QMON 提交延伸和進階作業 68
 - 延伸範例 68
 - 進階範例 71
 - 資源需求定義 76
 - Sun Grid Engine 系統如何配置資源 78
 - 延伸至一般的 Shell Script 79
 - 如何選擇指令解譯器 79
 - 輸出重新導向 79
 - 作用中的 Sun Grid Engine 註釋 80
 - ▼ 如何從指令行提交作業 82
 - 預設要求 83
 - 陣列作業 84
 - ▼ 如何從指令行提交陣列作業 84

- ▼ 如何使用 QMON 提交陣列作業 85
- 提交互動作業 85
 - 使用 QMON 提交互動式作業 86
- ▼ 如何使用 QMON 提交互動作業 86
 - 使用 qsh 提交互動式作業 88
- ▼ 如何使用 qsh 提交互動作業 89
 - 使用 qlogin 提交互動式作業 89
- ▼ 如何使用 qlogin 提交互動作業 89
 - 平行作業 89
 - Sun Grid Engine 作業的排程方式 89
 - 作業優先權 90
 - 相同等份排程 (Equal-share-scheduling) 90
 - 佇列選擇 90
- 透明遠端執行 91
 - 使用 qrsh 遠端執行 91
 - qrsh 用法 91
 - 使用 qtcsh 的透明作業分配 92
 - qtcsh 用法 92
 - 帶有 qmake 的平行檔案製作過程 (Parallel Makefile Processing) 94
 - qmake 用法 95
- 5. 檢查點、監視和控制作業 97
 - 關於檢查點作業 97
 - 使用者等級檢查點 97
 - 核心等級檢查點 98
 - 檢查點作業的遷移 98
 - 組成檢查點作業 Script 98
 - ▼ 如何從指令行提交、監視或刪除檢查點作業 99

- ▼ 如何使用 QMON 提交檢查點作業 100
 - 檔案系統需求 101
- 監視和控制 Sun Grid Engine 作業 101
- ▼ 如何使用 QMON 監視和控制作業 101
 - QMON 物件瀏覽器的額外資訊 110
- ▼ 如何使用 qstat 監視作業 110
- ▼ 如何使用電子郵件監視作業 112
 - 從指令行控制 Sun Grid Engine 作業 113
- ▼ 如何從指令行控制作業 113
- 作業相依 114
- 控制佇列 114
- ▼ 如何用 QMON 控制佇列 114
- ▼ 如何用 qmod 控制佇列 118
- 自訂 QMON 119

部分 IV. 管理

- 6. 主機和叢集配置 123
 - 關於主要和備用主要配置 124
 - 關於常駐程式和主機 125
 - 關於配置主機 126
 - ▼ 如何以 QMON 配置管理主機 126
 - ▼ 如何刪除管理主機 128
 - ▼ 如何新增管理主機 128
 - ▼ 如何從指令行配置管理主機 128
 - ▼ 如何以 QMON 配置提交主機 128
 - ▼ 如何刪除提交主機 130
 - ▼ 如何新增提交主機 130

- ▼ 如何從指令行配置提交主機 130
- ▼ 如何以 QMON 配置執行主機 130
- ▼ 如何刪除執行主機 132
- ▼ 如何關閉執行主機常駐程式 132
- ▼ 如何新增或修改執行主機 132
- ▼ 如何從指令行配置執行主機 136
- ▼ 如何使用 qhost 監視執行主機 137
- ▼ 如何從指令行刪除常駐程式 138
- ▼ 如何從指令行重新啟動常駐程式 139

基礎叢集配置 139

- ▼ 如何從指令行顯示基礎叢集配置 139
- ▼ 如何從指令行修改基礎叢集配置 140
- ▼ 如何使用 QMON 顯示叢集配置 140
- ▼ 如何使用 QMON 刪除叢集配置 141
- ▼ 如何使用 QMON 顯示全域叢集配置 141
- ▼ 如何使用 QMON 修改全域和主機配置 142

7. 配置佇列與佇列行事曆 145

關於佇列配置 145

- ▼ 如何以 QMON 配置佇列 146
- ▼ 如何配置一般參數 147
- ▼ 如何配置執行法參數 148
- ▼ 如何配置檢查點參數 149
- ▼ 如何配置載入 / 暫停臨界值 150
- ▼ 如何配置限制 151
- ▼ 如何配置使用者聚合體 152
- ▼ 如何配置次級佇列 154
- ▼ 如何配置使用者存取 155

- ▼ 如何配置所有人 156
 - ▼ 如何從指令行配置佇列 156
 - 關於佇列行事曆 157
 - ▼ 如何以 QMON 配置佇列行事曆 158
 - ▼ 如何從指令行配置行事曆 160
8. 聚合體概念 163
- 關於聚合體 163
 - ▼ 如何新增或修改聚合體配置 164
 - 聚合體種類 165
 - 佇列聚合體 166
 - 主機聚合體 166
 - 全域聚合體 168
 - 使用者定義聚合體 168
 - 可耗損資源 172
 - ▼ 如何設定可耗損資源 172
 - 設定可耗損資源範例 174
 - 配置聚合體 182
 - ▼ 如何從指令行修改聚合體配置 182
 - qconf 指令範例 183
 - 載入參數 183
 - 預設載入參數 183
 - 新增站點特定載入參數 184
 - ▼ 如何自行撰寫載入感應器 184
 - 規定 184
 - Script 範例 185
9. 管理使用者存取與策略 189

關於設定使用者	189
關於使用者存取	190
▼ 如何以 QMON 配置帳號	191
▼ 如何以 QMON 配置管理員帳號	191
▼ 如何從指令行配置管理員帳號	192
可用的開關	192
▼ 如何以 QMON 配置操作員帳號	193
▼ 如何從指令行配置操作員帳號	194
可用的開關	194
關於佇列所有人帳號	194
關於使用者存取權限	195
▼ 如何以 QMON 配置使用者存取清單	195
▼ 如何從指令行配置使用者存取清單	197
可用的選項	197
關於排程	197
排程策略	198
佇列分類	198
作業分類	198
在排程器間隔時間內發生的事	198
排程器監視	199
預設排程	199
排程替代	199
▼ 如何以 QMON 配置變更排程器配置	202
關於路徑別名	205
檔案格式	206
如何解譯路徑別名檔	206
路徑別名檔之範例	206

關於配置預設要求	207
預設要求檔案之格式	207
預設要求檔案範例	208
關於收集運算與利用數據	208
關於檢查點支援	209
檢查點環境	210
▼ 如何以 QMON 配置檢查點環境	210
檢視配置的檢查點環境	211
刪除配置的檢查點環境	211
修改配置的檢查點環境	211
新增檢查點環境	213
▼ 如何從指令行配置檢查點環境	214
qconf 檢查點選項	214
10. 管理平行環境	215
有關平行環境	215
▼ 如何以 QMON 配置 PE	216
▼ 顯示 PE 的內容	216
▼ 刪除 PE	216
▼ 修改 PE	216
▼ 新增 PE	217
▼ 如何從指令行配置 PE	220
qconf PE 選項	220
▼ 如何從指令行顯示配置的 PE 介面	220
▼ 如何以 QMON 顯示配置的 PE 介面	220
PE 啟動程序	222
終止 PE	223
PE 與 Sun Grid Engine 軟體之緊密整合	223

11. 錯誤訊息 225

Sun Grid Engine 5.3 軟體如何取得錯誤回報 225

不同錯誤或退出代碼之結果 226

於除錯模式 (Debug Mode) 中執行 Sun Grid Engine 程式 228

前言

《*Sun Grid Engine 5.3 管理和使用者指南*》是一本提供產品背景資訊、安裝指示、以及如何完整使用產品的指示的綜合手冊。

本書編排架構

因為本指南是針對 Sun Grid Engine 5.3 產品的使用者和產品責任與使用者的產品責任不盡相同的系統管理者使用，所以本指南分為四部分。每個文件都包含對使用者和管理者有特殊利益的資訊。

這些部分和其目標讀者的說明如下。

- 第 1 部分 – 背景和定義

針對給使用者和管理員使用，《*Sun Grid Engine 5.3 管理和使用者指南*》這個部分對產品使用、組件、專有名詞等等有詳盡的解釋。

- 第 2 部分 – 入門

針對給安裝產品的人使用 — 一般來說是管理者 — 指南的這個部分對「快速啟動」、完整和升級安裝有詳細的指示。

- 第 3 部分 – 使用 Sun Grid Engine 5.3 軟體

指南的這個部分是針對給使用者和管理者使用。包含有涵蓋許多作業的指示和背景資訊。

- 第 4 部分 – 管理

指南的這個部分裡的背景資訊和指示是針對給有經驗的系統管理員使用。

使用 UNIX 指令

本文件可能不包含基本 UNIX® 指令及程序的相關資訊，如關閉系統、啓動系統及配置裝置等。

請參考下列資訊：

- 《Solaris Handbook for Sun Peripherals》
- 關於 Solaris™ 作業環境的 AnswerBook2™ 線上說明文件
- 其他隨著系統一起收到的軟體說明文件

排印慣例

字型	意義	範例
AaBbCc123	指令、檔案和目錄名稱；電腦螢幕輸出。	編輯您的 .login 檔案。 使用 <code>ls -a</code> 列出所有檔案。 % You have mail。
AaBbCc123	您鍵入的內容，與電腦螢幕輸出不同	% su Password:
<i>AaBbCc123</i>	書名、新字或專有名詞、要強調的文字。用真實名稱或數值替換指令行變數。	請閱讀《使用者指南》的第六章。 這些是類別選項。 您必須是超級使用者才能執行此項操作。 若要刪除某個檔案，請鍵入 <code>rm 檔案名稱</code> 。

Shell 提示符號

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

相關說明文件

適用文件	標題	文件編號
參考文件	《 <i>Sun Grid Engine 5.3 和 Sun Grid Engine</i> ，企業版 5.3 參考手冊》	816-7479-10

線上存取 Sun 出版文件

部分精選之 Sun 系統說明文件位於：

<http://www.sun.com/products-n-solutions/hardware/docs>

整套的 Solaris 說明文件及其他主題的說明文件則位於：

<http://docs.sun.com>

在該網站上，您也可以找到如何訂購本指南印刷版的相關資訊。

Sun 歡迎您的指教

Sun 一直致力於改善相關的說明文件，因此竭誠歡迎您提出批評和建議。您可以將意見透過電子郵件寄給 Sun：

`docfeedback@sun.com`

請將文件編號 (816-7467-10) 放到您 email 的主題中。

部分 I | 背景和定義

《*Sun Grid Engine 5.3 管理和使用者指南*》的這個部分包含單一章節：

- 第一章 – 第 1 頁的「Sun Grid Engine 5.3 簡介」。

本章很簡單扼要，可能會讓讀者誤解其對使用者和管理者的重要性，如果能熟悉本章內容，兩者的需求就能獲得滿足。本章包含的內容如下。

- 在複雜的計算環境內，對 Sun™ Grid Engine 5.3 軟體的主要角色說明
- 產品主要組件和每項組件的功能定義清單
- 在 Sun Grid Engine 5.3 環境中，須知的重要詞彙

Sun Grid Engine 5.3 簡介

本章提供關於 Sun Grid Engine 5.3 系統的背景資訊，該資訊對使用者和管理員同樣有用。除了說明產品在管理叢集化電腦之混亂世界的角色外，本章包含了以下的主題。

- 簡單說明格線計算
- QMON 的簡介，Sun Grid Engine 5.3 圖形使用者介面
- 每個產品重要組件的解釋
- 使用者和管理員可使用的詳細客戶端指令清單
- Sun Grid Engine 5.3 專有名詞的完整詞彙

什麼是格線運算？

格線在概念上是很簡單的。那是執行作業之計算資源的集合。對使用者來說像是大型的系統，其提供強力分配資源的單一存取點。使用者視格線為單一運算資源。資源管理軟體，如 Sun Grid Engine，會接受使用者所提出的工作，並會依照資源管理政策，排定在格線中、適當的系統上執行的時間。使用者可以一次提出數千個作業，而不用擔心它們執行的地方。

沒有兩個格線是相同的；一種尺寸不能符合所有的情況。共有三個主要格線類別，階層從單一系統到使用數千個處理器的超級電腦類別的運算中心 (compute farm)。叢集格線，由許多共同運作以提供單一專案或部門的單一點存取的運算資源所組成，為 Sun Grid Engine 5.3 系統協助您建立及管理之物。

(其種兩種更複雜的格線 — *campus* 格線 和 *全域* 格線 — 是由 Sun 的相關產品，Sun Grid Engine，企業版，所建立與管理。)

Sun Grid Engine 5.3 軟體會妥善安排以組織技術和管理人員設定之企業資源策略為基礎的運算能力的傳送。Sun Grid Engine 系統使用這些策略檢查格線裡的可用運算資源、收集這些資源，然後以將橫跨格線用法最佳化的方式自動配置並傳送資源。

藉由管理資源和策略來管理工作量

Sun Grid Engine 軟體提供使用者提交需要計算的工作予 Sun Grid Engine 系統做為相關工作量透明分配的方式。使用者可提交批次作業 (batch job)、互動作業和平行作業予 Sun Grid Engine 系統。

本產品也支援檢查點程式 (checkpointing program)。檢查點作業會在不受使用者載入需求的介入下在工作站間遷移。

對管理員來說，此軟體提供了各式各樣監視和控制 Sun Grid Engine 作業的工具。

系統作業的方式

Sun Grid Engine 系統能從外界接收工作 — 使用者對電腦資源的需求、把這些作業放置在等候處 (holding area) 等候執行、將作業從等候處傳送到執行裝置、在作業執行中進行管理，並在完成時記錄執行的紀錄。

想像世界上首都的大型「金錢中心」的銀行。

使資源與需求相符

在銀行建築物的大廳有很多顧客，每個都有不同的需求，每個都等著被服務。有個客戶只想從他的戶頭提出少量的金錢。在他之後來的另一個客戶跟銀行的一個投資專員有約；她希望在進行複雜的投資之前能夠獲得一些建議。在長長的隊伍之中，他們倆個的前面還有另一個客戶想要申請大型的貸款 — 如同在*她*前面的八個客戶一樣。

不同客戶和不同需求需要不同類型和層級的銀行資源。或許，在這特別的一天，銀行有許多員工有充足的時間來應付顧客從帳戶的簡單提款。但在同一天，只有一兩個貸款行員能夠協助處理許多貸款申請人。情況可能在別的日子裡會有不同。

影響當然是顧客必須等候服務 — 即使當中有許多人的需求符合可用資源而能夠獲得立即的服務。

如果 Sun Grid Engine 系統是銀行經理，它就會以不同方式組織服務。

- 在進入銀行大廳時，顧客就會被要求宣告其名稱、單位（如代表的公司）及其需求。
- 然後紀錄顧客到達的時間。
- 根據顧客在大廳所提供的資訊、需求和立即可用資源相符的顧客、需求有最高優先權的顧客，以及在大廳等候時間最久的顧客可以獲得服務。

- 當然，在「Sun Grid Engine 銀行」裡，一個銀行行員可以在同一時間為數個顧客提供服務。Sun Grid Engine 系統可能會把新的顧客分派到工作量最少且最適合的銀行行員。

作業和佇列：Sun Grid Engine 世界

在 Sun Grid Engine 系統中，*作業* 對應銀行顧客，作業在電腦等候處而不是在大廳等候，而位在電腦伺服器的 *佇列* 則取代銀行行員的位置，並提供服務給作業。就如同類比中銀行顧客的例子，每個工作的需求 — 通常包含了可用的記憶體、執行速度、可用軟體版權和類似需求 — 可能大不相同，而僅有某些佇列可以提供對應的服務。

對應此類比，Sun Grid Engine 軟體會以以下的方式對可用資源和工作需求進行仲裁 (arbitrate)。

- 透過 Sun Grid Engine 系統提交工作的使用者會宣告作業的需求簡介。此外，使用者的身分和其 *專案* 或 *使用者群組* 的單位會由系統取得。使用者提交作業的時間也會儲存起來。
- 在佇列排定給新作業執行的時程時，Sun Grid Engine 系統會判定適合佇列的作業，並會立刻以最高的優先權或最長的等候時間分配作業。
- Sun Grid Engine 佇列可允許許多作業同時執行。Sun Grid Engine 系統會嘗試在負荷最少和最適合的佇列啟動新作業。

Sun Grid Engine 5.3 組件

圖 1-1 一圖顯示最重要的 Sun Grid Engine 組件和在系統內的互動關係。以下各節解釋組件的功能。

主機

有四類型的主機對 Sun Grid Engine 5.3 系統是十分重要的。

- 主要
- 執行
- 管理
- 提交

主要主機

主要主機是全部叢集活動的中心。它會執行主要常駐程式，`sge_qmaster`，和排程器常駐程式，`sge_schedd`。這兩個常駐程式控制所有的 Sun Grid Engine 組件，如佇列和作業，並維護組件狀態、使用者存取權等等的表格。

根據預設值，主要主機也是管理主機和提交主機。請參閱與這些主機相關的各節。

執行主機

執行主機是擁有執行 Sun Grid Engine 作業權限的節點。因此，它們管理 Sun Grid Engine 佇列並執行 Sun Grid Engine 執行常駐程式，`sge_execd`。

管理主機

主機能被賦予執行 Sun Grid Engine 系統任何種類之管理活動之權限。

提交主機

提交主機只對批次作業的提交和控制做考慮。尤其是，登入提交主機的使用者可以透過 `qsub` 提交作業、透過 `qstat` 控制作業狀態，還可以使用 Sun Grid Engine OSF/1 Motif 圖形使用者介面，`QMON`，在第 7 頁的「`QMON`，Sun Grid Engine 圖形使用者介面」一節中有其說明。

注意 – 主機在上述的類別中可以歸類到一個以上的類別。

常駐程式

四個常駐程式提供 Sun Grid Engine 5.3 系統的功能性。

`sge_qmaster` – 主要常駐程式

叢集管理和排程活動的中心，`sge_qmaster` 會維護主機、佇列、作業、系統負載和使用者權限的表格。它會從 `sge_schedd` 接收排程決定，而從適當執行主機上的 `sge_execd` 接收需求活動。

sge_schedd – 排程器常駐程式

排程常駐程式在 `sge_qmaster` 的協助下維護最新叢集狀態的視區。它會做出以下的排程決定：

- 哪項作業分派到哪個佇列

然後它會將這些決定轉寄到 `sge_qmaster`，其會初始啓動所要求的活動。

sge_execd – 執行常駐程式

執行常駐程式負責主機上的佇列和這些佇列中作業的執行。它會定期將資訊，如作業狀態或主機上的負載，轉寄到 `sge_qmaster`。

sge_commd – 通訊常駐程式

通訊常駐程式在著名的 TCP 連接埠上通訊。它用於所有 Sun Grid Engine 組件之間的通訊。

佇列

Sun Grid Engine 佇列是允許在特別主機上同時執行的作業類別的容器 (container)。佇列決定某些作業屬性；例如，是否能夠遷移。執行作業在它們的生命週期期間與佇列有關。與佇列的關聯性會影響可能發生於作業的某些事項。例如，如果佇列已經暫停，所有與該佇列相關的作業也會跟著暫停。

在 Sun Grid Engine 系統中，您不需將作業直接提交給佇列。您只需要指定作業的需求簡介（例如，記憶體、作業系統、可用軟體等等），而 Sun Grid Engine 軟體會自動將作業分派到低負載主機上的合適佇列。如果作業提交到特別的佇列，則作業會受限於此佇列和其主機，因此 Sun Grid Engine 常駐程式將無法選擇低負載或較合適的裝置。

客戶端指令

Sun Grid Engine 的指令行使用者介面是您可以讓您管理佇列、提交及刪除作業、檢查作業狀態和暫停 / 啓動佇列和作業的輔助程式（指令）集。Sun Grid Engine 系統利用以下的輔助程式集。

- `qacct` – 此指令會從叢級日誌檔案取得仲裁計算 (arbitrary accounting) 的資訊。
- `qalter` – 此指令會變更已經提交但在擱置狀態的作業的屬性。
- `qconf` – 此指令提供叢集和佇列配置使用者介面。
- `qdel` – 此指令提供使用者、操作者或管理者傳送信號給作業或其子集的方法。
- `qhold` – 此指令阻止已提交的作業執行。

- `qghost` – 此指令顯示 Sun Grid Engine 執行主機的狀態資訊。
- `qlogin` – 此指令以自動選擇低負載和合適的主機來初始啟動 `telnet` 或類似的登入階段作業。
- `qmake` – 此指令是標準 `make` 功能的替代指令。它會依照其功能而延伸 `make`，好將獨立的 `make` 步驟分配到合適機器的叢集。
- `qmod` – 此指令能使所有人暫停或啟動佇列（也會將信息傳到所有目前與此佇列有關，且在作用中的過程）。
- `qmon` – 此指令提供 X-windows Motif 指令介面和監視功能。
- `qresub` – 此指令藉由複製執行或擱置的作業來建立新的作業。
- `qrls` – 此指令會將作業從之前指定的暫停狀態中釋放出來；例如，透過 `qhold`（請見上述）。
- `qrsh` – 此指令可以用於多種用途如下。
 - 欲透過 Sun Grid Engine 系統提供互動應用程式的遠端執行 — 相當於標準 UNIX 功能，`rsh`
 - 欲考慮提交再執行時支援終端機 I/O（標準 / 錯誤輸出和標準輸入）和終端機控制的批次作業
 - 欲提供作業結束前一直維持在作用中的批次作業提交客戶端
 - 欲考慮 Sun Grid Engine 軟體控制之平行作業的遠端作業執行
- `qselect` – 此指令列出對應指定選擇標準的佇列名稱清單。`qselect` 的輸出通常會傳輸到其他 Sun Grid Engine 的指令，好將動作套用在選取的佇列集上。
- `qsh` – 此指令會在低負載的主機上開啓互動的 shell（在 `xterm` 之中）。任何類型的互動作業都可在此 shell 執行。
- `qstat` – 此指令提供所有與叢集相關的作業和佇列的狀態清單。
- `qsub` – 此指令是用於提交作業於 Sun Grid Engine 系統的使用者介面。
- `qtcsch` – 此指令是眾所皆知並廣泛使用的 Unix C-Shell (`csch`) 導數，`tcsch` 完全相容的替代品。它透過 Sun Grid Engine 提供指令 shell 給適當且低負載的主機指定應用程式之透明分配執行的延伸。

所有的程式會透過 `sge_commd` 與 `sge_qmaster` 通訊。這會反映在圖 1-1 所描述，Sun Grid Engine 系統中的組件互動的圖解視區中。

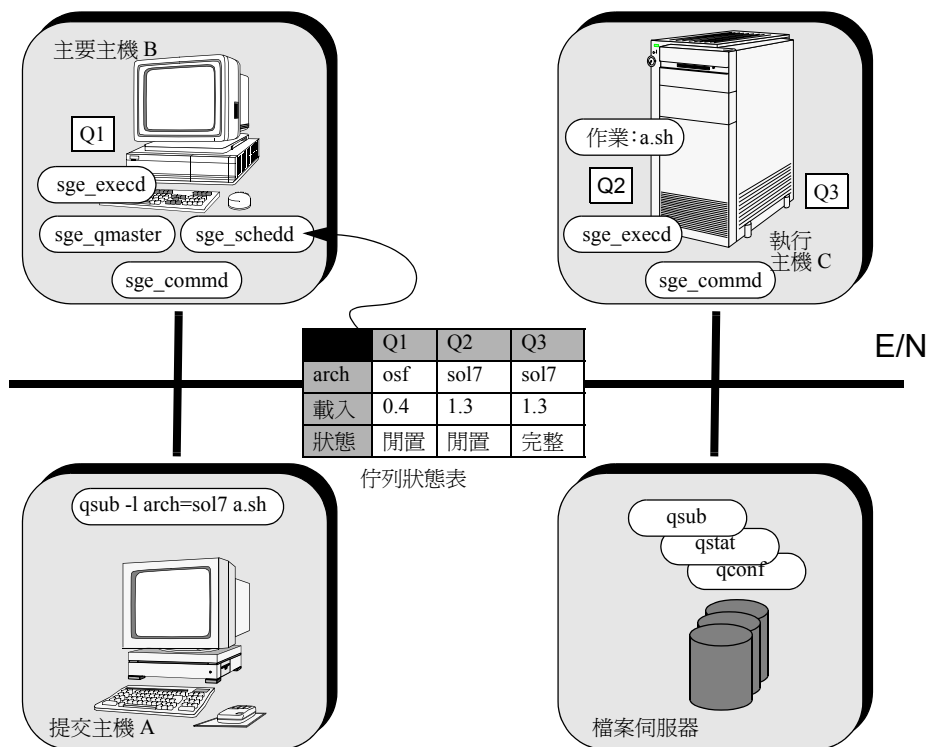


圖 1-1 Sun Grid Engine 系統的組件互動

QMON，Sun Grid Engine 圖形使用者介面

使用 QMON，圖形使用者介面 (GUI) 工具，您可以完成大部分 — 如果不是全部 — 的 Sun Grid Engine 5.3 作業。圖 1-2 會顯示 QMON 主功能表，其通常為使用者和管理者功能的起始點。每個主要功能表上的圖示都是您可以按下初始啟動各種作業的 GUI 按鈕。每個按鈕的名稱（當您將滑鼠指標移到按鈕上方會在螢幕上出現文字）也會對其功能有所敘述。

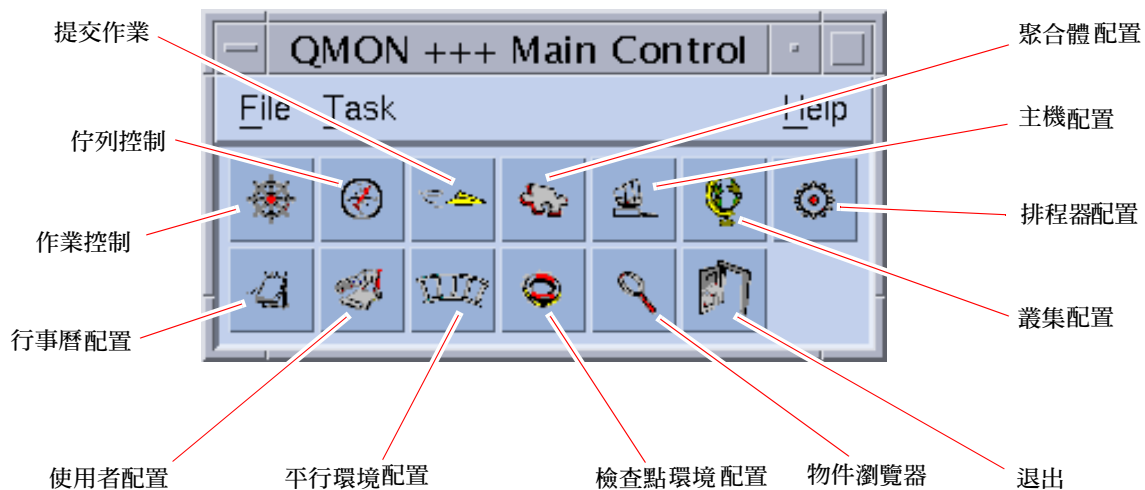


圖 1-2 QMON 主要功能表，定義的

自訂 QMON

qmon 的觀感大部分由特別設計的資源檔案所定義。參考的資源檔是放在 `<sgc_root>/qmon/Qmon` 中，而合理的預設值已經編譯在其中。

叢集管理可以藉由將 qmon 特定資源定義包含到標準 `.Xdefaults` 或 `.Xresources` 檔案中，或是將站點特定 Qmon 檔案放到標準搜尋路徑，如 `XAPPLRESDIR`，所提及的位置上，來把站點特定預設值安裝在標準位置上，如 `/usr/lib/X11/app-defaults/Qmon`。詢問您的管理員看看上述的任何事物是否和您的例子有關，

此外，使用者可以藉由將 Qmon 檔案複製並修改至起始目錄（或至另一個私人 `XAPPLRESDIR` 搜尋路徑所指定的位置）或將必要的資源定義包含至使用者私人的 `.Xdefaults` 或 `.Xresources` 檔案中，來配置個人的喜好設定。私人的 Qmon 資源檔案也可以在作業或 X11 環境（例如在 `.xinitrc` 資源檔案中）的啟動期間透過 `xrdb` 指令安裝。

欲取得可能自訂 (customization) 的詳細資訊，請參閱參考 Qmon 檔案裡的注釋行。

另一個自訂 qmon 的方式已為圖 5-3 和圖 5-13 裡的「作業控制 (Job Control)」和「佇列控制 (Queue Control)」自訂對話方塊做了解釋。在兩個對話方塊中，您可用「儲存」鈕儲存過濾並顯示配置自訂對話方塊至使用者起始目錄下的檔案，`.qmon_preferences` 的定義。重新開始之際，qmon 會讀取這個檔案並重新啟動先前定義的運作方式。

Sun Grid Engine 專有名詞的詞彙

詞彙提供了在 Sun Grid Engine 上下文和一般的資源管理中經常使用的專有名詞的簡介。目前有許多專有名詞尚未使用，但將會出現在 Sun Grid Engine 文件的其他部分。

存取清單	被授與或拒絕資源存取，如佇列或某一主機，的使用者和 UNIX 群組清單。使用者和群組可能屬於多重存取清單，而相同的存取清單可以在不同的情況下使用。
儲存格	有個別配置和主要機器的個別 Sun Grid Engine 叢集。儲存格可用來鬆散地連接個別的管理單元。
檢查點	將作業執行狀態儲存到所謂的檢查點，並使作業中斷，而在之後恢復後不會遺失資訊和已經完成的工作的程序。若檢查點再執行恢復前就已經移到另一個主機，則此過程稱為遷移。
檢查點環境	Sun Grid Engine 配置實體，其定義與檢查點方法相關的事件、介面和動作。
叢集	機器的集合，名為主機，主機上會有 Sun Grid Engine 功能產生。
聚合體	一套與佇列、主機或整個叢集相關的屬性。
群組	UNIX 群組。
硬資源 (hard resource)	
需求	在作業啟動前必須配置好的資源。與軟資源 (soft resource) 需求相反。
主機	會有 Sun Grid Engine 功能發生的機器。
作業	批次作業是可以不受使用者介入而執行並且不需要存取終端機的 UNIX shell script。 互動作業是以 Sun Grid Engine 指令， <code>qrsh</code> 、或 <code>qlogin</code> 啟動的階段作業，其會分別為使用者互動開啓 <code>xterm</code> 視窗或提供遠端登入階段作業的對等物。
作業陣列	包含一個範圍的獨立相同工作的作業。每個工作與個別的作業非常類似。作業陣列工作的唯一不同處是獨特工作辨識器（整數）。
作業類別	某種意義上相同並做類似處理的一套作業。在 Sun Grid Engine 之中，作業類別是由對應作業的相同需求和適合這些工作的佇列特性所定義的。
管理員	可以利用 Sun Grid Engine 所有層面的使用者。主要主機和其他宣告為管理主機的超級使用者擁有管理員的權限。管理員權限也可以指定給非 <code>root</code> 使用者帳號。
遷移	在作業恢復執行前，將檢查點從一個主機移往另一個主機的過程。

操作員	除了無法變更配置卻可以維護作業之外，能執行和管理員一樣指令的使用者。
所有人	可以暫停 / 解除暫停和停用 / 啓動他們所擁有的佇列的使用者。通常使用者是常駐在工作站的佇列的所有人。
平行環境	Sun Grid Engine 配置實體，其爲 Sun Grid Engine 定義必要的介面，以正確處理平行作業。
平行作業	包含一個以上有緊密關係之工作的作業。工作可跨多重主機分配。平行作業通常使用通訊工具，如共享記憶體或訊息傳遞 (MPI, PVM)，將作業同步化並使其產生關聯性。
策略	一套 Sun Grid Engine 管理者可以用來定義 Sun Grid Engine 運作方式的規則和配置。策略將自動由 Sun Grid Engine 實行。
優先權	Sun Grid Engine 作業比較其他作業的相關重要性的層級。
佇列	能同時在 Sun Grid Engine 執行主機上執行的某些類別和數目的作業的容器。
資源	執行作業所消耗或佔用的計算裝置。典型的例子是記憶體、CPU、I/O 頻寬、檔案空間、軟體授權等等。
軟資源	
需求	作業所需，但不必在作業啓動前配置的資源。依照可用的原則配置給作業。與硬資源需求相反。
暫停	暫停執行中的作業但使其維持在執行機器上的過程（和檢查點相反，其中作業已經中止）。暫停的作業仍會消耗一些資源，如交換記憶體或檔案空間。
使用者	如果該人員至少在一個提交主機或執行主機上有一個有效的登入，則可以使用 Sun Grid Engine 來提交作業或執行作業。
使用者集 (userset)	存取清單（請見上述說明）。

部分 II 入門

《*Sun Grid Engine 5.3 管理和使用者指南*》的這個部分包含單一章節。

- 第二章 – 第 13 頁的「安裝」

本章包含 Sun Grid Engine 5.3 產品的首次安裝指示，以及將舊版的產品升級到新版次的指示。

安裝

本章說明並提供三個安裝作業的詳細指示：

- 「快速啟動」安裝 — 不建議所有站點使用（請參閱第 13 頁的「快速啟動安裝簡介」）
- Sun Grid Engine 5.3 軟體的完整、首次安裝（請參閱第 19 頁的「完整安裝簡介」）
- 附有特殊加密功能的安全性安裝（請參閱第 30 頁的「如何安裝和設定 CSP 安全化系統 (CSP-Secured System)」）

注意 – 本章的指示假設您正在執行 Solaris™ 作業環境的電腦上安裝軟體。任何由其它 Sun Grid Engine 所執行的作業環境架構建立之功能的差異都紀錄在 `<sge_root>/doc` 目錄中，以 `arc_depend_` 字串開頭的檔案。檔案名稱剩餘的部分表示檔案中注釋所應用的作業系統結構。

快速啟動安裝簡介

注意 – 本節說明要讓快速啟動安裝程序有效所必須實行於您站點的條件。若您的環境不包含規劃的*所有*先決條件，您就*無法*使用快速安裝程序。在此情況下，要取得如何在更嚴格的條件下安裝 Sun Grid Engine 5.3 軟體的詳細資訊，請至第 19 頁的「完整安裝簡介」一節。

快速啓動安裝的先決條件

注意 – 這些指示只用於首次 Sun Grid Engine 5.3 安裝。要取得如何對舊版的 Sun Grid Engine 產品進行現存安裝的升級指示，請參閱 《*Sun Grid Engine 5.3 and Sun Grid Engine, Enterprise Edition 5.3 Release Notes*》。

本節說明在您經歷快速啓動安裝程序前幾個必須存在或需在站點建立的條件。

- 安裝帳號
- 安裝目錄
- 通訊埠編號

安裝帳號

對快速啓動安裝程序來說，需有 *管理員* 帳號或是您必須根據以下的指示建立一個管理員帳號。

- 管理員可以是現存的管理登入或新的登入，如 `sgeadmin`。

本帳號將擁有 Sun Grid Engine 安裝和多工緩衝處理目錄中的所有檔案，一旦安裝就可以用來配置和管理叢集。

- 該使用者不應該是 *root*。

然而，若您想要使用 *root* 做為檔案所有權，使用者 *root* 必須在安裝程式 Sun Grid Engine 之目錄具有所有主機的完整寫入權限。通常共享的 (NFS) 檔案系統不會匯出給使用者 *root* 以允許寫入權限。

- 這個帳號必須在安裝前就存在。

安裝目錄

若要準備使用管理員帳號，您應該建立安裝目錄，最好是在全網路的共享檔案系統上。

▼ 如何建立安裝目錄

- 請使用以下的指令序列。

```
% mkdir -p < 安裝目錄 >
% chown < 管理使用者 > < 安裝目錄 >
% chmod 755 < 安裝目錄 >
```

注意 – 爲此或類似程序建立的目錄在本手冊的剩餘部分叫做 Sun Grid Engine *root* 目錄。

通訊埠編號

Sun Grid Engine 系統使用 TCP 埠做為通訊之用。叢集內的*所有*主機必須使用相同的連接埠編號。您可以將連接埠編號放到以下任一位置中。以下為兩個範例。

- NIS 服務或 NIS+ 資料庫

把以下加入服務資料庫中。

```
% sge_commd 535/tcp
```

- 每個機器上的 `/etc/services`

如果 NIS 沒在您的站點執行，則您可以在每個機器上將以上的服務加到 `/etc/services` 檔案。

最好使用低於 600、具有權限的連接埠，以避免與連接接近或低於 1024 或高於 1024 的應用程式產生動態衝突。

讀取分發媒體 (Distribution Media)

Sun Grid Engine 5.3 軟體透過網際網路下載以文件資料庫檔案進行分發。分發包含直接寫入媒體的磁帶文件資料庫 (tar)。

▼ 如何解開 Sun Grid Engine 5.3 分發

1. 登入您選擇計劃讀進 Sun Grid Engine 分發媒體之安裝（請參閱第 14 頁的「快速啟動安裝的先決條件」一節）至主機的帳號。
2. 將您的工作目錄變更至 Sun Grid Engine root 目錄。
3. 使用以下指令讀進分發媒體。

```
% cd sge_root_dir  
% tar -xvpf 分發來源
```

在以上的指令中，`sge_root_dir` 為 Sun Grid Engine root 目錄的路徑名稱，而 `分發來源` 是硬碟上磁帶文件資料庫檔案的名稱。

為您的叢機安裝預設的 Sun Grid Engine 系統

預設的 Sun Grid Engine 系統包含一個**主要主機**和**執行主機**的仲裁編號。主要主機控制全數的叢集活動，而執行主機控制由主要主機指定給執行主機的作業之執行。單一主機可以同時擔任主要主機和執行主機。

注意 – 先安裝主要主機，然後在仲裁序列中以執行主機的安裝做為結束。

▼ 如何安裝主要主機

1. 選擇做為主要主機的機器。

依循指示選擇主要主機的機器。

- 選取的系統不應該過度負載其它的作業。
- 主要主機應提供足夠的可用主要記憶體，以執行必要的 Sun Grid Engine 常駐程式。

所需的數量完全依據您叢集的大小以及系統中預期的作業數目。對擁有很多主機與擁有 100 個作業排列的叢集而言，10 MB 的空白記憶體應已足夠。

對非常大型的叢集來說—擁有 1 千或更多主機和數萬個作業排列—您需要 1 GB 的記憶體。

2. 登入選取的機器。

若要進行具所有功能的安裝，您必須使用 *root* 帳號安裝（檔案仍為由第 14 頁的「快速啟動安裝的先決條件」所建立的管理員帳號所擁有）。

若為測試安裝，您也可以安裝為管理員使用者，但只有管理員能夠執行作業，而 Sun Grid Engine 系統將有與監視系統負載和系統控制相關之限制功能。

3. 將目錄 (cd) 變更至 Sun Grid Engine root 目錄。

4. 使用以下指令執行主要主機安裝程序。

```
% ./install_qmaster
```

若產生錯誤，則安裝程序會列出錯誤狀況的說明。

5. 回應具欲安裝的主機名稱之安裝 script。

安裝 script 會要求您提供您想初始安裝的主機清單。列出所有這類的主機，因為這些主機會以**提交**主機和**管理**主機新增。

執行主機的安裝（請參閱下一節）需要所有的主機做為管理主機。若您計劃在許多主機上安裝 Sun Grid Engine 程式，則安裝 script 將給您機會提供包含所有主機名稱的檔案路徑，一行一個主機。

6. 回應所有剩餘的安裝 script 提示符號。

安裝程序需要一些額外的資訊。大部分的問題需要提供有用的預設值，其您可以按 Return 來確認。

▼ 如何安裝執行主機

1. 根據以下指示，選取用來安裝的帳號。
 - 因為主要主機安裝的緣故，您應該使用 *root* 帳號安裝執行主機來獲得所有 Sun Grid Engine 功能的存取權。

安裝為 *root* 仍會維持第 14 頁的「快速啟動安裝的先決條件」一節所建立的管理員帳號的檔案所有權。
 - 藉由管理員帳號的安裝只能用於測試的目的，並禁止管理員之外的使用者執行作業。它不會允許 Sun Grid Engine 程式提供完整的系統監視和控制功能。
2. 登入在主要主機安裝程序期間選取為安裝至指定執行主機之一的帳號。
3. 將目錄 (cd) 變更至 Sun Grid Engine root 目錄。
4. 輸入以下指令以初始啟動執行主機安裝程序：

```
% ./install_execd
```

安裝程序會指出任何的錯誤。

5. 回應安裝 script 提示符號。

安裝程序會詢問您的主機是否應該配置預設佇列。在此例子定義的佇列將有第 17 頁的「預設系統配置」一節所說明的特性。
6. 當安裝 script 通知您執行主機安裝程序安裝成功時，請於每個在主要主機安裝期間命名的執行主機上進行相同的程序。

一旦您完成清單上的主機安裝，您的預設 Sun Grid Engine 系統就已配置於叢集上，並可準備使用。若要使用該程式進行練習，請參閱第 61 頁的「如何從指令行執行簡單作業」一節。

下節為已安裝的預設配置簡介。

預設系統配置

注意 – 以下為執行快速啟動安裝程序以將 Sun Grid Engine 系統配置在您的環境之說明。此為測試目的的最小設定，之後可隨時變更或延伸。

在完成主要主機和執行主機的安裝後，以下的基礎 Sun Grid Engine 系統就已配置在您的叢集中。

- 主要主機

您執行主要主機安裝程序的主機配置已為叢集的主要主機。沒有配置備用主要主機在主要主機故障時接管主要主機的作業。

- 執行主機

在主要主機安裝期間，機器將詢問您是否想安裝 Sun Grid Engine 執行代理程式的機器清單。在安裝這些執行主機的期間，您可以允許安裝程序在這些主機上自動建立佇列。佇列說明可以在特殊主機上執行的作業簡介（屬性和需求清單）。根據預設值配置給執行主機的佇列有以下的重要特性。

- 佇列名稱：< 不具資格的主機名稱 >.q
- 插槽（同時執行的作業）：< 處理器的數量 >
- 佇列提供作業無限的系統資源（如記憶體、CPU 時間等等）。
- 佇列不會實施特定使用者或使用者群組的存取限制。任何具有有效帳號的使用者可以執行佇列中的作業。
- 將配置每個 CPU 的 1.75 負載臨界值（即 1.75 負載臨界值試圖平均存取每個 CPU）。

注意 – 佇列配置，如任何其他 Sun Grid Engine 配置，可以在系統操作時，在之後直接變更。

注意 – 如果您也在主要主機上呼叫執行主機安裝程序，主要主機可以扮演主要和執行主機。

- 管理帳號和主機

主要主機和所有執行主機都配置允許執行管理 Sun Grid Engine 指令。允許管理 Sun Grid Engine 的唯一使用者為使用者 *root* 和第 14 頁的「快速啟動安裝的先決條件」一節所說明的管理員帳號。若沒有特權的使用者安裝 Sun Grid Engine 程式，他會加到 Sun Grid Engine 管理員的清單中。

- 提交帳號和主機

若您在 *root* 帳號之下安裝程式，任何具有有效帳號的使用者可以提交和控制 Sun Grid Engine 作業。在底下安裝 Sun Grid Engine 軟體的使用者將是唯一有存取權的使用者（請參閱第 14 頁的「快速啟動安裝的先決條件」）。提交作業、控制 Sun Grid Engine 系統活動或刪除作業的作業可以從任一主要主機或執行主機執行。

- 常駐程式

以下的常駐程式會在不同主機上的系統安裝期間啟動或是在一般系統操作期間呼叫。

- *sge_qmaster* 只在主要主機上執行。它是中央叢集活動控制常駐程式。
- *sge_schedd* 只在主要主機上啟動。本常駐程式負責分發 Sun Grid Engine 叢集中的工作量。

- `sge_execd` 負責在執行主機上執行作業，並因此在所有執行主機上執行。
- `sge_shepherd` 的一個範例為實際在主機上執行的每個作業執行。
`sge_shepherd` 控制作業過程階層並在作業結束後蒐集帳號資料。
- `sge_commd` 在每個執行主機和主要主機上執行。所有 `sge_commd` 的網路會形成 Sun Grid Engine 叢集的網路通訊骨幹。

完整安裝簡介

注意 – 這些指示只用於首次 Sun Grid Engine 5.3 安裝。要取得如何對舊版的 Sun Grid Engine 產品進行現存安裝的升級指示，請參閱 《*Sun Grid Engine 5.3 and Sun Grid Engine, Enterprise Edition 5.3 Release Notes*》。

完整安裝包含以下廣義的作業。

- 計劃 Sun Grid Engine 配置和環境
- 從外部媒體讀取 Sun Grid Engine 分配檔案到工作站上。
- 在主要主機和每個 Sun Grid Engine 系統中的執行主機上執行安裝 script。
- 關於管理和提交主機的註冊資訊
- 驗證安裝

安裝應該由熟悉 Solaris 作業環境的人來執行。整個過程會經由三個階段完成。

階段 1 – 計劃

注意 – 若您想用加強的安全性安裝系統，請在繼續安裝前參閱第 29 頁的「用加強的安全性安裝」一節。

安裝的計劃階段包含以下作業。

- 決定您的 Sun Grid Engine 環境是否為單一叢集或名為儲存格的子叢集集合
- 選擇將為 Sun Grid Engine 主機的機器。決定每個機器將成為的主機種類 — 主要主機、備用主要主機、管理主機、提交主機、執行主機或主機的組合
- 確認所有的 Sun Grid Engine 使用者在所有提交和執行主機上都有相同的使用者 id
- 決定 Sun Grid Engine 目錄組織的角色為何 — 例如，每個工作站上做為完整樹的組織目錄、cross-mounted 目錄、某些工作站上的部分目錄樹 — 以及每個 Sun Grid Engine root 目錄的所在位置。
- 決定站點的佇列結構

- 決定網路服務是否定義為 NIS 檔案或為 `/etc/services` 中每個工作站的本端
- 完成安裝工作表（請參考步驟 1，第 25 頁的「在開始安裝前，在類似下表的表格中寫下您的安裝計劃。」），以用在之後的安裝步驟

階段 2 – 安裝軟體

安裝階段包含以下作業。

- 建立安裝目錄，並將分配檔案載入
- 安裝主要主機
- 安裝所有執行主機
- 註冊所有管理主機
- 註冊所有提交主機

階段 3 – 驗證安裝

驗證階段包含以下作業。

- 檢查常駐程式是否在主要主機上執行
- 檢查常駐程式是否在所有的執行主機上執行
- 檢查 Sun Grid Engine 有否執行簡單指令
- 提交測試作業

計劃安裝

在您開始安裝 Sun Grid Engine 5.3 軟體之前，您必須仔細計劃如何達到完全符合您的環境的結果。本節會幫助您做出會影響本程序其餘部分的重大決定。

先決條件作業

以下各節說明您需用來安裝生產 Sun Grid Engine 系統的資訊。

安裝目錄 <sg_e_root>

準備好目錄讀進 Sun Grid Engine 分發媒體的內容。此目錄稱為 Sun Grid Engine *root* 目錄，而之後當 Sun Grid Engine 系統在作業中時，此目錄將用來儲存目前叢集配置和所有需進一步多工緩衝至磁碟的資料。

使用所有主機上正確參照的目錄之路徑名稱。例如，若使用自動裝載器裝載檔案系統，請將 <sg_e_root> 設為 /usr/SGE，而非 /tmp_mnt/usr/SGE。（在整份文件中，在參照安裝目錄時，就會使用 <sg_e_root> 環境變數。）

<sg_e_root> 是 Sun Grid Engine 目錄樹的最高層級。每個儲存格裡的 Sun Grid Engine 元件（請參閱第 24 頁的「儲存格」一節）需要對啓動上面的 <sg_e_root>/<儲存格>/common 有讀取存取。要取得所需權限的說明，請參閱第 23 頁的「檔案存取權限」一節。

為了安裝和管理的簡便，在您想執行 Sun Grid Engine 安裝程序的所有主機上都需可讀取此目錄。例如，您可以透過網路檔案系統（如 NFS）選擇可用的目錄。若您選擇選取主機的本地檔案系統，您將必須在為特定機器啓動安裝程序前把安裝目錄複製到每個主機。

root 目錄下的多工緩衝處理目錄

- 在 Sun Grid Engine 主要主機上，多工緩衝處理目錄維護在 <sg_e_root>/<儲存格>/spool/qmaster 和 <sg_e_root>/<儲存格>/spool/schedd 之下。
- 在每個執行主機上會維護名為 <sg_e_root>/<儲存格>/spool/<exec_host> 的多工緩衝處理目錄。

您不需要將這些目錄匯出至其它機器。然而，匯出整個 <sg_e_root> 樹並使主要和所有執行主機可寫入存取即可增加管理的簡便性。

目錄組織

決定 Sun Grid Engine 目錄組織的角色為何（例如，每個工作站上的完整樹、cross-mounted 目錄、某些工作站上的部分目錄樹），以及每個 Sun Grid Engine root 目錄，<sg_e_root>，的所在位置。

注意 – 因為安裝目錄和 / 或多工緩衝處理目錄的變更基本上需要新的系統安裝（儘管能保存所有前次安裝的重要資訊），所以您必須更小心選擇適合的前置 (upfront) 安裝目錄。

根據預設值，Sun Grid Engine 安裝程序將在安裝目錄下的目錄階層（請參閱第 22 頁的「樣本目錄階層」的圖 2-1）安裝 Sun Grid Engine 系統、手冊、多工緩衝處理區和配置檔案。若您接受此預設運作方式，您應該安裝 / 選擇允許第 23 頁的「檔案存取權限」所述之存取權限的目錄。

您可以選擇多工緩衝處理區域以於主要安裝期間放置其它位置（相關指示請參閱第 123 頁的「主機和叢集配置」的第六章）。

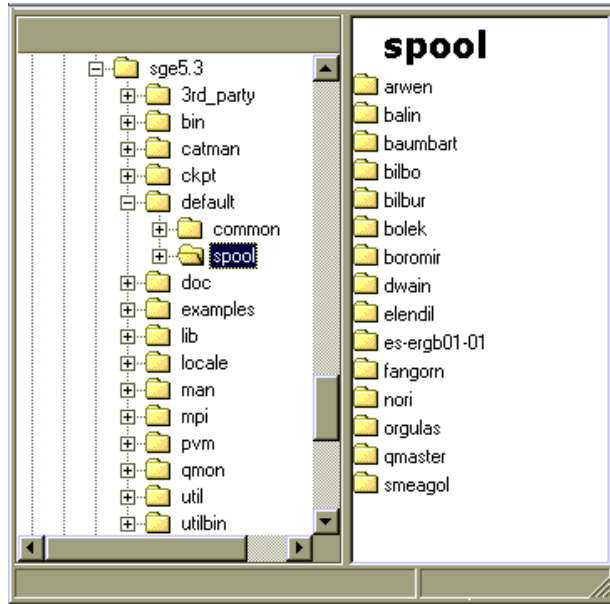


圖 2-1 樣本目錄階層

磁碟空間需求

Sun Grid Engine 目錄樹有以下固定的磁碟空間需求。

- 沒有任何二進位的安裝工具箱（包含說明文件）需要 40 MB
- 每一套的二進位（Cray 架構除外，其中二進位消耗約 35 MB）需要 10 到 15 MB

Sun Grid Engine 日誌檔案的理想磁碟空間如下。

- 主要主機多工緩衝處理目錄需要 30 到 200 MB，依據叢集的大小而定
- 每個執行主機需要 10 到 20 MB

注意 – 可以配置主要主機和執行主機的多工緩衝處理目錄，而這些目錄不需常駐在 `<sge_root>` 之下（就是它們的預設位置）。變更多工緩衝處理目錄的位置應該在主要安裝後完成（相關指示請參閱第 123 頁的「主機和叢集配置」的第六章）。

安裝帳號

您可以將 Sun Grid Engine 安裝在 root 帳號或無權限的（例如您自己的）帳號下。若您安裝在無權限的帳號下，該安裝僅會允許特定使用者執行 Sun Grid Engine 作業。存取至所有其它帳號將被拒絕。安裝在 root 帳號下將解決這項限制；但是，完整安裝程序需要 root 權限。

檔案存取權限

若您安裝為 root，您可能在共享的檔案系統上有為全部主機配置 root 讀取 / 寫入的問題，因此，您可能會有將 `<sge_root>` 放到全網路檔案系統的問題。您可以強制 Sun Grid Engine 軟體透過非 root 的管理使用者帳號（例如，名為 `sgeadmin`）執行所有 Sun Grid Engine 元件的完整檔案處理。因此，您只需給特定的使用者共享 root 檔案系統讀取 / 寫入存取權。Sun Grid Engine 安裝程序會詢問您是否要在管理使用者帳號下處理檔案。若您回答「是」並提供有效的 *使用者名稱*，檔案處理將透過該使用者名稱執行。否則，將使用您執行安裝程序的使用者名稱。

您必須確定在所有的範例中，用做檔案處理的帳號必須在所有主機上有 Sun Grid Engine root 目錄的讀取 / 寫入存取權。此外，安裝程序會假設您要讀進 Sun Grid Engine 分發媒體的主機能夠存取該目錄。

網路服務

決定您站點的網路服務要定義為 NIS 檔案或在 `/etc/services` 之工作站的本端。若您的站點使用 NIS，請找出 NIS 伺服器主機，好讓您將項目新增至服務 NIS 對映。

Sun Grid Engine 服務為 `sge_commd`。若要新增服務到 NIS 對映，請選擇一個保留、未使用過的連接埠號碼 —1024 之下的任一數字。以下是一個 `sge_commd` 項目的範例。

```
sge_commd 536/tcp
```

主要主機

這是控制 Sun Grid Engine 的主機，其執行主要常駐程式，`sge_qmaster`。對超大型的叢集來說 — 就是任何時間在系統內包含數百個或數千個主機和上萬個作業的叢集 — 則需要 1 GB 或更多的未使用之主要記憶體，而有兩個 CPU 會有更多好處。

- 它必須是個穩定的平台。
- 它不能因其他的處理過程而過度忙碌。
- 它至少要有 20 MB 的未使用的主要記憶體來執行 Sun Grid Engine 常駐程式。
- 它可選擇是否要有 Sun Grid Engine 目錄，`<sge_root>`，為其本端以減少網路流量。

備用主要主機

這些主機會備份 `sge_qmaster` 的功能，以防主要主機或主要常駐程式故障。做為備份主要主機，該機器必須有以下的特性。

- 它必須執行 `sge_shadowd`。
- 它必須分享 `sge_qmaster` 的狀態、作業和紀錄在磁碟的佇列配置資訊。特別是，備份主要主機需要至 `sge_qmaster` 的多工緩衝處理目錄和 `<sge_root>/<儲存格>/common` 目錄的讀取 / 寫入 `root` 或管理使用者存取權。
- `<sge_root>/<儲存格>/common/shadow_masters` 檔案必須包含一行定義主機為備用主要主機的數字。

這些條件符合之後就會啟動主機的備份主要主機功能。所以您不需要重新啟動 Sun Grid Engine 常駐程式來將主機變成備用主機。

執行主機

這些主機執行提交給 Sun Grid Engine 的作業。您將在每個執行主機上執行安裝 `script`。

管理主機

Sun Grid Engine 操作員和管理員執行管理作業，如重新配置佇列或從這些主機新增 Sun Grid Engine 使用者。主要主機安裝 `script` 會自動將主要主機變成管理主機。

提交主機

Sun Grid Engine 作業可能從提交主機提交及控制。主要主機安裝 `script` 會自動將主要主機變成提交主機。

儲存格

您可以將 Sun Grid Engine 設定為單一叢集或名為儲存格的鬆散組合叢集的集合。SGE_CELL 環境變數指出參照的叢集。當 Sun Grid Engine 安裝為單一叢集時，SGE_CELL 並無設定，而儲存格數值假定為數值 `default`。

使用者名稱

為了讓 Sun Grid Engine 驗證提交作業的使用者有提交作業和使用他們所需的執行主機的權限，使用者在相關的提交和執行主機的名稱必須相同。這個需求可能讓變更某些機器上的使用者 `id` 變成不可避免。

注意 – 主要主機上的使用者 id 和檢查權限沒有關係，名稱不需相同，甚至不用存在。

佇列

計劃符合您站點需求的佇列結構。這表示決定哪個佇列要放在哪個執行主機上、您的連續、互動、平行和其他作業類型是否需要佇列、每個佇列需要多少作業插槽和其他佇列配置決定。

Sun Grid Engine 管理員也可能讓安裝程序建立預設的佇列結構，其適合做為系統熟悉以及日後調整的起點。

注意 – 除了安裝進 Sun Grid Engine 的目錄，所有的 Sun Grid Engine 安裝程序所建立的設定可以在系統作業期間進行同步變更。

除非您已經熟悉 Sun Grid Engine 或是您之前已經決定您要在叢集使用的佇列結構，否則您不該允許安裝程序替您安裝預設的佇列結構。反之，您應該準備一個指定該佇列結構的文件，並在完成安裝過程後直接前進到第 145 頁的「配置佇列與佇列行事曆」的第七章。

▼ 如何計劃安裝

1. 在開始安裝前，在類似下表的表格中寫下您的安裝計劃。

參數	數值
<sg_e_root>	
管理使用者	
管理群組	
sg_e_commd 連接埠編號	
主要主機	
備用主要主機	
執行主機	
管理主機	
提交主機	

2. 確認檔案系統和目錄將包含 Sun Grid Engine 分發，而透過上述的定義設定存取權限，多工緩衝處理與配置檔案已正確完成設定。

▼ 如何讀取分發媒體

Sun Grid Engine 軟體會透過網際網路下載以文件資料庫檔案進行分發。網路分發也以 compress (副檔名 .Z) 或 gzip (副檔名 .gz) 的進行之最終壓縮的 tar 檔案格式提供。您必須在使用以下步驟進行前，將檔案解壓縮 (使用 uncompress 或 gunzip)。

1. 提供至分發媒體的存取並登入系統中 — 最好是直接連接檔案伺服器的系統。
2. 請依照第 21 頁的「安裝目錄 <sg_e_root>」一節所述，建立安裝目錄以讀進 Sun Grid Engine 安裝工具箱，並確定安裝目錄的存取權限已經正確設定。

3. 從指令提示符號執行以下程序。

```
% cd 安裝目錄
% tar -xvpf 分發來源
```

其中*安裝目錄*是安裝目錄的路徑名稱，而*分發來源*是硬碟上解壓縮磁帶文件資料庫檔案的名稱。這會讀進 Sun Grid Engine 安裝工具箱。

▼ 如何安裝主要主機

1. 以 root 身分登入主要主機。
2. 根據安裝工具箱常駐的目錄是否能從主要主機看見，請執行以下任一作業。
 - a. 若安裝工具箱常駐的目錄*能*從主要主機看見，請將目錄 (cd) 變更至安裝目錄，然後再繼續進行步驟 3。
 - b. 如果*不能*看見目錄而且不能變為可見，請執行以下事項。
 - i. 在主要主機上建立本端安裝目錄。
 - ii. 透過網路（例如，使用 ftp 或 rcp）將安裝工具箱複製到本端安裝目錄。
 - iii. 將目錄 (cd) 變更至本端安裝目錄。
3. 執行以下指示。

注意 – 若您要透過認證安全性通訊協定 (Certificate Security Protocol) 方法執行安裝，您必須將 `-csp` 旗號新增至以下指令中（請參閱第 30 頁的「如何安裝和設定 CSP 安全化系統 (CSP-Secured System)」）。

```
% ./install_qmaster
```

這會初始啟動主要安裝程序。您將被詢問數個問題，並可能需要執行幾個管理動作。這些問題和動作項目是能自我解釋的。

注意 – 有第二個作用中的終端機階段作業來執行管理作業會很方便。

主要安裝程序會建立 `sgc_qmaster` 和 `sgc_schedd` 所要求的合適目錄階層。該程序會在主要主機上啟動 Sun Grid Engine 元件 `sgc_commd`, `sgc_qmaster` 和 `sgc_schedd`。主要主機也會登錄為具有管理和提交權限的主機。

若您確信有異狀發生，您可以隨時中斷並重複安裝程序。

▼ 如何安裝執行主機

1. 以 root 身分登入執行主機。
2. 至於主要安裝，請將安裝工具箱複製到本端安裝目錄，或使用網路安裝目錄。
3. 將目錄 (cd) 變更至安裝目錄，並執行以下指令。

注意 – 若您要透過認證安全性通訊協定方法執行安裝，您必須將 `-csp` 旗號新增至以下指令中（請參閱第 30 頁的「如何安裝和設定 CSP 安全化系統 (CSP-Secured System)」）。

```
% ./install_execd
```

這會初始啟動執行主機安裝程序。執行主機安裝程序的運作方式和處理和主要主機的非常像。

4. 回應來自安裝 script 的提示符號。

注意 – 您也可以使用主要主機執行作業。您只要實行主要機器的執行主機安裝即可。

注意 – 若您使用一台非常慢的機器做為主要主機，或如果您的叢集相當龐大，您應該只使用主要機器執行主要作業。

執行主機安裝程序會建立 `sge_execd` 所要求的合適目錄階層。該程序會在執行主機上啟動 Sun Grid Engine 元件 `sge_commd` 和 `sge_execd`。

▼ 如何安裝管理和提交主機

主要主機暗中允許執行管理作業，並提交、監視和刪除作業。它不需要任何種類的額外安裝做為管理或提交主機。相反地，純粹的管理和提交主機則需要登記。

- 從管理主機（如主要主機）到透過管理帳號（如超級使用者帳號），請輸入以下指令。

```
% qconf -ah 管理主機名稱 [...]  
% qconf -as 提交主機名稱 [...]
```

要取得配置其它主機類型的詳細細節和其他方法，請參考第 125 頁的「關於常駐程式和主機」一節。

用加強的安全性安裝

您可以使用以下指示更安全設定系統。這些指示會協助您使用**認證安全性通訊協定 (CSP)** 式的加密來設定系統。

Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 產品都能利用這個安全設定，而這些指示適用於這兩個產品。為了達成簡單扼要的目的，這些指示只引用 Sun Grid Engine 產品。

除了以明文方式傳送訊息外，在這個較安全的系統內的訊息都以私密金鑰 (secret key) 加密。私密金鑰透過公共 / 私人金鑰通訊協定交換。使用者透過 Sun Grid Engine 系統提出證明文件以證明身分，並從 Sun Grid Engine 系統接收證明文件來證實使用者正與正確的系統通訊。在初始宣佈階段之後，通訊會以加密的形式繼續透明進行。階段作業的有效性只維持一段時間，在這之後，階段作業必須重新宣佈。

所需的額外設定

設定 Sun Grid Engine 系統的認證安全性通訊協定加強版的步驟非常類似標準設定。您通常依循第 25 頁的「如何計劃安裝」、第 26 頁的「如何讀取分發媒體」、第 27 頁的「如何安裝主要主機」、第 28 頁的「如何安裝執行主機」和第 28 頁的「如何安裝管理和提交主機」等幾節的指示。

然而，以下額外的作業是必要的。

- 主要主機上認證權限 (CA) 系統金鑰和證明文件的產生
這是透過呼叫具有 `-csp` 旗號的安裝 script 而完成的。
- 分發系統金鑰和證明文件至執行和提交主機
安全執行是系統管理員的工作；即金鑰必須以安全的方式傳輸到執行主機和提交主機，如透過 `ssh`。
- 使用者金鑰和證明文件的產生
這可以在主要安裝後自動由系統管理員執行。

- 系統管理員允許新使用者進入

▼ 如何安裝和設定 CSP 安全化系統 (CSP-Secured System)

1. 用第 19 頁的「完整安裝簡介」、第 20 頁的「計劃安裝」、第 27 頁的「如何安裝主要主機」、第 28 頁的「如何安裝執行主機」和第 28 頁的「如何安裝管理和提交主機」各節所規劃的方式安裝 Sun Grid Engine 系統，但有以下的例外：在呼叫不同安裝 script 時，使用額外的旗號，`-csp`。

例如，在安裝主要主機的基本安裝指示告訴您輸入 `./install_qmaster` 來呼叫 script 之處，您可加入 `-csp` 旗號來修改該指示。因此，要安裝 CSP 安全性系統，您可輸入以下指令變更主要主機安裝程序。

```
% ./install_qmaster -csp
```

2. 回應來自安裝 script 的提示符號。

若要產生 CSP 證明文件和金鑰，您必須提供以下的資訊。

- 兩個字母的國碼 — 例如，US 代表美國
- 州
- 位置 — 如城市
- 組織
- 組織單位
- CA 電子郵件地址

當安裝繼續進行時，認證權限也會建立。Sun Grid Engine 特定 CA 建立在主要主機上。含有安全性相關資訊的目錄如下。

- 在 `$SGE_ROOT/{預設| $SGE 儲存格}/common/sgeCA` 之下，儲存可公開存取的 CA 和常駐程式證明文件。
- 在 `/var/sgeCA/{sge 服務| 連接埠 $COMM 連接埠}/{預設| $SGE 儲存格}/private` 之下，儲存對應的私密金鑰。
- 在 `/var/sgeCA/{sge 服務| 連接埠 $COMM 連接埠}/{預設| $SGE 儲存格}/userkeys/$USER` 之下，儲存使用者金鑰和證明文件。

在這個過程中，script 輸出看起來類似程式範例 2-1 的範例。

程式範例 2-1 CSP 安裝 Script — 目錄建立

為 OpenSSL 安全性架構初始啟動認證權限 (CA)

```
-----  
Creating /scratch2/eddy/sge_sec/default/common/sgeCA  
Creating /var/sgeCA/port6789/default  
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/certs  
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/crl  
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/newcerts  
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/serial  
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/index.txt  
Creating /var/sgeCA/port6789/default/userkeys  
Creating /var/sgeCA/port6789/default/private  
Hit Return to continue >>
```

設定目錄之後，會產生 CA 特定的證明文件和私密金鑰。Sun Grid Engine 系統使用來自特別檔案的虛擬隨機資料 (pseudo random data) 或用 `/dev/random` (如果有的話)，來催化虛擬隨機編號產生器 (pseudo random number generator/PRNG)。(要取得更多隨機編號的詳細資訊，請參閱 <http://www.openssl.org/support/faq.html> 和 <http://www.cosy.sbg.ac.at/~andi>。)

在安裝 CA 基礎架構後，應用程式證明文件、使用者證明文件和私密金鑰由 CA 建立並簽署給管理使用者、虛擬常駐程式使用者和使用者，`root`。script — 其輸出類似程式範例 2-2 中的範例 — 對網站資訊的第一個查詢。

程式範例 2-2 CSP 安裝 Script — 資訊蒐集

```
Creating CA certificate and private key
-----

Please give some basic parameters to create the distinguished name (DN)
for the certificates.

We will ask for

    - the two letter country code
    - the state
    - the location, e.g city or your buildingcode
    - the organization (e.g. your company name)
    - the organizational unit, e.g. your department
    - the email address of the CA administrator (you!)

Hit Return to continue >>

Please enter your two letter country code, e.g. >US< >> DE
Please enter your state >> Bavaria
Please enter your location, e.g city or buildingcode >> Regensburg
Please enter the name of your organization >> Myorg
Please enter your organizational unit, e.g. your department >> Mydept
Please enter the email address of the CA administrator >> admin@my.org

You selected the following basic data for the distinguished name of
your certificates:

Country code:C=DE
State:ST=Bavaria
Location:L=Regensburg
Organization:O=Myorg
Organizational unit:OU=Mydept
CA email address:emailAddress=admin@my.org

Do you want to use these data (y/n) [y] >>
```

在確認您提供的資訊正確後，安裝程式會繼續 CA 證明文件和私密金鑰的產生，從設定 CA 基礎架構開始。script 輸出類似程式範例 2-3 中的範例。

程式範例 2-3 CSP 安裝 Script—CA 基礎架構建立

```
Creating RANDFILE from >/kernel/genunix< in
>/var/sgeCA/port6789/default/private/rand.seed<

1513428 semi-random bytes loaded
Creating CA certificate and private key

Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/var/sgeCA/port6789/default/private/akey.pem'
-----
Hit Return to continue >>
```

在安裝 CA 基礎架構後，CA 會建立並簽署應用程式、使用者證明文件和私密金鑰給虛擬常駐程式使用者和 root 使用者。Script 輸出與下幾頁所顯示的輸出類似。請注意，範例中有些行列都會進行縮寫，好放進這幾頁的每個單一行列中。縮寫的部份以 (...) 表示。

程式範例 2-4 CSP 安裝 Script—證明文件和私密金鑰建立

```
Creating Daemon certificate and key
-----

Creating RANDFILE from >/kernel/genunix< in >/var/sgeCA/(...)/rand.seed<

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/var/sgeCA/port6789/default/private/key.pem'
-----
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName          :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Bavaria'
localityName         :PRINTABLE:'Regensburg'
```

程式範例 2-4 CSP 安裝 Script— 證明文件和私密金鑰建立 (接上頁)

```
organizationName      :PRINTABLE:'Myorg'
organizationalUnitName:PRINTABLE:'Mydept'
uniqueIdentifier      :PRINTABLE:'root'
commonName           :PRINTABLE:'SGE Daemon'
emailAddress         :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:50:57 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
created and signed certificate for SGE daemons
Creating RANDFILE from >/kernel/genunix< in>/var/(...)/userkeys/root/rand.seed<

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/var/sgeCA/port6789/default/userkeys/root/key.pem'
-----
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName          :PRINTABLE:'DE'
stateOrProvinceName :PRINTABLE:'Bavaria'
localityName         :PRINTABLE:'Regensburg'
organizationName     :PRINTABLE:'Myorg'
organizationalUnitName:PRINTABLE:'Mydept'
uniqueIdentifier      :PRINTABLE:'root'
commonName           :PRINTABLE:'SGE install user'
emailAddress         :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:50:59 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
created and signed certificate for user >root< in >/var/(...)/userkeys/root<
Creating RANDFILE from >/kernel/genunix< in >/var/(...)/userkeys/eddy/rand.seed<
1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/var/sgeCA/port6789/default/userkeys/eddy/key.pem'
-----
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
```

程式範例 2-4 CSP 安裝 Script—證明文件和私密金鑰建立 (接上頁)

```
The Subjects Distinguished Name is as follows
countryName      :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Bavaria'
localityName      :PRINTABLE:'Regensburg'
organizationName   :PRINTABLE:'Myorg'
organizationalUnitName :PRINTABLE:'Mydept'
uniqueIdentifier   :PRINTABLE:'root'
commonName        :PRINTABLE:'SGE install user'
emailAddress       :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:50:59 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
created and signed certificate for user >root< in >/var/(...)/userkeys/root<
Creating RANDFILE from >/kernel/genunix< in >/var/(...)/userkeys/eddy/rand.seed<

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_cal4364.tmp
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/var/sgeCA/port6789/default/userkeys/eddy/key.pem'
-----
Using configuration from /tmp/sge_cal4364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName      :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Bavaria'
localityName      :PRINTABLE:'Regensburg'
organizationName   :PRINTABLE:'Myorg'
organizationalUnitName :PRINTABLE:'Mydept'
uniqueIdentifier   :PRINTABLE:'eddy'
commonName        :PRINTABLE:'SGE admin user'
emailAddress       :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:51:02 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
created and signed certificate for user >root< in >/var/(...)/userkeys/root<
Creating RANDFILE from >/kernel/genunix< in >/var/(...)/userkeys/eddy/rand.seed<

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_cal4364.tmp
Generating a 1024 bit RSA private key
.....++++++
.....++++++
```

程式範例 2-4 CSP 安裝 Script— 證明文件和私密金鑰建立 (接上頁)

```
writing new private key to '/var/sgeCA/port6789/default/userkeys/eddy/key.pem'
-----
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName          :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Bavaria'
localityName         :PRINTABLE:'Regensburg'
organizationName     :PRINTABLE:'Myorg'
organizationalUnitName :PRINTABLE:'Mydept'
uniqueIdentifier     :PRINTABLE:'eddy'
commonName           :PRINTABLE:'SGE admin user'
emailAddress         :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:51:02 2003 GMT (365 days

Write out database with 1 new entries
Data Base Updated
created and signed certificate for user >eddy< in >/var/(...)/userkeys/eddy<
Hit Return to continue >>
```

在主要主機，sge_qmaster，的安全性相關設定完成後，script 會提示繼續進行安裝程序的其餘部分，其與程式範例 2-5 中的範例類似。

程式範例 2-5 CSP 安裝 Script – 安裝的延續

```
SGE startup script
-----

Your system wide SGE startup script is installed as:

    "/scratch2/eddy/sge_sec/default/common/rcsge"

Hit Return to continue >>
```

3. 請執行以下其中一項。

- a. 若您認為共享檔案系統的安全性不足以將 CSP 安全性資訊維持在一個執行常駐程式可存取的地方，則請前進至步驟 4。
- b. 若您認為共享檔案系統的安全性是足夠的，請繼續進行第 28 頁的「如何安裝執行主機」一節所規劃的基礎安裝程序。
當您為執行主機安裝呼叫 './inst' script 時，請記得加入 -csp 旗號。
在完成所有其餘的安裝步驟後，請參照第 39 頁的「如何產生使用者證明文件和私密金鑰」一節的指示。

4. (選擇性的) 若共享檔案系統的安全性也不足以將 CSP 安全性資訊維持在一個執行常駐程式可存取的地方，您就必須將包含常駐程式私密金鑰和隨機檔案的目錄傳輸到執行主機。

- a. 做為主要主機上的 root，請輸入以下指令來準備將私密金鑰複製到您將設定為執行主機的機器上。

```
# umask 077
# cd /
# tar cvpf /var/sgeCA/port6789.tar /var/sgeCA/port6789/default
```

- b. 做為每個執行主機上的 root，請輸入以下指令來複製檔案。

```
# umask 077
# cd /
# scp < 主要主機 >:/var/sgeCA/port6789.tar .
# umask 022
# tar xvpf /port6789.tar
# rm /port6789.tar
```

- c. 輸入以下指令驗證檔案權限。

```
# ls -lR /var/sgeCA/port6789/
```

輸出應該類似程式範例 2-6 中的範例。

程式範例 2-6 檔案權限驗證

```
/var/sgeCA/port6789/:
total 2
drwxr-xr-x  4 eddy      other          512 Mar  6 10:52 default
/var/sgeCA/port6789/default:
total 4
drwx----- 2 eddy      staff          512 Mar  6 10:53 private
drwxr-xr-x  4 eddy      staff          512 Mar  6 10:54 userkeys
/var/sgeCA/port6789/default/private:
total 8
-rw-----  1 eddy      staff          887 Mar  6 10:53 cakey.pem
-rw-----  1 eddy      staff          887 Mar  6 10:53 key.pem
-rw-----  1 eddy      staff         1024 Mar  6 10:54 rand.seed
-rw-----  1 eddy      staff          761 Mar  6 10:53 req.pem
/var/sgeCA/port6789/default/userkeys:
total 4
dr-x----- 2 eddy      staff          512 Mar  6 10:54 eddy
dr-x----- 2 root      staff          512 Mar  6 10:54 root
/var/sgeCA/port6789/default/userkeys/eddy:
total 16
-r-----  1 eddy      staff         3811 Mar  6 10:54 cert.pem
-r-----  1 eddy      staff          887 Mar  6 10:54 key.pem
-r-----  1 eddy      staff         2048 Mar  6 10:54 rand.seed
-r-----  1 eddy      staff          769 Mar  6 10:54 req.pem
/var/sgeCA/port6789/default/userkeys/root:
total 16
-r-----  1 root      staff         3805 Mar  6 10:54 cert.pem
-r-----  1 root      staff          887 Mar  6 10:54 key.pem
-r-----  1 root      staff         2048 Mar  6 10:53 rand.seed
-r-----  1 root      staff          769 Mar  6 10:54 req.pem
```

d. 請輸入以下指令繼續進行 Sun Grid Engine 安裝。

```
# cd $SGE_ROOT
# ./install_execd -csp
```

e. 請依照以第 28 頁的「如何安裝執行主機」一節的步驟 3 為開頭的安裝指示之其餘部分，記得加入 `-csp` 旗號以啟動安裝 script。

在完成所有其餘的安裝步驟後，請參照第 39 頁的「如何產生使用者證明文件和私密金鑰」一節的指示。

▼ 如何產生使用者證明文件和私密金鑰

要使用 CSP 安全性系統，使用者必須能存取使用者特定證明文件和私密金鑰。執行此步驟最方便的方法是建立辨識使用者的文字檔案。

1. 建立並儲存辨識使用者的文字檔案。

使用 `myusers.txt` 檔案的格式，如以下範例所示。（檔案的欄位是 `UNIX_username:Gecos_field:email_address`。）

```
eddy:Eddy Smith:eddy@my.org
sarah:Sarah Miller:sarah@my.org
leo:Leo Lion:leo@my.org
```

2. 做為主要主機上的 `root`，請輸入以下指令。

```
# $SGE_ROOT/util/sgeCA/sge_ca -usercert myusers.txt
```

3. 請輸入以下指令確認。

```
# ls -l /var/sgeCA/port6789/default/userkeys
```

此目錄列表應該產生跟以下類似的輸出資訊。

```
dr-x-----  2 eddy  staff          512 Mar  5 16:13 eddy
dr-x-----  2 sarah staff          512 Mar  5 16:13 sarah
dr-x-----  2 leo   staff          512 Mar  5 16:13 leo
```

4. 請跟每個您列於檔案（在本範例是 `myusers.txt`）中的使用者輸入下列指令，將安全性相關檔案安裝進 `$HOME/.sge` 目錄中。

```
% source $SGE_ROOT/default/common/settings.csh
% $SGE_ROOT/util/sgeCA/sge_ca -copy
```

使用者應該看到以下的確認事項（在本範例是使用者 eddy）。

```
Certificate and private key for user eddy have been installed
```

每個 Sun Grid Engine 的安裝都有安裝對應 COMMD_PORT 編號的子目錄。輸出前發出指令將導致以下的範例，以 myusers.txt 檔案為基礎。

```
% ls -lR $HOME/.sge
/home/eddy/.sge:
total 2
drwxr-xr-x  3 eddy staff      512 Mar  5 16:20 port6789

/home/eddy/.sge/port6789:
total 2
drwxr-xr-x  4 eddy staff      512 Mar  5 16:20 default

/home/eddy/.sge/port6789/default:
total 4
drwxr-xr-x  2 eddy staff      512 Mar  5 16:20 certs
drwx----- 2 eddy staff      512 Mar  5 16:20 private

/home/eddy/.sge/port6789/default/certs:
total 8
-r--r--r--  1 eddy staff      3859 Mar  5 16:20 cert.pem

/home/eddy/.sge/port6789/default/private:
total 6
-r-----  1 eddy staff        887 Mar  5 16:20 key.pem
-r-----  1 eddy staff      2048 Mar  5 16:20 rand.seed
```

▼ 如何檢查證明文件

- 依照您要執行的事項，輸入以下一至數個指令。

顯示證明文件

以一個字串鍵入下列文字（該指令在本指南中因為太長而無法放到一行內），並在 `-in` 和 `~/ .sge` 元件間加入空格。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -in  
~/ .sge/port6789/default/certs/cert.pem -text
```

檢查發出者

以一個字串鍵入下列文字（該指令在本指南中因為太長而無法放到一行內），並在 `-in` 和 `~/ .sge` 元件間加入空格。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -issuer -in  
~/ .sge/port6789/default/certs/cert.pem -noout
```

檢查主題

以一個字串鍵入下列文字（該指令在本指南中因為太長而無法放到一行內），並在 `-in` 和 `~/ .sge` 元件間加入空格。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -subject -in  
~/ .sge/port6789/default/certs/cert.pem -noout
```

顯示證明文件的電子郵件

以一個字串鍵入下列文字（該指令在本指南中因為太長而無法放到一行內），並在 `-in` 和 `~/ .sge` 元件間加入空格。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -email -in  
~/ .sge/default/port6789/certs/cert.pem -noout
```

顯示有效性

以一個字串鍵入下列文字（該指令在本指南中因為太長而無法放到一行內），並在 `-in` 和 `~/ .sge` 元件間加入空格。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -dates -in  
~/ .sge/default/port6789/certs/cert.pem -noout
```

顯示特點

以一個字串鍵入下列文字（該指令在本指南中因為太長而無法放到一行內），並在 `-in` 和 `~/ .sge` 元件間加入空格。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -fingerprint -in  
~/ .sge/port6789/default/certs/cert.pem -noout
```

▼ 如何驗證安裝

主要主機上

1. 登入主要主機。
2. 根據您執行的作業系統，執行以下指令之一。
 - a. 在 BSD 式 UNIX 系統上，輸入以下指令。

```
% ps -ax
```

- b. 在執行 UNIX System 5 式之作業系統的系統上（如 Solaris 作業環境），輸入以下指令。

```
% ps -ef
```

3. 透過輸出尋找類似以下範例的 `sge` 字串。

在 BSD 式 UNIX 系統上，您應該可以看到與下面類似的輸出資訊。

```
14673 p1 S < 2:12 /gridware/sge/bin/solaris/sge_commd
14676 p1 S < 4:47 /gridware/sge/bin/solaris/sge_qmaster
14678 p1 S < 9:22 /gridware/sge/bin/solaris/sge_schedd
```

若是 UNIX System 5 式的系統，您應該可以看到與下面類似的輸出。

```
root 439 1 0 Jun 2 ?3:37 /gridware/sge/bin/solaris/sge_commd
root 439 1 0 Jun 2 ?3:37 /gridware/sge/bin/solaris/sge_qmaster
root 446 1 0 Jun 2 ?3:37 /gridware/sge/bin/solaris/sge_schedd
```

若您沒看到合適的字串，則主要主機上所需之一或數個 Sun Grid Engine 常駐程式就沒有在這機器上執行（不管您是否真的在主要主機上，您都可以查閱檔案 `<sge_root>/<儲存格>/common/act_qmaster`）。前進至下一步。

4. (選擇性的) 手動重新啟動常駐程式。

要取得如何繼續進行的指示，請參閱第 125 頁的「關於常駐程式和主機」一節。

執行主機上

1. 登入您執行 Sun Grid Engine 執行主機安裝程序的執行主機。
2. 請在指令主機程序參考步驟 2，以決定適合您系統的 `ps` 指令，然後輸入該指令。
3. 在輸出中尋找 `sge` 字串。

在 BSD 式 UNIX 系統上，您應該可以看到與下面類似的輸出。

```
14685 p1 S < 1:13 /gridware/sge/bin//sge_commd
14688 p1 S < 4:27 /gridware/sge/bin/solaris/sge_execd
```

若是 UNIX System 5 式的系統，如 Solaris 作業環境，您應該可以看到與下面類似的輸出。

```
root 169 1 0 Jun 22 ? 2:04 /gridware/sge/bin/solaris/sge_commd
root 171 1 0 Jun 22 ? 7:11 /gridware/sge/bin/solaris/sge_execd
```

如果您沒看到類似輸出，執行主機上所需的一或數個常駐程式就沒有在執行。前進至下一步。

4. (選擇性的) 手動重新啓動常駐程式。

要取得如何繼續進行的指示，請參閱第 125 頁的「關於常駐程式和主機」一節。

試驗指令

若兩個必要的常駐程式在主要和執行主機上執行，則 Sun Grid Engine 系統應該為可操作的。發出試驗指令檢查。

1. 登入主要主機或另一個管理主機。

確定將您安裝 Sun Grid Engine 二進位的路徑加入您的標準搜尋路徑中。

2. 從指令行輸入指令。

```
% qconf -sconf
```

這個 `qconf` 指令會顯示目前的全域叢集配置（請參閱第 139 頁的「基礎叢集配置」一節）。若該指令失敗，則最可能是您的 `SGE_ROOT` 環境變數設定不適當或 `qconf` 未能聯絡與 `sge_qmaster` 相關的 `sge_commd`。前進至下一步。

3. 檢查是否 `script` 檔，`<sge_root>/<儲存格>/common/settings.csh` 或 `<sge_root>/<儲存格>/common/settings.sh` 設定環境變數，`COMMD_PORT`。

如果是這樣，請確定環境變數 `COMMD_PORT` 已經設為您再度嘗試上述指令前的特別數值。如果 `COMMD_PORT` 變數沒有用在設定檔中，則從您執行指令之機器上的服務資料庫（如，`/etc/services` 或是 NIS 服務對映）必須提供一個 `sge_commd` 項目。若不是這回事，請將這樣的項目新增至機器的服務資料庫中，並給予其配置在 Sun Grid Engine 主要主機上的數值，然後繼續進行下一步。

4. 重試 `qconf` 指令。

準備提交作業

在您開始提交批次 `script` 至 Sun Grid Engine 系統前，請檢查您站點的標準及您個人的 `shell` 資源檔（`.cshrc`、`.profile` 或 `.kshrc`）是否包含指令，如 `stty`（批次作業沒有預設的終端機連線，因此對 `stty` 的呼叫會造成錯誤）。

1. 登入主要主機。

2. 輸入以下指令。

```
% rsh 執行主機 date
```


執行主機表示其中一個您將使用之已安裝的執行主機（您應該檢查所有的執行主機看看您的登入或初始目錄是否各個主機都不同）。`rsh` 指令應該給您一個非常類似於在主要主機上本端執行之 `date` 指令的輸出。若有任何額外包含錯誤訊息的行列，您必須消除錯誤的原因才能成功執行批次作業。

對所有的指令解譯器來說，您可以在執行指令，如 `stty`，之前，檢查實際的終端機連線。以下為如何執行的 Bourne-/Korn-Shell 範例：

```
tty -s
if [ $?= 0 ]; then
    stty erase ^H
fi
```

C-Shell 語法非常類似：

```
tty -s
if ( $status = 0 ) then
    stty erase ^H
endif
```

3. 提交包含在 `<sge_root>/examples/jobs` 目錄其中一個樣本 `script`。

輸入以下指令。

```
% qsub script 路徑
```

4. 使用 Sun Grid Engine `qstat` 指令監視作業的運作模式。

要取得更多提交和監視批次作業的資訊，請參閱第 67 頁的「提交批次作業」。

5. 在作業完成執行後，請在初始目錄檢查重新導向的 `stdout/stderr` 檔案，`<script 名稱>.e< 作業 id>` 和 `<script 名稱>.o< 作業 id>`，而 `< 作業 id>` 為指定給每個作業之連續獨一無二的整數。

如果有任何問題，請參閱第十一章「疑難排解」。

部分 III 使用 Sun Grid Engine 5.3 軟體

主要針對給使用者使用 — 也就是，不執行系統管理員工作的人員（請參閱第四部分第 121 頁的「管理」） — 《*Sun Grid Engine 5.3 管理和使用者指南*》的這個部分包含三個章節。

- 第三章 – 第 49 頁的「導覽 Sun Grid Engine 5.3 程式」
本章介紹一些 Sun Grid Engine 5.3 的基礎，並包含如何列出不同資源的指示。
- 第四章 – 第 61 頁的「提交作業」
本章提供藉由 Sun Grid Engine 5.3 系統提交作業的完整指示，並以作業提交「練習」開始，讓你能熟悉此程序。
- 第五章 – 第 97 頁的「檢查點、監視和控制作業」
本章解釋作業控制的概念並包含完成不同作業控制作業的指示。

在第三部分的每個章節包含透過 Sun Grid Engine 5.3 系統完成各種作業的背景資訊和詳盡指示。

導覽 Sun Grid Engine 5.3 程式

本章介紹一些協助您使用本軟體的基礎 Sun Grid Engine 5.3 概念和專有名詞。要取得完整的產品背景資訊，包括綜合的字彙，請參閱第一章第 1 頁的「Sun Grid Engine 5.3 簡介」。

本章也包含完成以下作業的指示。

- 第 51 頁的「如何啓動 QMON 瀏覽器」
- 第 51 頁的「如何顯示佇列清單」
- 第 52 頁的「如何顯示佇列屬性」
- 第 55 頁的「如何找出主要主機名稱」
- 第 55 頁的「如何顯示執行主機清單」
- 第 55 頁的「如何顯示管理主機清單」
- 第 55 頁的「如何顯示提交主機清單」
- 第 57 頁的「如何顯示可要求屬性清單」

Sun Grid Engine 使用者種類和操作

使用者種類在 Sun Grid Engine 分爲四個類別。

- **管理員 (Manager)** – 管理員可全權操作 Sun Grid Engine。根據預設值，任何管理佇列機器之超級使用者擁有管理員的特權。
- **操作員 (Operator)** – 操作員可以執行許多和管理員相同的指令，除了進行配置變更，如新增、刪除或修改佇列之外。
- **所有人 (Owners)** – 佇列所有人被允許暫停或啓動其所擁有的佇列或作業，但沒有更進一步的管理權限。
- **使用者 (User)** – 使用者有其特定權限，請參見第 59 頁的「使用者存取權限」，但沒有叢集或佇列管理的能力。

表 3-1 會顯示不同使用者類別可用的 Sun Grid Engine 5.3 指令功能。

表 3-1 使用者類別和相關指令功能

指令	管理員	操作員	所有人	使用者
qacct	完整的	完整的	僅擁有作業	僅擁有作業
qalter	完整的	完整的	僅擁有作業	僅擁有作業
qconf	完整的	無系統設定修改	僅顯示配置和存取權限	僅顯示配置和存取權限
qdel	完整的	完整的	僅擁有作業	僅擁有作業
qhold	完整的	完整的	僅擁有作業	僅擁有作業
qhost	完整的	完整的	完整的	完整的
qlogin	完整的	完整的	完整的	完整的
qmod	完整的	完整的	僅擁有作業和所擁有的佇列	僅擁有作業
qmon	完整的	無系統設定修改	無配置變更	無配置變更
qrexec	完整的	完整的	完整的	完整的
qselect	完整的	完整的	完整的	完整的
qsh	完整的	完整的	完整的	完整的
qstat	完整的	完整的	完整的	完整的
qsub	完整的	完整的	完整的	完整的

佇列和佇列屬性

爲了能在您的站點以最理想狀態使用 Sun Grid Engine 系統，您應該熟悉佇列結構與配置給您 Sun Grid Engine 系統的佇列屬性。

QMON 瀏覽器

Sun Grid Engine 主要功能是圖形使用者介面 (GUI) 指令工具，QMON 瀏覽器。QMON 瀏覽器提供各種的 Sun Grid Engine 功能，包含作業提交、作業控制和重要資訊蒐集。

▼ 如何啓動 QMON 瀏覽器

- 從指令行輸入以下指令

```
% qmon
```

在顯示訊息視窗之後，會出現 QMON 主控制面板，其與以下圖示類似（請參閱圖 1-2 來辨識圖示的意義）。



圖 3-1 QMON 主要控制功能表

本手冊的許多指示需要使用 QMON 瀏覽器。圖示按鈕的名稱，其說明了它們的功能，在滑鼠游標經過時會出現在螢幕上。

（要取得如何自訂 QMON 瀏覽器的指示，請參閱第 8 頁的「自訂 QMON」。）

「佇列控制」 QMON 對話方塊

第 114 頁的「如何用 QMON 控制佇列」一節所顯示和說明的 QMON 「佇列控制」對話方塊提供已安裝的佇列及其目前狀態的快速簡介。

▼ 如何顯示佇列清單

- 輸入以下指令。

```
% qconf -sql
```

▼ 如何顯示佇列屬性

您可以使用 QMON 或指令行顯示佇列屬性。

使用 QMON 瀏覽器

1. 從主要 QMON 功能表，按一下「瀏覽器」圖示。
2. 按一下「佇列」按鈕。
3. 在「佇列控制」對話中，將滑鼠游標移到圖示之上以取得合適的佇列。

圖 3-2 是顯示佇列屬性資訊的部分範例。

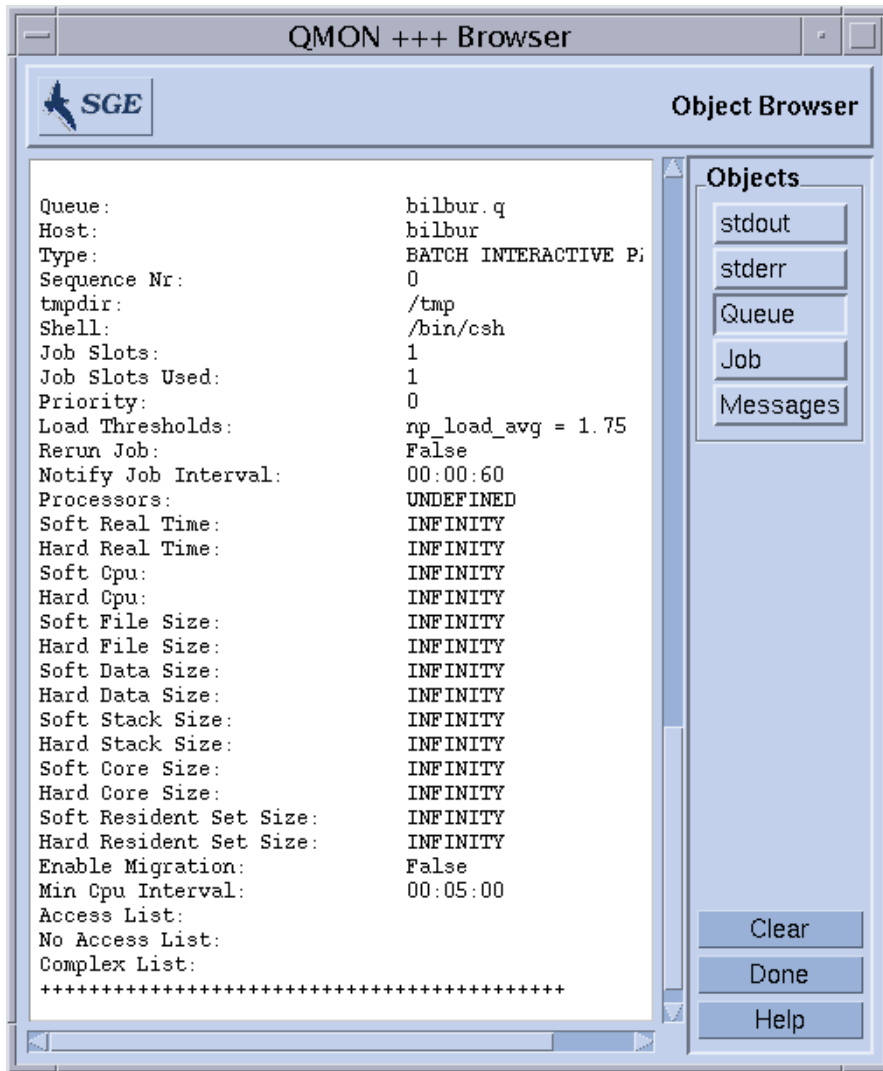


圖 3-2 QMON 瀏覽器的佇列屬性顯示

從指令行

- 輸入以下指令。

```
% qconf -sq 佇列名稱
```

螢幕出現類似顯示於圖 3-2 的資訊。

解譯佇列屬性資訊

您可以在 `queue_conf` 線上說明手冊和《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》的 `queue_conf` 一節找到每個佇列屬性的詳細說明。

以下是一些最重要的參數清單。

- `qname` – 要求的佇列名稱。
- 主機名稱 – 佇列的主機。
- 處理器 – 多重處理器系統的處理器，佇列對其有存取權。
- `qtype` – 允許在此作業執行的作業種類。目前，此為批次、互動、檢查點、平行或以上的任何組合或是傳輸
- 插槽 – 可以同時在該佇列執行的作業數目。
- 所有人清單 – 如第 60 頁的「管理員、操作員和所有人」一節所解釋的佇列所有人
- 使用者清單 – 列在此參數的使用者存取清單（請參閱第 59 頁的「使用者存取權限」）中的使用者或群組辨識器可存取佇列。
- 非使用者清單 (`xuser_lists`) – 列在此參數的使用者存取清單（請參閱第 59 頁的「使用者存取權限」）中的使用者或群組辨識器無法存取佇列。
- 聚合體清單 – 列在此參數的聚合體與此佇列有關，而包含在這些聚合體的屬性會組成一套佇列的可要求屬性（請參閱第 56 頁的「可要求的屬性」）。
- 聚合體數值 – 指定為特定聚合體屬性提供給該佇列的容量（請參閱第 56 頁的「可要求的屬性」）。

主機功能

按一下 QMON 主功能表的「主機配置」鈕會顯示您 Sun Grid Engine 叢集中與主機相關的功能簡介。然而，沒有 Sun Grid Engine 管理者權限，您可能無法將任何變更套用到現有的配置。

主機配置對話在第 125 頁的「關於常駐程式和主機」一節有說明。以下各節提供從指令行擷取這類資訊的指令。

▼ 如何找出主要主機名稱

主要主機的位置對使用者來說應該是透明的 (transparent)，因為主要主機可能隨時會在目前的主要主機和其中一個備用主要主機之間進行遷移。

- 使用文字編輯器打開 `<sgc_root>/<儲存格>/common/act_qmaster` 檔案。
目前主要主機的名稱位在此檔案中。

▼ 如何顯示執行主機清單

要顯示叢集中配置為執行主機的主機清單，請使用指令：

```
% qconf -sel
% qconf -se 主機名稱
% qhost
```

第一個指令顯示目前所有配置為執行主機的名稱清單。第二個指令顯示指定執行主機的詳細資訊。第三個指令顯示執行主機的狀態和載入資訊。要取得透過 `qconf` 顯示的資訊細節，請參考 `host_conf` 線上說明手冊，而如要取得其輸出和進一步選項的細節，請參考 `qhost` 線上說明手冊。

▼ 如何顯示管理主機清單

具管理權限的主機清單可以使用下述指令顯示：

```
% qconf -sh
```

▼ 如何顯示提交主機清單

提交主機清單可以使用下述指令顯示：

```
% qconf -ss
```

可要求的屬性

提交 Sun Grid Engine 作業時，您可指定作業的需求簡介。使用者可指定要使作業順利執行的主機或佇列的屬性或特性。Sun Grid Engine 會將這些作業需求對映到 Sun Grid Engine 叢集的主機和叢集配置，並因此為作業找到合適的主機。

用來指定作業需求的屬性與 Sun Grid Engine 叢集（如網路共享磁碟上所需的空間）、主機（如作業系統架構）或佇列（如允許的 CPU 時間）有關，或這些屬性是由站點策略衍生而來，如僅在某些主機上安裝軟體的可用性。

可用的屬性包含佇列屬性清單（請參閱第 50 頁的「佇列和佇列屬性」）、全域和主機相關屬性清單（請參閱第 165 頁的「聚合體種類」），以及管理員定義的屬性。但是，為了方便，Sun Grid Engine 管理員通常會選擇只定義所有可要求可用屬性的子集。

目前可要求的屬性顯示於 QMON 提交對話方塊的「要求資源 (Requested Resources)」子對話方塊（請參閱圖 3-3）中（要取得如何提交作業的詳細資訊，請參考第 67 頁的「提交批次作業」一節）。它們列在「可用資源 (Available Resources)」選擇清單中。

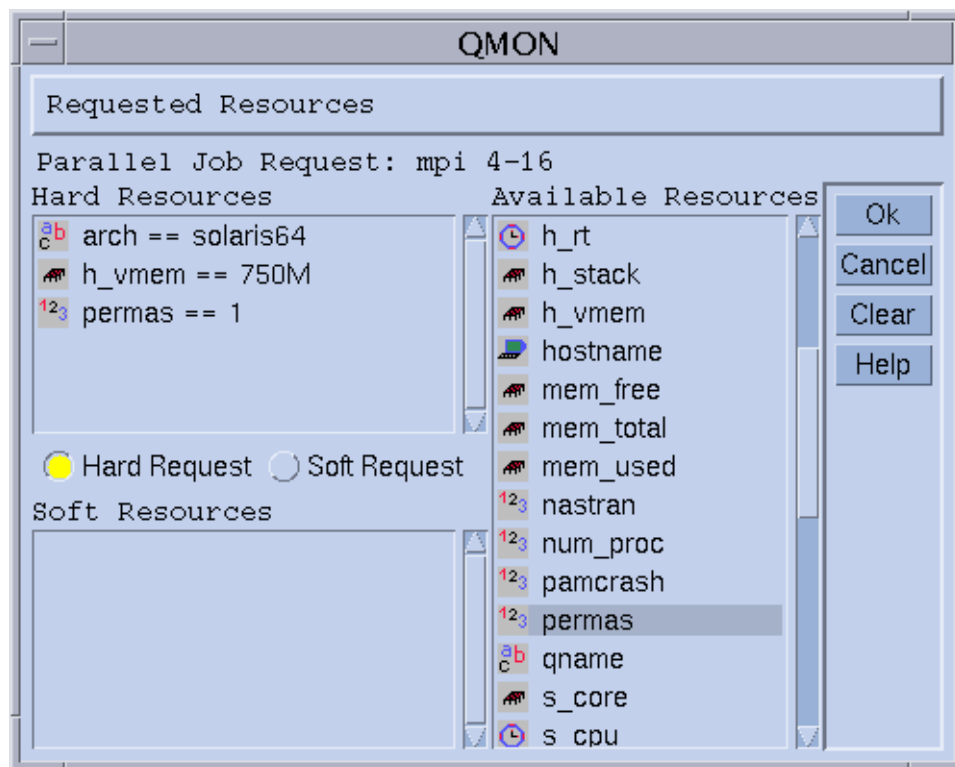


圖 3-3 要求資源對話方塊

▼ 如何顯示可要求屬性清單

1. 從命令行輸入以下指令來顯示配置的聚合體清單：

```
% qconf -scl
```

聚合體包含一套屬性的定義。有三個標準的聚合體：

- global – 對叢集全域屬性來說（選擇性的）
- host – 對主機特定屬性來說
- queue – 對佇列屬性的屬性來說

任何因上述指令而列出之進一步的聚合體名稱即為管理員定義的聚合體（請參閱第 163 頁的「聚合體概念」的第八章，或《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》的聚合體格式說明，以取得更多聚合體的資訊）。

2. 輸入以下指令顯示特定聚合體的屬性：

```
% qconf -sc 聚合體名稱 [...]
```

佇列聚合體的輸出可能看起來如表 3-2 所顯示。

表 3-2 顯示的 queue 聚合體屬性

#Name	Shortcut	Type	Value	Relop	Requestable	Consumable	Default
qname	q	STRING	NONE	==	YES	NO	NONE
hostname	h	HOST	unknown	==	YES	NO	NONE
tmpdir	tmp	STRING	NONE	==	NO	NO	NONE
calendar	c	STRING	NONE	==	YES	NO	NONE
priority	pr	INT	0	>=	NO	NO	0
seq_no	seq	INT	0	==	NO	NO	0
resun	re	INT	0	==	NO	NO	0
s_rt	s_rt	TIME	0:0:0	<=	NO	NO	0:0:0
h_rt	h_rt	TIME	0:0:0	<=	YES	NO	0:0:0
s_cpu	s_cpu	TIME	0:0:0	<=	NO	NO	0:0:0
h_cpu	h_cpu	TIME	0:0:0	<=	YES	NO	0:0:0
s_data	s_data	MEMORY	0	<=	NO	NO	0
h_data	h_data	MEMORY	0	<=	YES	NO	0
s_stack	s_stack	MEMORY	0	<=	NO	NO	0

表 3-2 顯示的 queue 聚合體屬性 (接上頁)

#Name	Shortcut	Type	Value	Relop	Requestable	Consumable	Default
qname	q	STRING	NONE	==	YES	NO	NONE
hostname	h	HOST	unknown	==	YES	NO	NONE
tmpdir	tmp	STRING	NONE	==	NO	NO	NONE
calendar	c	STRING	NONE	==	YES	NO	NONE
priority	pr	INT	0	>=	NO	NO	0
seq_no	seq	INT	0	==	NO	NO	0
h_stack	h_stack	MEMORY	0	<=	NO	NO	0
s_core	s_core	MEMORY	0	<=	NO	NO	0
h_core	h_core	MEMORY	0	<=	NO	NO	0
s_rss	s_rss	MEMORY	0	<=	NO	NO	0
h_rss	h_rss	MEMORY	0	<=	YES	NO	0
min_cpu_interval	mci	TIME	0:0:0	<=	NO	NO	0:0:0
max_migr_time	mmt	TIME	0:0:0	<=	NO	NO	0:0:0
max_no_migr	mmn	TIME	0:0:0	<=	NO	NO	0:0:0

name 一欄基本上和 `qconf -sq` 指令所顯示的第一欄一樣。佇列屬性包含大部分的 Sun Grid Engine 佇列屬性。shortcut 一欄包含第一欄完整名稱的管理員可定義縮寫。使用者可在 `qsub` 指令要求的選項中填入完整名稱或捷徑。

requestable 一欄會告知對應的項目是否在 `qsub` 使用。因此，例如管理員只需將 `qname` 和 / 或 `qhostname` 項目設為不可要求，就可以禁止叢集的使用者直接為其作業要求某機器 / 佇列。這麼做暗示可用的使用者要求一般可以透過多重佇列達成，其會執行 Sun Grid Engine 的載入平衡功能。

relop 一欄定義所用的關係操作以計算佇列是否符合使用者要求。執行的對照 (comparison) 為：

■ User_Request relop Queue/Host/...-Property

如果對照的結果是錯的，則使用者的作業就無法在考慮的佇列中執行。例如，讓佇列 `q1` 配置 100 秒的軟 `cpu` 時間限制（要取得使用者過程限制的說明，請參閱 `queue_conf` 和 `setrlimit` 線上說明手冊），而佇列 `q2` 配置提供 1000 的軟 `cpu` 時間限制。

consumables 和 default 欄位對管理員是有意義的，可讓其宣告所謂的可消耗資源（請參閱第 172 頁的「可耗損資源」一節）。使用者會像其他屬性要求消耗品。然而，資源的 Sun Grid Engine 內部簿記 (bookkeeping) 是不同的。

假設使用者提交以下要求。

```
% qsub -l s_cpu=0:5:0 nastran.sh
```

`s_cpu=0:5:0` 要求（有關語法的細節，請參閱 `qsub` 線上說明手冊）會要求至少授與 5 分鐘軟限制 `cpu` 時間的佇列。因此，只有至少提供 5 分鐘軟 CPU 執行時間限制的佇列才能正確配置以執行作業。

注意 – 如果有一個以上的佇列可以執行作業，則 Sun Grid Engine 只會考慮排程過程中的工作量資訊。

使用者存取權限

Sun Grid Engine 管理員可以限制某些使用者或使用者群組存取佇列或其他 Sun Grid Engine 功能（如，平行環境介面；請參閱第 89 頁的「平行作業」）。

注意 – Sun Grid Engine 會自動考慮由叢集管理配置的存取限制。以下各節在您想要查詢個人的存取權限時才會對您有重要性。

爲了限制存取權限，管理員建立並維護所謂的存取清單（或簡稱 *ACL*）。ACL 包含仲裁使用者和 UNIX 群組名稱。然後，在佇列或平行環境介面配置中將 ACL 加入存取允許或存取拒絕清單（請分別參閱《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》第五節的 `queue_conf` 或 `sge_pe`）。

屬於列於存取允許清單的 ACL 之使用者有存取佇列或平行環境介面的權限。做爲存取拒絕清單中 ACL 成員之使用者無法存取相關的資源。

透過 QMON 主功能表的「使用者配置」圖示鈕開啓的「使用者集配置」對話方塊可讓您透過「使用者集配置」對話方塊查詢您擁有存取權的 ACL。相關細節請參考第九章第 189 頁的「管理使用者存取與策略」。

您可用以下的指令從指令行取得目前配置的 ACL 清單：

```
% qconf -sul
```

在一或多重存取清單的項目可用以下指令列出：

```
% qconf -su acl 名稱 [...]
```

ACL 由使用者帳號名稱和以字首 "@" 符號辨識爲 UNIX 群組名稱的 UNIX 群組名稱所組成。如此您可以決定您的帳號要屬於哪個 ACL。

注意 – 如果您有使用 `newgrp` 指令交換主 UNIX 群組的權限，則您的存取權可能會改變。

您現在可以檢查您有存取權或存取權被拒絕的佇列或平行環境介面。請查詢第 50 頁的「佇列和佇列屬性」和第 216 頁的「如何以 QMON 配置 PE」所述的佇列或平行環境介面配置。存取允許清單名為 `user_lists`。存取拒絕清單名為 `xuser_lists`。若您的使用者帳號或主要 UNIX 群組和存取允許清單有關，則您被允許存取相關資源。若您與存取拒絕清單有關，則您無法存取佇列或平行環境介面。若兩個清單都是空的話，則每個擁有有效帳號的使用者都可以存取相關資源。

管理員、操作員和所有人

Sun Grid Engine 管理員清單可以透過以下方式取得：

```
% qconf -sm
```

操作員清單則可透過以下方式取得：

```
% qconf -so
```

注意 – Sun Grid Engine 管理主機的超級使用者被視為預設的管理員。

使用者，其為某個佇列的所有人，包含在第 50 頁的「佇列和佇列屬性」一節所述的佇列配置資料庫。要取得該資料庫您可執行：

```
% qconf -sq 佇列名稱
```

相關的佇列配置項目稱為 `owners`。

提交作業

本章提供使用 Sun Grid Engine 5.3 提交作業以供處理的相關背景資訊和指示。本章以執行簡單作業的範例開始，然後會繼續介紹執行複雜作業的指令。

本章包含完成以下作業的指令。

- 第 61 頁的「如何從指令行執行簡單作業」
- 第 63 頁的「如何提交作業，從圖形使用者介面，QMON」
- 第 82 頁的「如何從指令行提交作業」
- 第 84 頁的「如何從指令行提交陣列作業」
- 第 85 頁的「如何使用 QMON 提交陣列作業」
- 第 86 頁的「如何使用 QMON 提交互動作業」
- 第 89 頁的「如何使用 qsh 提交互動作業」
- 第 89 頁的「如何使用 qlgin 提交互動作業」

執行簡單作業

使用本節的資訊和指令來熟悉提交 Sun Grid Engine 5.3 作業的基本程序。

注意 – 若您已在未經授權的帳號 (unprivileged account) 下安裝 Sun Grid Engine 程式，您必須以該特殊使用者的身分登入才能執行作業（詳情請參閱第 20 頁的「先決條件作業」）。

▼ 如何從指令行執行簡單作業

在執行任何 Sun Grid Engine 指令之前，您必須先正確設定可執行搜尋路徑和其他環境條件。

1. 根據您的指令解譯器，輸入以下兩個指令之一：

a. 若您使用 `csh` 或 `tcsh` 做為指令解譯器：

```
% source sge_root_dir/default/common/settings.csh
```

`sge_root_dir` 指定在安裝程序之初，所選取的 Sun Grid Engine root 目錄的位置。

b. 若您使用 `sh`、`ksh` 或 `bash` 做為指令解譯器：

```
# . sge_root_dir/default/common/settings.sh
```

注意 – 您可以將以上的指令加到 `.login`、`.cshrc` 或 `.profile` 檔中（哪一個都適合），以確保您之後啟動所有的互動階段作業會有正確的 Sun Grid Engine 設定。

2. 將以下的簡單作業 script 提交到 Sun Grid Engine 叢集中。

您可以在 Sun Grid Engine root 目錄中的 `examples/jobs/simple.sh` 檔案找到以下的作業。

```
#!/bin/sh
#This is a simple example of a Sun Grid Engine batch script
#
# Print date and time
date
# Sleep for 20 seconds
sleep 20
# Print date and time again
date
# End of script file
```

輸入以下指令，其假設 `simple.sh` 是儲存上述 script 的 script 檔的名稱，而該檔案位於您目前工作的目錄下。

```
% qsub simple.sh
```

`qsub` 指令應該以下列方式確認成功的作業提交。

```
your job 1 ("simple.sh") has been submitted
```

3. 輸入以下指令以擷取作業上的狀態資訊。

```
% qstat
```

您應該收到包含目前所有 Sun Grid Engine 系統所辨識的作業相關的狀態報告，以及每個作業所謂的*作業 ID*（包含在提交確認中之獨一無二的號碼）、作業 *script* 的名稱、作業所有人、狀態資訊（*r* 表示執行中）、提交或啟動時間，和最後，作業執行的佇列名稱。

如果 *qstat* 指令沒有產生輸出的話，則系統無法實際辨識作業。例如，您的作業可能已經結束。您可以檢查 *stdout* 和 *stderr* 重新導向 (*redirection*) 檔案來控制完成作業的輸出。根據預設值，這些檔案是在執行作業的主機上之作業所有人起始目錄中產生的。這些檔案名稱是由作業 *script* 檔案名稱、附加的點符號，此符號之後的 "o" 表示 *stdout* 檔案、"e" 表示 *stderr* 檔案，而最後是獨一無二的作業 ID。因此，您可以分別在 *simple.sh.o1* 和 *simple.sh.e1* 的名稱下找到作業的 *stdout* 和 *stderr* 檔案（如果該作業是在新安裝的 Sun Grid Engine 系統上首次執行的話）。

▼ 如何提交作業，從圖形使用者介面，QMON

提交和控制 Sun Grid Engine 作業以及取得 Sun Grid Engine 系統簡介的簡單方法就是圖形使用者介面，QMON。相較其他的軟體，QMON 為提交和監視作業的作業提供了一個作業提交功能表和「作業控制」對話方塊。

From the command line prompt, 鍵入以下指令。

```
% qmon
```

在電腦啟動期間，會顯示一個訊息視窗並出現 QMON 主要功能表。

4. 按「作業控制」鈕，然後按「提交作業」鈕。



圖 4-1 QMON 主要功能表

會出現「作業提交」和「作業控制」對話方塊（請分別參閱圖 4-2 和圖 4-3）。當您將滑鼠指標移到按鈕上時，畫面就會顯示按鈕名稱（如「作業控制」）。

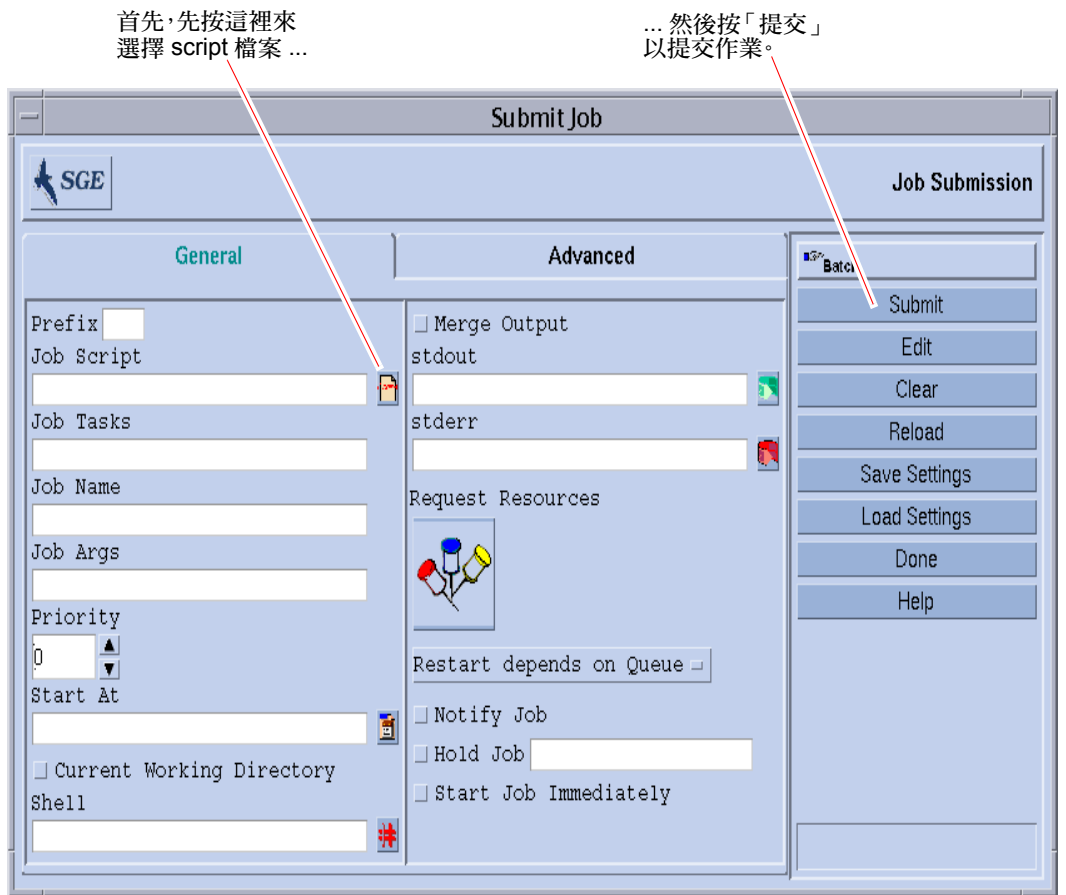


圖 4-2 QMON 作業提交對話方塊



圖 4-3 QMON 作業控制對話方塊

5. 在作業提交功能表中，按「作業 Script」檔案選擇圖示來開啓檔案選擇方塊。
6. 按下適當的檔名，以選擇 script 檔案（例如，來自指令行範例的檔案 *simple.sh*）。
7. 按下「作業提交」功能表底部的「提交」鈕。

過了幾秒後，您應該就可以在「作業控制」畫面監視您的作業了。您會先在「擱置中作業」下看到作業；而當作業啟動時，它會迅速移動到「執行中作業」。

提交批次作業

以下各節說明如何透過 Sun Grid Engine 5.3 程式提交較複雜的作業。

關於 Shell Script

Shell script，又稱為批次作業，原則是組合在檔案中命令行指令的序列。Script 檔是因 `chmod` 指令變成可以執行。若呼叫指令，就會啟動正確的指令解譯器（例如，`csh`、`tcsh`、`sh` 或 `ksh`），而每個指令都由都解譯為使用者手動鍵入執行 script。您可以從 shell script 中呼叫隨意的指令、應用程式和其他 shell script。

適當的指令解譯器是否呼叫為 `login-shell`，是根據它的名稱（`csh`、`tcsh`、`sh`、`ksh`、...）是否事實上包含在執行作業的特定主機和佇列的 Sun Grid Engine 配置 `login_shells` 選項的 [數值] 清單 (value list) 中。

注意 – 配置於叢集的不同主機和佇列的 Sun Grid Engine 配置可能不同。您可以透過 `qconf` 指令的 `-sconf` 和 `-sq` 選項來顯示有效的配置（詳細的資訊請參考《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》）。

如果指令解譯器被呼叫為 `login-shell`，您作業的環境將和您登入並執行 script 的環境完全相同。例如，在使用 `csh` 時，除了系統預設啟動資源檔之外，將執行 `.login` 和 `.cshrc`（就好像 `/etc/login`），但如果 `csh` 不是呼叫為 `login-shell`，則僅會執行 `.cshrc`。是否呼叫為 `login-shell` 的不同處之相關說明，請參考您選擇的指令解譯器的手冊。

Script 檔的範例

程式範例 4-1 是簡單 shell script 的範例，其會先從 Fortran77 來源編譯應用程式，`flow`，然後加以執行：

```
#!/bin/csh

# This is a sample script file for compiling and
# running a sample FORTRAN program under Sun Grid Engine.

cd TEST

# Now we need to compile the program 'flow.f' and
# name the executable 'flow'.

f77 flow.f -o flow
```

程式範例 4-1 簡單 Shell Script

您的本端系統使用者指南會提供建立和自訂 shell script 的詳細資訊（您可能也想看看 sh、ksh、csh 或 tcsh 線上說明手冊）。在以下各節中，重點是在需要考慮的特性，好為 Sun Grid Engine 準備批次 script。

一般來說，您可以把您要手動從指令提示符號執行的所有 shell script 提交到 Sun Grid Engine，只要它們不需要終端機連接（除了標準錯誤和輸出裝置外，其可以自動重新導向）且不需要互動使用者介入。因此，程式範例 4-1 已經準備好要提交到 Sun Grid Engine，並會執行所需的動作。

使用 QMON 提交延伸和進階作業

在嘗試較複雜的作業提交形式前 — 延伸或進階 — 了解一些重要的程序背景資訊是有用的。以下各節提供該資訊。

延伸範例

「作業提交」對話方塊的標準形式（請參閱圖 4-2）提供了配置以下延伸作業的參數的方法：

- 用於嵌入式 script Sun Grid Engine 提交選項的字首字串（愈取得詳細資訊請參閱第 80 頁的「作用中的 Sun Grid Engine 註釋」一節）
- 要使用的作業 script
 - 若按下相關檔案的按鈕，檔案選擇方塊就會開啓（請參閱圖 4-3）
- 提交陣列作業的作業 ID 範圍（請參閱第 84 頁的「陣列作業」）
- 作業名稱（選擇作業 script 之後就會設定預設值）
- 作業 script 的引數

- 作業的初始優先權數值
沒有管理者或操作者權限只能降低他們的初始優先權數值。
- 作業認為適合執行的時機
若按下相關的檔案按鈕，就會開啓說明對話方塊，以便輸入正確格式的時間（請參閱圖 4-4）
- 表示作業是否在目前工作的目錄下執行的旗號（僅針對提交和可能的執行主機間相同的目錄階層）
- 用於執行作業 script 的指令解譯器（請參閱第 79 頁的「如何選擇指令解譯器」）
若按下相關的按鈕，就會開啓說明對話方塊，以便輸入作業的指令解譯器規格（請參閱圖 4-5）。
- 表示作業的標準輸出和標準錯誤輸出是否合併成標準輸出串流的旗號
- 要使用的標準輸出重新導向（請參閱第 79 頁的「輸出重新導向」）
如果沒指定，將使用預設值。若按下相關的檔案按鈕，就會出現說明者對話方塊，以便輸入輸出重新導向的替代方式（請參閱第 79 頁的「輸出重新導向」）。
- 要使用的標準錯誤輸出重新導向 — 和標準輸出重新導向非常相似
- 作業的資源需求
若要定義您作業所需的資源，請按對應的圖像鈕。如果作業要求資源，則圖像鈕的顏色會改變。
- 定義作業在系統當機或類似事件後而中止的作業是否能重新啓動，或重新啓動運作方式是根據佇列或由作業所要求的選擇清單鈕
- 表示作業在即將暫停或取消時，是否作業由 SIGUSR1 或 SIGUSR2 信號通知的旗號
- 表示要分派給作業的使用者保留 (user hold) 或作業相依 (job dependency) 的旗號
只要被指定任何類型的保留，作業就不適合執行（若要取得更多關於保留的資訊，請參閱第 101 頁的「監視和控制 Sun Grid Engine 作業」一節）。附接於「保留 (Hold)」旗號的輸入欄位僅將保留限制在特定陣列作業的作業範圍中（請參閱第 84 頁的「陣列作業」）。
- 強迫作業立刻啓動（如果可能的話）或被拒絕的旗號
如果選擇此旗號，則作業不會排在佇列中。

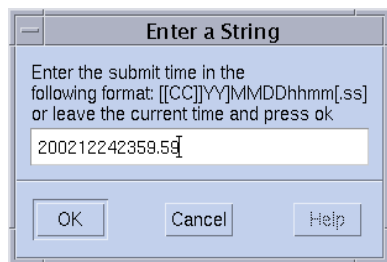


圖 4-4 在時間輸入方塊中

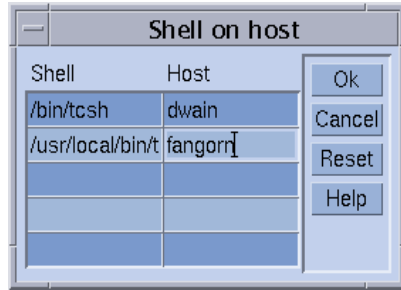


圖 4-5 Shell 選擇方塊

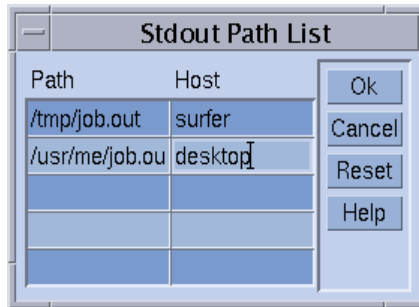


圖 4-6 輸出重新導向方塊

「作業提交」螢幕右側的按鈕讓您初始啓動不同的動作：

- 提交 – 依照提交對話方塊的指定來提交作業。
- 修改 – 使用 vi 或 \$EDITOR 環境變數所定義的編輯器修改 X 終端機裡選擇的 script 檔。
- 清除 – 清除在「作業提交」對話方塊中的所有設定值，包括任何的指定資源需求。
- 重新載入 – 重新載入指定的 script 檔、語法解析 (parse) 任何嵌入式 script 選項（請參閱第 80 頁的「作用中的 Sun Grid Engine 註釋」一節）、語法解析預設設定值（請參閱第 83 頁的「預設要求」一節）並刪除這些設定值的中間手動變更。該動作相當於「清除」動作，其規格為前一個 script 檔的規格。若已經選擇 script 檔，則該選項只會顯示一種效果。
- 儲存設定值 – 將目前的設定值儲存到檔案。檔案選擇方塊已開啓，以便選擇檔案。儲存的檔案可能之後會明確載入（請參閱下文）或可能用為預設要求（請參閱第 83 頁的「預設要求」一節）。
- 載入設定值 – 載入之前用「儲存設定值」鈕所儲存的設定值（請參閱上文）。載入的設定值會覆寫目前的設定值。
- 完成 – 關閉「作業提交」對話方塊。
- 說明 – 顯示對話方塊特定說明。

圖 4-7 顯示已設定大部分參數的「作業提交」對話方塊。

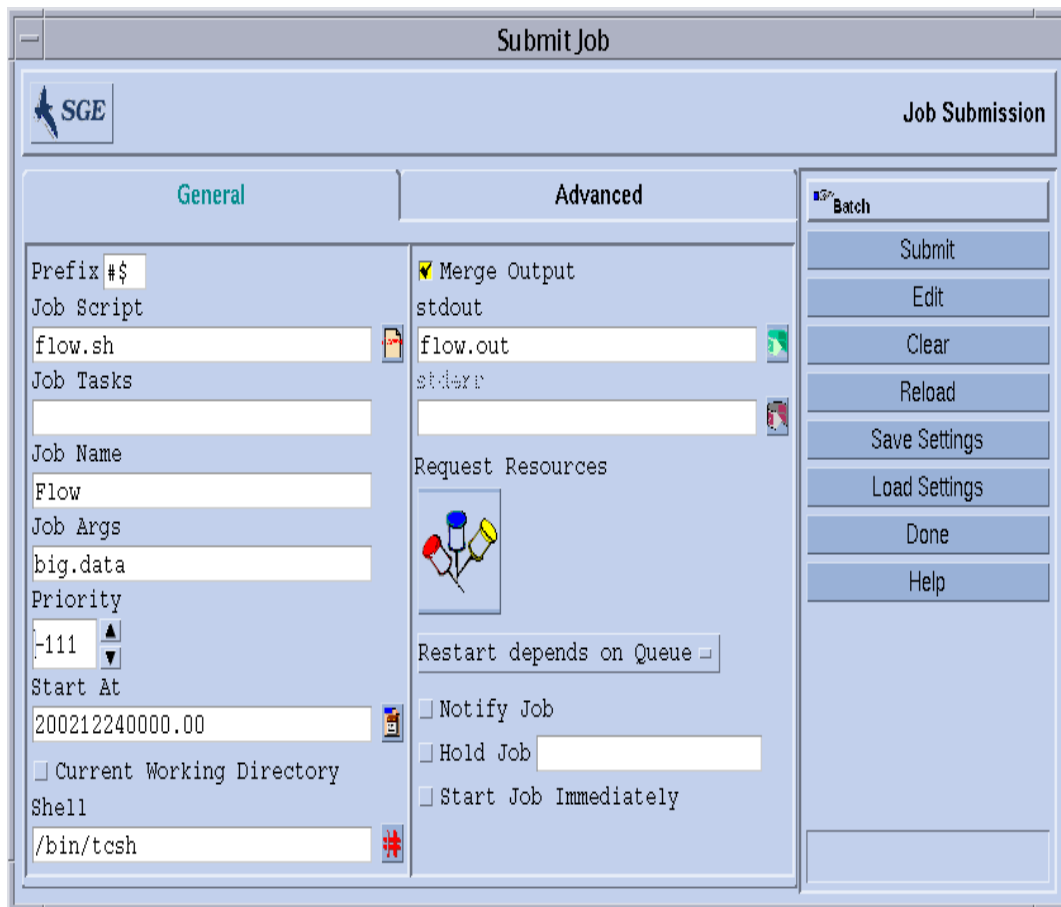


圖 4-7 延伸作業提交範例

配置在範例中的作業擁有 script 檔，flow.sh，其必須常駐在 QMON 工作目錄中。此作業叫做 Flow，script 檔採用單一引數，big.data。本作業以優先權 -111 啟動，並在 2002 年 12 月 24 日午夜前都適合執行。本作業會在提交工作目錄中執行，並將使用 tcsh 指令解譯器。最後，標準輸出和標準錯誤輸出將合併進檔案，flow.out，該檔案也會建立在目前的工作目錄中。

進階範例

「進階」提交螢幕允許以下額外的參數定義：

- 將使用平行環境介面和所需的處理範圍（請參閱第 89 頁的「平行作業」一節）
- 作業執行前需為作業設定的一套環境變數。

若按下相關的按鈕，就會出現說明者對話方塊，供要輸出的環境變數定義使用（請參閱圖 4-8）。環境變數能從 QMON 的執行時間環境取得，或可以定義仲裁的環境變數。

- 名為環境 (Context) 的名稱 / 數值配對的清單（請參閱圖 4-9），可以用來儲存並傳送可以從 Sun Grid Engine 叢集中存取的作業相關資訊。

環境變數可以透過 `-ac/-dc/-sc` 選項至 `qsub`、`qrsh`、`qsh`、`qlogin` 或 `qalter` 從指令行修改，並可透過 `qstat -j` 取得。

- 檢查點適合作業的情況下，要使用的檢查點環境（請參閱第 97 頁的「關於檢查點作業」一節）
- 與作業相關的帳號字串

帳號字串將加到為作業而做的帳號紀錄，並在之後可用於帳號分析。

- 「驗證 (Verify)」旗號，其判定作業一致性檢查模式

若要檢查作業要求的一致性，Sun Grid Engine 會採用空白且未載入的叢集，並嘗試至少找出一個作業可執行的佇列。可能的檢查模式為：

- 略過 - 完全沒有一致性的檢查。
 - 警告 - 雖然有不一致性的報告，但仍接受作業（如果叢集配置在作業提交後進行變更會比較符合要求）。
 - 錯誤 - 有不一致性的報告，如碰上任何作業，該作業會遭拒絕。
 - 剛驗證 (Just verify) - 將不會提交作業，但會產生一份作業對叢集中每個主機和佇列之適合性的廣泛報告。
- 關於使用者是透過電子郵件通知的事件

作業的事件啟動 / 結束 / 中斷 / 暫停目前已定義。

- 通知郵件寄送的電子郵件位址清單

如果按下相關按鈕，就會出現說明者對話來定義寄件清單（請參閱圖 4-10）。

- 要求強制選擇做為作業執行之佇列名稱清單。

在符號清單項目 (bulleted list item)，第 69 頁的「作業的資源需求」，所述的對應資源需求將硬佇列 (Hard Queue) 清單和軟佇列 (Soft Queue) 清單看做同樣的事物。

- 適合做為平行作業的主要佇列的佇列名稱清單。

平行作業在主要佇列啟動。所有作業產生至平行作業的其他佇列稱為副佇列 (*slave queues*)。

- 需要在啟動提交作業前順利完成的作業之 ID 清單

剛建立的作業取決於這些作業的順利完成。

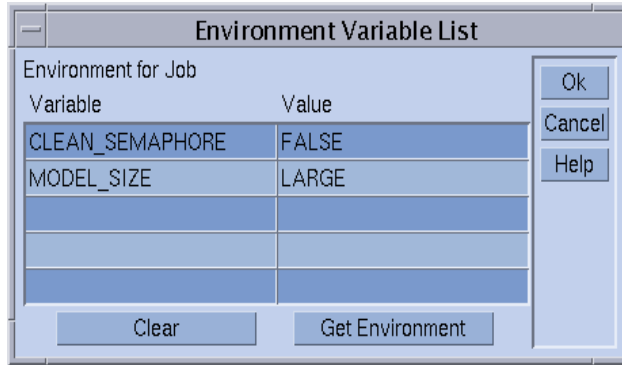


圖 4-8 作業環境定義

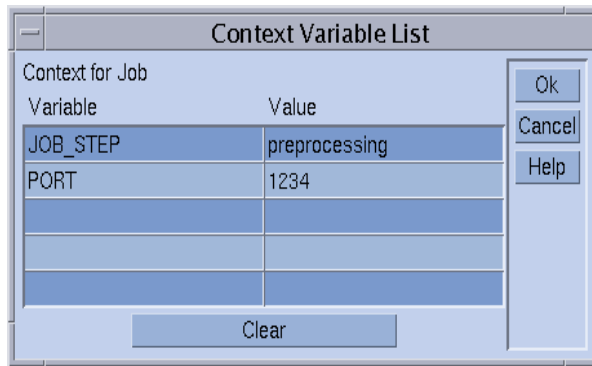


圖 4-9 作業上下文定義

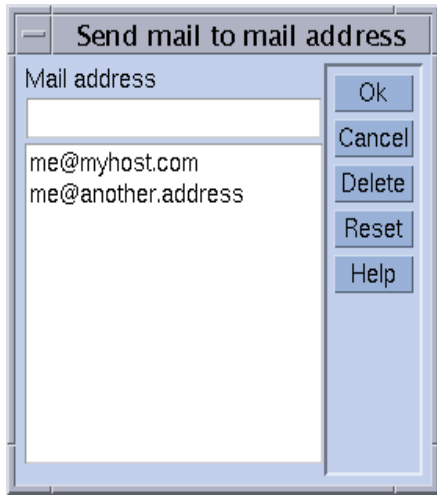


圖 4-10 郵件地址規格

定義在圖 4-11 的作業和第 68 頁的「延伸範例」一節的作業定義相比，有以下額外的特性。

- 作業需要使用平行環境 `mpi`。它需要建立至少 4 個平行過程，而最多可以使用 16 個過程（如果有的話）。
- 已設定並輸出兩個作業環境變數。
- 已設定兩個上下文變數。
- 帳號字串 `FLOW` 將新增至作業帳號紀錄中。
- 如果因為系統當機造成作業失敗，則該作業將重新啟動。
- 如果偵測到作業要求和叢集配置間有不一致之處，則應該列印出警告。
- 在作業開始和結束時，郵件應該傳送到兩個電子郵件地址的清單。
- 作業最好是在佇列 `big_q` 中執行。

圖 4-11 顯示進階作業提交的範例。

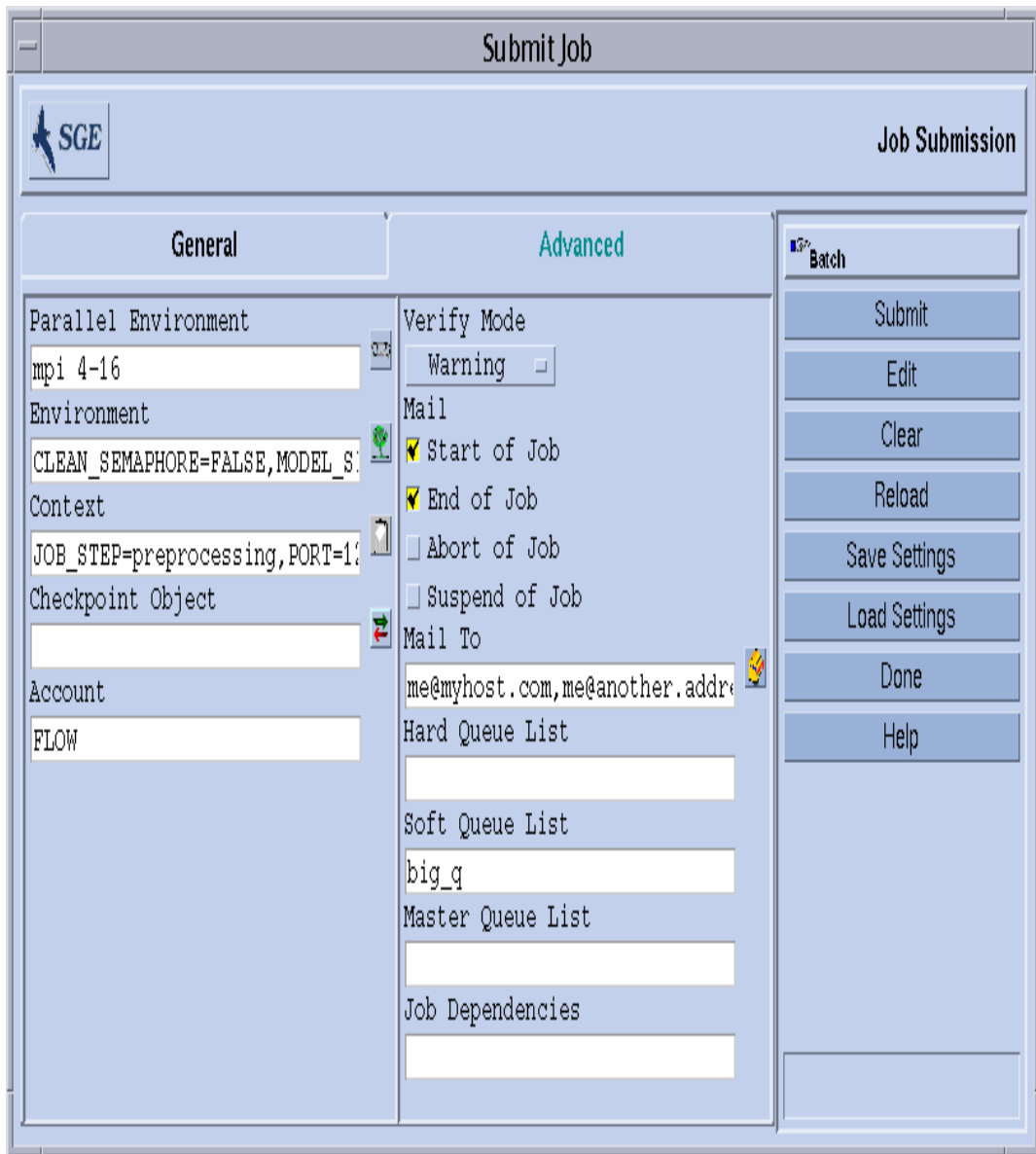


圖 4-11 進階作業提交範例

資源需求定義

在目前的範例中，使用的提交選項沒有表達任何對執行作業的主機的需求。Sun Grid Engine 會假設這樣的作業可以在任何主機上執行。然而在練習中，大多數的作業要求在執行的主機上能達成一定的先決條件，才能夠順利完成。這樣的先決條件為足夠的可用記憶體、要求軟體的安裝或一定的作業系統架構。另外，叢集管理通常會限制叢集內機器的使用。例如，作業所允許消耗的 CPU 時間通常會有限制。

在使用者對叢集的裝備及其使用策略不了解的情況下，Sun Grid Engine 會提供使用者找出適合使用者作業之主機的方法。使用者必須做的是指定使用者作業的需求，並讓 Sun Grid Engine 管理找尋適合且低負載主機的作業。

資源需求透過第 56 頁的「可要求的屬性」一節所說明的 *可要求屬性* 來指定。QMON 提供一個非常簡便的指定作業需求的方法。在按下「作業提交」對話方塊中的「要求資源」鈕後會開啓「要求資源」對話方塊（請參閱圖 4-12 的範例）只會顯示目前在「可用資源」選擇清單中合適的屬性。在屬性上連按兩下可使該屬性新增至作業的硬或軟（請參閱以下解釋）資源清單中，而且（除了設為 True 的 BOOLEAN 類型屬性之外）會開啓說明者對話方塊，以引導您輸入相關屬性的數值規格。

顯示在圖 4-12 的範例「要求資源」對話方塊作業的資源簡介，其中要求一個 solaris64 主機和至少提供 750 MB 記憶體的可用 permas 授權。如果找到一個以上符合此規格的佇列，則任何定義的軟資源需求都算在內（本範例中沒有）。然後，如果沒找到符合硬和軟需求的佇列，任何認可硬資源的佇列都可視為合適的佇列。

注意 – 只有在一個以上的佇列適合作業時，載入標準會決定從哪裡啓動作業。

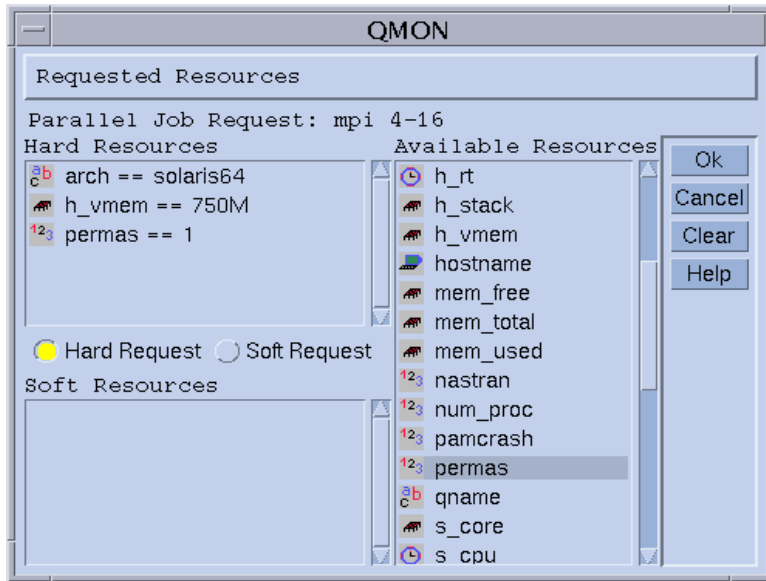


圖 4-12 要求資源對話方塊

注意 – INTEGER 屬性 `permas` 是透過管理員延伸引進到「全域」聚合體，STRING 屬性 `arch` 是從「主機」聚合體匯入的，而 MEMORY 屬性 `h_vmem` 是從「佇列」聚合體匯入的。

相等的資源需求簡介也可以從 `qsub` 指令行提交：

```
% qsub -l arch=solaris64,h_vmem=750M,permas=1 \
permas.sh
```

注意 – 在第一個 `-l` 選項之前的隱含 (implicit) `-hard` 開關已被略過。

750 MB 的記號 `750M` 是 Sun Grid Engine 數量語法 (quantity syntax) 的範例。對要求記憶體消耗的屬性來說，您可以指定由所謂的乘數由後新增的 (appended) 整數小數 (integer decimal)、浮點小數 (floating point decimal)、八進位整數 (integer octal) 和十六進位整數 (integer hexadecimal)。

- `k` – 將數值乘以 1000。
- `K` – 將數值乘以 1024。
- `m` – 將數值乘以 1000 次 1000。
- `M` – 將數值乘以 1024 次 1024。

八進位常數由開頭的 0 (zero) 和 0 到 7 之間的數字所指定。指定十六進位的常數需要由前新增 (prepend) 0x 的數字，然後使用 0 到 9、a 到 f 和 A 到 F 等數字。如果沒有乘數由後新增 (append)，則數值是以位元組來計算的。若使用浮點小數 (floating point decimal)，結果的數值在整數後的數值將遭刪除。

對那些有加上時間限制的屬性來說，您可以使用時數、分數或秒數和三者任何的組合來指定時間數值。時數、分數和秒數是以冒號分隔的十進位數字所指定的。3:5:11 的時間即表示 11111 秒。如果指定時數、分數或秒數的數字是 0，則該數字在保有冒號的情況下可以留空。因此 :5: 的數值即表示 5 分鐘。用於上述的「要求資源」對話方塊的表格是為延伸的部分，只在 QMON 有效。

Sun Grid Engine 系統如何配置資源

如上一節所示，了解 Sun Grid Engine 軟體如何處理資源要求和如何配置資源是很重要的。以下提供了 Sun Grid Engine 軟體的資源配置演算法的概略檢視。

1. 在所有預設要求檔案中讀取並進行語法分析（請參閱第 83 頁的「預設要求」一節）。
2. 為嵌入式選項處理 script 檔（請參閱第 80 頁的「作用中的 Sun Grid Engine 註釋」一節）。
3. 當作業提交時，讀取所有 script 嵌入式選項，不管它們在 script 檔的位置。
4. 從指令行讀取所有的要求並進行語法分析。

一旦收集所有的 qsub 要求，硬和軟要求就會分開處理（會先處理硬的）。這些要求會對應以下的優先順序進行評估。

1. script/ 預設要求檔案從左到右
2. script/ 預設要求檔案從上到下
3. 指令行從左到右

換句話說，指令行可以用來置換嵌入式旗號。

已配置要求為硬的資源。如果要求無效，提交將遭拒絕。如果在提交時間無法符合一或多個要求（例如，要求的佇列忙碌中），此作業就會被置於佇列中並在之後重新排程。如果符合所有的硬要求，這些要求就會配置而作業就可以執行。

已檢查要求為軟的資源。即使沒有符合一些或全部的要求，作業仍然可以執行。如果多重佇列（已經符合硬需求）提供部分的軟體資源清單（重疊或不同的部分），Sun Grid Engine 軟體將選擇提供最多軟要求的佇列。

此作業將啟動，並涵蓋配置資源。

使用執行 UNIX 指令如 hostname 或 date 的小型測試 script 檔來實驗取得引數清單選項和嵌入式選項或硬與軟需求如何影響對方的經驗是有用的。

延伸至一般的 Shell Script

如果在 Sun Grid Engine 的控制下執行，則有些延伸部分會到影響 script 運作模式一般 shell script 的部分。以下幾節說明這些延伸部分。

如何選擇指令解譯器

用來處理作業 script 檔案的指令解譯器可以在提交時間指定（請參閱，例如，圖 4-7）。然而，如果沒有指定，則配置變數，`shell_start_mode`，會決定如何選擇指定解譯器：

- 如果 `shell_start_mode` 設為 `unix_behavior`，script 檔的第一行—如果以 `"#!"` 序列開始的話—將會被評估來決定指令解譯器。如果第一行沒有 `"#!"` 序列，Bourne Shell `sh` 會根據預設值使用。
- 對其他的 `shell_start_mode` 設定值來說，使用的是用 `shell` 參數為作業啟動的佇列配置的預設指令解譯器（請參閱第 50 頁的「佇列和佇列屬性」一節和 `queue_conf` 線上說明文件）。

輸出重新導向

因為批次作業沒有裝端機連線，所以其標準輸出和標準錯誤輸出必須重新導入檔案中。Sun Grid Engine 允許使用者定義輸出重新導入檔案的位置，但如果沒有指定，就會使用預設值。

檔案的標準位置在目前作業執行的工作目錄中。預設標準輸出檔案名稱為 `< 作業名稱 > .o < 作業 id >`，預設標準錯誤輸出重新導向至 `< 作業名稱 > .e < 作業 id >`。`< 作業名稱 >` 是從 script 檔案名稱建立或可以由使用者定義（範例請參閱 `qsub` 線上說明手冊的 `-N` 選項）。`< 作業 id >` 是由 Sun Grid Engine 指定給作業的獨一無二的辨識器。

如果是陣列作業的工作（請參閱第 84 頁的「陣列作業」一節），作業辨識器會加到以句點符號分隔的檔案名稱中。因此，產生的標準重新導向路徑為 `< 作業名稱 > .o < 作業 id > . < 工作 id >` 和 `< 作業名稱 > .e < 作業 id > . < 工作 id >`。

如果標準位置不適合，使用者可使用圖 4-11 和圖 4-6 所顯示的 `QMON` 來指定輸出方向，或使用 `-e` 和 `-o qsub` 選項。標準輸出和標準錯誤輸出可以合併到一個檔案，而更改方向是可以以執行主機為單位而指定的。例如，根據作業執行的主機，輸出更改方向的檔案位置也變的不同。要建立自訂但獨一無二的更改方向檔案路徑，虛擬環境變數可用來和 `qsub -e` 與 `-o` 選項一起使用。這些變數的清單如下。

- `$HOME` – 執行機器上的起始目錄
- `$USER` – 作業所有人的使用者 ID
- `$JOB_ID` – 目前的作業 ID
- `$JOB_NAME` – 目前的作業名稱（請參閱 `-N` 選項）
- `$HOSTNAME` – 執行主機名稱
- `$TASK_ID` – 陣列作業作業索引編號

這些變數在作業執行期間擴張至真正的數值，而更新路徑會隨著它們建立。

欲知更進一步的細節，請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》的 `qsub` 項目。

作用中的 Sun Grid Engine 註釋

之前有 `#` 符號的行被視為 `shell script` 裡的註釋。然而，`Sun Grid Engine` 會辨識特殊的註釋行，並以特別的方式使用它們。這種 `script` 行的其他部分會被視為 `Sun Grid Engine` 提交指令 `qsub` 的指令行引數清單的一部份。在這些特殊註釋行裡所供應的 `qsub` 選項也是由 `QMON` 「作業提交」對話方塊所解譯的，而當選取 `script` 檔時，對應的參數就會預先設定。

預設值的特殊註釋行是靠 `#$` 字首字串辨識的。字首字串可以使用 `qsub -C` 選項重新定義。

說明的機制稱為提交引數的 `script` 嵌入。以下為使用 `script` 嵌入指令行選項的 `script` 檔案範例。

```
#!/bin/csh
#Force csh if not Sun Grid Engine default #shell
#$ -S /bin/csh

# This is a sample script file for compiling and
# running a sample FORTRAN program under Sun Grid Engine.
# We want Sun Grid Engine to send mail when the job begins
# and when it ends.
#$ -M EmailAddress
#$ -m b,e

# We want to name the file for the standard output
# and standard error.
#$ -o flow.out -j y

# Change to the directory where the files are located.
cd TEST

# Now we need to compile the program 'flow.f' and
# name the executable 'flow'.
f77 flow.f -o flow

# Once it is compiled, we can run the program.
flow
```

程式範例 4-2 使用 `Script` 嵌入式指令行選項

環境變數

在執行 Sun Grid Engine 作業時，預設到作業環境的變數數目如下所列。

- ARC – 作業執行節點的 Sun Grid Engine 架構名稱；此名稱會編碼進 `sge_execd` 二進位
- COMMD_PORT – 指定 TCP 埠，而 `sge_commd(8)` 預料將在其上聽取通訊需求
- SGE_ROOT – 在啓動或預設 `/usr/SGE` 之前，Sun Grid Engine root 目錄設為 `sge_execd`
- SGE_CELL – 作業執行的 Sun Grid Engine 儲存格
- SGE_JOB_SPOOL_DIR – `sge_shepherd(8)` 所用來在作業執行期間儲存作業相關資料的目錄
- SGE_O_HOME – 作業提交主機上之作業所有人的起始目錄
- SGE_O_HOST – 作業提交的主機
- SGE_O_LOGNAME – 作業提交主機上之作業所有人的登入名稱
- SGE_O_MAIL – 在作業提交指令上下文中 MAIL 環境變數的內容
- SGE_O_PATH – 在作業提交指令上下文中 PATH 環境變數的內容
- SGE_O_SHELL – 在作業提交指令上下文中 SHELL 環境變數的內容
- SGE_O_TZ – 在作業提交指令上下文中 TZ 環境變數的內容
- SGE_O_WORKDIR – 作業提交指令的工作目錄
- SGE_CKPT_ENV – 指定檢查點作業執行的檢查點環境（如使用 `qsub -ckpt` 選項選擇一般）
- SGE_CKPT_DIR – 只為檢查點作業設定；包含檢查點介面的路徑 `ckpt_dir`（請參閱 checkpoint 線上說明手冊）
- SGE_STDERR_PATH – 作業的標準錯誤串流轉移的檔案路徑名稱；一般用來增加來自 `prolog`、`epilog`、平行環境啓動 / 停止或檢查點 `script` 的錯誤訊息的輸出
- SGE_STDOUT_PATH – 作業的標準錯誤串流轉移的檔案路徑名稱；一般用來增加來自 `prolog`、`epilog`、平行環境啓動 / 停止或檢查點 `script` 的訊息的輸出
- SGE_TASK_ID – 此作業代表的陣列作業裡的作業辨識器
- ENVIRONMENT – 請每次都設為 `BATCH`；此變數表示 `script` 在批次模式下執行
- HOME – 來自 `passwd` 檔案的使用者起始目錄路徑
- HOSTNAME – 作業執行節點的主機名稱
- JOB_ID – 當作業提交時，由 `sge_qmaster` 指定的獨一無二的辨識器；作業 ID 是範圍到 99999 的十進位整數
- JOB_NAME – 作業名稱，是由 `qsub script 檔案名稱`、時期 (period) 和作業 ID 的數字所建立的；此預設值可能會被 `qsub -N` 所覆寫
- LOGNAME – 來自 `passwd` 檔案的使用者登入名稱
- NHOSTS – 平行作業所使用的主機數目
- NQUEUES – 配置給作業的佇列數目（序列作業請永遠配置一個）
- NSLOTS – 平行作業所使用的佇列插槽數目

- PATH – 預設的 shell 搜尋路徑：/usr/local/bin:/usr/ucb:/bin:/usr/bin
- PE – 作業執行的平行環境（僅限平行作業）
- PE_HOSTFILE – 包含由 Sun Grid Engine 指定給平行作業的虛擬平行機器之定義的檔案路徑
有關檔案格式的細節，請參閱 sge_pe 裡 \$pe_hostfile 參數的說明。環境變數僅供平行作業使用。
- HOSTNAME – 作業執行佇列的主機名稱
- REQUEST – 作業的要求名稱，該名稱是作業 script 檔案名稱或是透過 qsub -N 選項清楚指定給作業的名稱
- RESTARTED – 表示，不管檢查點作業是否已經重新啓動；如果設定（為數值 1），作業至少已中斷一次，並因此重新啓動。
- SHELL – 來自 passwd 檔案的使用者登入 shell

注意 – 這不見得就是作業使用中的 shell。

- TMPDIR – 至作業暫時工作目錄的完整路徑
- TMP – 和 TMPDIR 相同；為 NQS 相容性之用
- TZ – 從 sge_execd 匯入的時區變數，如設定的話
- USER – 來自 passwd 檔案的使用者登入名稱

▼ 如何從命令行提交作業

- 輸入 qsub 指令，和適當的引數。

例如，使用 script 檔案名稱 flow.sh 的簡單作業 — 如第 61 頁的「如何從命令行執行簡單作業」一節所說明 — 可以使用以下指令提交：

```
% qsub flow.sh
```

然而，放棄延伸 QMON 作業提交的對等結果 — 如圖 4-7 所示 — 看起來如下：

```
% qsub -N Flow -p -l11 -a 200012240000.00 -cwd \  
-S /bin/tcsh -o flow.out -j y flow.sh big.data
```

可增加進一步的指令行選項，以組成更複雜的要求。例如，顯示在圖 4-11 的進階作業要求看起來如下：

```
% qsub -N Flow -p -l11 -a 200012240000.00 -cwd \  
-S /bin/tcsh -o flow.out -j y -pe mpi 4-16 \  
-v SHARED_MEM=TRUE,MODEL_SIZE=LARGE \  
-ac JOB_STEP=preprocessing,PORT=1234 \  
-A FLOW -w w -r y -m s,e -q big_q\  
-M me@myhost.com,me@other.address \  
flow.sh big.data
```

預設要求

上一節的最後一個範例顯示進階的作業要求可能會變的複雜而不方便，特別是如果類似的需求需要經常提交的話。為避免輸入這種指令行而造成難以處理且容易產生錯誤的作業，使用者可在 `script` 檔插入 `qsub` 選項（請參閱第 80 頁的「作用中的 Sun Grid Engine 註釋」）或使用所謂的**預設要求**。

叢集管理可能為所有的 Sun Grid Engine 使用者設定預設要求檔案。另一方面，使用者能在使用者的起始目錄建立私人的預設要求檔案，並能在工作目錄中建立應用程式專用預設要求檔案。

預設要求檔案只包含根據預設值在單一或多重行上應用到 Sun Grid Engine 作業的 `qsub` 選項。叢集全域預設要求檔案的位置為 `<sge_root>/<儲存格>/common/sge_request`。私人的一般預設要求檔案位在 `$HOME/.sge_request` 之下，而應用程式專用的預設要求檔案預期放在 `$cwd/.sge_request`。

如果這些檔案中可使用一個以上，它們就可以依照優先順序合併成一個預設的要求：

1. 全域預設要求檔案。
2. 一般私人預設要求檔案。
3. 應用程式專用的預設要求檔案。

注意 – `Script` 嵌入和 `qsub` 指令行比預設要求檔案有更高的優先權。因此，`script` 嵌入會覆寫掉預設要求檔案的設定值，而 `qsub` 指令行選項可能會再次覆寫掉這些設定值。

注意 – `qsub -clear` 選項可以在任何時候在預設要求檔案、嵌入式 `script` 指令和 `qsub` 指令中使用，來刪除任何先前的設定值。

下面是一個私人預設要求檔案的範例。

```
-A 我的專案 -cwd -M me@myhost.com -m b,e  
-r y -j y -S /bin/ksh
```

除非已經覆寫，特定使用者的所有作業之帳號字串將為我的專案、作業將在目前的工作目錄執行、在作業開始和結束時，郵件通知將傳送到 *me@myhost.com*、作業將在系統當機後重新啟動、標準輸出和標準錯誤輸出將合併，而 *ksh* 將被當成指令解譯器使用。

陣列作業

參數化並重複執行相同的一套操作（包含在作業 *script* 中）對 Sun Grid Engine 陣列作業功能是理想的應用程式。這種應用程式的典型範例位在作業（如描繪）可在 Digital Content Creation 產業找到。在此範例中，動畫演算會分割成數個畫面，而相同的描繪運算可為每個畫面單獨執行。

陣列作業功能提供了一個方便提交、監視和控制這類應用程式的方法。另一方面，Sun Grid Engine 提供陣列作業有效的實行方法，在獨立作業的陣列加入單一作業時處理運算。陣列作業的作業是經陣列索引編號提供做為參考的。所有作業的索引橫跨整個陣列作業的索引範圍，此範圍是在單一 *qsub* 指令提交陣列作業期間所定義的。

陣列作業可以以一個總數或由獨立作業或作業子集來監視和控制（如暫停、恢復或取消），在此情況下，對應的索引編號尾端會加上作業 ID 以供做作業參考。當作業執行時（跟一般作業非常相似），它們可以使用環境變數 *\$SGE_TASK_ID* 來取得它們自己的作業索引編號，並存取指定給該作業辨識器的輸入資料集。

▼ 如何從命令行提交陣列作業

- 輸入 *qsub* 指令和適當的引數。

以下為提交陣列作業的範例。

```
% qsub -l h_cpu=0:45:0 -t 2-10:2 render.sh data.in
```

-t 選項定義作業索引範圍。在此情況下，使用的是每兩個索引時（規格的 *:2* 部分），*2-10:2* 會指定 *2* 是最低而 *10* 是最高的索引編號。因此，陣列作業包含 5 個作業，其作業索引為 2、4、6、8 和 10。每個作業會要求 45 分鐘的硬 CPU 時間。（*-l* 選項），而一旦由 Sun Grid Engine 發送並啟動，將會執行作業 *script render.sh*。作業可使用 *\$SGE_TASK_ID* 找出它們是否為作業 2、4、6、8 或 10，而作業可以使用其索引編號在資料檔案 *data.in* 中找出它們的輸入資料記錄。

▼ 如何使用 QMON 提交陣列作業

- 依照第 63 頁的「如何提交作業，從圖形使用者介面，QMON」的指示，並另外考慮以下的注意事項。

注意 – 從 QMON 提交的陣列作業事實上與第 63 頁的「如何提交作業，從圖形使用者介面，QMON」所述的運作方式一致。唯一的不同是在圖 4-7 顯示的作業的作業輸入視窗需要包含和 `qsub -t` 選項一樣語法的作業範圍規格。若要取得陣列索引語法的詳細資訊，請參考《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `qsub` 項目。

第 101 頁的「監視和控制 Sun Grid Engine 作業」與第 113 頁的「從指令行控制 Sun Grid Engine 作業」，和有關 `qstat`、`qhold`、`qrls`、`qmod` 和 `qdel` 的《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》各節，包含一般監視和控制 Sun Grid Engine 作業和特別的陣列作業之持續資訊。

注意 – 陣列作業提供所有一般作業可辨識的 Sun Grid Engine 功能的完整存取。特別是，它們同時可以是平行作業，或可與其他作業產生相互依賴。

提交互動作業

提交互動作業而非批次作業在您的作業需要您的直接輸入來影響作業結果的情況時有用。這個例子通常是針對 X 視窗應用程式（其定義為互動），或針對需要立即結果的說明，來控制進一步運算的作業。

三個存在於 Sun Grid Engine 系統以建立互動作業的方法。

- `qlogin` – 這是個類似 `telnet` 的階段作業，其在由 Sun Grid Engine 軟體選取的主機上啟動。
- `qrsh` – 這相當於標準的 UNIX `rsh` 功能。指令不是在 Sun Grid Engine 系統選取的主機上遠端執行，就是如果沒有指定執行指令，則遠端 `rlogin` (`rlogin`) 階段作業就會在遠端主機上啟動。
- `qsh` – 這是從執行作業的機器上叫出的 `xterm`，其顯示根據您的 `DISPLAY` 環境變數的規格或設定值而設定。如果沒有設定 `DISPLAY` 變數，而顯示目的也有明確定義，Sun Grid Engine 會在提交互動作業的主機上將 `xterm` 導向 X 伺服器的 0.0 畫面。

注意 – 如要正確運作，所有功能都需要正確的 Sun Grid Engine 叢集參數配置。`qsh` 需定義正確的 `xterm` 執行路徑，而此類型的作業需能使用互動佇列。請聯絡您的系統管理員，看看您的叢集是否已經準備好執行互動作業了。

互動作業的預設處理和批次作業的處理不同，因為互動作業如果在提交時不能執行，它們就不會進入排程。這會立刻顯示沒有足夠的適當資源來分派提交後的互動作業。使用者在這樣的情況下會收到 Sun Grid Engine 叢集目前太過忙碌的通知。

這套預設的運作方式可以使用 `-now no` 選項變更至 `qsh`、`qlogin` 和 `qrsh`。若指定該選項，則互動作業會像批次作業一樣進入排程。使用 `-now yes`，使用 `qsub` 提交的批次作業也可以像互動作業般處理，並會立刻分派執行或被拒絕。

注意 – 互動作業只能在 INTERACTIVE（互動）類型的佇列中執行（詳情請參考第 145 頁的「關於佇列配置」）。

接下來的各節簡述 `qlogin` 和 `qsh` 功能的用法。`qrsh` 指令在第 91 頁的「透明遠端執行」一節較廣義的上下文 (broader context) 中有解釋。

使用 QMON 提交互動式作業

唯一一種可以從 QMON 提交的互動式作業是那些在 Sun Grid Engine 選取的主機上叫出 `xterm` 的作業。

▼ 如何使用 QMON 提交互動式作業

- 按住「作業提交」對話方塊右邊按鈕欄上的圖示，直到出現「互動 (Interactive)」圖示。這會使「提交作業」對話方塊準備好提交互動作業（請參閱圖 4-13 和圖 4-14）。

對話方塊中選項的意義和用法和第 67 頁的「提交批次作業」一節中批次作業的意義與用法相同。根本的不同處是幾個輸入欄位設為不回應，因為它們不需套用到互動作業

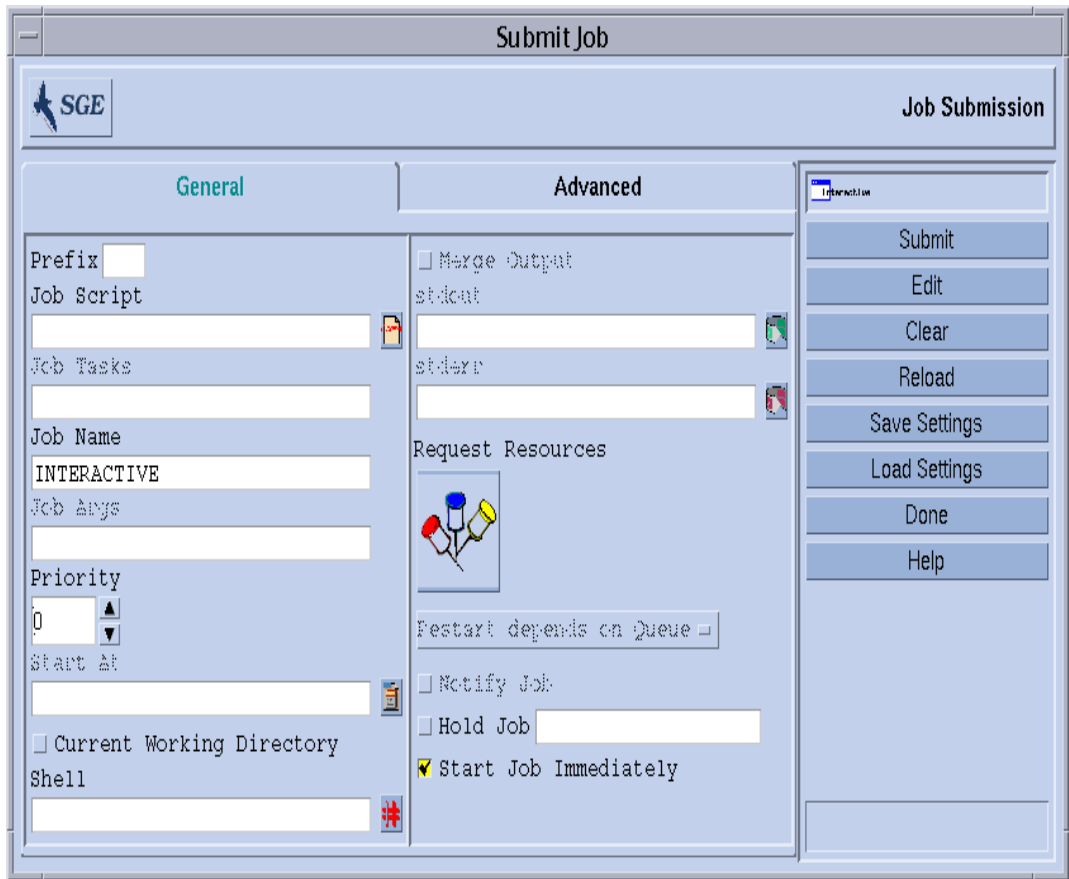


圖 4-13 互動作業提交對話方塊，一般

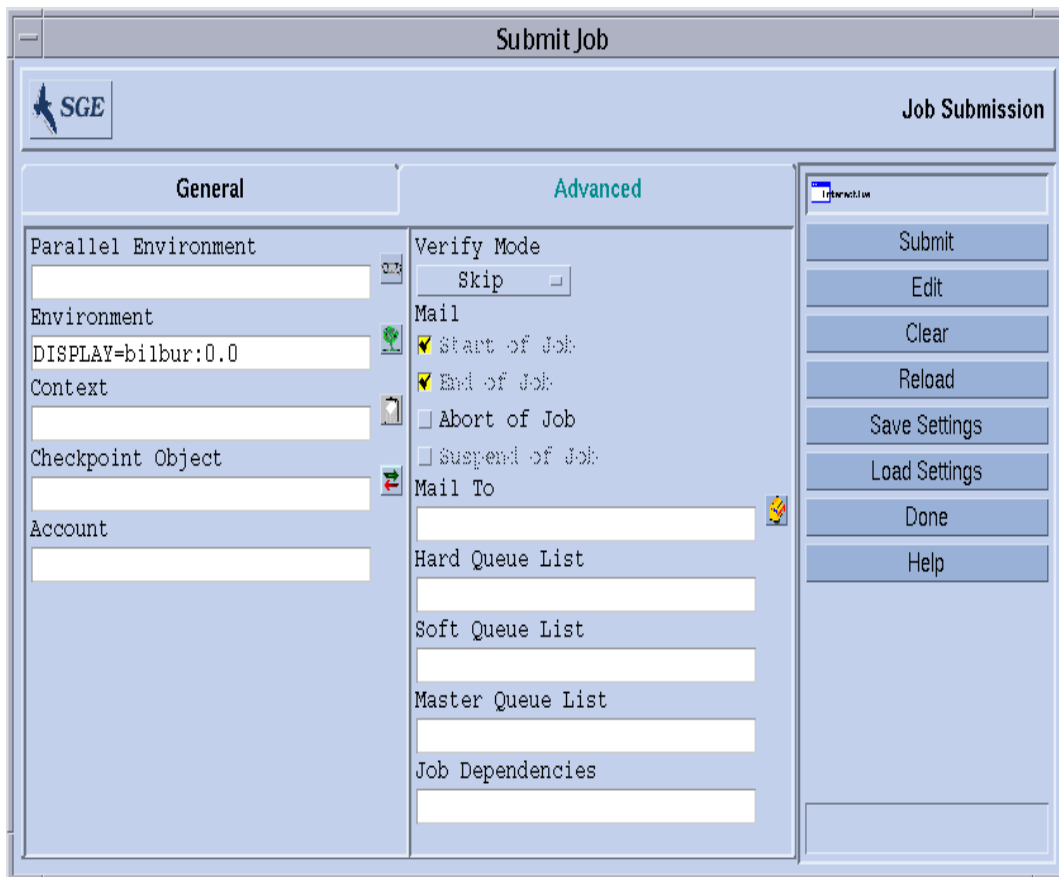


圖 4-14 互動作業提交對話方塊，進階

使用 qsh 提交互動式作業

Qsh 和 qsub 非常類似，並支援數個 qsub 選項，以及額外的開關 `-display` 來指向呼叫的 `xterm` 的顯示（詳細資訊請參考《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `qsh` 項目）

▼ 如何使用 qsh 提交互動作業

- 輸入以下指令在任何可用的 Sun Solaris 64 位元作業系統主機上啟動 `xterm`。

```
% qsh -l arch=solaris64
```

使用 qlogin 提交互動式作業

您可以從任何終端機或終端模擬使用 `qlogin` 指令，從 Sun Grid Engine 控制下初始啟動互動階段作業。

▼ 如何使用 qlogin 提交互動作業

- 輸入以下指令來找出有 Star-CD 授權和一個至少可提供 6 小時硬 CPU 時間限制的低負載主機。

```
% qlogin -l star-cd=l,h_cpu=6:0:0
```

注意 – 根據配置給 Sun Grid Engine 系統使用的遠端登入功能，您可能必須在登入提示符號輸入使用者名稱、密碼或以上兩者。

平行作業

Sun Grid Engine 系統提供如何使用類似 PVM 或 MPI 之仲裁訊息傳遞環境（請見《*PVM User's Guide*》與《*MPI User's Guide*》以取得詳細資訊）、或利用單一佇列中或跨多重佇列與機器而分配的多重插槽之共享記憶體平程式，來執行平行作業的方法。不同的平行環境 (PE) 介面之仲裁數目可以同時進行配置。PE 的相關細節請參閱第十章，第 215 頁的「管理平行環境」。

Sun Grid Engine 作業的排程方式

Sun Grid Engine 5.3 軟體實質上使用兩套標準進行作業排程：

- 作業優先權
- 相同等份

作業優先權

有關不同作業的排程優先順序，會根據預設值使用先進先出(fifo)規則。所有擱置中（尚未排程）的作業都插進一個清單中，第一個提交的作業放在清單的開頭，緊接著是第二個提交的作業，並依此順序排列。第一個提交的作業會最先嘗試排程。若至少有一個合適的佇列可用，則該作業就會排入時程。不管第一個是否已分派，Sun Grid Engine 軟體都會在之後為第二個作業排程。

叢集管理可以使用指定給作業的優先權數值駁回擱置中作業的優先順序。實際的優先權數值可以使用 `qstat` 指令顯示（優先權數值包含在名為 `p` 的擱置中作業顯示之最後一欄裡；相關細節請參考第 110 頁的「如何使用 `qstat` 監視作業」一節）。在提交時，指定給作業的預設優先權數值為 0。優先權數值為正負整數，而擱置中作業清單會以遞減的優先權數值順序對應分類。指定作業相對的高優先權數值會使該作業移至擱置中作業清單的頂端。即使剛提交作業，還是會插入優先權數值為負的作業。如果有幾個作業的優先權數值相同，則 `fifo` 規則會套用在該優先權類型裡。

相同等份排程 (Equal-share-scheduling)

`fifo` 規則有時會造成問題，特別是如果使用者容易在同一時間提交一連串的作業（即透過 `shell-script` 一個接著一個提交）。所有之後提交和指定至佇列相同群組的作業都需等候非常長的一段時間。**相同等份排程** 會將已經擁有執行作業的使用者的作業分配到優先清單的末端來避免此問題。作業分類只會在具有相同優先權數值種類的作業間執行。如果 Sun Grid Engine 排程器配置項目 `user_sort`（相關細節請參考 `sched_conf` 線上說明手冊）設為 `TRUE`，相同等份排程就會啟動。

佇列選擇

Sun Grid Engine 系統不會分派要求非指定佇列的作業，如果這些作業不能立刻啟動。這樣的作業會在 `sge_qmaster` 標示多工緩衝處理，並會不時重新排程。因此，這樣的作業會分派到下個可用的適合佇列。

相反地，由名稱要求至某的佇列的作業會直接前往這個佇列，不管它們是否會啟動或進入多工緩衝處理。因此，以資訊工程**批次佇列**檢視 Sun Grid Engine 佇列是對名稱要求的作業唯一有效的方法。用非特定要求提交的作業會使用 `sge_qmaster` 的多工緩衝處理機制做為佇列排序，然後利用一個較抽象和較有彈性的佇列排序概念。

如果作業已經過排程，而多重、未使用的佇列符合其資源要求，作業通常會分派給屬於承載量最低的主機之佇列（最合適的）。藉由將 Sun Grid Engine 排程器配置項目 `queue_sort_method` 設定為 `seq_no`，叢集管理可能變更這個載入相依計劃到固定順序演算法中：佇列配置項目 `seq_no` 是用最低序列編號來定義指定佇列最高優先權的在佇列間的優先順序。

透明遠端執行

Sun Grid Engine 提供一套支援某些運算作業之透明遠端執行的緊密相關的功能。此功能的核心工具是第 91 頁的「使用 `qrsh` 遠端執行」一節所說明的 `qrsh` 指令。兩個建立在 `qrsh` 之上的高階功能 — `qtcs` 和 `qmake` — 允許透過 Sun Grid Engine 的隱含運算作業做透明分配，因此加強了標準 UNIX 功能 `make` 和 `csh`。`qtcs` 的解釋位於第 92 頁的「使用 `qtcs` 的透明作業分配」一節中，而 `qmake` 的解釋位於第 94 頁的「帶有 `qmake` 的平行檔案製作過程 (Parallel Makefile Processing)」一節中。

使用 `qrsh` 遠端執行

`Qrsh` 建立在 `rsh` 功能的周圍（有關 `rsh` 範圍的細節可參閱 `<sg_e_root>/3rd_party` 裡的資訊），並可以用於多種用途。

- 透過可與標準 UNIX 功能 `rsh`（又稱為 HP-UX 的 `remsh`）相比的 Sun Grid Engine 來提供互動應用程式的遠端執行。
- 透過可類似標準 UNIX 功能 `rlogin` 的 Sun Grid Engine，來提供互動登入階段作業功能（請注意，`qlogin` 仍需要為 UNIX `telnet` 功能的 Sun Grid Engine 代表）。
- 欲考慮提交再執行時支援終端機 I/O（標準 / 錯誤輸出和標準輸入）和終端機控制的批次作業。
- 提供一個方法來提交沒有嵌入 `shell-script` 的獨立程式。
- 提供批次作業提交客戶端，其在作業擱置或執行時會保持活動，而當作業完成或取消時才會結束。
- 允許 Sun Grid Engine 系統控制的作業遠端執行作業（如平行作業的同步作業）在平行作業配置的分散資源架構內（請參閱第 223 頁的「PE 與 Sun Grid Engine 軟體之緊密整合」一節）。

藉由這些功能，`qrsh` 是 `qtcs` 和 `qmake` 功能實施，以及 Sun Grid Engine 與平行環境（如 MPI 或 PVM）所謂的緊密整合的主要啟動基本架構。

`qrsh` 用法

`qrsh` 指令的一般表格為：

```
% qrsh [options] program|shell-script [arguments] \  
      [> stdout_file] [>&2 stderr_file] [< stdin_file]
```

`qrsh` 幾乎了解全部的 `qsub`，並提供幾個額外的選項。

- `-now yes|no` – 該選項控制作業是否立刻接受排程，並在沒有適當資源時就會拒絕該作業，因此通常適用於互動作業（因此此為預設值），或如果作業無法在提交時間啟動時，是否該作業會像批次作業般進入排程。
- `-inherit -qrsh` 不會通過 Sun Grid Engine 排程過程來啟動作業，但它會假設其已經嵌入在平行作業的上下文中，而該平行作業在指定的遠端執行主機上已經有配置的適當資源。這個 `qrsh` 形式通常用在 `qmake` 和緊密平行環境整合之中。預設值不會繼承外部的作業資源。
- `-verbose` – 這個選項可顯示排程過程上的輸出。此選項主要是除錯，因此根據預設值是關閉的。

使用 `qtcsch` 的透明作業分配

`qtcsch` 是大家熟知並廣為使用的 UNIX 之完全相容的替代品。C-Shell (`csch`) 導數 `tcsh` (`qmake` 建立於 `tcsh` 的周圍 - 有關 `tcsh` 範圍的細節可參閱 `<sgc_root>/3rd_party` 裡的資訊)。它透過 Sun Grid Engine 提供指令 `shell` 給適當且低負載的主機指定應用程式之透明分配執行的延伸。哪個應用程式要遠端執行而哪些條件適用在執行主機的選取都定義在名為 `.qtask` 的配置檔案。

這樣的應用程式對使用者是透明的，但可透過 `qrsh` 功能提交 Sun Grid Engine 執行。因為 `qrsh` 提供標準輸出、錯誤輸出和標準輸入處理以及到遠端執行應用程式的終端控制連線，所以遠端執行此類的應用程式和在 `shell` 相同主機上執行這類的應用程式只有三點明顯的差異處。

- 遠端主機可能比本端主機更加合適（功能更強、負載更低、已安裝需求硬 / 軟體資源），而本端主機可能連應用程式的執行都不允許。當然這是預想到的不同處。
- 遠端啟動作業或透過 Sun Grid Engine 的處理會導致小小的延遲。
- 管理員可以透過互動作業 (`qrsh`) 和 `qtcsch` 限制資源的使用。如果沒有足夠的適當資源來透過 `qrsh` 功能啟動應用程式，或如果所有的系統都已過度載入，則隱含的 `qrsh` 提交將會失敗，並傳回對應的錯誤訊息 (`Not enough resources ... try later`)。

除了標準的使用外，`qtcsch` 也是協力廠商代碼和工具整合的適當平台。在單一執行表格中使用 `qtcsch` 在整合環境中的 `qtcsch -c 應用程式名稱` 呈現一個幾乎不需要變更的持續介面。所有所需的應用程式、工具、整合、站點，甚至是使用者特定配置都包含在適當定義的 `.qtask` 檔中。更進一步的優點是該介面可以從任何類型的 `shell script`、C 程式，甚至是 Java 應用程式之內使用。

`qtcsch` 用法

`qtcsch` 的呼叫和對 `tcsh` 的呼叫完全相同。`Qtcsch` 藉由對 `.qtask` 檔的支援和提供一套特別化的 `shell` 內建模式來延伸 `tcsh`。

.qtask 檔定義如下。檔案中的每一行都有以下的格式：

```
% [!]appl_name qrsh_options
```

開頭選擇性的驚嘆號 (!) 定義叢集全域 .qtask 檔和 qtcsh 使用者的個人 .qtask 檔之中衝突定義間的優先權。若在叢集全域檔案中沒有驚嘆號的話，使用者檔案中最後衝突的定義將被拒絕。若驚嘆號在叢集全域檔案中，則不能覆寫對應的定義。

行的其他部分指定應用程式名稱，在指令行鍵入 qtcsh 時，該名稱會提交 Sun Grid Engine 做為遠端執行之用，以及做為 qrsh 功能的選項之用，該功能將為應用程式所用，並為應用程式定義資源需求。

注意 – 出現在指令行的應用程式名稱必須和 .qtask 檔所定義的一致。若有絕對值或相對的目錄規格做為字首，這是假設已輸入本端二進位且不進行遠端執行。

注意 – 然而，在執行帶有應用程式名稱的對照前，Csh 別名已經擴展。用於遠端執行的應用程式也可能在 qtcsh 指令行的任何地方出現，特別是在標準 I/O 重新導向的前後。

因此，以下的範例為有效且有意義的語法：

```
# .qtask file
netscape -v DISPLAY=myhost:0
grep -l h=filesurfer
```

若為 .qtask 檔，則以下 qtcsh 指令行：

```
netscape
~/mybin/netscape
cat very_big_file | grep pattern | sort | uniq
```

將暗中造成：

```
qrsh -v DISPLAY=myhost:0 netscape
~/mybin/netscape
cat very_big_file | qrsh -l h=filesurfer grep pattern | sort | uniq
```

qtcsh 能在開關所影響的不同模式下作業，其中這些開關可以是開或關：

- 本端或遠端指令執行（遠端為預設值）

- 立即或批次遠端執行（立即為預設值）
- 冗長或非冗長輸入（非冗長為預設值）

這些模式的設定可以在啓動時間使用 `qtcsch` 的選項引數，或在執行期間，使用 `shell` 內建指令 `qrshmode` 進行變更。要取得更多的資訊，請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》裡的 `qtcsch` 項目。

帶有 `qmake` 的平行檔案製作過程 (Parallel Makefile Processing)

`qmake` 是標準 UNIX `make` 功能的替代品。它會延伸 `make` 的功能將獨立的 `make` 步驟分配至合適機器的叢集。`qmake` 建立在熱門的 GNU-`make` 功能，`gmake`，的周圍。要取得 `gmake` 所包含的細節，請參閱 `<sge_root>/3rd_party` 所提供的資訊。

要確認聚合體分散 `make` 過程可以執行到完成，`qmake` 會先以類似平行作業的類比形式配置所需的資源。然後，`Qmake` 不需和 Sun Grid Engine 排程產生互動就能管理這一套資源。當資源因啓動具有 `-inherit` 選項的 `qrsh` 功能而變成可用時，它會分配 `make` 步驟。

因為 `qrsh` 提供標準輸出、錯誤輸出和標準輸入處理，以及終端機控制連線至遠端執行 `make` 步驟，所以在本端執行 `make` 程序或使用 `qmake` 之間只有三個顯著的不同：

- 假設獨立的 `make` 步驟有一定的有效期，且有足夠的獨立 `make` 步驟可供處理，則 `make` 過程的平行化明顯加快速度。當然這是預想到的不同處。
- 在遠端啓動的 `make` 步驟中，將會有 `qrsh` 所造成的隱含小型零碎空間和這類的遠端執行。
- 要利用 `qmake` 的 `make` 步驟分配，使用者必須指定為最小程度的平行化；如可同時執行的 `make` 步驟的數量。此外，使用者可指定 `make` 步驟所要求的資源特性，如可用軟體授權、機器架構、記憶體或 CPU 時間等需求。

`make` 最常用的用法是聚合體套裝軟體的編譯。然而，這可能不是 `qmake` 的主要應用程式。程式檔案通常都很小（因為好的程式練習的緣故），因此單一程式檔案的編譯（其為單一 `make` 步驟）通常只花幾秒鐘。此外，編譯通常會暗示有很多無法加速的檔案存取（包含巢狀檔案）（如果這些檔案存取是以平行方式為多重 `make` 步驟而製作），因為檔案伺服器可以有效成為將所有檔案存取序列化的障礙。所以，有時您無法期待編譯過程能有令人滿意的加速。

其他可能的 `qmake` 應用程式更為適合。例如，相互依賴的控制和透過檔案製作之聚合體分析作業的工作流程。在某些領域中，如 EDA，這是很常見的事，而這樣的環境中，每個 `make` 步驟都是帶有非常重要的資源和運算時間需求的模擬或資料分析作業。這樣的例子中，可達成相當程度的加速。

qmake 用法

qmake 的指令行語法看起來跟 qrsh 的指令行語法很像：

```
% qmake [-pe pe 名稱 pe 範圍][further options] \  
-- [gnu-make-options][ 目標 ]
```

注意 – qmake 也支援 `-inherit` 選項，如本節稍後所述。

您必須特別注意 `-pe` 選項的用法及其與 `gmake -j` 選項的關係。兩個選項都可用來表示要達到的平行之數量。不同處在於 `gmake` 不會用 `-j` 提供指定像是平行環境做為使用的可能性。因此，`qmake` 會假定已為平行製作配置了名為 `make` 的預設的環境。此外，`gmake` 的 `-j` 不允許一個範圍的規格，只允許單一數字。`qmake` 將用 `-j` 解譯所給的數字，做為 `1-<給予的數字>` 的範圍。相反地，`-pe` 允許所有這些參數的詳細規格。因此，以下的指令行範例是相同的。

```
% qmake -- -j 10  
% qmake -pe make 1-10 --
```

當以下的指令行無法透過 `-j` 選項表示時：

```
% qmake -pe make 5-10,16 --  
% qmake -pe mpi 1-99999 --
```

除了該語法，`qmake` 還支援兩種模式的呼叫：從指令行（沒有 `-inherit`）或在批次作業內（有 `-inherit`）的交互方式。這兩個模式初始啟動不同的動作序列：

- **互動式** – 當 `qmake` 在指令行呼叫時，這類的 `make` 過程會透過將 `qmake` 指令行指定的資源需求列入考慮的 `qrsh` 暗中提交給 `Sun Grid Engine`。`Sun Grid Engine` 然後會選擇一個執行與平行 `make` 作業相關的平行作業的主機器，然後在那裡啟動 `make` 程序。這個是必要的，因為 `make` 過程架構相依的，而所需的架構指定於 `qmake` 指令行。主機器上的 `qmake` 過程然後會將獨立 `make` 步驟委託其它由 `Sun Grid Engine` 配置工作並透過平行環境主機檔案傳遞 `qmake` 的主機執行。
- **批次** – 在此情況下，`qmake` 會與 `-inherit` 選項出現在批次 `script` 之內（如果 `-inherit` 選項不存在，將如上述的第一種情況，產生新的作業）。這會造成 `qmake` 利用已分配至嵌入 `qmake` 的作業的資源。它將直接使用 `qrsh -inherit` 來啟動 `make` 步驟。在批次模式呼叫 `qmake` 時，資源要求或 `-pe` 和 `-j` 選項就會被忽略。

注意 – 單一的 CPU 作業也必須要求平行環境 (`qmake -pe make 1 --`)。若不需要平行執行，請呼叫具 `gmake` 指令行語法的 `qmake` (沒有 Sun Grid Engine 選項和 "--")，其運作方式將像 `gmake`。

若要取得 `qmake` 的進一步的詳細資料，請參考《*Sun Grid Engine 5.3 和 Sun Grid Engine*，企業版 5.3 參考手冊》裡的 `qmake` 項目。

檢查點、監視和控制作業

在您使用 Sun Grid Engine 5.3 系統提交作業後，您需要懂得如何監視並控制這些作業。本章提供完成這些作業的背景資訊和指示。

本章包含以下特定作業的指示：

- 第 99 頁的「如何從指令行提交、監視或刪除檢查點作業」
- 第 100 頁的「如何使用 QMON 提交檢查點作業」
- 第 101 頁的「如何使用 QMON 監視和控制作業」
- 第 110 頁的「如何使用 qstat 監視作業」
- 第 112 頁的「如何使用電子郵件監視作業」
- 第 113 頁的「如何從指令行控制作業」
- 第 114 頁的「如何用 QMON 控制佇列」
- 第 118 頁的「如何用 qmod 控制佇列」

關於檢查點作業

本節探索兩個不同種類的作業檢查點。

- *使用者等級*
- *核心等級*

使用者等級檢查點

許多應用程式，特別是通常消耗很多 CPU 時間的程式，已實行檢查點並重新啓動機制以增加容錯。狀態資訊和已處理資料的重要部分會在特定運算階段中重複寫入一或數個檔案。與檢查點之前的狀況相比，如果應用程式可以中斷，然後稍後重新啓動，並達成一

致的狀態，就可處理這些檔案（即重新啓動檔案）。當使用者大部分必須處理重新啓動檔案，好將這些檔案移至正確的位置時，這種的檢查點稱爲*使用者等級* 檢查點。

對沒有整合的（使用者等級）檢查點之應用程式來說，您可以使用公共領域（例如，請參閱 University of Wisconsin 的 *Condor* 專案）或一些硬體廠商所提供所謂的*檢查點檔案庫* 做爲替代。重新將應用程式連結這樣的檔案庫會把檢查點機制安裝到應用程式中，而不需變更來源碼。

核心等級檢查點

有些作業系統在作業系統核心內有提供檢查點支援。在此情況下，應用程式不需準備也不需重新連結。核心等級檢查點通常適用於單一過程也可用於完整過程階層。換言之，相依過程的階層會隨時進行檢查點檢查並重新啓動。通常可用使用者指令和 C 資料庫 (C-library) 介面初始啓動檢查點。

Sun Grid Engine 會支援作業系統檢查點，如果有的話。要取得目前支援的核心等級檢查點功能的資訊，請參考《Sun Grid Engine Release Notes》。

檢查點作業的遷移

檢查點作業可隨時中斷，因爲它們重新啓動的功能確定僅少數已完成工作必須重複。此功能用來建立 Sun Grid Engine 的遷移和動態裝載平衡機制。若有要求，檢查點 Sun Grid Engine 作業會依照要求中斷並遷移到 Sun Grid Engine 儲存區的其他機器，並藉動態的方式平衡叢集中的裝載。檢查點作業會因下列原因中斷和遷移。

- 執行佇列或作業應由 `qmod` 或 `qmon` 指令明確暫停。
- 執行佇列或作業會自動暫停，因爲已超過佇列的暫停臨界值（請參閱第 150 頁的「如何配置載入 / 暫停臨界值」一節），而作業的檢查點事件規格則包含暫停的例子（請參閱第 99 頁的「如何從指令行提交、監視或刪除檢查點作業」一節）。

您可以辨識即將由遷移狀態 `m` 在 `qstat` 輸出中遷移的作業。遷移作業會移回 `sge_qmaster` 並在之後分配到另一個合適的佇列，如果有任何合適佇列的話。

組成檢查點作業 Script

核心等級檢查點的 Shell script 和一般 shell script 沒有不同。

使用者等級檢查點作業的 Shell script 與一般 Sun Grid Engine 批次 script 間唯一的不同點在於一旦重新啓動，它們能正確處理該案的能力。環境變數 `RESTARTED` 設定在作業重新啓動時，會對作業進行檢查點檢查。它可以用來跳過作業 script 的區段，其僅可以在初始呼叫時執行。

因此，透明檢查點的作業 script 看起來可能類似程式範例 1。

```
#!/bin/sh
#Force /bin/sh in Sun Grid Engine
#$ -S /bin/sh

# Test if restarted/migrated
if [ $RESTARTED = 0 ]; then
    # 0 = not restarted
    # Parts to be executed only during the first
    # start go in here
    set_up_grid
fi

# Start the checkpointing executable
fem
#End of scriptfile
```

程式範例 1 檢查點作業 Script 範例

如果使用者等級檢查點作業已遷移，記下作業 script 從頭開始重新啟動是很重要的。使用者負責將 shell-script 的程式流程導向作業中斷的位置，因此需跳過 script 中這些執行一次以上的重要行列。

注意 – 核心等級檢查點作業是可隨時中斷的，而包含的 (embracing) shell script 會由上個檢查點發生的地方重新啟動。因此，RESTARTED 環境變數和核心等級檢查點作業沒有相關。

▼ 如何從指令行提交、監視或刪除檢查點作業

使用合適的開關輸入以下指令。

```
#qsub 選項 引數
```

提交檢查點作業的運作方式和一般批次 script 的作業方式相同，qsub -ckpt 和 -c 開關除外，這些開關要求檢查點機制並定義為作業產生的檢查點事件。-ckpt 選項採用使用檢查點環境（第 209 頁的「關於檢查點支援」）名稱的引數。-c 選項並非強制性，並也採用一個引數。它可以用來覆寫檢查點環境配置裡 when 參數的定義（細節請參閱《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》中的 checkpoint 項目）。

-c 選項的引數可以是以下單字母選項（或字母的任何組合）中的一個或替代的時間值：

- n – 沒有執行檢查點。這有最高的優先權

- s – 檢查點只會在關閉作業主機上的 `sgexecd` 才會產生。
- m – 以對應佇列配置所定義的最少 CPU 間隔時間產生檢查點（請參閱 `queue_conf` 線上說明手冊的 `min_cpu_interval` 參數）。
- x – 若作業暫停，檢查點就會產生。
- interval – 在指定的間隔時間產生檢查點，但該間隔時間不能比 `min_cpu_interval`（請上述）所定義的還要頻繁。時間數值必須指定為 `hh:mm:ss`（以冒號分隔的二位數的時數、分數和秒數）。

檢查點作業*監視*與一般作業的不同處是這些作業會不時地遷移（由狀態 `m` 表示在 `qstat` 輸出中遷移，請見上述），因此，這些作業不能受限於單一佇列。然而，唯一的作業辨識編號與作業名稱都維持相同。

刪除檢查點作業的運作方式如同第 113 頁的「從命令行控制 Sun Grid Engine 作業」一節所述。

▼ 如何使用 QMON 提交檢查點作業

- 依照第 71 頁的「進階範例」的指示，注意以下的額外資訊。

透過 QMON 的檢查點作業提交與一般加上指定合適的檢查點環境之批次作業的提交相同。如同第 71 頁的「進階範例」程序中所解釋的，「作業提交」對話方塊提供與作業相關的檢查點環境一個輸入視窗。輸入視窗之外還有一個圖示鈕，其可以開啓顯示於圖 5-1 的「選擇」對話方塊。您可以從可用的清單選擇合適的檢查點環境。有關安裝於您站點之檢查點環境屬性的資訊，請向系統管理員查詢，或參考第 209 頁的「關於檢查點支援」一節。

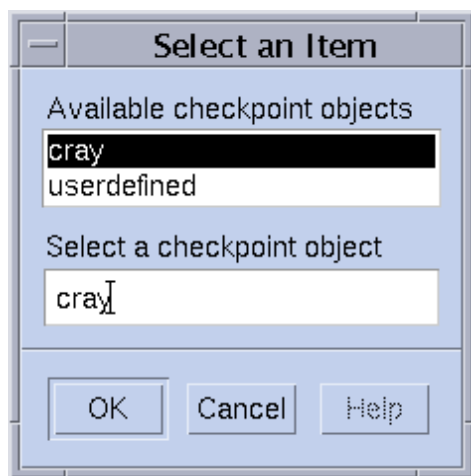


圖 5-1 檢查點物件選擇

檔案系統需求

當寫入檢查點檔案庫式的使用者等級或核心等級檢查點時，進行檢查點之過程或作業涵蓋的完整虛擬記憶體影像則需轉儲。要達到這個目的，您需要足夠的磁碟空間。若檢查點環境配置參數 `ckpt_dir` 已設定，則檢查點資訊會轉儲到 `ckpt_dir` 之下的作業私人位置。若 `ckpt_dir` 設為 `NONE`，則會使用檢查點作業啟動的目錄。要取得檢查點環境配置的詳細資訊，請參考《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `checkpoint` 項目。

注意 – 若 `ckpt_dir` 設為 `NONE`，您應該使用 `qsub -cwd script` 啟動檢查點作業。

引起檔案系統如何組織之額外需求的原因為，為了順利遷移並重新啓動作業，所有的機器上都必須看得到檢查點檔案和重新啓動檔案。因此需要 NFS 或類似的檔案系統。請詢問您的叢集管理看看這個需求是否符合您的站點。

若您的站點沒有執行 NFS 或如果因某些因素而沒使用，您應該能在使用者等級檢查點作業的情況下，從 `shell script` 的開端（例如，透過 `rcp` 或 `ftp`）明確傳輸重新啓動檔案。

監視和控制 Sun Grid Engine 作業

原則上有三種監視提交作業的方式。

- 使用 Sun Grid Engine 圖形使用者介面，`QMON`
- 從指令行使用 `qstat` 指令
- 使用電子郵件

▼ 如何使用 `QMON` 監視和控制作業

`Sun Grid Engine` 圖形使用者介面，`QMON`，提供特別為控制作業設計的對話方塊。

- 在 `QMON` 主功能表中，按一下「作業控制」鈕，然後根據以下各節詳細記載的額外資訊繼續進行。

此對話方塊的一般目的是提供監視所有執行、擱置和系統或零件能辨識之完成作業的可配置數量。對話方塊也可以用來操作作業，如變更其優先權、暫停、恢復或取消作業。共可顯示三個環境，一個用於執行作業、另一個用於等待分派至合適資源的擱置作業、而第三個則用於最近完成的作業。您可以按螢幕上方的對應 `tab` 標籤在三個列出環境中做選擇。

在其預設的格式中（圖 5-2），它會顯示每個執行和擱置作業的作業 Id、優先權、作業名稱、和佇列欄位。

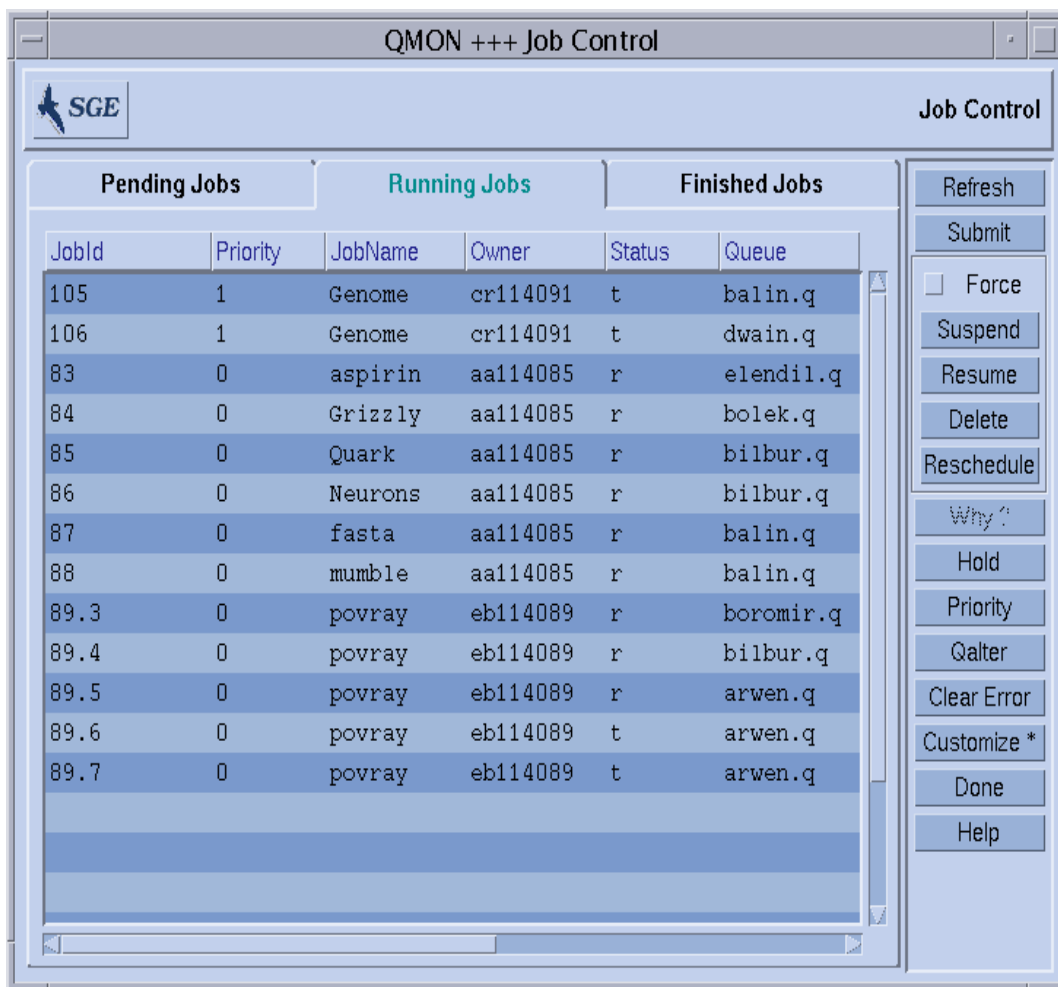


圖 5-2 作業控制對話方塊 — 標準格式

您可以配置與「自訂」對話方塊一同顯示的資訊集，（請參閱圖 5-3），您只需按「作業控制」對話方塊中的「自訂」鈕即可開啓此對話方塊。

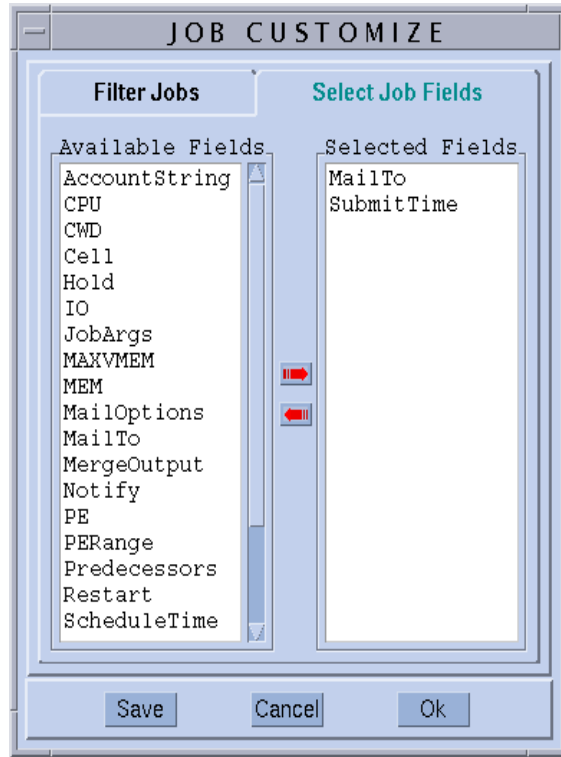


圖 5-3 作業控制自訂對話方塊

有了「自訂」對話方塊，您就可以選擇將顯示並過濾利益作業之 Sun Grid Engine 作業物件的進一步的項目。圖 5-3 的範例選擇「寄至 (MailTo)」和「提交時間 (Submit Time)」等額外的欄位。

顯示於圖 5-4 的「作業控制」對話方塊描述在「完成作業」清單使用自訂後加強的外觀。

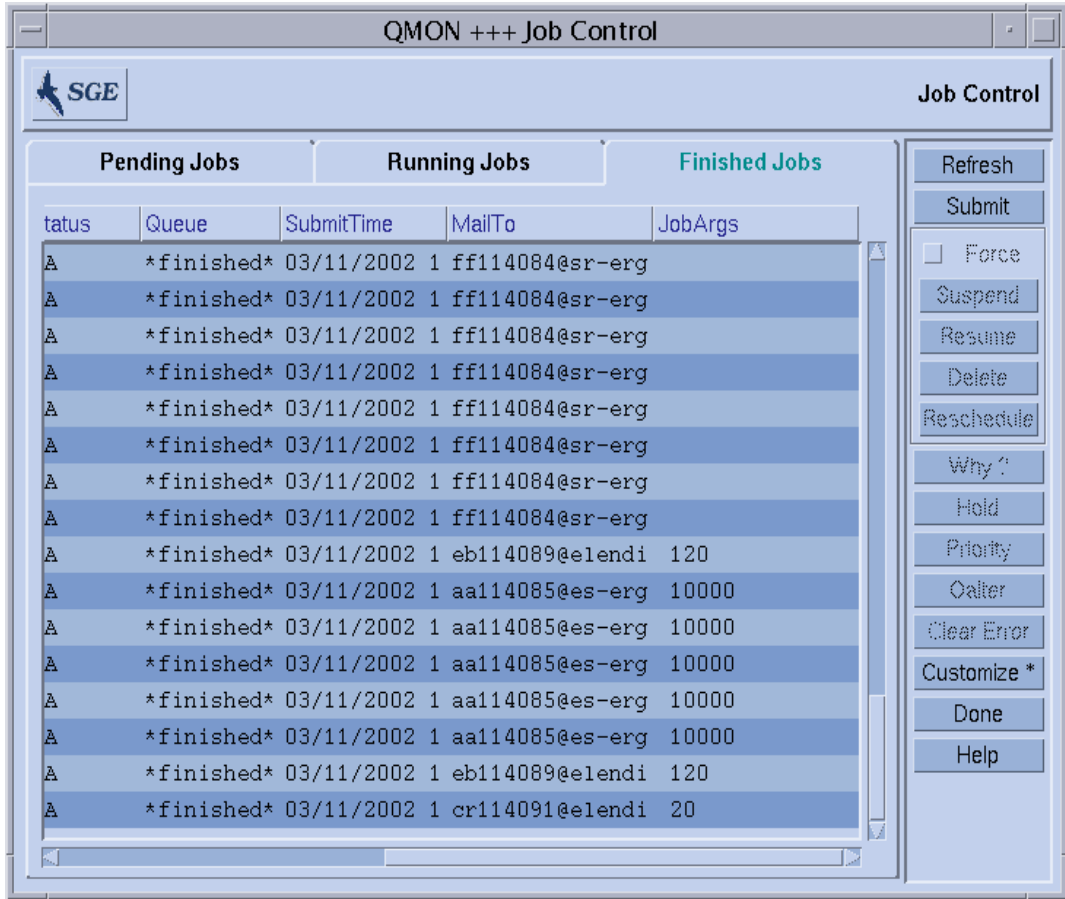


圖 5-4 作業控制對話方塊完成作業 — 加強

圖 5-5 中的過濾功能範例只選擇執行或適合 solaris64 架構之 `ferst1` 所擁有的作業。

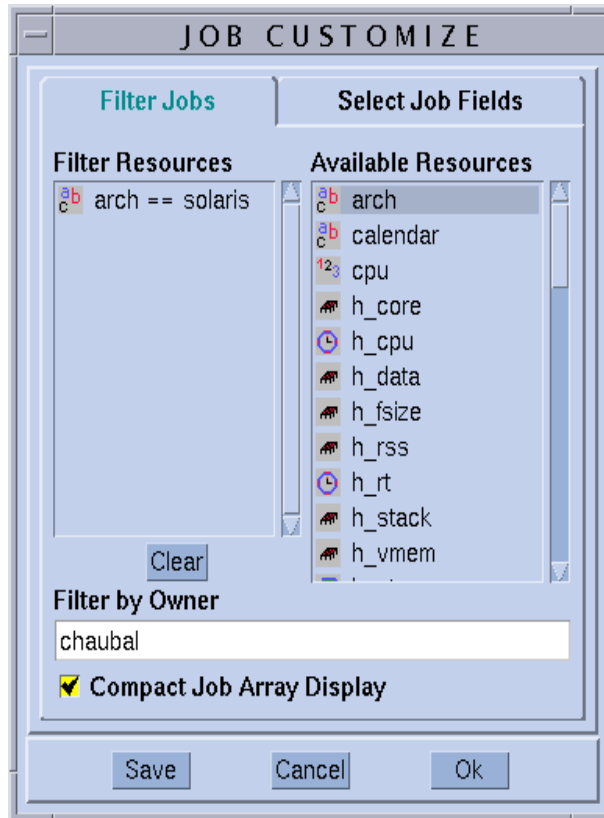


圖 5-5 作業控制過濾

產生顯示「擱置作業」的「作業控制」對話方塊顯示於圖 5-6。

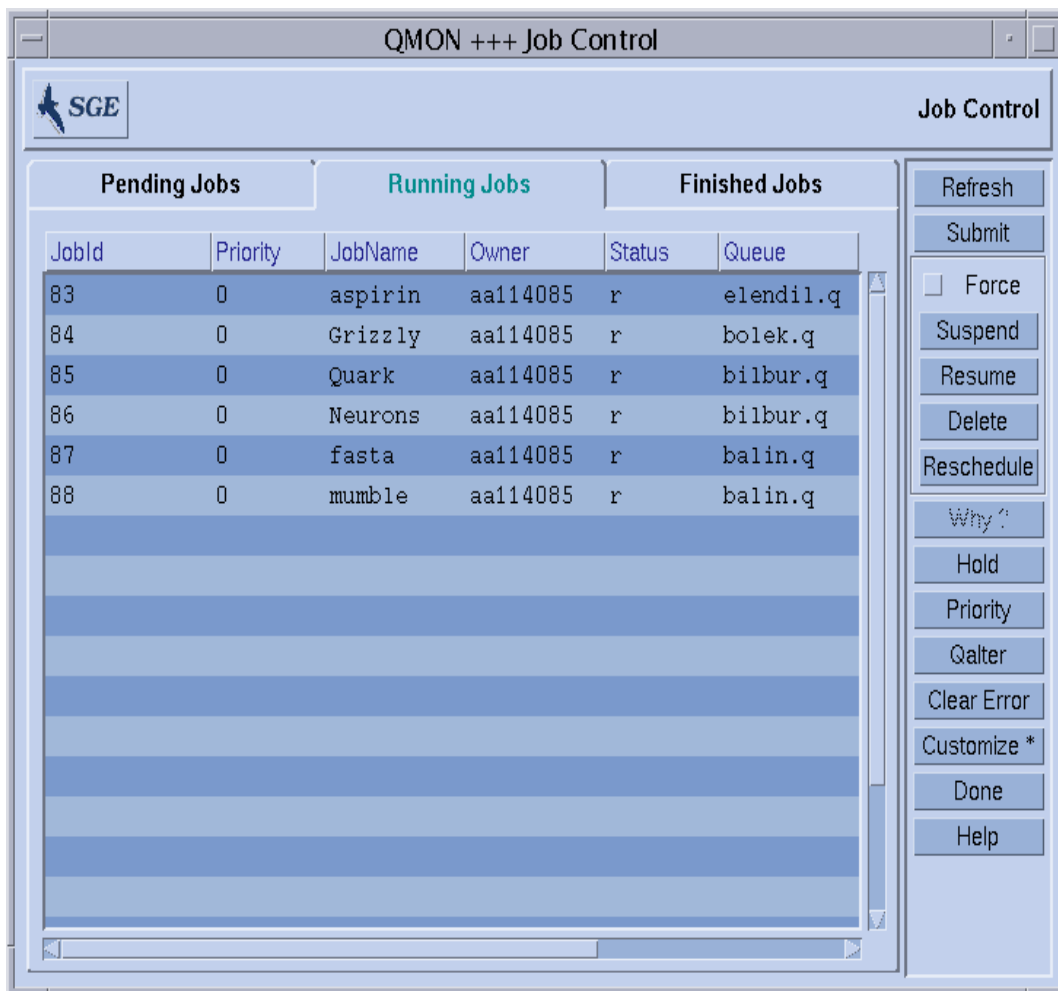


圖 5-6 作業控制對話方塊 — 過濾後

注意 – 例如，顯示於圖 5-3 的「自訂」對話方塊中的 Save 鈕會將自訂儲存於使用者初始目錄的 .qmon_preferences 檔案中，並因此重新定義「作業控制」對話方塊的預設外觀。

圖 5-6 中的「作業控制」對話方塊也是陣列作業如何在 QMON 顯示的範例。

作業可用以下滑鼠 / 鍵的組合選擇（做為日後操作之用）：

- 在按住「控制」鍵時，用滑鼠左鍵按一下作業可啟動多重作業的選項。

- 在按住「Shift」鍵時，用滑鼠左鍵按一下另一個作業可以在選項開始與目前作業之間、和包含選項開始與目前作業來選擇所有作業。
- 在按住「控制」鍵時，用滑鼠左鍵按一下作業可切換單一作業的選擇狀態。

您可以透過螢幕右邊的對應鈕暫停、恢復（解除暫停）、刪除、阻止（和放開）、重設優先權並修改 (Qalter) 選取的作業。

暫停、解除暫停、刪除、保留、修改優先權和修改作業的動作只適用於作業所有人、或 Sun Grid Engine 管理員和操作員的作業（請參閱第 60 頁的「管理員、操作員和所有人」）。只有執行的作業才能暫停 / 恢復；只有擱置作業可以阻止和修改（在優先權和其他屬性中）。

暫停作業表示相當於傳送帶有 UNIX kill 指令的 SIGSTOP 訊號給作業的處理群組，該指令會中止作業且不再消耗 CPU 時間。解除暫停作業會傳送 SIGCONT 訊號，並因此恢復作業（欲取得更多發出訊號過程的資訊，請參閱系統的 kill 線上說明手冊）。

注意 – 您可以強制暫停、解除暫停和刪除；例如，註冊 sge_qmaster，但因網路問題而不通知控制作業的 sge_execd，例如，在對應的 sge_execd 無法達成時。使用 Force 旗號來達到此目的。

若在選取的擱置作業使用保留鈕，就會開啓「設定保留」子對話方塊（請參閱圖 5-7）。

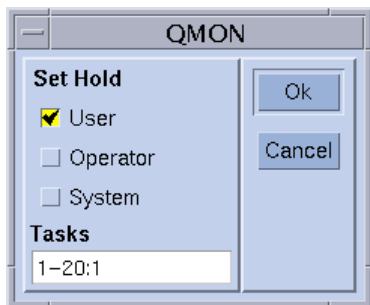


圖 5-7 作業控制保留

「設定保留」子對話方塊能啓動設定和重設使用者、系統和操作員保留。作業所有人和 Sun Grid Engine 操作員和管理員可以設定 / 重設使用者保留 (User holds)。管理者和操作者可以設定 / 重設操作者保留 (Operator holds)，而管理者保留 (managers holds) 只能由管理者設定 / 重設。只要作業指定有任何的保留，該作業就不適合執行。設定 / 重設保留的替代方法為 qalter、qhold 和 qrls 指令（請參閱《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》的對應項目）。

若按住優先權 (Priority) 鈕，就會開啓另一個子對話方塊（圖 5-8），其可以啓動輸入選取的擱置中作業的新優先權。在 Sun Grid Engine 之中，優先權可決定擱置作業清單的作業順序和由 Job Control 對話方塊所顯示的擱置作業順序。使用者只能在 0 到 -1024 的範圍間設定優先權。Sun Grid Engine 操作員和管理員也可以將優先權階層最多增加至 1023（要取得作業優先權的詳細資訊，請參閱第 90 頁的「作業優先權」一節）。

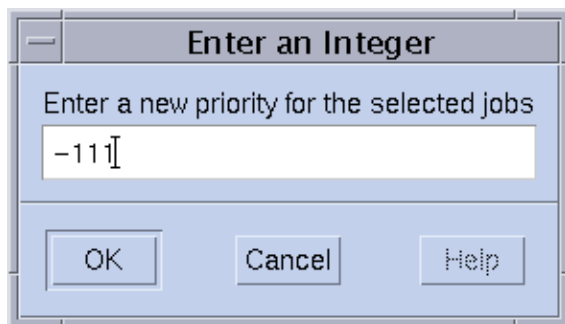


圖 5-8 作業控制優先權定義

為擱置作業按 **Qalter** 鈕時，就會開啓第 63 頁的「如何提交作業，從圖形使用者介面，**QMON**」說明的「作業提交」畫面，以及對應提交時定義之作業屬性的所有對話方塊集的項目。這些無法變更的項目設為無回應。其他的您可以進行編輯，並在「作業提交」對話方塊按 **Qalter** 鈕（可用來取代「提交」鈕）來用 **Sun Grid Engine** 紀錄變更。

在「作業提交」畫面中，「驗證 (Verify)」旗號使用於 **Qalter** 模式時有特殊的意義。您可以檢查擱置作業的一致性，並調查它們還沒排程的原因。您只需為「驗證」旗號選擇想要的一致性檢查模式，然後按 **Qalter** 鈕。系統會根據選擇的檢查模式顯示發生不一致時的警告。要取得更進一步的資訊，請參閱第 71 頁的「進階範例」一節和 **qalter** 線上說明手冊的 **-w** 選項。

另一個檢查作業為何仍舊擱置的方法是選擇作業，然後按「作業控制」對話方塊的 **Why?** 鈕。這會開啓「物件瀏覽器 (Object Browser)」對話方塊，並顯示阻止 **Sun Grid Engine** 排程器在最近一次的 **pass** 分派作業的原因清單。顯示這類訊息的範例瀏覽器畫面顯示在圖 5-9。

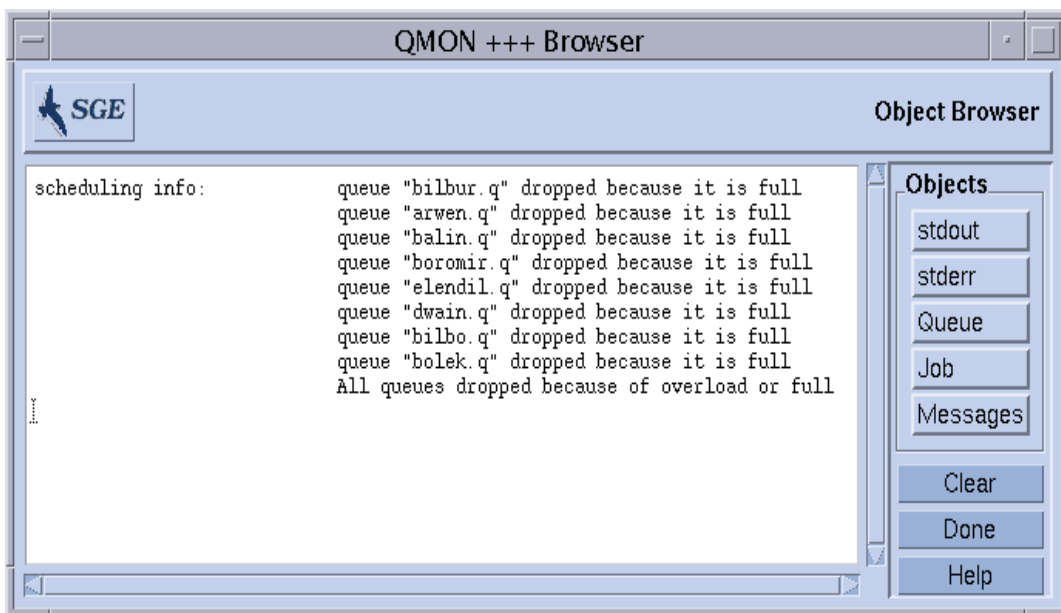


圖 5-9 顯示排程資訊的瀏覽器

注意 – Why? 按鈕只會傳送有意義的輸出資訊，若排程器配置參數 `schedd_job_info` 設為 `true` 的話（請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine, 企業版 5.3 參考手冊*》的 `sched_conf`）。顯示的排程資訊與上次的排程間隔時間有關。在您調查作業沒有進行排程時，該資訊可能已經不正確了。

「清除錯誤 (Clear Error)」按鈕可以用來移除選取之擱置作業的錯誤狀態，該作業曾在之前嘗試啟動，可是因為作業相依問題而失敗（如，權限不足以寫入指定的作業輸出檔案）。

注意 – 錯誤狀態使用擱置作業清單中的紅色字體顯示，並應該在修正錯誤狀況後移除；如透過 `qalter`。如果作業要求在作業中斷時要傳送電子郵件，則這樣的錯誤狀況會自動透過電子郵件報告（如透過 `qsub -m a` 選項）。

為維持顯示最新資訊，QMON 使用輪詢方案從 `sgemaster` 取得作業狀態。按「重新顯示」鈕可以強制更新。

最後，該鈕提供 QMON 「作業提交」對話方塊的連結（例如，請參閱圖 5-10）。

QMON 物件瀏覽器的額外資訊

QMON 物件瀏覽器可以用來快速取得 Sun Grid Engine 作業的額外資訊，而不需要如第 101 頁的「如何使用 QMON 監視和控制作業」一節所述的自訂「作業控制」對話方塊。

按一下 QMON 主功能表的「瀏覽器」圖示鈕就可開啓「物件瀏覽器」。如果選取瀏覽器的「作業」鈕，且如果滑鼠指標移到「作業控制」對話方塊中的作業行上，則瀏覽器會顯示 Sun Grid Engine 作業的相關資訊（請參閱圖 5-2 的範例）。

圖 5-10 的瀏覽器畫面會提供在此情況下所顯示的資訊的範例。

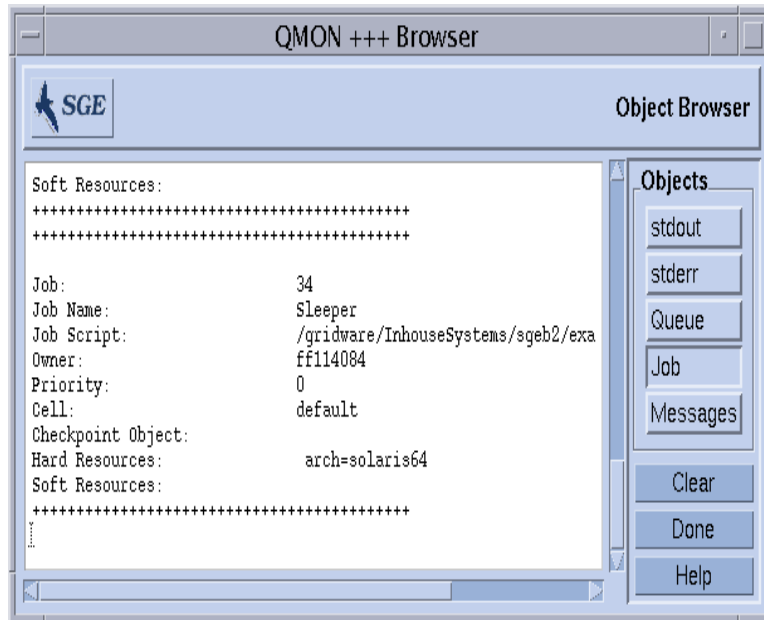


圖 5-10 物件瀏覽器 — 作業

▼ 如何使用 qstat 監視作業

- 從命令行透過詳述於以下各節的資訊的引導，使用以下其中一個指令。

```
% qstat
% qstat -f
```

第一個表格只提供提交作業的簡介（請參閱表 5-1）。第二個表格包含目前另外配置的佇列資訊（請參閱表 5-2）。

在第一個表格中，標題行表示欄位的意義。大部分欄位的目的應該可以自我解釋。然而，state 一欄包含具以下意義的單一字元碼：r 表示執行、s 表示暫停、q 表示已進入排程、而 w 則表示等待（要取得 qstat 輸出格式的詳細解釋，請參閱《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》中的 qstat 項目）。

第二個表格分為兩個部分，第一個顯示所有可用佇列的狀態，第二個（以 - PENDING JOBS - ... 分隔元為名）顯示 sge_qmaster 作業多工緩衝區域的狀態。佇列部分的第一行定義與列出與佇列相關欄位的意義。佇列由水平線分隔線分隔。若作業在佇列中執行，則作業會以第一個表格中以和 qstat 指令相同的格式，列印於相關佇列之下。第二個輸出部分的擱置作業也會如同 qstat 第一個表格的擱置作業的列印方式印出。

以下各欄的佇列說明需要更多的解釋。

- qtype – 此為佇列種類—B（批次）、I（互動）、P（平行）、和 C（檢查點）或組合之一。
- used/free – 此為佇列中已使用 / 未使用作業插槽的計算。
- states – 此為佇列的狀態—u（未知）、a（警告）、s（暫停）、d（關閉）、E（錯誤）、或組合之一。

此外，qstat 線上說明手冊包含更詳細的 qstat 輸出格式說明。

qstat 指令的各種額外選項會增加兩個版本的功能。-r 選項可用來顯示提交作業的資源需求。此外，輸出可以限制於某個使用者，或限制於特定的佇列，而 -l 選項可以用來指定 qsub 指令的資源需求，如第 76 頁的「資源需求定義」一節所述。若使用資源需求，只有符合 qstat 指令行中的資源需求規格的佇列（而作業在這些佇列中執行）才會顯示。

表 5-1 qstat 輸出的範例

job-ID	prior	name	user	state	submit/start at	queue	function
231	0	hydra	craig	r	07/13/96 20:27:15	durin.q	MASTER
232	0	compile	penny	r	07/13/96 20:30:40	durin.q	MASTER
230	0	blackhole	don	r	07/13/96 20:26:10	dwain.q	MASTER
233	0	mac	elaine	r	07/13/96 20:30:40	dwain.q	MASTER
234	0	golf	shannon	r	07/13/96 20:31:44	dwain.q	MASTER
236	5	word	elaine	qw	07/13/96 20:32:07		
235	0	andrun	penny	qw	07/13/96 20:31:43		

表 5-2 qstat -f 輸出的範例

queue	qtype	used/free	load_avg	arch	states
dq	BIP	0/1	99.99	sun4	au
durin.q	BIP	2/2	0.36	sun4	
231	0	hydra	craig	r	07/13/96 20:27:15 MASTER
232	0	compile	penny	r	07/13/96 20:30:40 MASTER
dwain.q	BIP	3/3	0.36	sun4	
230	0	blackhole	don	r	07/13/96 20:26:10 MASTER
233	0	mac	elaine	r	07/13/96 20:30:40 MASTER
234	0	golf	shannon	r	07/13/96 20:31:44 MASTER
fq	BIP	0/3	0.36	sun4	
#####					
- PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS -					
#####					
236	5	word	elaine	qw	07/13/96 20:32:07
235	0	andrun	penny	qw	07/13/96 20:31:43

▼ 如何使用電子郵件監視作業

- 從指令行透過詳述於以下各節的資訊的引導，用合適的引數輸入以下指令。

```
% qsub 引數
```

qsub -m 開關要求如果有事件發生時，需傳送電子郵件給提交作業的使用者或 -M 旗號所指定的電子郵件地址（有關旗號的說明，請參閱 qsub 線上說明手冊）。-m 選項的引數指定事件。有以下選項可用：

- b - 郵件在作業開始時寄送。
- e - 郵件在作業結束時寄送。
- a - 郵件在作業中斷時寄送（如，透過 qdel 指令）。
- s - 郵件在作業暫停時寄送。
- n - 不寄送任何郵件（預設值）。

多重選項可以在以逗點分隔的清單中使用單一 -m 選項選取。

相同的郵件事件可由 QMON 「作業提交」對話方塊的說明配置。請參閱第 71 頁的「進階範例」一節。

從指令行控制 Sun Grid Engine 作業

第 101 頁的「如何使用 QMON 監視和控制作業」一節解釋如何使用 Sun Grid Engine 圖形使用者介面 QMON 刪除、暫停和恢復 Sun Grid Engine 作業。

相同的功能也可以從指令行取得，本節有相關說明。

▼ 如何從指令行控制作業

- 從指令行透過詳述於以下各節的資訊的引導，用合適的引數輸入以下其中一個指令。

```
% qdel 引數  
% qmod 引數
```

您可使用 `qdel` 指令取消 Sun Grid Engine 作業，不管它們正在執行或在多工緩衝處理中。`qmod` 指令提供執行中作業的暫停或解除暫停（恢復）的方法。

對兩個指令來說，您需要知道作業辨識號碼，該號碼會回應成功的 `qsub` 指令而顯示。若您忘記該號碼，則您可以透過 `qstat` 取得（請參閱第 110 頁的「如何使用 `qstat` 監視作業」一節）。

以下包含兩個指令的數個範例：

```
% qdel 作業 id  
% qdel -f 作業 id1、作業 id2  
% qmod -s 作業 id  
% qmod -us -f 作業 id1、作業 id2  
% qmod -s 作業 id. 作業 id 範圍
```

要刪除、暫停或解除暫停作業，您必須是作業的所有人、Sun Grid Engine 管理員或操作員（請參閱第 60 頁的「管理員、操作員和所有人」）。

對兩個指令來說，`-f` 強制選項可以用來記錄作業在 `sge_qmaster` 的狀態變更而不通知 `sge_execd`（例如，`sge_execd` 因為網路問題之類的緣故而無法達成時）。`-f` 選項是用來給管理員使用的。然而，如果是 `qdel`，則使用者可以強制刪除他們自己的作業，若叢集配置 `qmaster_params` 項目中的旗號 `ENABLE_FORCED_QDEL` 已設定（要取得更多資訊，請參閱《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》中的 `sge_conf` 線上說明手冊）。

作業相依

建立聚合體作業最方便的方法通常是將作業分割為子作業。在這些例子中，子作業需靠其它子作業成功完成才能啟動。有一個範例是之前的作業產生後繼作業才能讀取和處理的輸出檔案。

Sun Grid Engine 使用其作業相依功能來支援相依作業。作業可以配置來依靠一個或多重作業的成功完成。此功能由 `qsub -hold_jid` 選項實行。作業清單可指定可依靠的提交作業。作業清單也可以包含陣列作業的子集。除非相依清單中的所有作業都已經順利完成，否則提交作業將無法執行。

控制佇列

如第 50 頁的「佇列和佇列屬性」一節所述，佇列所有人有暫停 / 解除暫停或停用 / 啟動佇列的權限。如果這些使用者不時需要某些機器執行重要工作，且如果他們會被在背景執行的 Sun Grid Engine 作業強烈影響，這情況就非常理想。

暫停或啟動佇列有兩個方法。

- QMON 佇列控制對話方塊
- `qmod` 指令

▼ 如何用 QMON 控制佇列

- 從 QMON 主要功能表上按下「佇列控制」鈕。

螢幕就會顯示「佇列控制」對話方塊，類似圖 5-11 所示的對話方塊。

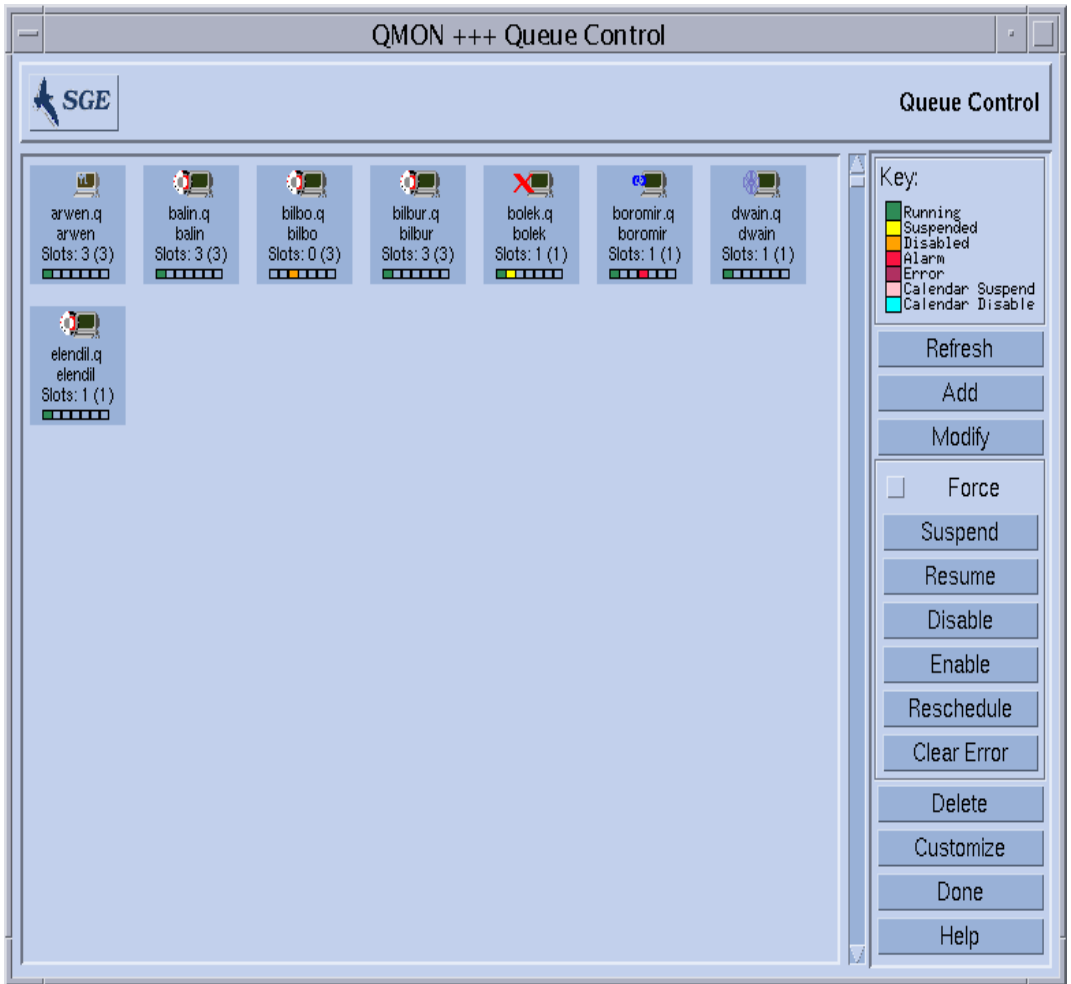


圖 5-11 佇列控制對話方塊

「佇列控制」對話方塊的目的是提供叢集中可用資源和活動的快速簡介。它也提供暫停 / 解除暫停、停用 / 啟動佇列，以及配置佇列的方法。每個顯示的圖示代表一個佇列。若主要顯示區域是空的，則表示沒有配置佇列。每個佇列圖示都貼上佇列名稱、佇列常駐的主機之名稱、和佔用的作業插槽數。如果 `sge_execd` 在佇列主機上執行，並已經用 `sge_qmaster` 登錄，則佇列圖示上的圖片表示佇列主機的作業系統架構，而圖示底部的指示條說明佇列的狀態。「佇列控制」對話方塊右邊的圖示顯示顏色代表的意義。

對這些佇列來說，使用者可以按住鍵盤上的「Shift」鍵，然後在佇列圖示上按一下滑鼠左鍵，就可取得佇列並隱含取得管理佇列的機器的目前屬性、負載和資源消耗的資訊。螢幕上將呈現類似於顯示於圖 5-12 的資訊畫面。

按一下滑鼠左鍵或點進一個包圍佇列圖示鈕的長方形區域就可選擇佇列。「刪除」、「暫停」/「解除暫停」、「停用」/「啟動」鈕可用來在選擇的佇列上執行對應的操作。暫停/解除暫停和停用/啟動操作需要通知對應的 `sge_execd`。如果這不可能做到（如，因為主機故障），則如果打開「強制 (Force)」切換鈕，就可強制 `sge_qmaster` 內部狀態變更。

如果佇列暫停，則該佇列就會關閉不能用於以後的作業，而已經在佇列執行的作業也會如同一節第 101 頁的「如何使用 QMON 監視和控制作業」解釋一般暫停。佇列及其作業在佇列解除暫停時就會恢復。

注意 – 如果在暫停佇列中的作業已另外明確暫停，則該作業在佇列解除暫停時也不會恢復。該作業需要再度明確解除暫停。

停用的佇列會關閉，但是，在這些佇列執行的作業被允許繼續進行。停用佇列通常是用來「耗盡 (drain)」佇列。在啟動佇列後，佇列就可以再度用於作業執行。在仍舊執行的作業上沒有執行動作。

暫停/解除暫停和停用/啟動操作需要佇列所有人或 Sun Grid Engine 管理者或操作者權限（請參閱第 60 頁的「管理員、操作員和所有人」一節）。

顯示於「佇列控制」對話方塊的資訊會定期更新。按「重新顯示」鈕可以強制更新。「完成 (Done)」鈕會關閉對話方塊。

「自訂」鈕能讓您選擇透過過濾操作而顯示的佇列。圖 5-13 中的樣本畫面顯示只屬於架構 `osf4`（如，Compaq UNIX version 4）的主機上執行的佇列選項。「自訂」對話方塊的「儲存」鈕讓您將設定儲存在初始目錄中的 `.qmon_preferences` 檔中，在之後呼叫 QMON 做為標準重新啟動之用。

對配置的目的來說，在您按「佇列控制」畫面右邊的「新增」或「修改」鈕時，子對話方塊就會開啓（相關細節請參閱第 146 頁的「如何以 QMON 配置佇列」一節）。

Attribute	Slot-Limits/Fixed Attributes	Load(scaled)/Consumable
arch	solaris64	none
num_proc		1
load_avg		0.113
load_short		0.094
load_medium		0.113
load_long		0.121
np_load_avg		0.113
np_load_short		0.094
np_load_medium		0.113
np_load_long		0.121
mem_free		49.000M
mem_total		256.000M
swap_free		380.000M
swap_total		513.000M
virtual_free		429.000M
virtual_total		769.000M
mem_used		207.000M
swap_used		133.000M

圖 5-12 佇列屬性顯示

所有附接佇列的屬性（包含從主機或叢集得到的）都列在「屬性」欄。「插槽限制 / 固定屬性 (Slot-Limits/Fixed Attributes)」欄顯示已每個佇列插槽限制或以固定聚合體屬性定義的屬性值。負載（依比例）/ 可消耗的欄位說明報告的（如依比例配置）負載參數（請參閱第 183 頁的「載入參數」一節）及根據 Sun Grid Engine 可消耗資源功能的可用資源容量（請參閱第 172 頁的「可耗損資源」一節）。

注意 – 若負載屬性配置為可消耗的資源，則負載報告和可消耗容量可能相互覆寫。顯示的是兩者的最小值，通常用於作業分派演算法。

注意 – 顯示的負載和可消耗數值目前不考慮如第 24 頁的「執行主機」一節所述的負載調整修正。

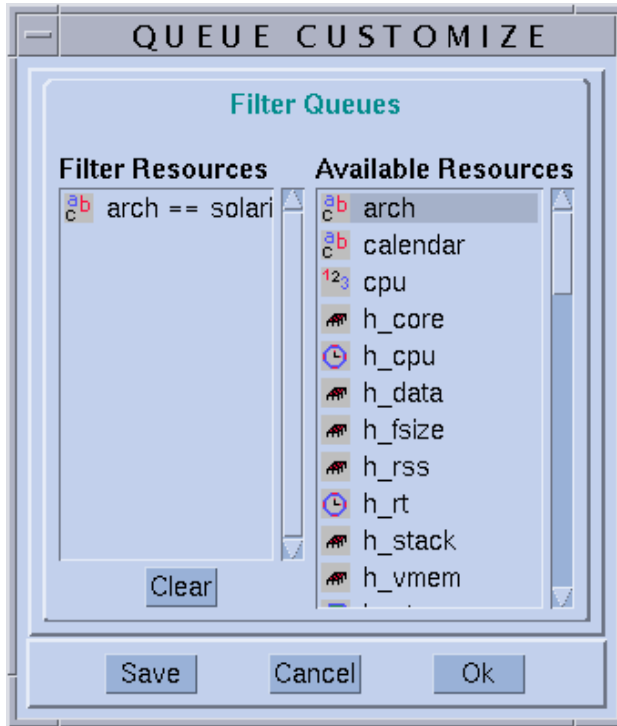


圖 5-13 佇列控制自訂

▼ 如何用 `qmod` 控制佇列

第 113 頁的「如何從指令行控制作業」一節解釋 Sun Grid Engine 指令，`qmod`，如何可用來暫停 / 解除暫停 Sun Grid Engine 作業。然而，`qmod` 指令會額外提供使用者暫停 / 解除暫停或停用 / 啟動佇列的方法。

- 透過詳述於以下各節的資訊的引導，用合適的引數輸入以下指令。

```
% qmod 引數
```

以下指令為 `qmod` 用於此目的的範例：

```
% qmod -s q 名稱  
% qmod -us -f q 名稱1、q 名稱2  
% qmod -d q 名稱  
% qmod -e q 名稱1、q 名稱2、q 名稱3
```

前兩個指令會暫停或解除暫停佇列，而第三和第四個指令則可停用並啟動佇列。第二個指令除了強制登錄 `sge_qmaster` 的狀態變更外，還使用 `qmod -f` 選項（萬一對應的 `sge_execd` 因網路問題而無法達成時。）

注意 – 暫停 / 解除暫停和停用 / 啟動佇列需要佇列所有人或 Sun Grid Engine 管理者或操作者權限（請參閱第 60 頁的「管理員、操作員和所有人」一節）。

注意 – 您可以用 `crontab` 或 `at` 作業使用 `qmod` 指令。

自訂 QMON

QMON 的觀感大部分由特別設計的資源檔案所定義。參考的資源檔是放在 `<sge_root>/qmon/Qmon`，而合理的預設值已經編譯在其中。

叢集管理可以藉由將 QMON 特定資源定義包含到標準 `.Xdefaults` 或 `.Xresources` 檔案中，或是將站點特定 `Qmon` 檔案放到標準搜尋路徑，如 `XAPPLRESDIR`，所提及的位置上，來把站點特定預設值安裝在標準位置上，如 `/usr/lib/X11/app-defaults/Qmon`。詢問您的管理員看看上述的任何事物是否和您的例子有關，

此外，使用者可以藉由將 `Qmon` 檔案複製並修改至起始目錄（或至另一個私人 `XAPPLRESDIR` 搜尋路徑所指定的位置），或將必要的資源定義包含進使用者私人的 `.Xdefaults` 或 `.Xresources` 檔案中，來配置個人的喜好設定。私人的 `Qmon` 資源檔案也可以在作業或 X11 環境的（例如在 `.xinitrc` 資源檔案中）啟動期間透過 `xrdb` 指令安裝。

欲取得可能自訂的詳細資訊，請參閱參考 `Qmon` 檔案裡的注釋行。

另一個自訂 QMON 的方式已為圖 5-2 和圖 5-13 裡的「作業控制」和「佇列控制自訂」對話方塊做了解釋。在兩個對話方塊中，您可用「儲存」鈕儲存過濾並顯示配置自訂對話方塊至使用者起始目錄下的檔案，`.qmon_preferences` 的定義。重新開始之際，QMON 會讀取這個檔案並重新啟動先前定義的運作方式。

部分 IV 管理

《Sun Grid Engine 5.3 管理和使用者指南》的這個部分包含六個章節，是針對給管理者所使用的。

■ 第六章 — 第 123 頁的「主機和叢集配置」

本章提供配置 Sun Grid Engine 5.3 主機和叢集的一般背景和詳盡的指示。

■ 第七章 — 第 145 頁的「配置佇列與佇列行事曆」

本章包含 *佇列* 的重要概念的說明—佇列的功用是不同種類的 Sun Grid Engine 5.3 作業的「容器」。本章包含完整的配置佇列的指示。

■ 第八章 — 第 163 頁的「聚合體概念」

本章解釋 Sun Grid Engine 5.3 系統如何使用 *聚合體 (complex)* 來定義所有使用者可能要求的作業資源屬性之相關資訊。管理員會配置不同的聚合體以符合環境的需求，而本章提供此做法的詳盡指示。

■ 第九章 — 第 189 頁的「管理使用者存取與策略」

本章提供透過 Sun Grid Engine 5.3 系統取得的使用者策略類型的完整背景資訊，並提供如何將這些策略配對到計算環境的指示。

■ 第十章 — 第 215 頁的「管理平行環境」

除了說明 Sun Grid Engine 5.3 系統如何與 *平行環境* 一致之外，本章還提供處理它們的完整配置指示。

■ 第十一章 — 第 225 頁的「錯誤訊息」

本章解釋 Sun Grid Engine 5.3 取得錯誤訊息的程序，並說明如何在除錯模式中執行軟體。

主機和叢集配置

本章提供配置系統 Sun Grid Engine 5.3 不同層面的背景資訊和指示。您會在本章找到以下作業的指示。

- 第 126 頁的「如何以 QMON 配置管理主機」
- 第 128 頁的「如何刪除管理主機」
- 第 128 頁的「如何新增管理主機」
- 第 128 頁的「如何從指令行配置管理主機」
- 第 128 頁的「如何以 QMON 配置提交主機」
- 第 130 頁的「如何刪除提交主機」
- 第 130 頁的「如何新增提交主機」
- 第 130 頁的「如何從指令行配置提交主機」
- 第 130 頁的「如何以 QMON 配置執行主機」
- 第 132 頁的「如何刪除執行主機」
- 第 132 頁的「如何關閉執行主機常駐程式」
- 第 132 頁的「如何新增或修改執行主機」
- 第 136 頁的「如何從指令行配置執行主機」
- 第 137 頁的「如何使用 qhost 監視執行主機」
- 第 138 頁的「如何從指令行刪除常駐程式」
- 第 139 頁的「如何從指令行重新啓動常駐程式」
- 第 139 頁的「如何從指令行顯示基礎叢集配置」
- 第 140 頁的「如何從指令行修改基礎叢集配置」
- 第 140 頁的「如何使用 QMON 顯示叢集配置」
- 第 141 頁的「如何使用 QMON 刪除叢集配置」
- 第 141 頁的「如何使用 QMON 顯示全域叢集配置」
- 第 142 頁的「如何使用 QMON 修改全域和主機配置」

關於主要和備用主要配置

備用主要主機名稱檔案，`<sg_e_root>/<儲存格>/common/shadow_masters` 包含主要的主要主機（Sun Grid Engine 主要常駐程式 `sg_e_qmaster` 初始啟動的機器）和備用主要主機的名稱。主要主機名稱檔案的格式如下。

- 檔案的第一行定義主要的主要主機
- 以下各行指定備用主要主機，每行一個。

（備用）主要主機出現的順序是很重要的。如果主要的主要主機（檔案的第一行）無法繼續執行，則定義在第二行的備用主要主機將會接替。如果這個也故障，則定義在第三行的會執行工作，以此類推。

若要將主機準備成 Sun Grid Engine 備用主機，需先符合以下的需求：

- 備用主機需要執行 `sg_e_shadowd`。
- 備用主要主機需要共享 `sg_e_qmaster` 的狀態資訊、作業和紀錄至磁碟的佇列配置。特別是，（備用）主要主機需要至主機的多工緩衝處理目錄和 `<sg_e_root>/<儲存格>/common` 目錄的讀取/寫入 `root` 存取權。
- 備用主要主機名稱檔案必須包含定義主機為備用主要主機的資料行。

一旦符合這些要求，則該主機的備用主要主機功能就會啟動。要啟動這個功能不需重新啟動 Sun Grid Engine 常駐程式。

備用主要主機上 `sg_e_qmaster` 的自動錯誤修復啟動需要花費一些時間（以一分鐘為順序）。同時每當執行 Sun Grid Engine 指令時，您就會得到對應的錯誤訊息。

注意 – `<sg_e_root>/<儲存格>/common/act_qmaster` 檔案包含實際執行 `sg_e_qmaster` 常駐程式的主機名稱。

爲了要啟動備用 `sg_e_qmaster`，Sun Grid Engine 必須確定原來的 `sg_e_qmaster` 已經終止或它將會終止，且不會執行動作來干預剛啟動的備用 `sg_e_qmaster`。這個鮮少無法達成。在這些例子中，對應的錯誤訊息將紀錄在備用主要主機上 `sg_e_shadowd` 的訊息日誌檔（請參閱第 225 頁的「錯誤訊息」的第十一章）而任何開啓至 `sg_e_qmaster` 常駐程式的 `tcp` 連線都會永久失敗。如果發生的話，請確認主要常駐程式現在沒有在執行，並在任何的備用主要機器上手動重新啟動 `sg_e_qmaster`（請參閱第 138 頁的「如何從指令行刪除常駐程式」一節）。

關於常駐程式和主機

Sun Grid Engine 根據系統上執行的常駐程式以及主機登錄方法，主機可以分為四個群組 `sge_qmaster`

- **主要主機** – 主要主機是全部叢集活動的中心。它會執行主要常駐程式 `sge_qmaster`。`sge_qmaster` 來控制所有 Sun Grid Engine 元件，如佇列和作業，並維護元件狀態、使用者存取權等等的表格。第 27 頁的「如何安裝主要主機」一節說明如何初始設定主要主機，而第 124 頁的「關於主要和備用主要配置」一節則顯示如何配置動態主要主機變更。主要主機通常執行 Sun Grid Engine 排程器 `sge_schedd`。主要主機不需要安裝程序之外所執行的進一步配置。
- **執行主機** – 執行主機通常是有執行 Sun Grid Engine 作業權限的節點。因此，它們管理 Sun Grid Engine 佇列並執行 Sun Grid Engine 執行常駐程式，`sge_execd`。（執行主機由第 28 頁的「如何安裝執行主機」一節所述的執行主機安裝程序進行初始設定）。
- **管理主機** – 權限可以給予主要主機之外的其他主機，好在 Sun Grid Engine 執行任何種類的管理活動。管理主機可以用下述指令設定：

```
qconf -ah 主機名稱
```

細節請參閱 `qconf` 線上說明手冊。

- **提交主機** – 提交主機只允許提交和控制批次作業。特別是，登入提交主機的使用者可以透過 `qsub` 提交作業、並透過 `qstat` 控制作業狀態或執行 Sun Grid Engine 的 OSF/1 Motif 圖形使用者介面，`QMON`。提交主機可以用下述指令設定：

```
qconf -as 主機名稱
```

細節請參閱 `qconf` 線上說明手冊。

注意 – 主機在上述的類別中可以歸類到一個以上的類別。主要主機是預設的管理和提交主機。

關於配置主機

Sun Grid Engine 維護主要主機之外所有主機種類的物件清單。若是管理和提交主機，則這些清單只提供主機是否有管理或提交權限的資訊。如果是執行主機物件，則在主機上執行的物件，如 `sge_execd` 所回報的負載資訊，就會儲存在那裡，而負載參數比例因數將由 Sun Grid Engine 管理員所提供。

以下各節解釋如何使用 Sun Grid Engine 圖形使用者介面，QMON，的說明並從指令行配置不同的主機物件。

GUI 管理由一套主機配置對話方塊所提供，其可以按 QMON 主功能表的「主機配置」圖示鈕呼叫。可用的對話方塊為管理主機配置（請參閱圖 6-1）、提交主機配置（請參閱圖 6-2）和執行主機配置（請參閱圖 6-3）。對話方塊可使用螢幕頂端的選擇清單鈕切換。

`qconf` 指令為主機物件管理提供指令行介面。

▼ 如何以 QMON 配置管理主機

1. 按一下 QMON 主功能表頂端的「管理主機」標籤。
「管理主機配置 (Administration Host Configuration)」對話方塊（與下圖類似）開啓。

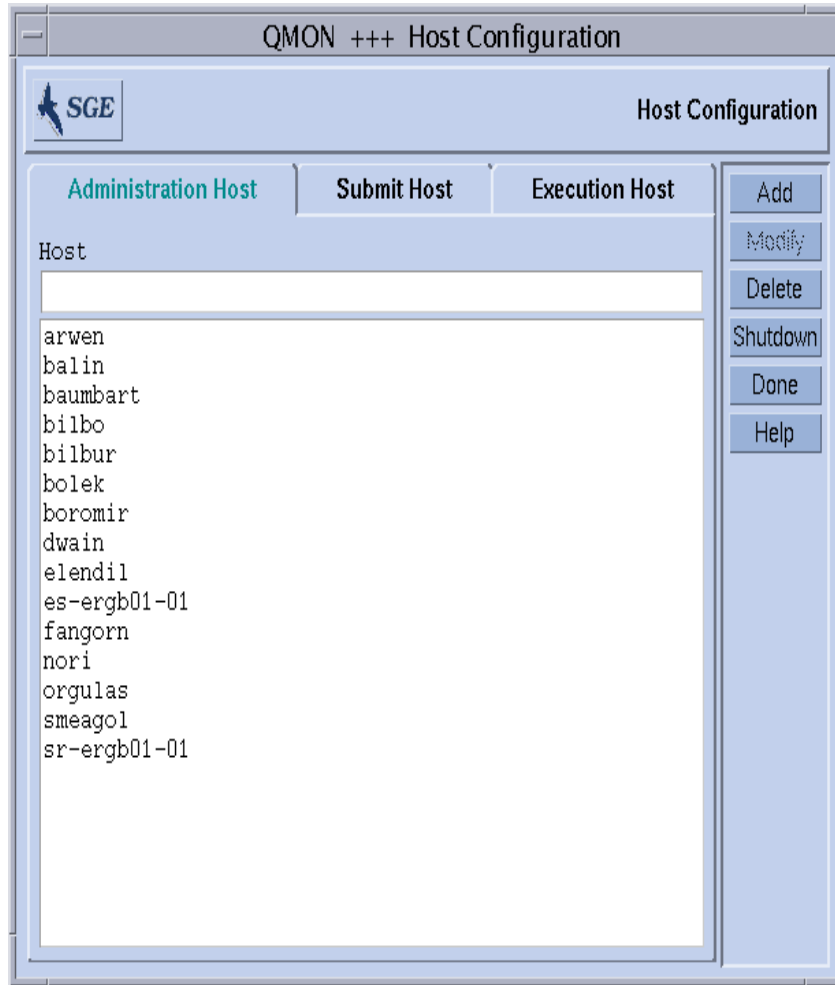


圖 6-1 管理主機配置對話方塊

注意 – 當第一次按下「主機配置」鈕時，預設就會開啓「管理主機配置」對話方塊。

2. 根據您要配置主機的方式，依照以下各節的指導進行。

有了這個對話方塊，您可以從允許 Sun Grid Engine 管理指令的主機上配置主機。在螢幕中間的選擇清單會顯示已宣告可提供管理權限之主機。

▼ 如何刪除管理主機

- 在現存主機名稱上按一下滑鼠左鍵，然後按對話方塊底部的「刪除」鈕就可從該清單刪除現存主機。

▼ 如何新增管理主機

- 將名稱輸入「主機名稱」輸入視窗，然後按「新增」鈕或「Return」鍵增加新的主機。

▼ 如何從命令行配置管理主機

- 依照您想如何配置主機的方式，用合適的引數輸入以下指令。

```
% qconf 引數
```

qconf 指令的引數和其結果如下。

- qconf -ah *主機名稱*
新增管理主機 — 將指定主機新增至管理主機的清單。
- qconf -dh *主機名稱*
刪除管理主機 — 將指定主機從管理主機的清單刪除。
- qconf -sh
顯示管理主機 — 顯示所有目前配置的管理主機清單。

▼ 如何以 QMON 配置提交主機

1. 按一下 QMON 主功能表頂端的「提交主機 (Submit Host)」標籤。
「提交主機配置」對話方塊（與下圖類似）開啓。

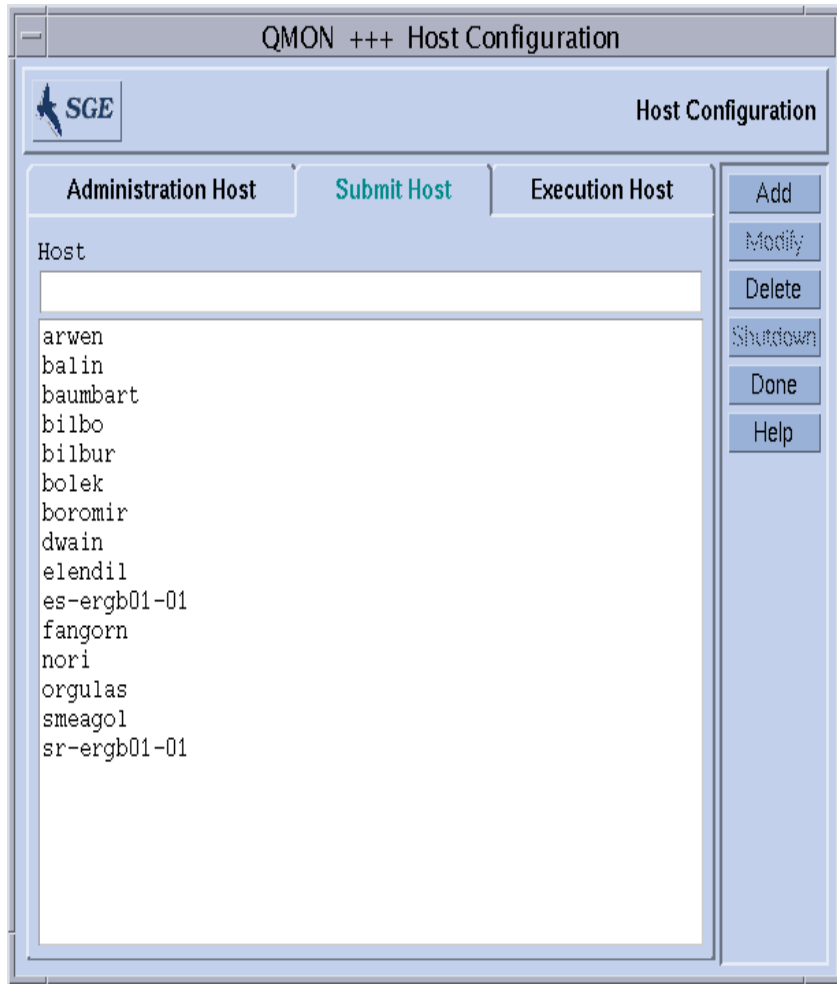


圖 6-2 提交主機配置

2. 根據您要配置主機的方式，依照以下各節的指導進行。

使用此對話方塊，您就可以從提交、監視和控制作業的主機上宣告主機。這些主機不允許管理 Sun Grid Engine 指令，除非他們也被宣告為管理主機（請參閱第 126 頁的「如何以 QMON 配置管理主機」）。在螢幕中間的選擇清單會顯示已宣告可提供提交權限之主機。

▼ 如何刪除提交主機

- 按一下「提交主機」對話方塊中現存主機的名稱，然後按對話方塊底部的「刪除」鈕就可從該清單刪除現存主機。

▼ 如何新增提交主機

- 將名稱輸入「提交主機」對話方塊中的「主機名稱」輸入視窗，然後按「新增」鈕或「Return」鍵增加新的主機。

▼ 如何從命令行配置提交主機

- 依照您想如何配置主機的方式，用合適的引數輸入以下指令。

```
% qconf 引數
```

qconf 指令的引數和其結果如下。

- qconf -as *主機名稱*
新增提交主機 — 將指定主機新增至提交主機的清單。
- qconf -ds *主機名稱*
刪除提交主機 — 將指定的主機從提交主機清單刪除。
- qconf -ss
顯示提交主機 — 顯示目前所有配置提供提交權限的主機名稱清單。

▼ 如何以 QMON 配置執行主機

1. 按一下 QMON 主功能表頂端的「執行主機」(Execution Host) 標籤。
「執行主機配置」對話方塊（與下圖類似）開啓。

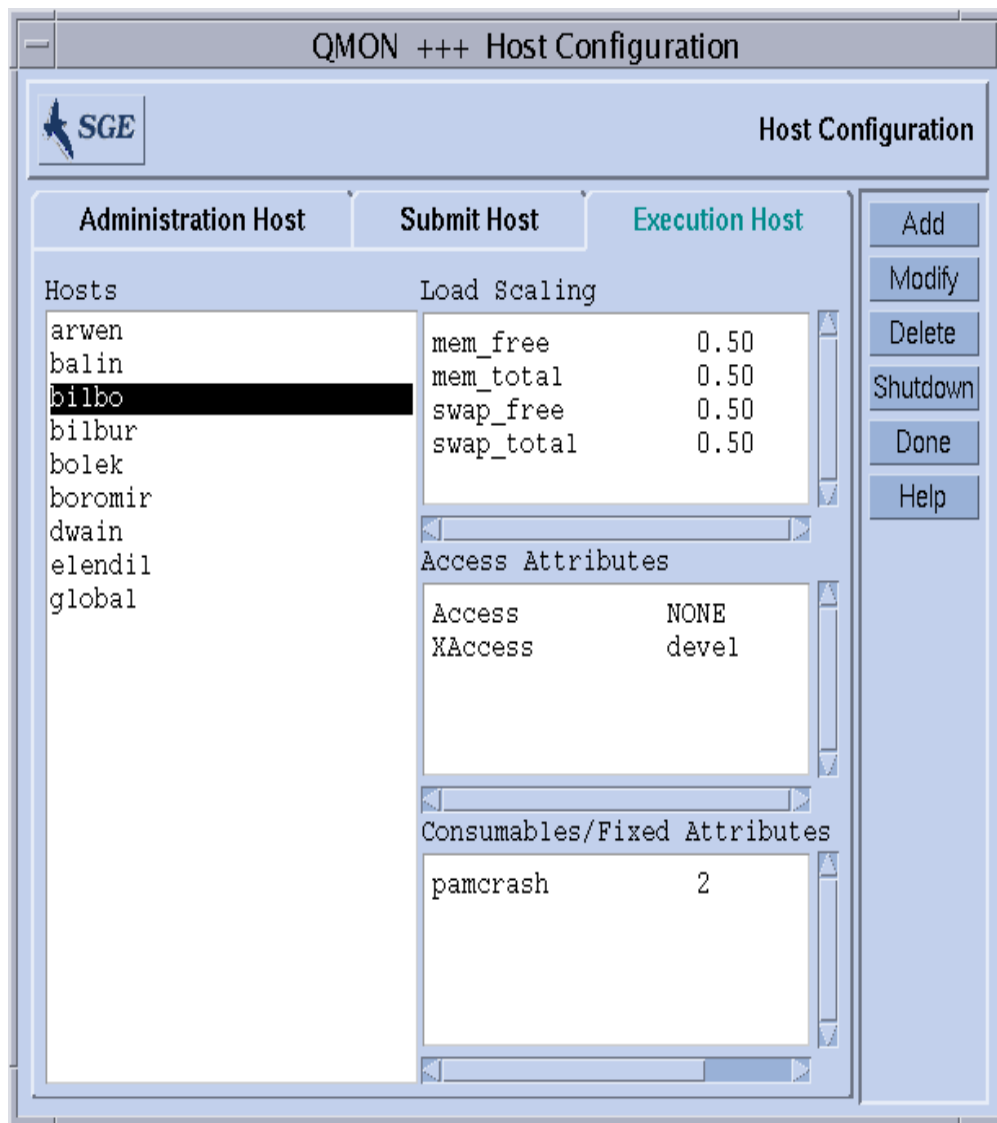


圖 6-3 執行主機配置

2. 根據您要配置主機的方式，依照以下各節的指導進行。

Sun Grid Engine 執行主機可以從此對話方塊配置。這些主機不會自動允許管理或提交指令，除非他們也被宣告為管理或提交主機（請參閱第 126 頁的「如何以 QMON 配置管理主機」和第 128 頁的「如何以 QMON 配置提交主機」）。

主機選擇清單會顯示已定義的執行主機。目前配置給主機相關的消耗和固定聚合體屬性的負載比例因數 (load scaling factors)、存取權限和資源可用性會顯示在「負載比例」(Load Scaling)，而「存取屬性」和「可消耗/固定屬性」會顯示選取執行主機的視窗。要取得聚合體屬性、使用者權限和負載參數的細節，請參考第 163 頁的「關於聚合體」、第 59 頁的「使用者存取權限」和第 183 頁的「載入參數」等節。

▼ 如何刪除執行主機

- 在「執行主機」對話方塊中，按一下要刪除的「執行主機」名稱，然後按對話方塊右邊按鈕欄中的「刪除」鈕。

▼ 如何關閉執行主機常駐程式

- 對任何選取的主機來說，請按「執行主機」對話方塊的「關閉」鈕。

▼ 如何新增或修改執行主機

1. 請按「執行主機」對話方塊按鈕欄中的「新增」或「修改」鈕。
螢幕上將出現類似於顯示於圖 6-4 的對話方塊。

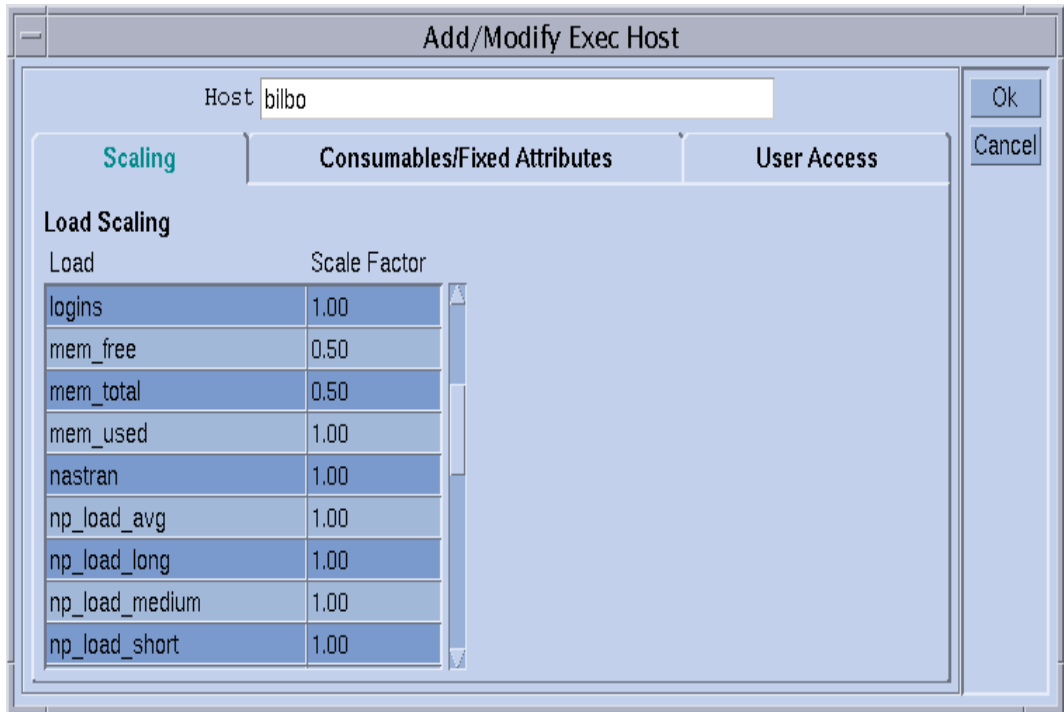


圖 6-4 修改負載比例

2. 根據您要修改主機的方式，依照以下各節的指導進行。

用來新增執行主機或修改現存執行主機的配置對話方塊可進行所有與主機相關之屬性修改。執行主機的名稱可以顯示或新增進「主機」輸入視窗。您可以選擇對話方塊中的「比例」標籤來定義比例因數（請參閱圖 6-4）。

所有可用的負載參數都顯示於「負載比例」表的「負載」欄中，而對應的比例定義可在「比例因數」欄找到。您可以編輯「比例因數」欄。有效的比例因數是固定點表示法或科學記號中正值浮點記號。

若在標籤工具集 (tab widget) 中選擇「可消耗 / 固定屬性」，則可定義與主機相關的聚合體屬性（請參閱圖 6-5）。與主機相關的聚合體（請參閱第 163 頁的「關於聚合體」一節）是全域和主機聚合體或是透過對話方塊左側底部的「聚合體選擇」(Complex Selection) 區而連接主機的**管理員定義聚合體**。可用的管理員定義聚合體顯示在左邊，而其可以透過紅色箭頭連接或脫離。若您需要目前聚合體配置的進一步資訊或您想進行修改，「聚合體」配置圖示鈕可打開最上層的「聚合體配置」對話方塊。

對話方塊右側底部區域的「可消耗 / 固定屬性」表會列出目前所有數值定義的聚合體屬性。按一下上方的「名稱」或「數值」鈕來增加清單。之後會開啓一選擇清單，列出所有隨附於主機的屬性（如上述之配置於全域、主機、與附接於該主機之管理員定義聚合

體的所有屬性的結合 (union)。「屬性選擇 (Attribute Selection)」對話方塊圖顯示於圖 6-6。選擇其中一個屬性，並按「確定」鈕確認選項，就會將此屬性新增至「可消耗/固定屬性」表中的「名稱」欄，並將指標放入對應的「數值」欄位中。您可以在「數值」欄位上連按兩下滑鼠左鍵來修改現存的數值。您可用滑鼠左鍵先選擇對應表格的行列來刪除屬性。鍵入 CTRL-D 或按滑鼠右鍵開啓刪除方塊並確認刪除動作，就可刪除選取的清單記錄。

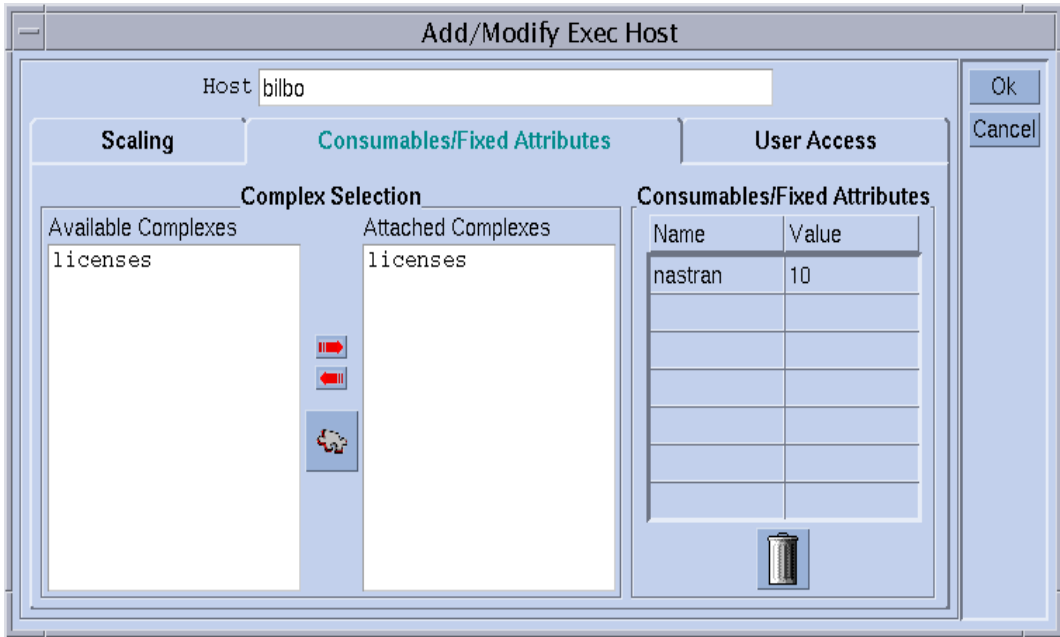


圖 6-5 修改可消耗 / 固定屬性

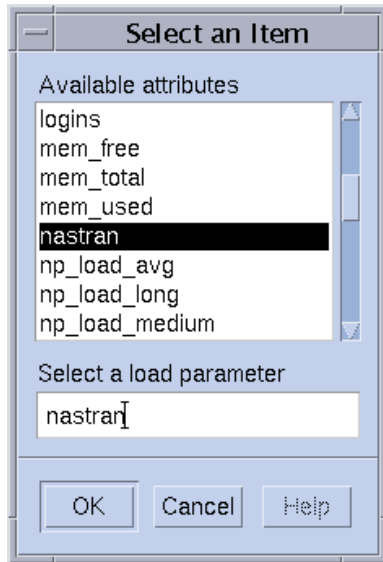


圖 6-6 可用的聚合體屬性

藉選擇「使用者存取」標籤（圖 6-7），您就可以定義以之前配置的使用者存取清單為基礎之執行主機的存取權。

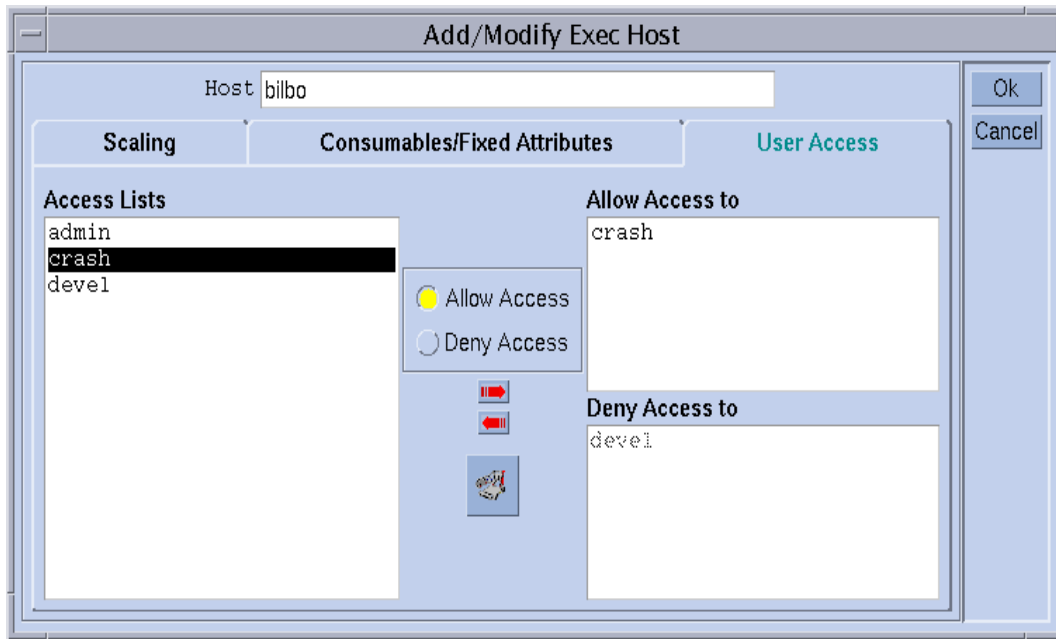


圖 6-7 修改使用者存取

▼ 如何從指令行配置執行主機

- 依照您想如何配置主機的方式，用合適的引數輸入以下指令。

```
% qconf 引數
```

維護執行主機清單的指令行介面由以下選項提供給 `qconf` 指令。

- `qconf -ae [執行主機範本]`

新增執行主機 — 使用執行主機配置範本叫出編輯器（預設的 `vi` 或對應至 `$EDITOR` 環境變數）。若選擇性參數 `執行主機範本`（已配置執行主機的名稱）已經存在，則該執行主機的配置可當成範本使用。變更範本並儲存至磁碟中可配置執行主機。請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》的 `host_conf` 項目以取得將變更的範本項目之詳細說明。

- `qconf -de 主機名稱`

刪除執行主機 — 將指定主機從執行主機的清單刪除。執行主機配置中的所有項目都已遺失。

- `qconf -me` *主機名稱*

修改執行主機 — 使用指定執行主機的配置做為範本叫出編輯器（預設的 `vi` 或對應至 `$EDITOR` 環境變數）。變更範本並儲存至磁碟中就可修改執行主機配置。請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine, 企業版 5.3 參考手冊*》的 `host_conf` 線上說明手冊以取得將變更的範本項目之詳細說明。

- `qconf -Me` *檔案名稱*

修改執行主機 — 使用 *檔案名稱* 的內容做為執行主機配置範本。指定檔案中的配置必須參照現存的執行主機。該執行主機的配置可用檔案內容置換。這個 `qconf` 選項對離線的執行主機配置變更有效；如在 `cron` 作業中，因為其不需要手動互動 (`manual interaction`)。

- `qconf -se` *主機名稱*

顯示執行主機 — 如 `host_conf` 所定義的顯示指定執行主機的配置。

- `qconf -sel`

顯示執行主機清單 — 顯示配置為執行主機的主機名稱清單。

▼ 如何使用 `qhost` 監視執行主機

`qhost` 指令提供簡單取得執行主機狀態之快速簡介的方法。

- 輸入以下指令。

```
% qhost
```

該指令產出類似下列的輸出。

表 6-1 範例 qhost 輸出

HOSTNAME	ARCH	NPROC	LOAD	MEMTOT	MEMUSE	SWAPTO	SWAPUS
global	-	-	-	-	-	-	-
BALROG.genias.de	solaris6	2	0.38	1.0G	994M	900M	891M
BILBUR.genias.de	solaris	1	0.18	96M	70M	164M	9.0M
DWAIN.genias.de	irix6	1	1.13	149M	55.8M	40M	0.0
GLOIN.genias.de	osf4	2	0.05	768M	701M	1.9G	13.5M
SPEEDY.genias.de	alinux	1	0.08	248M	60.6M	125M	232.0K
SARUMAN.genias.de	solaris	1	0.11	96M	77M	192M	9.0M
FANGORN.genias.de	linux	1	2.01	124.8M	49.9M	127.7M	4.3M

要取得輸出格式的說明和更多的選項，請參考《*Sun Grid Engine 5.3 和 Sun Grid Engine*，企業版 5.3 參考手冊》的 qhost 項目。

▼ 如何從指令行刪除常駐程式

- 請使用以下其中一項指令。請注意，您需要 Sun Grid Engine 管理員或操作者權限才能執行這些作業（請參閱第九章，第 189 頁的「管理使用者存取與策略」）。

```
% qconf -kej  
% qconf -ks  
% qconf -km
```

- 第一個指令將刪除所有目前活動中的作業並結束所有的 Sun Grid Engine 執行常駐程式。

注意 – 若使用 qconf -ke 置換該指令，則 Sun Grid Engine 執行常駐程式將中斷，但是活動中的作業不會取消。當 sge_execd 沒在該系統上執行時，結束的作業會到 sge_execd 再度重新啟動後才會報告給 sge_qmaster。然而，作業報告不會遺失。

- 第二個指令會關閉 Sun Grid Engine 排程器 sge_schedd。
- 第三個指令會強制 sge_qmaster 過程終止。

若您有執行作業，且您想用 Sun Grid Engine 的關機程序等到目前活動的作業都結束，您可以在執行上述的 qconf 序列前，針對每個佇列使用下述指令。

```
% qmod -d 佇列名稱
```

qmod 停用指令會防止新作業排程進入停用的佇列。然後您應該等候刪除常駐程式，直到佇列中不再有作業執行。

▼ 如何從指令行重新啟動常駐程式

1. 以 root 身分登入您想要 Sun Grid Engine 5.3 常駐程式重新啟動的機器。
2. 執行以下的 script。

```
% <sgc_root>/< 儲存格 >/common/rcsgc
```

該 script 會尋找在該主機正常執行的常駐程式，然後啟動對應的常駐程式。

基礎叢集配置

基礎的 Sun Grid Engine 叢集配置是一套資訊，配置以反映站點相依，如 mail 或 xterm 的程式有效路徑，並影響 Sun Grid Engine 運作方式。提供的全域配置是用於 Sun Grid Engine 儲存區中的 Sun Grid Engine 主要主機和每個主機。此外，Sun Grid Engine 系統可以配置使用每個主機的本端配置來置換全域配置中的特別項目。

《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》的 sgc_conf 項目包含配置項目的詳細說明。Sun Grid Engine 叢集管理員應在安裝後直接根據站點需求調整全域和本端配置，並在之後維持更新狀態。

▼ 如何從指令行顯示基礎叢集配置

顯示目前配置的 Sun Grid Engine 指令是 qconf 程式的顯示配置選項。以下為幾個範例（詳細說明請參閱《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》）。

- 請輸入以下其中一項指令。

```
% qconf -sconf
% qconf -sconf global
% qconf -sconf < 主機 >
```

前兩個指令是相等的，並會顯示全域配置。第三個指令則顯示主機的本端配置。

▼ 如何從指令行修改基礎叢集配置

注意 – 變更叢集配置的 Sun Grid Engine 指令 —qconf— 只能為 Sun Grid Engine 管理員所用。

- 請輸入以下其中一項指令。

```
% qconf -mconf global
% qconf -mconf < 主機 >
```

- 第一個指令範例修改全域配置。
- 第二個範例在指定的執行或主要主機之本端配置上操作。

上述兩個指令為許多可用 qconf 指令之範例。其他指令請參考 《*Sun Grid Engine 5.3 和 Sun Grid Engine, 企業版 5.3 參考手冊*》。

▼ 如何使用 QMON 顯示叢集配置

1. 在 QMON 主要功能表中按下 「叢集配置」 鈕。
螢幕上顯示類似圖 6-8 範例的 「叢集配置」 對話方塊。

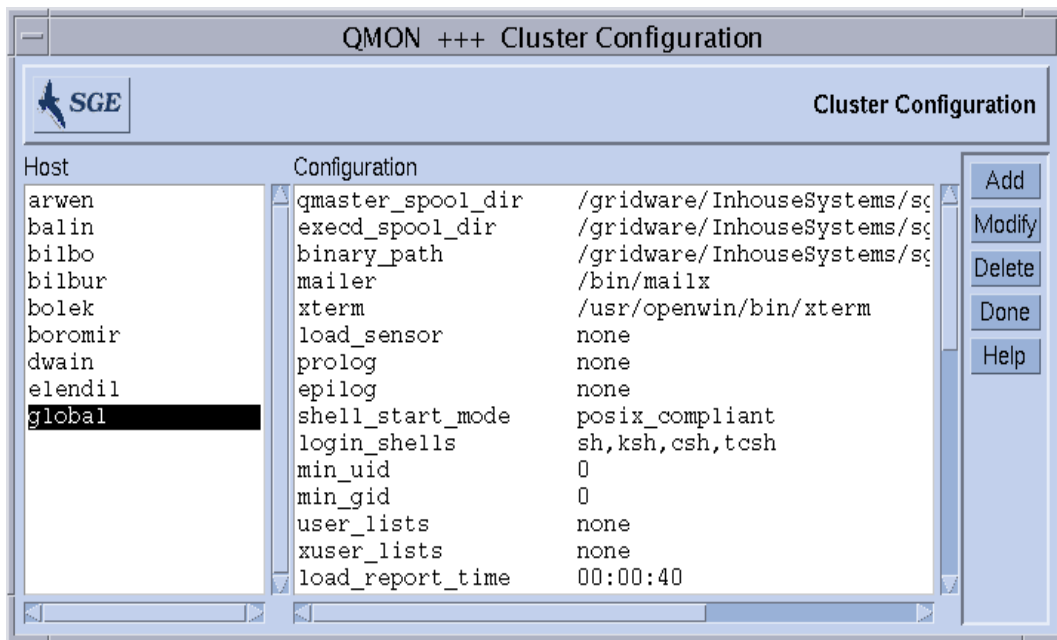


圖 6-8 「叢集配置」對話方塊

2. 在螢幕左側的「主機」選擇清單中，按一下主機名稱可顯示該主機目前的配置。

▼ 如何使用 QMON 刪除叢集配置

1. 在 QMON 主要功能表中按一下「叢集配置」鈕。
2. 在螢幕左側的「主機」選擇清單中，按一下您想刪除配置之主機的名稱。
3. 請按「刪除」鈕。

▼ 如何使用 QMON 顯示全域叢集配置

- 在「主機」選擇清單中，選擇名稱，global。

配置以 `sge_conf` 線上說明手冊所述的格式顯示。使用「修改」鈕來修改選取的全域或主機本端配置。使用「新增」鈕為特定主機新增新的配置。

▼ 如何使用 QMON 修改全域和主機配置

1. 在「叢集配置」對話方塊中（如第 140 頁的「如何使用 QMON 顯示叢集配置」一節所述），按一下「新增」鈕或「修改」鈕。

類似圖 6-9 範例的「叢集設定」對話方塊就會開啓。

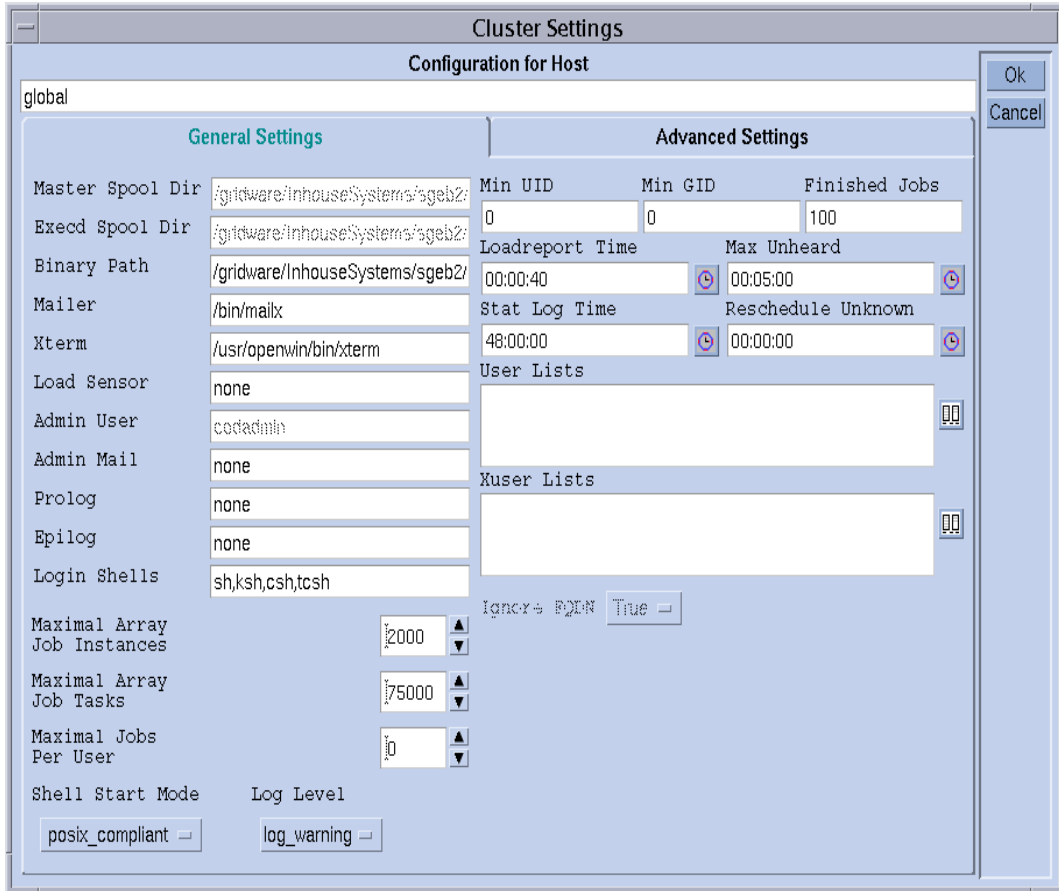


圖 6-9 叢集設定對話方塊 — 一般設定

2. 請參照以下各節的詳細資訊進行變更。

「叢集設定」對話方塊提供變更全域或主機本端配置的所有參數的方法。所有的項目欄位只在變更全域配置時才能存取；即選取主機、全域，然後按下「修改」。如果修改一般主機，則其實際的配置會反映在對話方塊中，而只有適合主機本端變更的參數才能修改。如果增加新的主機本端配置，則對話方塊項目會是空的欄位。

「進階設定」標籤（圖 6-10）會根據變更的是全域、主機本端或新的配置來顯示對應的運作方式。它會提供較少用到的叢集配置參數的存取。

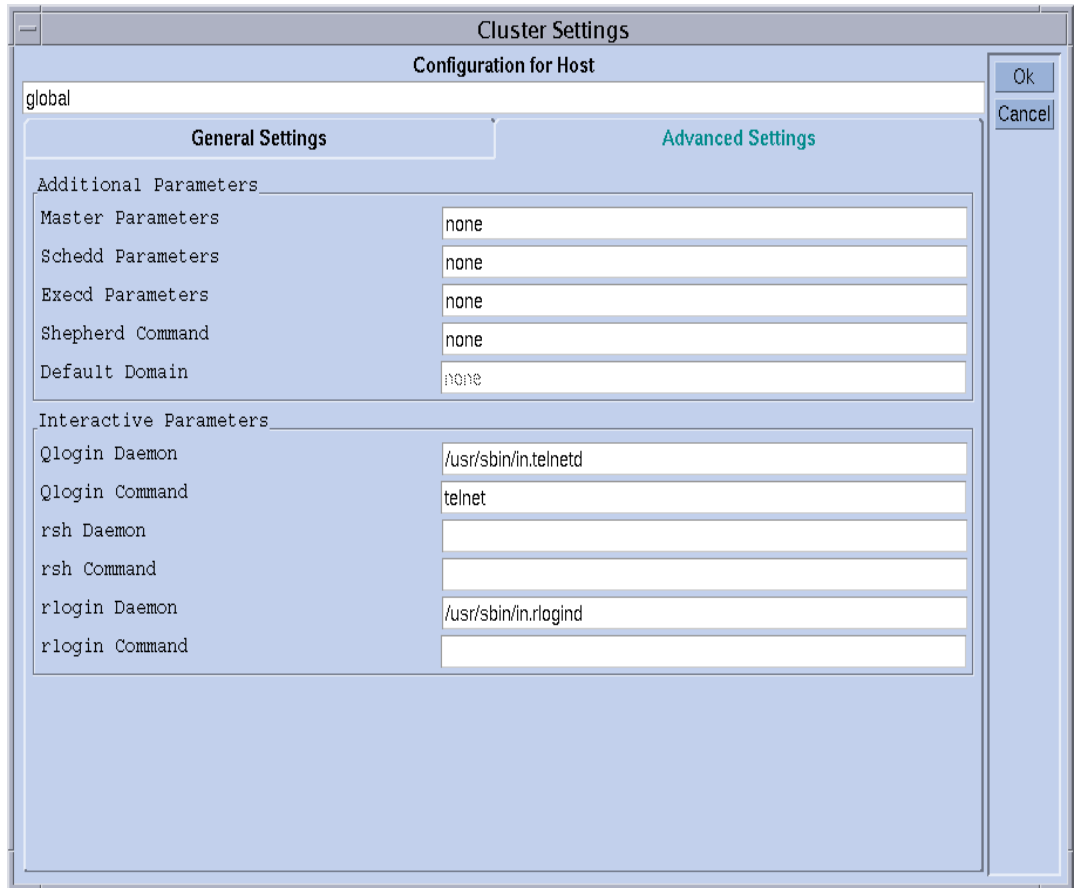


圖 6-10 叢集設定對話方塊 — 進階設定

在完成修改後，右上角的「確定」鈕會紀錄修改的配置。按「取消」可刪除任何變更。這兩個例子都會關閉對話方塊。

要取得所有叢集配置參數的完整說明，請參考 `sge_conf` 線上說明手冊。

配置佇列與佇列行事曆

本章提供配置 Sun Grid Engine 5.3 佇列的相關背景資訊與指示。

以下為本章涵蓋指示的特定作業清單。

- 第 146 頁的「如何以 QMON 配置佇列」
- 第 147 頁的「如何配置一般參數」
- 第 148 頁的「如何配置執行法參數」
- 第 149 頁的「如何配置檢查點參數」
- 第 150 頁的「如何配置載入 / 暫停臨界值」
- 第 151 頁的「如何配置限制」
- 第 152 頁的「如何配置使用者聚合體」
- 第 154 頁的「如何配置次級佇列」
- 第 155 頁的「如何配置使用者存取」
- 第 156 頁的「如何配置所有人」
- 第 156 頁的「如何從指令行配置佇列」
- 第 158 頁的「如何以 QMON 配置佇列行事曆」
- 第 160 頁的「如何從指令行配置行事曆」

關於佇列配置

Sun Grid Engine 佇列為不同種類的作業容器，為屬於同一種類但同時執行之不同作業提供對應資源。作業不會停留在 Sun Grid Engine 佇列中，反而會在分配至佇列當下立刻執行。Sun Grid Engine 排程器作業擱置清單是 Sun Grid Engine 作業的唯一等候區。

配置 Sun Grid Engine 佇列將會以 `sge_qmaster` 記錄佇列屬性。一旦佇列配置完畢，就會立即顯示在全部叢集上，且位於 Sun Grid Engine 儲存區所有主機上之全部 Sun Grid Engine 使用者皆可看到。

▼ 如何以 QMON 配置佇列

1. 從 QMON 主要功能表上按下「佇列控制」鈕。
2. 在「佇列控制」對話方塊中按下「新增」或「修改」鈕，

就會出現「佇列配置」的對話方塊。第 114 頁的「如何用 QMON 控制佇列」會詳加說明「佇列控制」對話方塊，和其監視與操作佇列狀態的工具。如果是第一次開啓「佇列配置」對話方塊，就會出現一般參數表（請參閱第 147 頁的「如何配置一般參數」）。

3. 請參照以下幾節的詳細資訊決定如何配置。

受到理想操作影響的佇列會顯示或定義於螢幕上方的「佇列」與「主機名稱」視窗。如果一個佇列被修改了，在「佇列配置」對話方塊開啓前，必須先從「佇列控制」對話方塊裡選出另一個現有佇列。如果要新增一個新的佇列，就必須定義佇列名稱與佇列位於的主機。

在「主機名稱」視窗下有三個按鈕可讓您輕鬆使用「佇列配置」對話方塊。「複製」鈕可透過佇列選取清單輸入一現有佇列的所有參數，「重新設定」鈕可以載入範本佇列的配置，而「重新顯示」鈕則可在對話方塊開啓時載入其餘修改過的物件配置（請參閱第 152 頁的「如何配置使用者聚合體」與第 155 頁的「如何配置使用者存取」兩節以得知更多「重新顯示」鈕的詳細資訊）。

「佇列配置」對話方塊右上方的「確定」按鈕會以 `sge_qmaster` 記錄這些變更，而右下方的「取消」按鈕則會捨棄任何變更。以上兩個按鈕都會關閉對話方塊。

有九組參數組可用來定義一佇列。

- 一般（請參閱第 147 頁的「如何配置一般參數」）
- 執行法（請參閱第 148 頁的「如何配置執行法參數」）
- 檢查點（請參閱第 149 頁的「如何配置檢查點參數」）
- 載入 / 暫停臨界值（請參閱第 150 頁的「如何配置載入 / 暫停臨界值」）
- 限制（請參閱第 151 頁的「如何配置限制」）
- 聚合體（請參閱第 152 頁的「如何配置使用者聚合體」）
- 次級（請參閱第 154 頁的「如何配置次級佇列」）
- 使用者存取（請參閱第 155 頁的「如何配置使用者存取」）
- 所有人（請參閱第 156 頁的「如何配置所有人」）

您可透過「佇列參數」標籤選擇理想的參數組。

▼ 如何配置一般參數

- 選擇「一般」參數組。

以下顯示的圖 7-1 為範例圖示。

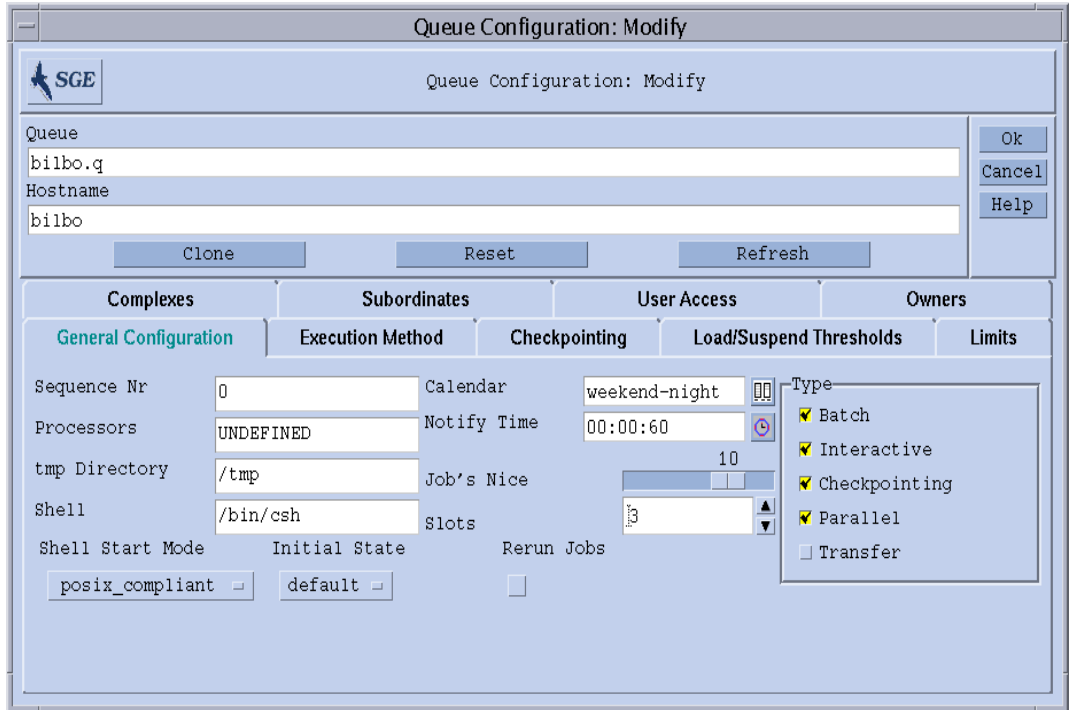


圖 7-1 佇列配置 — 一般參數

提供的輸入欄位可用來設定以下參數：

- 佇列的序列編號。
- 處理器 — 為一個可設定佇列作業使用處理器之指定器。對一些作業系統架構而言，這可以是一範圍（例如 1-4、8、10）或是處理器組的整數辨識器。請參閱您 Sun Grid Engine 分配之 doc 目錄中的 arc_depend_*.asc 檔以得知更多相關資訊。
- 暫存目錄路徑。
- 用以執行作業 script 的預設指令解譯器 (Shell)。
- 佇列隨附之行事曆：可為佇列定義作業中與結束作業的時間。
- 傳達 SIGUSR1/SIGUSR2 通知訊號與暫停 / 刪除訊號之間的等候時間 (Notify)。
- 用以啟動佇列作業的 nice 值（0 表示使用系統預設值）。
- 佇列中可同時執行的作業數量（作業插槽）。
- 佇列中可執行之佇列與作業種類。有多重選擇可供使用。
- Shell Start Mode，即啟動作業 script 的模式。

- Initial State：可新增佇列，或者在佇列主機上執行的 `sge_execd` 啟動時可儲存佇列。
- 佇列預設之 rerun 策略：當捨棄作業時會執行之策略，例如當系統故障時。使用者可透過 `qsub -r` 選項或者「作業提交」對話方塊覆寫此策略（請參閱圖 4-7）。

請參閱 `queue_conf` 線上說明手冊以得知更多參數相關資訊。

▼ 如何配置執行法參數

- 選擇「執行法」參數組。

以下顯示的圖 7-2 為範例圖示。

The screenshot shows a window titled "Queue Configuration: Modify" for the Sun Grid Engine (SGE). The window contains the following elements:

- Queue:** `bilbo.q`
- Hostname:** `bilbo`
- Buttons:** Clone, Reset, Refresh, Ok, Cancel, Help
- Configuration Tabs:**
 - Complexes
 - Subordinates (Active)
 - General Configuration
 - Execution Method (Active)
 - Prolog
 - Epilog
 - Starter Method
 - Suspend Method
 - Resume Method
 - Terminate Method
 - Checkpointing
 - Load/Suspend Thresholds
 - Limits
 - User Access
 - Owners

圖 7-2 佇列配置 — 執行法參數

提供的輸入欄位可用來設定以下參數：

- 在作業 script 開始與結束時，與作業擁有相同執行環境之特定佇例 prologue 與 epilogue script。
- 可覆寫應用於作業中 Sun Grid Engine 預設值的啟動 / 暫停 / 恢復 / 終止法。

請參閱 `queue_conf` 線上說明手冊以得知更多參數相關資訊。

▼ 如何配置檢查點參數

- 選擇「檢查點」參數組。

以下顯示的圖 7-3 為範例圖示。

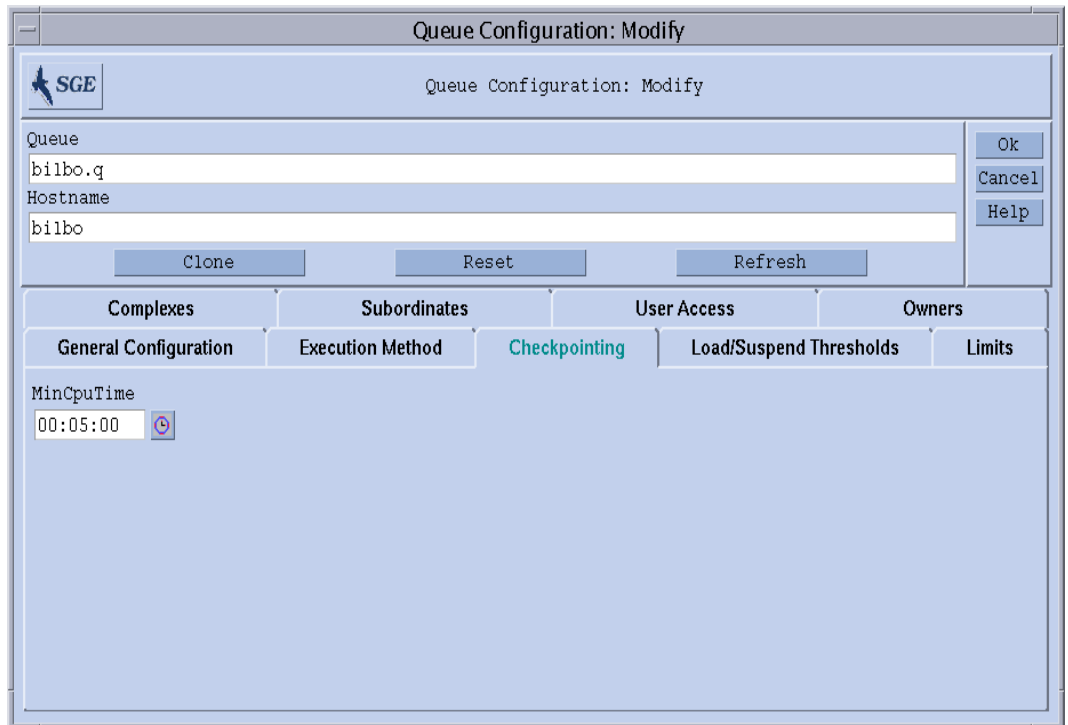


圖 7-3 佇列配置 — 檢查點參數

提供的輸入欄位可用來設定以下參數：

- 定期檢查點間隔時間 (MinCpuTime)

請參閱 `queue_conf` 線上說明手冊以得知更多參數相關資訊。

▼ 如何配置載入 / 暫停臨界值

- 選擇「載入 / 暫停臨界值」參數組。
- 以下顯示的圖 7-4 為範例圖示。

Load	Value
np_load_av	1.75
h_vmem	100M

Load	Value
np_load_av	4

Suspend interval: 00:05:00
Jobs suspended per interval: 1

圖 7-4 佇列配置 — 載入臨界值

提供的輸入欄位可用來設定以下參數：

- 載入臨界值與暫停臨界值表可用來定義載入參數的超載臨界值與可耗損聚合體屬性。（請參見第 163 頁的「關於聚合體」）。

載入臨界值的超載值可避免佇列透過 Sun Grid Engine 繼續接受作業。當超過一或二個以上的暫時臨界值時就會暫停佇列作業以減低載入。最新配置的臨界值顯示於圖表中。一個現有的臨界值可經由連接滑鼠左鍵兩下，即可選擇與變更對應「數值」欄位的設定。要增加新的臨界值，請按上方的「名稱」鈕或「數值」鈕。顯示佇列所有有效屬性的選擇清單將會開啓。「屬性選擇」對話方塊圖顯示於圖 6-6。選擇其中一個屬性，並按「確定」鈕確認選項，就會新增此屬性至相對臨界值圖表的「名稱」欄，並將游標放入數值欄位中。輸入 CTRL-D 或按滑鼠右鍵開啓刪除方塊並確認刪除動作，就可刪除選取清單記錄。

- 每次間隔時間暫時的作業數量：以減低配置佇列系統的負載。
- 更多作業的暫停間距時間：以避免暫停的臨界值仍然過多。

請參閱 queue_conf 線上說明手冊以得知更多參數相關資訊。

▼ 如何配置限制

- 選擇「限制」參數組。
以下顯示的圖 7-5 為範例圖示。

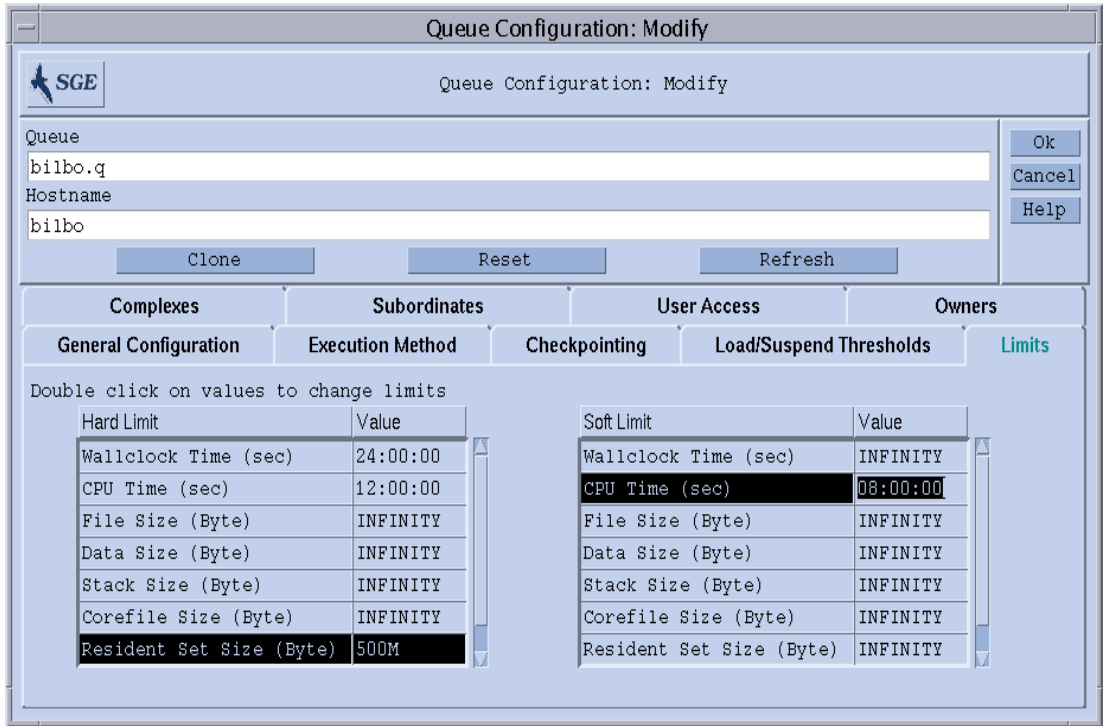


圖 7-5 佇列配置 — 限制

提供的輸入欄位可用來設定以下參數：

- 於佇列中執行作業的**硬與軟**限制。

要變更限制數值就連接兩下「限制」記錄的數值欄位。連接數值欄位四下開啓「記憶體」或「時間」限制數值的輸入對話方塊（請參見圖 7-6 與圖 7-7）。

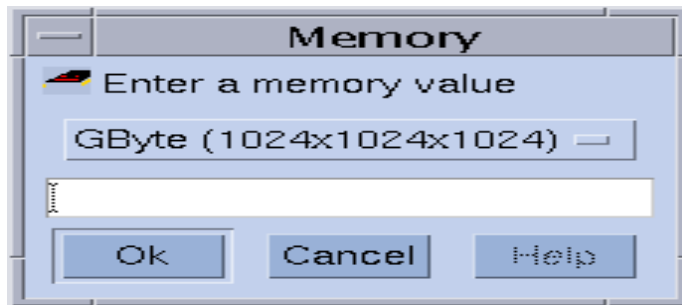


圖 7-6 記憶體輸入對話方塊

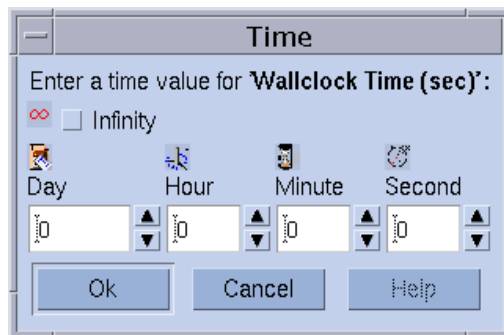


圖 7-7 時間輸入對話方塊

請參閱 `queue_conf` 與 `setrlimit` 線上說明手冊以得知更多個別限制參數，與其在不同作業系統架構中的解讀方式之詳細資訊。

▼ 如何配置使用者聚合體

- 選擇「使用者聚合體」參數組。
以下顯示的圖 7-8 為範例圖示。

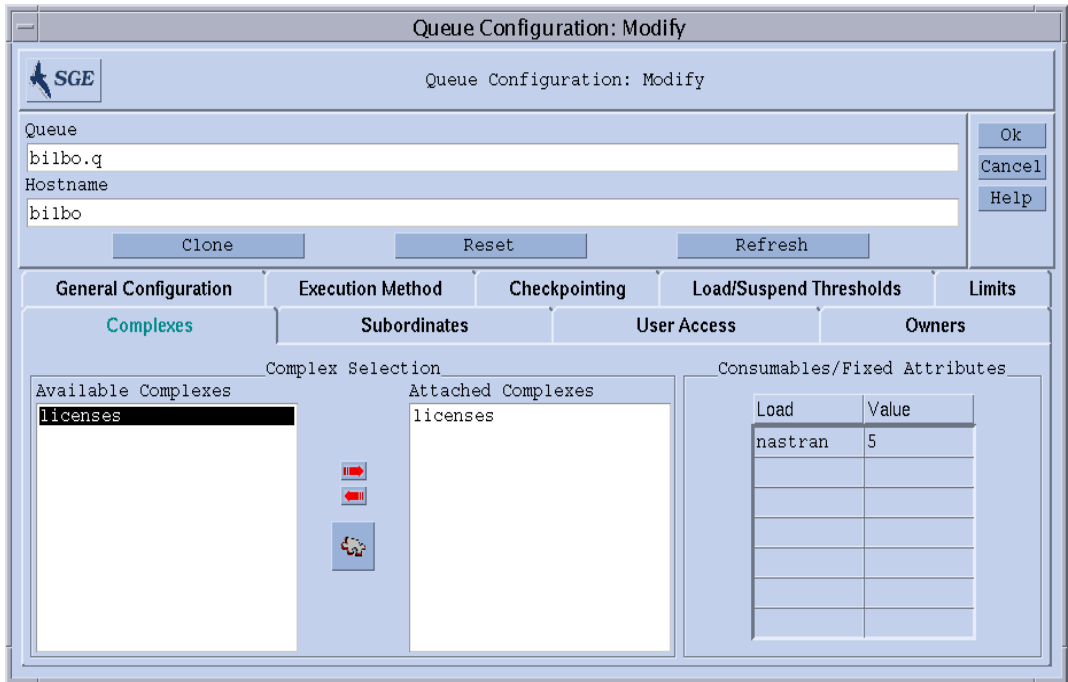


圖 7-8 佇列配置 — 使用者聚合體

提供的輸入欄位可用來設定以下參數：

- 使用者設定：定義隨附於佇列之聚合體（請參見第 168 頁的「使用者定義聚合體」）。

「聚合體選擇」方塊中的紅色箭號可從 / 到佇列中隨附或分離使用者定義之聚合體。

- 從佇列可用之聚合體參數組為選取的屬性定義數值。

可用之聚合體參數會從全域聚合體、主機聚合體、與隨附使用者定義之聚合體中集合每個預設值。屬性為可耗損或固定參數。就可耗損屬性、或固定屬性的固定佇列特定值而言，定義佇列值同時也定義了由佇列管理的容量（請見第 163 頁的「關於聚合體」得知詳細資訊）。有明確定義數值的屬性會顯示於「可耗損 / 固定屬性」表中。現有屬性可藉由連按兩下對應數值欄位做選擇與變更。要新增屬性定義，請按上方的「名稱」鈕或「數值」鈕。顯示佇列所有有效屬性的選擇清單將會開啓。「屬性選擇」對話方塊圖顯示於圖 6-6。選擇其中一個屬性，並按「確定」鈕確認選項，就會新增此屬性至屬性表之「名稱」欄，並將游標放入數值欄位中。輸入 CTRL-D 或按滑鼠右鍵開啓刪除方塊並確認刪除動作，就可刪除選取的清單記錄。

請參閱 `queue_conf` 線上說明手冊以得知更多參數相關資訊。

「聚合體配置」對話方塊（請參見第 163 頁的「聚合體概念」第八章的圖 8-5）在按下「聚合體配置」圖示按鈕時就會開啓。您可以在隨附或分離使用者定義之聚合體至佇列以前，檢查或修改目前聚合體配置。

▼ 如何配置次級佇列

- 選擇「次級」參數組。

以下顯示的圖 7-9 為範例圖示。

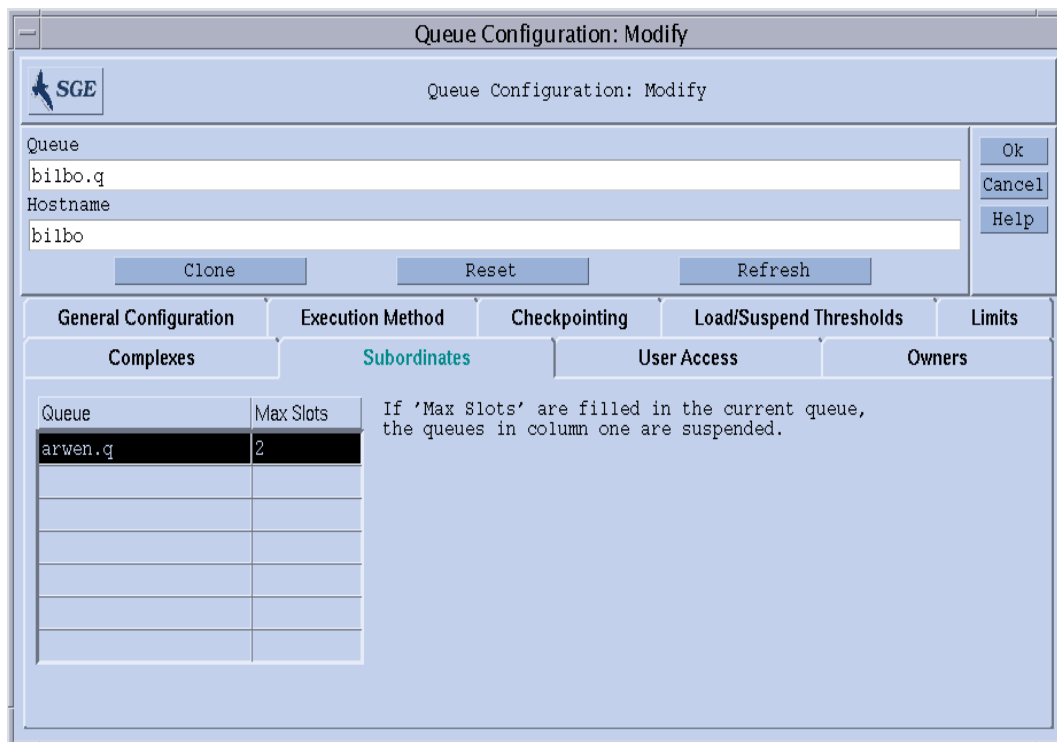


圖 7-9 佇列配置 — 次級參數

提供的輸入欄位可用來設定以下參數：

- 次於配置佇列之佇列

若配置佇列忙碌時，就會暫停次級佇列；若配置佇列不忙碌，就會解除暫停次級佇列。對次級佇列而言，每一配置佇列至少需配置作業插槽的數量以便啓動暫停功能。若無指定作業插槽數值，就必須填入所有插槽數值以啓動對應佇列之暫停功能。

請參閱 `queue_conf` 線上說明手冊以得知更多參數相關資訊。

使用次級佇列設備來執行高優先與低優先佇列，和獨立佇列。

▼ 如何配置使用者存取

- 選擇「使用者存取」參數組。
以下顯示的圖 7-10 為範例圖示。

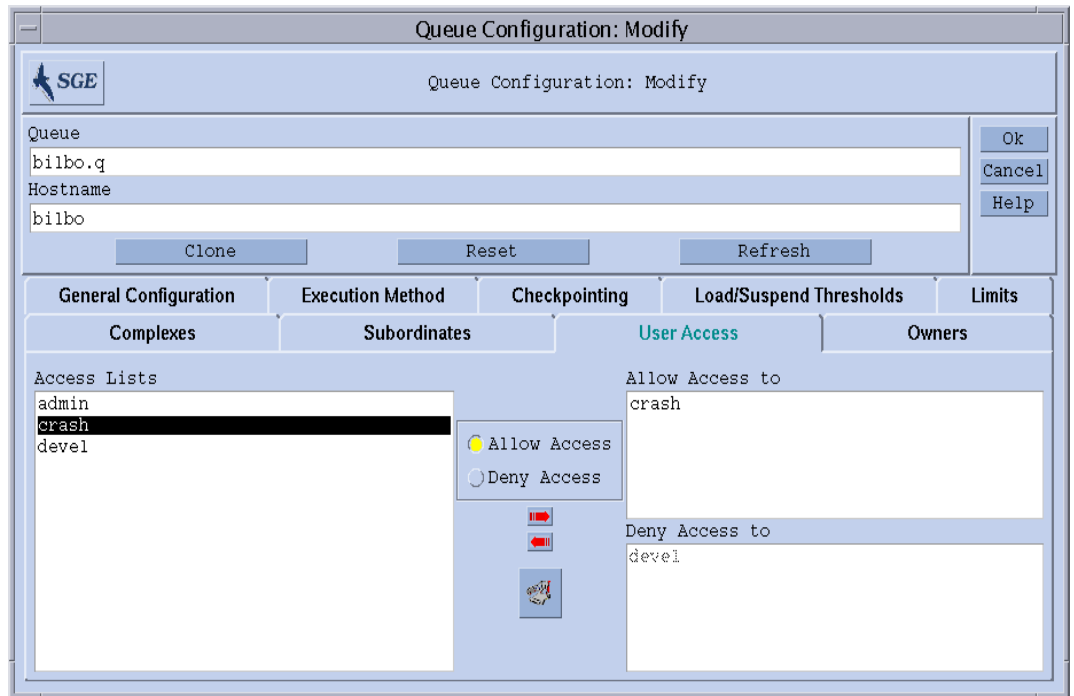


圖 7-10 佇列配置 — 使用者存取

提供的輸入欄位可用來設定以下參數：

- 隨附於佇列的「允許 (allow)」或「拒絕 (deny)」清單之使用者存取清單。

存取清單內允許清單之使用者或使用者群組可存取至佇列中，而位於拒絕清單中就無法存取至佇列。如果允許清單是空的，就不會限制存取，除非於拒絕清單中有明確指定。

請參閱 `queue_conf` 線上說明手冊以得知更多參數相關資訊。

按螢幕下方中間的按鈕開啓「存取清單配置」對話方塊（請參見第 59 頁的「使用者存取權限」）。

▼ 如何配置所有人

- 選擇「所有人」參數組。
以下顯示的圖 7-11 為範例圖示。

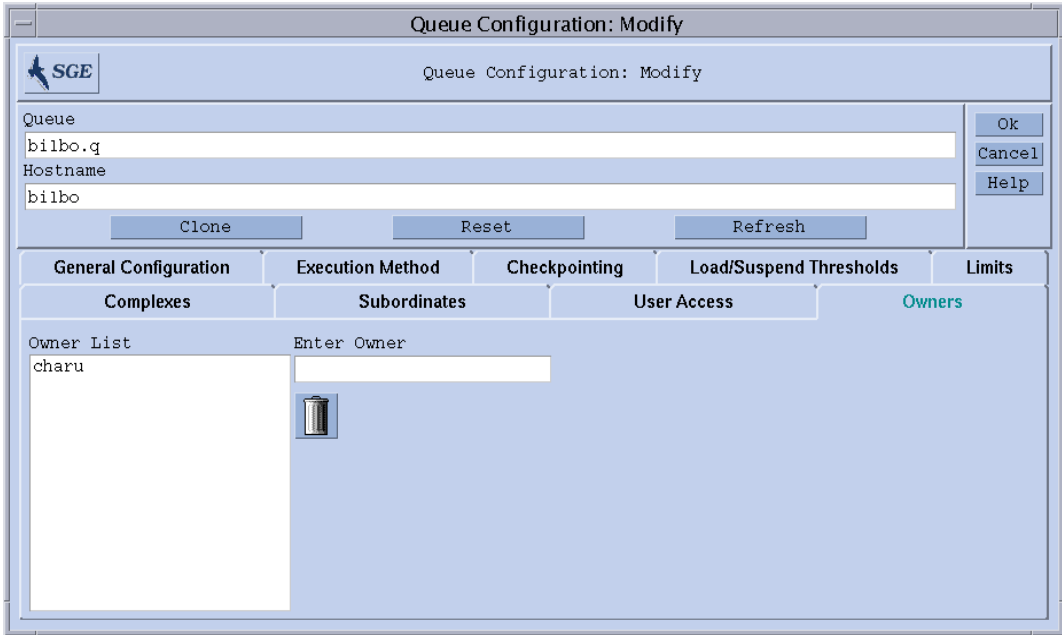


圖 7-11 佇列配置 — 所有人

提供的輸入欄位可用來設定以下參數：

- 佇列所有人清單

佇列之所有人擁有暫停 / 解除暫停，或停用 / 啟動佇列之權力。所有可用之使用者帳號皆為有效值，可新增至佇列所有人清單中。要從佇列所有人清單中刪除一使用者帳號，請於「所有人清單」視窗選取此帳號，然後按一下對話方塊右下方的垃圾筒圖示。

請參閱 `queue_conf` 線上說明手冊以得知更多參數相關資訊。

▼ 如何從指令行配置佇列

- 輸入以下指令與適合的選項（依您想如何配置佇列而定）。

```
# qconf 選項
```


qconf 指令有以下幾個選項。

- `qconf -aq [佇列名稱]`

新增佇列 — 在佇列配置範本叫出編輯器（預設 `vi` 或對應至 `$EDITOR` 環境變數）。如果選項參數 *佇列名稱* 出現，就使用佇列配置做為範本。變更範本並儲存至磁碟片中就完成配置佇列的工作。請參見《*Sun Grid Engine 5.3 和 Sun Grid Engine 企業版 5.3 參考手冊*》內的 `queue_conf` 記錄以得知改變範本記錄的詳細資訊。

- `qconf -Aq 檔案名稱`

新增佇列 — 使用這個檔案 *檔案名稱* 來定義佇列。定義檔也許是由 `qconf -sq 佇列名稱` 建立的（見下方）。

- `qconf -cq 佇列名稱 [...]`

清除佇列 — 清除特定佇列的狀態使其閒置以執行作業。重設狀態時並不會顧及到目前狀態。此選項有助於消除錯誤狀態，但不可用於一般操作模式。

- `qconf -dq 佇列名稱 [...]`

刪除佇列 — 從可用佇列清單中刪除引數中之特定佇列。

- `qconf -mq 佇列名稱`

修改佇列 — 修改指定佇列。在欲變更之佇列配置中叫出編輯器（預設 `vi` 或對應至 `$EDITOR` 環境變數）。變更配置並儲存至磁碟片中就完成修改佇列的工作。

- `qconf -Mq 檔案名稱`

修改佇列 — 使用這個檔案 *檔案名稱* 定義修改過的佇列配置。定義檔也許是由 `qconf -sq 佇列名稱`（見下方）與之後的修改所建立的。

- `qconf -sq [佇列名稱 [...]]`

顯示佇列 — 顯示預設平台佇列配置（如果沒有出現引數的話），或者顯示目前由逗點分開的引數清單中之佇列配置。

- `qconf -sql`

顯示佇列清單 — 顯示所有目前配置的佇列清單。

關於佇列行事曆

佇列行事曆會依據一年的某天、一星期的某天、或某天何時來定義 Sun Grid Engine 佇列的可用性。佇列可於時間中的仲裁點做配置以變更狀態。佇列狀態可以變更至停用、啟用、暫停、或恢復（解除暫停）。

Sun Grid Engine 提供定義行事曆特定站點組的功能，每個站點組皆包含仲裁狀態變更與事件發生的時間。這些行事曆與佇列是有關連的，即每個佇列須（或無須）隨附單一行事曆，才可接受隨附行事曆裡定義之檔案可用性。

行事曆格式的語法會於線上說明手冊 calendar_conf 中詳加說明。以下幾個範例會說明對應的管理設備。

▼ 如何以 QMON 配置佇列行事曆

1. 在 QMON 主要功能表上按一下「行事曆配置」。

「佇列行事曆配置」對話方塊與圖 7-12 很類似。

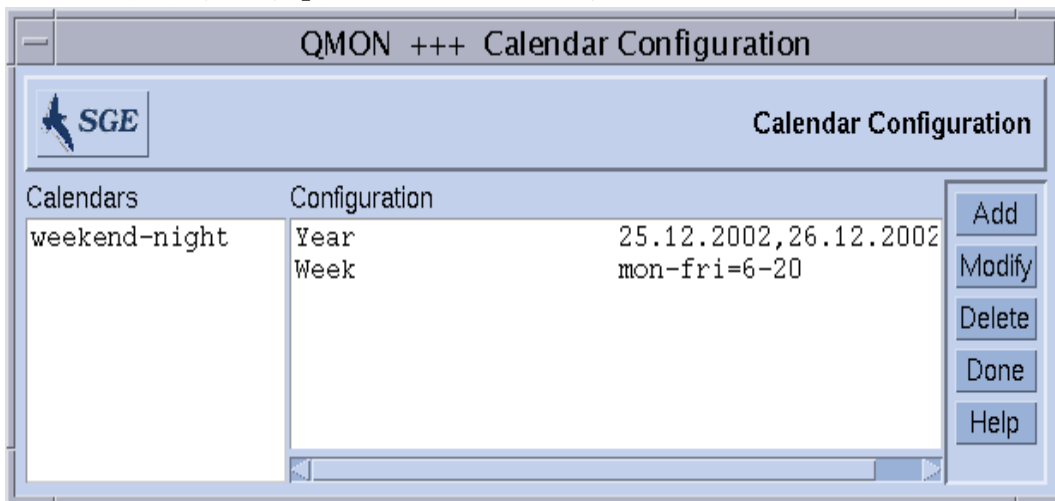


圖 7-12 行事曆配置

可用存取清單顯示於螢幕左方的「行事曆」選擇清單中。

2. 在「行事曆」選擇清單中，按一下您想修改或刪除的行事曆配置。
3. 依您想如何變更配置，執行以下其中一種步驟。
 - a. 按下螢幕右方的「刪除」按鈕來刪除選取的行事曆。
 - b. 按下「修改」按鈕修改選取的行事曆。
 - c. 按下「新增」按鈕新增存取清單。

在所有情況下皆會開啓「行事曆定義」對話方塊（如圖 7-13 所示），並提供刪除、修改、或新增的方法。

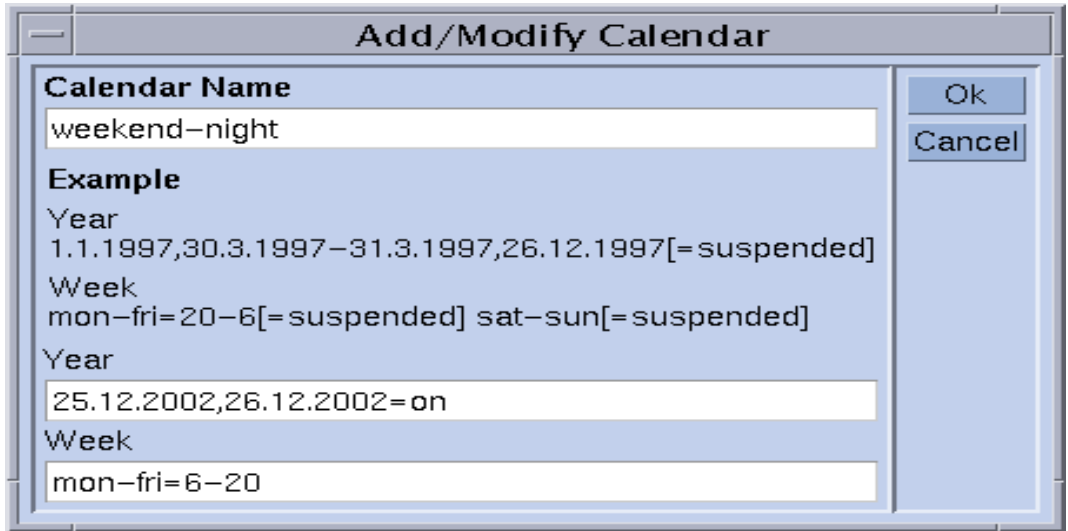


圖 7-13 新增、刪除、或修改行事曆

4. 依下節指導進行。

「行事曆名稱」輸入視窗會顯示修改作業中選取的行事曆名稱，或者您也可以自行輸入欲作業之行事曆名稱。「年」與「星期」輸入欄位讓您可利用 `calendar_conf` 線上說明手冊中提及的語法來定義行事曆事件。

對佇列而言，以上行事曆的配置範例也可適用於下班時間與週末，而且耶誕佳節被定義為比照週末辦理。

請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》裡的 `calendar_conf` 項目以得知更多範例與語法的詳細說明。

要於佇列上隨附行事曆配置，就得於佇列上建立行事曆定義之可用性檔案。隨附於一般參數佇列配置功能表的行事曆顯示於圖 7-14。Calendar 輸入欄位包含隨附之行事曆名稱，而輸入欄位旁的圖示鈕則會開啓選擇對話方塊與最近配置的行事曆清單。請參閱第 145 頁的「關於佇列配置」一節以得知更多配置佇列的詳細說明。

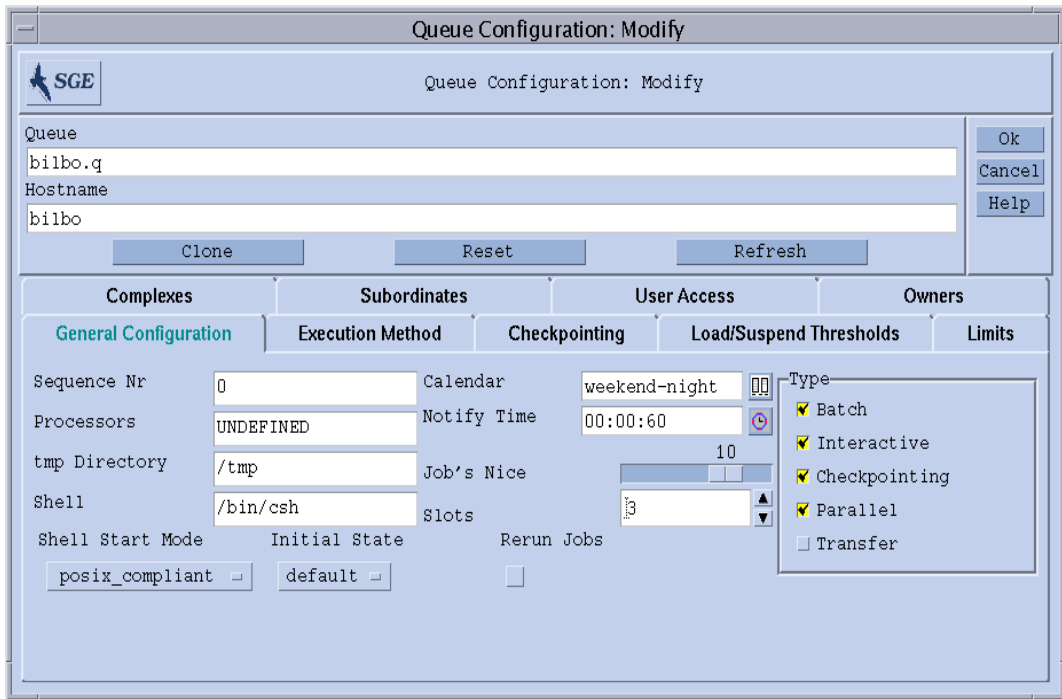


圖 7-14 於一般參數佇列配置功能表的行事曆配置

▼ 如何從指令行配置行事曆

- 輸入以下指令與合適的開關。

```
# qconf 開關
```

以下為四種可用之開關：

- `qconf -Acal, -acal`

新增行事曆 — 本指令會新增行事曆配置至 Sun Grid Engine 叢集。新增之行事曆會由檔案或者一開啓平台配置之編輯器讀出，讓您可輸入行事曆。

- `qconf -dcal`

刪除行事曆。

- `qconf -Mcal, -mcal`

修改行事曆 – 本指令可修改現有行事曆配置。欲修改之行事曆會由檔案 (`-Mcal`)，或者開啓之前配置的編輯器讀出，讓您可輸入新的定義 (`-mcal`)。

- `qconf -scal, -scall`

顯示行事曆 – 此指令會顯示現有行事曆配置 (`-scal`) 或列出所有配置的行事曆清單 (`-scall`)。

聚合體概念

本章會說明 Sun Grid Engine 5.3 聚合體 此重要概念。除了提供聚合體與其相關概念之背景資訊外，本章還包含如何完成下述作業之詳細指示。

- 第 164 頁的「如何新增或修改聚合體配置」
- 第 172 頁的「如何設定可耗損資源」
- 第 182 頁的「如何從指令行修改聚合體配置」
- 第 184 頁的「如何自行撰寫載入感應器」

關於聚合體

聚合體的定義可透過 `qsub` 或 `qalter -l` 選項，提供所有使用者可能要求的 Sun Grid Engine 作業資源屬性之相關資訊，與 Sun Grid Engine 系統內之相關參數解讀資訊。

聚合體還為 Sun Grid Engine 系統可耗損資源 工具建立架構，此功能允許叢集全域、特定主機、或可辨識資源與其相關容量之佇列相關屬性的定義。資源可用性與 Sun Grid Engine 作業之要求將於排程過程中列入考慮。Sun Grid Engine 同時也將執行必要之簿記與容量規劃，以避免過度消耗可耗損資源。典型具有可耗損屬性的範例有可用之空白記憶體、未被使用之軟體組版權、空白磁碟空間、或連接網路可用之頻寬。

簡易來說，Sun Grid Engine 聚合體可用來說明佇列、主機、或叢集屬性之原有意義。此說明包含了屬性名稱、可參閱之捷徑、屬性之設定值種類（例如 `STRING` 或 `TIME`）、用以指定聚合體屬性之預先定義值、由 Sun Grid Engine 排程器 `sge_schedd` 使用之關係操作員、確認一作業是否可由使用者提出可要求的旗號屬性、並辨別此屬性是否為可耗損屬性之可耗損旗號（若一作業無法明確提出屬性要求，就會考慮建立並預設可耗損屬性之要求值）。

顯示於圖 8-1 之「QMON 聚合體配置」對話方塊描述該如何定義聚合體屬性。

▼ 如何新增或修改聚合體配置

1. 在 QMON 主要功能表上按下「聚合體配置」鈕。

圖 8-1 為「聚合體配置」對話方塊範例。

2. 請遵照以下幾節的詳細資訊來新增或修改聚合體配置。

- 第 166 頁的「佇列聚合體」
- 第 166 頁的「主機聚合體」
- 第 168 頁的「全域聚合體」
- 第 168 頁的「使用者定義聚合體」

「聚合體配置」對話方塊提供變更現有聚合體定義與定義新使用者聚合體的方法。

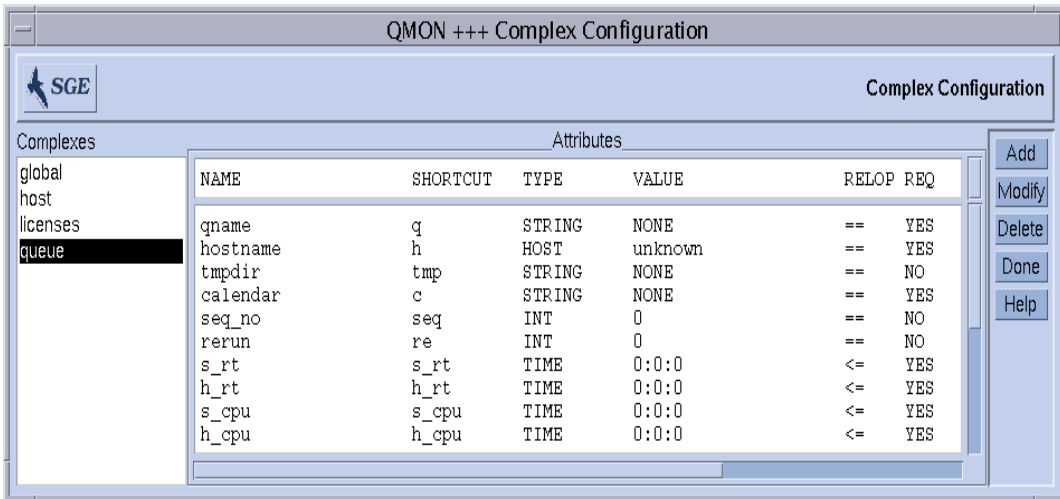


圖 8-1 聚合體配置對話方塊 — 佇列

在螢幕左方會顯示一選擇清單，內含系統可辨識之所有聚合體。可使用此清單來修改或刪除聚合體。您可由螢幕右方之相對按鈕選擇欲進行之操作（新增、修改、或刪除）。如果想建立新的聚合體，或修改現有聚合體，就會開啓如圖 8-2 一般的對話方塊。

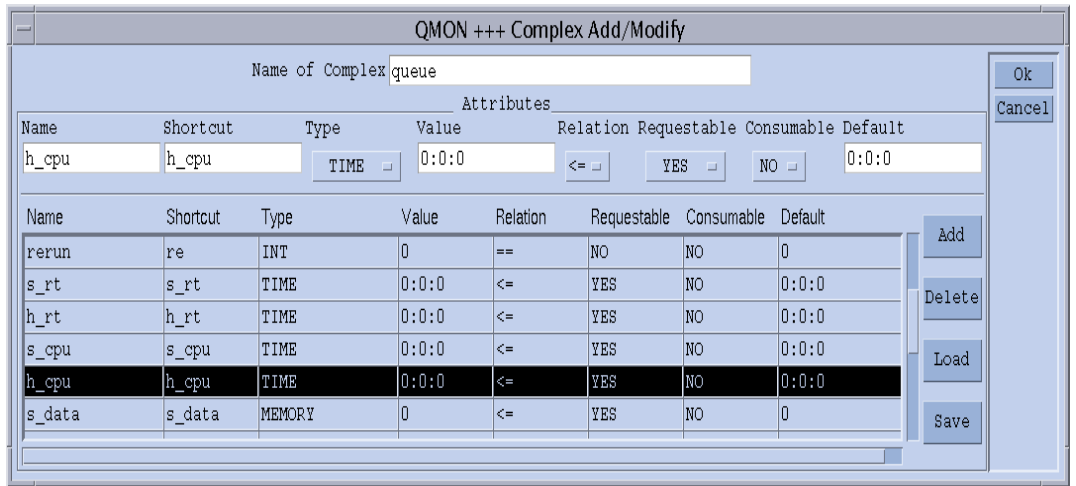


圖 8-2 聚合體新增 / 修改對話方塊

您必須自行輸入聚合體的名稱，或選擇顯示於上方「聚合體名稱」輸入視窗之名稱。只要按下滑鼠左鍵選取一行，您就可以於聚合體定義表上修改聚合體屬性。選取記錄將顯示於定義視窗上，而選取器將顯示於「屬性」方塊上方。變更定義後並按下「新增」鈕就可更新定義表上的改變。

只要填入定義視窗、使用選取器、然後按下「新增」鈕，就可新增一筆新記錄。新增新物件時，請勿選取屬性表上的任一行資料。

「載入」與「儲存」鈕可用來載入與儲存從一般檔案，或至一般檔案的聚合體配置。檔案選擇方塊將會開啓，以供您選取檔案。「刪除」鈕可用來刪除聚合體配置中選取的任一行資料。

請參閱聚合體線上說明手冊以得知圖表中欄與列意義的詳細資訊。最後，螢幕右上方的「確定」鈕將以 `sge_qmaster` 來記錄新增或變更的聚合體。

聚合體種類

Sun Grid Engine 聚合體物件分成以下四種不同的聚合體種類。

- 佇列聚合體
- 主機聚合體
- 全域聚合體
- 使用者定義聚合體

以下幾節會詳細說明各種類型。

佇列聚合體

佇列聚合體是由 `queue` 這個特殊名稱而來的。

於其預設格式中，可選擇定義於 `queue_conf` 裡的佇列配置參數。佇列聚合體的主要目的為定義如何解讀這些參數，並提供容器更多屬性，以供所有佇列使用。因此，使用者定義屬性可延伸佇列聚合體。

如果可由一特定佇列得知佇列聚合體，則此佇列的對應配置值就會於佇列聚合體中更換屬性值（會覆寫 `value` 欄）。

舉例來說，如果一佇列已設定此佇列聚合體為大，此預設值為 `unknown`（請見圖 8-1）的佇列屬性 `qname` 之數值欄就會設定成大。

不明確之設定值可使用佇列配置中的 `complex_values` 參數覆寫（請見第 145 頁的「關於佇列配置」）。這通常是用於可耗損資源（請見第 172 頁的「可耗損資源」一節）。舉例來說，關於虛擬記憶體大小限制，可使用佇列配置值 `h_vmem` 來限制一作業之總占用記憶體之數量，而 `complex_values` 清單中的對應記錄則會定義主機或指定給佇列之虛擬記憶體的總可用數量。

如果管理者想新增屬性至佇列聚合體，就會使用佇列的 `complex_values` 參數，或者由預設值使用位於佇列聚合體配置中的 `value` 欄來定義與特定佇列相關之數值。

主機聚合體

主機聚合體是由 `host` 這個特殊名稱而來的，內含於主機上受管理的所有屬性之特徵定義（請參閱圖 8-3）。與主機相關之屬性的標準組包含兩個種類，但也有可能會增加上節所述之佇列聚合體。第一種類型是由許多佇列配置屬性建立而成，特別適用於以主機為主之管理。這些屬性包含：

- 插槽
- 七
- `h_vmem`
- `s_fsize`
- `h_fsize`

（請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》內的 `queue_conf` 記錄以得知更多詳細資訊）。

注意 – 同時於主機聚合體與佇列配置中定義這些屬性並不會造成互相矛盾的現象，還可同時允許於主機等級與佇列等級中維持對應資源。舉例來說，總虛擬空白記憶體 (`h_vmem`) 可供主機管理使用，而總數量的一個子集則與此主機上的一個佇列有關。

標準主機聚合體第二種屬性類型為預設載入數值。所有 `sge_execd` 都會定期報告載入數值至 `sge_qmaster`。報告的載入數值為標準 Sun Grid Engine 載入數值，例如 CPU 載入平均值，或者是由 Sun Grid Engine 管理（請見第 183 頁的「載入參數」一節）定義之載入數值。標準載入數值之特徵定義為預設主機聚合體的一部份，而管理員定義之載入數值則需要延伸主機聚合體。

主機聚合體不僅可延伸以包含非標準載入參數，還可管理主機相關資源，例如指定至主機之軟體版權數量，或者主機本端檔案系統上的可用磁碟空間。

如果要結合主機聚合體與一主機或此主機之佇列，就必須由以下其中一種方法確認特定主機聚合體屬性的具體值。

- 佇列配置衍生屬性之佇列配置
- 回報之載入數值
- 對應主機配置之 `complex_values` 項目中，設定值的明確定義（請參見第 126 頁的「關於配置主機」一節）。

如果以上方法皆不可行（例如此設定值應為載入參數，但 `sge_execd` 並沒有回報其載入數值），就得利用主機聚合體配置的 `value` 欄位。

舉例來說，總空白虛擬記憶體屬性 `h_vmem` 會定義於佇列配置中以為限制，並同時以標準載入參數回報。主機與主機上隨附之佇列的虛擬記憶體總可用容量會定義於此主機與此佇列配置上的 `complex_values` 清單中。與定義 `h_vmem` 為可耗損資源一同（請見第 172 頁的「可耗損資源」），此法可有效利用機器的記憶體，無須擔心因交換而導致耗損記憶體，進而減低系統效能。

注意 – 只有 `Shortcut`、`Value`、`Relation`、`Requestable`、`Consumable` 與 `Default` 欄會因系統預設載入屬性而變更。請勿刪除預設屬性。

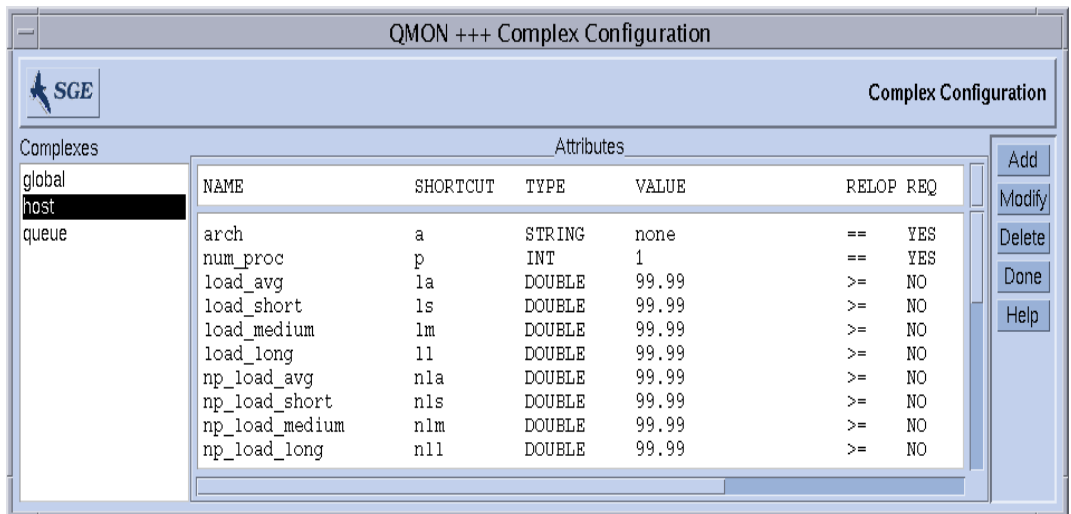


圖 8-3 聚合體配置對話方塊 — 主機

全域聚合體

全域聚合體是由 global 這個特殊名稱而來的。

於全域聚合體配置之記錄可用來參照叢集各式資源屬性，例如一檔案伺服器之可用網路頻寬、或者網路各式可用檔案系統之未用磁碟空間（請見圖 8-4）。如果對應載入報告包含 GLOBAL 辨識器，全域資源屬性也可與載入報告做結合（請見第 183 頁的「載入參數」一節）。全域載入數值可由叢集上任何主機報告得知。Sun Grid Engine 與預設值不會報告全域載入數值，所以就沒有預設全域聚合體配置。

全域聚合體屬性的明確數值會由全域載入報告，或者由 global 主機配置（請見第 126 頁的「關於配置主機」一節）之 complex_values 參數裡的明確定義，抑或結合一特定主機或佇列與一明確定義之對應 complex_values 清單，加以確認。如果以上方法無效（例如尚未報告載入數值），就得利用全域聚合體配置之 value 欄位。

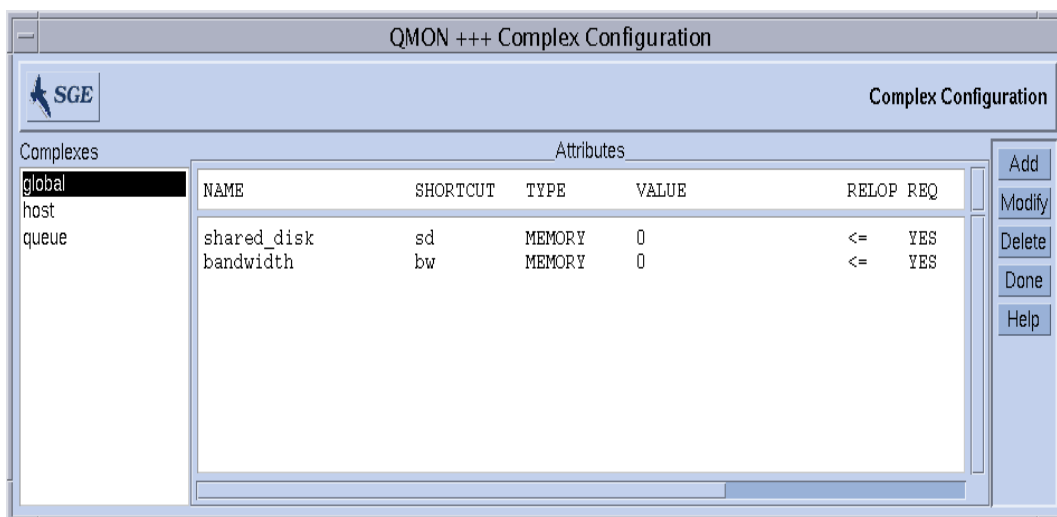


圖 8-4 聚合體配置對話方塊 — 全域

使用者定義聚合體

藉由設定使用者定義聚合體，就可延伸 Sun Grid Engine 管理至 Sun Grid Engine 管理之屬性組，進而限制屬性於特定佇列與 / 或主機上的影響力。使用者聚合體只是屬性名稱與其對應定義之總稱，顯示 Sun Grid Engine 如何管理這些屬性。透過 complex_list 佇列與主機配置參數（請見第 145 頁的「關於佇列配置」與第 126 頁的「關於配置主機」二節）即可隨附一個或多個使用者定義聚合體至一個佇列與 / 或主機中。所有指定聚合體內定義之屬性可分別用於佇列、主機、與預設聚合體屬性中。

與佇列和主機結合的使用者定義聚合體之明確值必須經由佇列與主機配置中的 complex_values 參數，或者利用使用者聚合體配置之 value 欄位，才可建立此數值。

舉例來說，定義以下使用者定義聚合體 licenses。

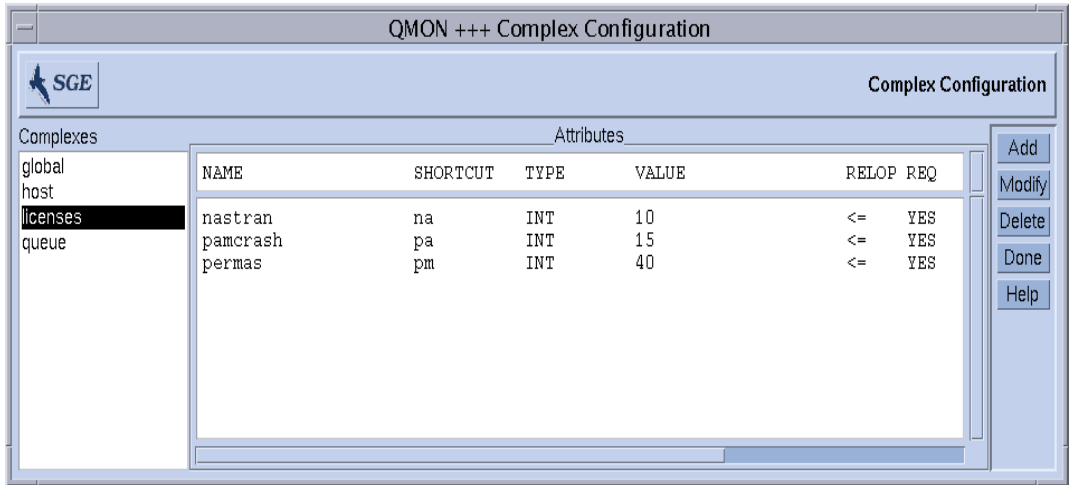


圖 8-5 聚合體配置對話方塊 — 版權

然後讓 licenses 聚合體新增至至少一個或多個佇列的相關使用者定義聚合體之清單，如同顯示於圖 8-6 的佇列配置 User Complexes 子對話方塊（請見第 145 頁的「關於佇列配置」與其相關節以得知如何配置佇列的詳細資訊）。

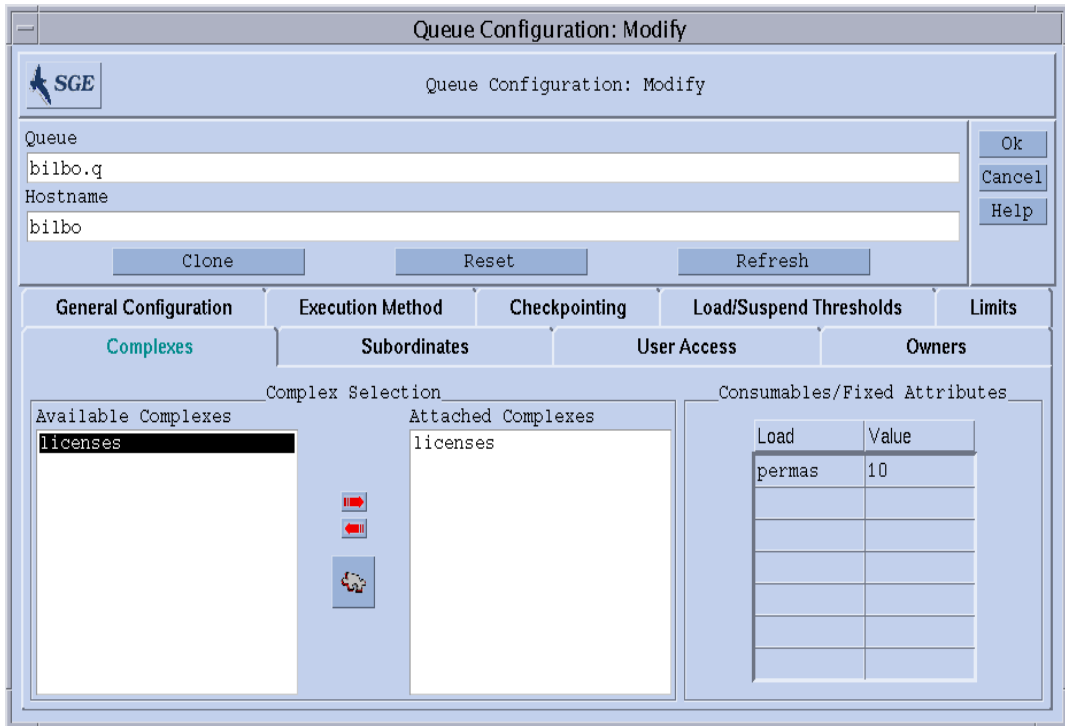


圖 8-6 使用者定義聚合體佇列配置

接著就會配置顯示的佇列，以管理最多 10 組的軟體組 `permas` 版權。再者，`licenses` 聚合體屬性 `permas` 之於 Sun Grid Engine 作業中就變成可要求，就如同圖 8-7 中，「提交」對話方塊中的「要求資源」子對話方塊之「可用資源」清單所示（請參閱第四章，第 61 頁的「提交作業」一節得知如何提交作業的詳細資訊）。

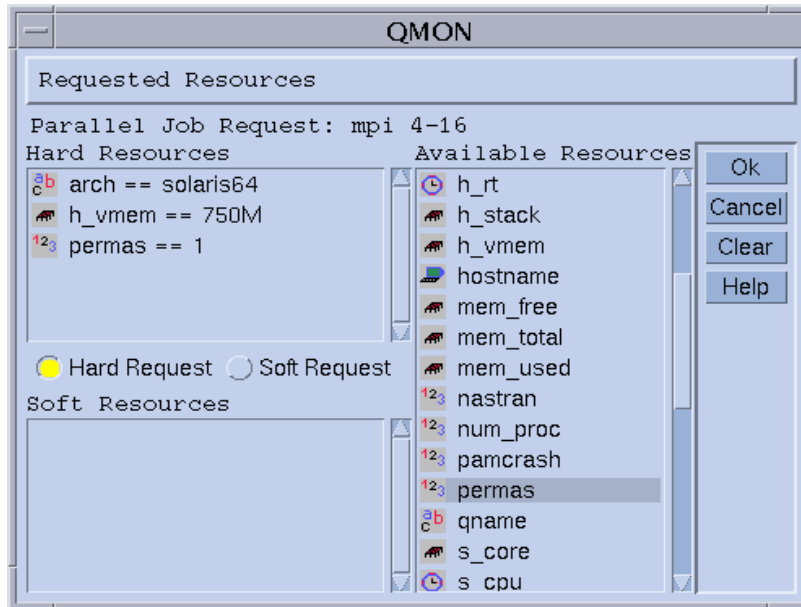


圖 8-7 「提交」子對話方塊之「要求資源」

不然，使用者也可從命令行提交作業，並要求 licenses 屬性，如下所示。

```
% qsub -l pe=1 permas.sh
```

注意 – 您可使用 pm 捷徑代替完整屬性名稱 permas。

此配置與相似作業要求的結果，唯一適合這些作業的佇列即為與使用者定義 licenses 聚合體結合之佇列，有配置可用之 permas 版權。

無效的使用者定義聚合體名稱

以下為保留的聚合體名稱清單，所以不可用於指示使用者定義聚合體名稱。

- global
- host
- queue

可耗損資源

可耗損資源，又名為 *可耗損資源*，為一可有效管理限制資源（如可用記憶體、檔案系統上的未用空間、網路頻寬、或使用中的軟體版權）的方法。可耗損資源的總可用容量會由 Sun Grid Engine 管理員定義，而對應資源之消耗也會由 Sun Grid Engine 內部簿記監視。Sun Grid Engine 會替所有執行中的作業計算此資源之消耗量，並確保僅在 Sun Grid Engine 內部簿記顯示有足夠的可耗損資源時才會發送作業。

可耗損資源可與預設或使用者定義載入參數結合（請見第 183 頁的「載入參數」）；即可為可耗損屬性報告載入數值，為載入屬性而建立「可耗損」旗號。Sun Grid Engine 可耗損資源管理會同時將負載（測量資源可用性）與內部簿記列入考慮，並確認兩者並未超出預定限制。

要啟動可耗損資源管理，您必須定義一個資源的總容量。您可由叢集全域、主機與佇列為主完成定義的工作，因為這些種類可以依序互相替換（即主機可以限制叢集資源，佇列可以限制主機與叢集資源）。資源容量的定義會與佇列和主機配置中的 `complex_values` 項目一同執行（請見《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》裡的 `host_conf` 與 `queue_conf` 項目，和第 145 頁的「關於佇列配置」與第 126 頁的「關於配置主機」）。`global` 主機的 `complex_values` 定義有明確說明叢集全域可耗損資源之設定。位於 `complex_values` 清單中的每個可耗損聚合體屬性都會得到一設定值，以表示資源的最大可用量。內部簿記將經由所有執行中的作業，即經過作業資源要求之作業，減去預計總消耗資源量。

▼ 如何設定可耗損資源

僅有數字之聚合體屬性（即屬於 INT、MEMORY 與 TIME 種類）才可配置為可耗損資源。

1. 在 QMON 主要功能表上按下「聚合體配置」鈕。

圖 8-1 為「聚合體配置」對話方塊範例。

2. 要啟動屬性的 Sun Grid Engine 可耗損資源管理，請於聚合體配置中建立 CONSUMABLE 旗號，如圖 8-8 描述之 `virtual_free` 記憶體資源一般。
3. 要設定其他可耗損資源，請參照以下幾節的詳細範例指示。
 - 第 174 頁的「範例 1：浮動軟體版權管理」
 - 第 178 頁的「範例 2：虛擬記憶體之空間分享」
 - 第 180 頁的「範例 3：管理可用磁碟空間」

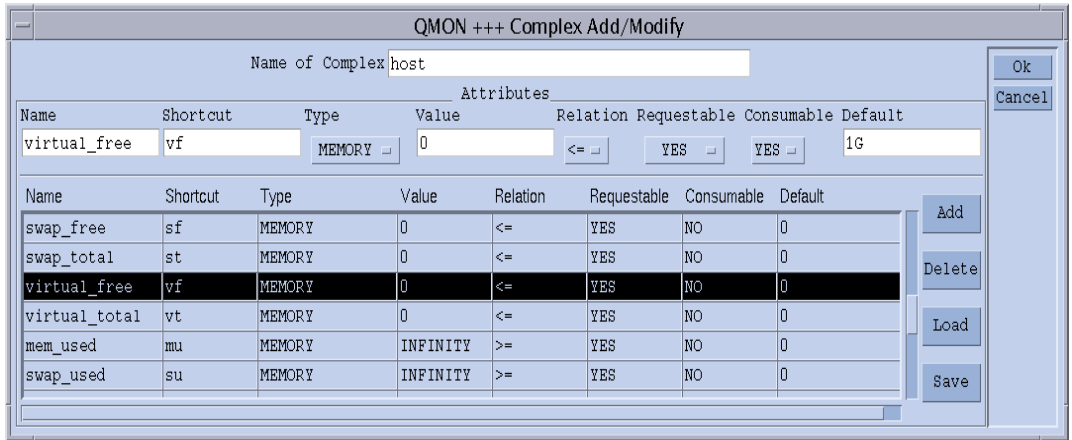


圖 8-8 聚合體配置對話方塊——virtual_free

接著，若要 Sun Grid Engine 於每一佇列與主機執行要求的容量規劃，就必須於 complex_values 清單中定義容量。圖 8-9 顯示如何定義 1 GB 之虛擬記憶體為目前主機之容量值。

(任何佇列裡) 主機上一同執行所有作業之虛擬記憶體需求將會累積，並自 1 GB 容量中減去，以確認可用之虛擬記憶體。如果作業對 virtual_free 的要求超出可用容量，此作業就不會被分派到主機上的佇列。

注意 – 藉由輸入「可要求」參數之強制值就可以強制作業要求資源，如此才可得知其預計消耗量（請見圖 8-8）。

注意 – 預設資源消耗量可藉由作業未明確要求之可耗損屬性的管理員事先完成定義（請見圖 8-8—200 MB 為預設消耗量）。這僅在尚未執行要求屬性時才有意義，如上說明。

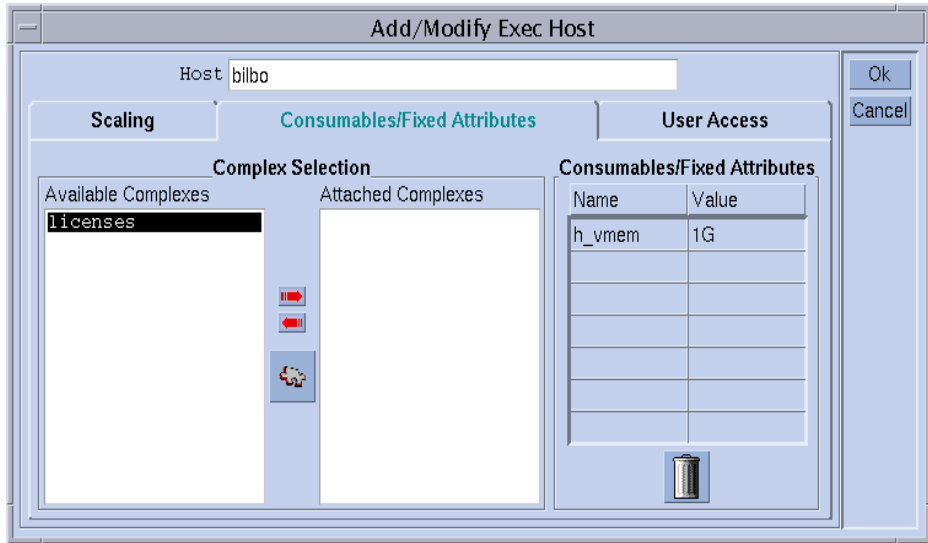


圖 8-9 執行主機配置 —virtual_free

設定可耗損資源範例

利用以下範例引導您如何為站點設定可耗損資源。

範例 1：浮動軟體版權管理

假設您於叢集中使用套裝軟體 `pam-crash`，而且您已存取至 10 個；意即只要此軟體總呼叫量未到 10，您就會於每個系統中使用 `pam-crash`，目的是為了在配置 Sun Grid Engine 期間，全部 10 個版權都已被其他執行中的 `pam-crash` 作業佔據時，避免繼續排程 `pam-crash` 作業。

只要有 Sun Grid Engine 可耗損資源，就可輕易達到此目的。首先，您必須新增可用 `pam-crash` 版權的數量以做為全域聚合體配置之可耗損資源，如圖 8-10 所示。

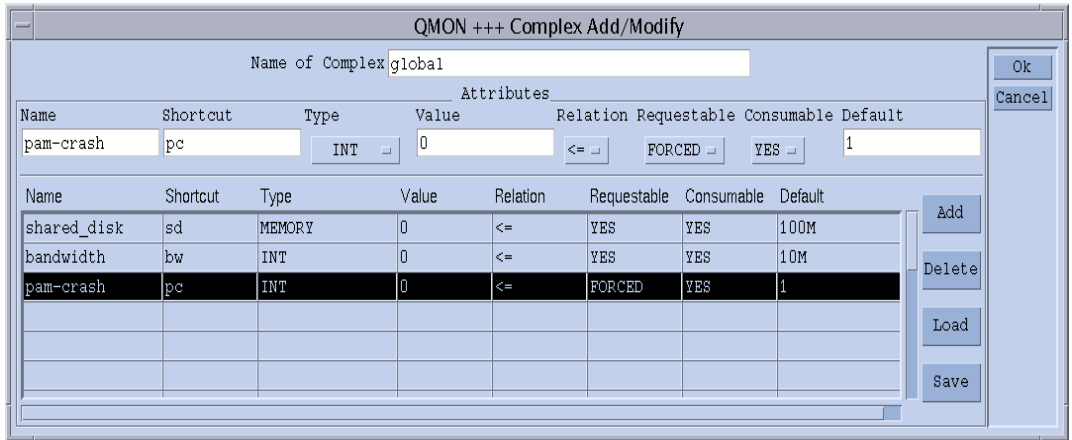


圖 8-10 聚合體配置對話方塊 —pam-crash

可耗損屬性的名稱會設為 `pam-crash`，而 `pc` 可於 `qalter`、`qselect`、`qsh`、`qstat`、或 `qsub -l` 選項中被用來當作捷徑。屬性種類則定義為整數計數器 (integer counter)。數值欄位的設定與可耗損資源無關，因可耗損資源是透過 `complex_values` 清單（如下）由全域、主機、或佇列配置接收其設定值。「可要求的」旗號會設定為 `FORCED`，代表使用者於提交作業時，必須要求會占用多少的 `pam-crash` 版權。「可耗損」旗號最後會定義屬性為可耗損資源，與 `Default` 設定無關，因為 `Requestable` 是設定為 `FORCED`，所以將會從任一作業上接收此屬性之要求值。

要執行此屬性與叢集之資源規劃，就必須於全域主機配置中定義可用 `pam-crash` 版權之數量，如圖 8-11 所示。此屬性 `pam-crash` 值會設定為 10，對應 10 個浮動版權。

注意 – `Consumable/Fixed Attributes` 表會對應於主機配置檔案格式 `host_conf` 的 `complex_values` 項目。

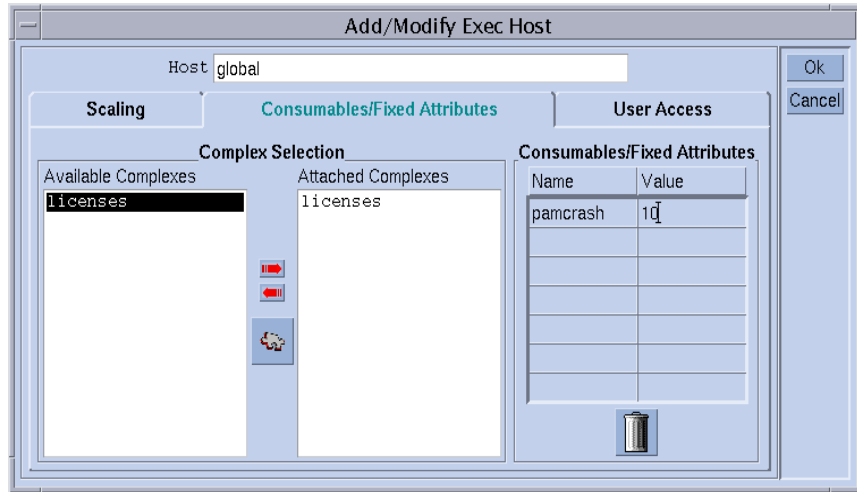


圖 8-11 全域主機配置 —pam-crash

假設使用者提交以下作業。

```
% qsub -l pc=1 pam-crash.sh
```

此作業僅可於 pam-crash 版權佔用量低於 10 時執行。但是此作業可於叢集任一地方執行，而且還會於執行時間內為其佔據一 pam-crash 版權。

如果您叢集中的主機無法包含於浮動授權中 — 例如，因您沒有浮動授權所需的 pam-crash 二進位 — 您就可將與可耗損屬性 pam-crash 相關之主機的容量設為 0，自 pam-crash 授權管理中排除此浮動授權。這必須在「執行主機配置 (Execution Host Configuration)」對話方塊中完成，如圖 8-12 所示的主機 bilbo。

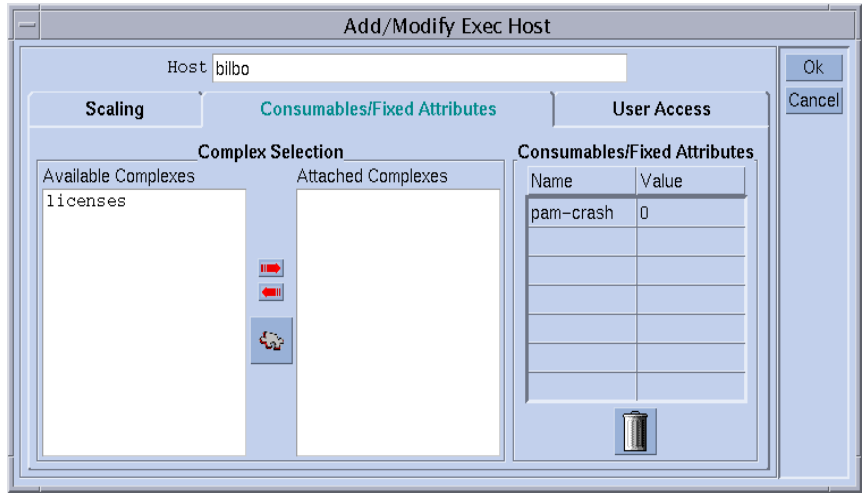


圖 8-12 執行主機配置 — pam-crash

注意 – pam-crash 屬性可用與否對執行主機而言很不明確，因 global 聚合體屬性會繼承至所有執行主機中。同樣地，只要將容量設定 0，您也可同時限制由特定主機管理之授權數量為叢集所有授權的一部分，並設定為非 0 數值，例如 2。在此情況中，最多 2 個 pam-crash 作業會同時存在於此主機中。

同樣地，您也可以藉由執行 pam-crash 作業以避免特定佇列；例如，因為此佇列很特別，有記憶體與 CPU 時間限制，所以不適用於 pam-crash。在此情況中，您只要於佇列配置中設定相對容量至 0 即可，如圖 8-13 所示。

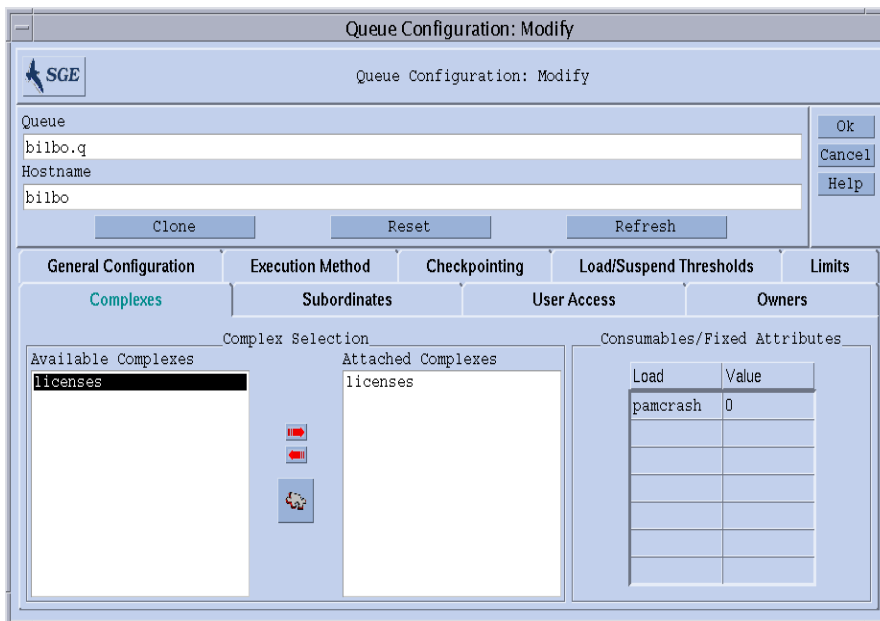


圖 8-13 佇列配置 — pam-crash

注意 – 佇列無法肯定 pam-crash 屬性是否可用，因 global 聚合體屬性會繼承至所有佇列中。

範例 2：虛擬記憶體之空間分享

系統管理員的一般任務即為調整系統，避免因過度耗損記憶體造成之機器交換而減低效能。Sun Grid Engine 軟體亦可透過「可耗損資源」工具支援此功能。

標準載入程式 virtual_free 可回報可用之空白虛擬記憶體，即可用交換空間與可用實體記憶體之結合。為避免交換，可以最小化交換空間的使用。在理想情況之下，執行主機中所有作業所需之全部記憶體應與實體記憶體相符合

Sun Grid Engine 軟體可保證所有作業皆可按如上啟動操作，完成以下預設與配置。

- virtual_free 會被配置成可耗損資源，而其位於每個主機上的容量都會設定為可用實體記憶體（或更低）。
- 作業會要求其參與之記憶體使用量，而於執行時間內不可超出要求之設定值。

圖 8-8 顯示一主機聚合體配置範例，而一對應之執行主機配置（主記憶體 1 GB 之主機）則顯示於圖 8-9 中。

注意 – Requestable 旗號會於主機配置範例中設定為 YES，與之前全域聚合體配置範例設定為 FORCED 不同。這代表使用者無需為其作業指定所需記憶體，但若無明確要求記憶體，就必須使用 Default 欄位裡的值。在此情況下預設值為 1 MB，代表無提出要求之作業會被預設將佔據所有可用之實體記憶體。

注意 – virtual_free 為 Sun Grid Engine 之標準載入參數之一。最新記憶體數據之額外可用性會自動為 Sun Grid Engine 列入虛擬記憶體容量規劃之考量。如果空白虛擬記憶體之載入報告低於 Sun Grid Engine 內部簿記的值，此載入數即便會用於避免記憶體過度耗損。如果作業沒有在 Sun Grid Engine 之下啟動，就很容易造成回報之載入數值與 Sun Grid Engine 內部簿記之間的差別。

如果您於單一機器上混合執行不同記憶體需求之作業等級，您也許會想分割此機器之記憶體，以供不同作業等級使用。此功能，一般稱為空間分享，可藉由為每一作業等級配置一佇列，並於主機上指定一部分記憶體為其所用而完成。

在此範例中，圖 8-14 顯示之佇列配置會隨附一半可用記憶體至主機 bilbo—500 MB，至佇列 bilbo.q。因此，執行於 bilbo.q 佇列的所有作業之預設累積記憶體就不會超出 500 MB。在其他佇列之作業就不列入考慮之中，但主機 bilbo 上所有執行作業之總預設記憶體仍不可超過 1 GB。

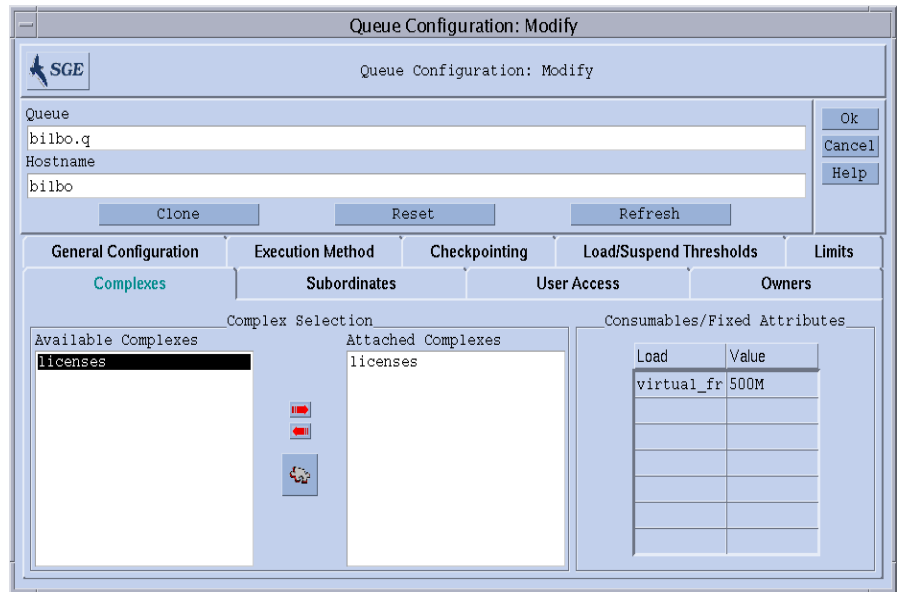


圖 8-14 佇列配置 — virtual_free

注意 – 藉由繼承自主機聚合體，此屬性 `virtual_free` 可適用於所有佇列。

使用者也許會提交作業至與範例相同之配置系統，用以下兩種形式其中一種：

```
% qsub -l vf=100M honest.sh
% qsub dont_care.sh
```

由第一個指令提交之作業會於可使用 100 MB 記憶體時開始執行，而且此容量也會列入 `virtual_free` 可耗損資源之容量規劃中。第二項作業僅會在系統無任何作業時執行，未明確要求所有可用之記憶體。除此之外，第二項作業將無法執行於佇列 `bilbo.q` 中，因其超出佇列之記憶體容量。

範例 3：管理可用磁碟空間

有些應用程式需要操作大量儲存於檔案中之資料組，因此就需依賴於其執行時間期間內，有效磁碟空間之可用性。此需求與可用記憶體之空間分享類似，請參見之前討論的範例。主要的差別即為 Sun Grid Engine 並沒有提供空白磁碟空間為其標準載入參數之一。這是因為磁碟通常會以站點特定方式被分割為檔案系統，以致無法自動辨識利益之檔案系統。

但是可用磁碟空間可透過 Sun Grid Engine 內的可耗損資源工具做有效管理。建議使用主機聚合體屬性 `h_fsize` 來達到此目的，稍後便會說明原因為何。首先，必須先將屬性配置成可耗損資源，如圖 8-15 範例所示。

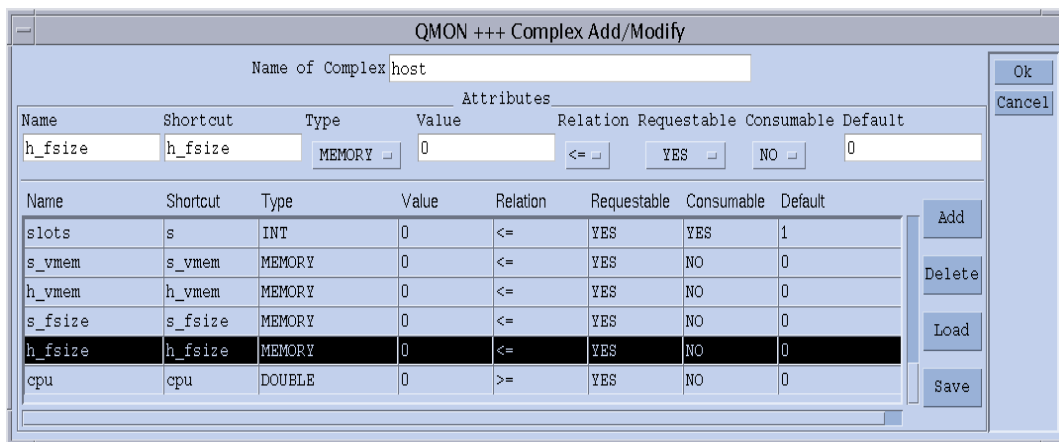


圖 8-15 Complex 配置 — `h_fsize`

如果要假設主機本端檔案系統，先將磁碟空間容量定義為可耗損，再置於主機配置上是很合理的，如圖 8-16 所示。

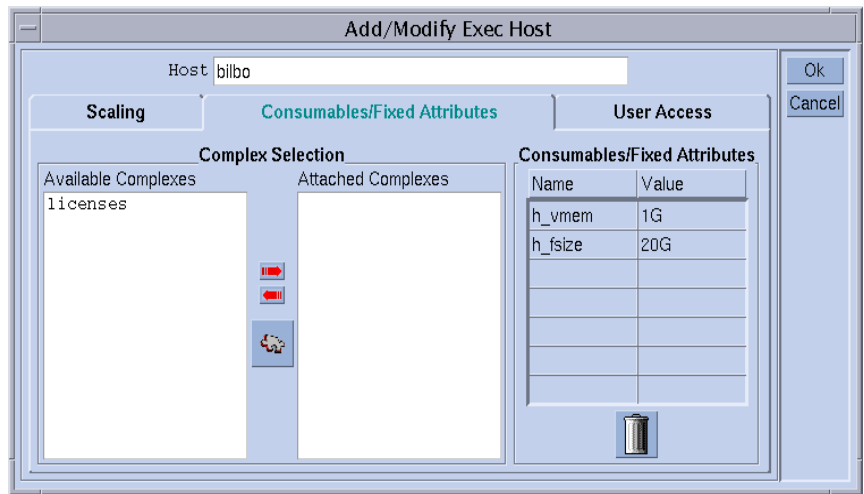


圖 8-16 執行主機配置 — h_fsize

提交作業至上述配置 Sun Grid Engine 系統之運作方式與前述範例很類似：

```
% qsub -l hf=5G big_sort.sh
```

為何本範例建議 h_fsize 屬性的原因為 h_fsize 可同時用於佇列配置中之硬檔案大小限制。檔案大小限制是用以避免作業於提交期間（前述範例為 20 GB）建立比指定檔更大的檔案，或者如果此作業沒有要求屬性，則用以限制由佇列配置傳送之對應值。h_fsize 之 Requestable 旗號於本範例中已被設立為 FORCED，這樣才會一直顯示要求。

只要利用佇列限制為可耗損資源，就可自動獲得由使用者指定之要求的控制權，與由作業 script 指定之實際資源消耗成對比。任何違反限制事宜都將受到處罰，而此作業也將中斷（請見 queue_conf 與 setrlimit 線上說明手冊得知詳細資訊）。如此便可確保以 Sun Grid Engine 內部容量規劃為主之資源要求是確實的。

注意 – 有些作業系統僅提供每個過程檔案大小限制。在此情況之下，作業就須建立多重大小直達限制之檔案。但是在支援每個作業檔案大小限制之系統中，Sun Grid Engine 會使用此功能與 h_fsize 屬性（請見 queue_conf 線上說明手冊以得知更多詳細資訊）。

如果您認為未提交至 Sun Grid Engine 的應用程式會同時佔據磁碟空間，Sun Grid Engine 內部簿記就會因缺乏磁碟空間而無法有效預防應用程式發生故障。要預防這個問題，請定期接收磁碟空間使用數據。數據會顯示總磁碟空間消耗量，包含發生在 Sun Grid Engine 外之消耗量。

Sun Grid Engine 載入感應器介面（請見第 184 頁的「新增站點特定載入參數」）可加強設定標準 Sun Grid Engine 載入參數與站點特定資訊，例如一特定檔案系統之可用磁碟空間。

藉由新增適合的載入感應器與回報 `h_fsize` 之空白磁碟空間，您就可結合可耗損資源管理與資源可用性數據。Sun Grid Engine 將會依磁碟空間與源於 Sun Grid Engine 內部資源規劃和最新回報載入數值的可用容量來比較作業需求。僅有符合以上兩項標準之作業才可分派至主機。

配置聚合體

Sun Grid Engine 聚合體可以透過圖表，自「QMON 聚合體配置」對話方塊中詳加定義並維護，如本節第 164 頁的「如何新增或修改聚合體配置」與以下所述，又或者可以由指令行執行。

▼ 如何從指令行修改聚合體配置

輸入以下指令與合適的選項。

```
% qconf 選項
```

請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》之聚合體項目，或聚合體線上說明手冊以得知 `qconf` 指令格式與有效值欄位語法之詳細資訊。

以下為有用之選項：

- `-ac`
- `-mc`
- `-Ac`
- `-Mc`

`qconf -Ac` 與 `-Mc` 選項將聚合體配置視為一引數，而 `-ac` 與 `-mc` 選項則使用編輯器於平台聚合體配置編輯，或修改一現有聚合體配置。

選項說明如下：

- `qconf -Ac, -ac`

新增一新的聚合體至可用聚合體清單中。

- `qconf -Mc, -mc`

現改一現有聚合體。

qconf 指令範例

以下指令：

```
% qconf -sc licenses
```

會列出 `nastran` 聚合體（如圖 8-5 裡定義）至檔案格式裡的標準輸出串流，如 `complex(5)` 線上說明手冊定義一般。表 8-1 顯示 `licenses` 聚合體之輸出範例。

#name	shortcut	type	value	relop	requestable	consumable	default
nastran	na	INT	10	<=	YES	NO	0
pam-crash	pc	INT	15	<=	YES	YES	1
permas	pm	INT	40	<=	FORCED	YES	1

#---- # start a comment but comments are not saved across edits -----

表 8-1 `qconf -sc` 輸出範例

載入參數

本節說明 Sun Grid Engine 5.3 載入參數之概念，並包含如何自行撰寫載入感應器之指示。

預設載入參數

在預設模式中，`sge_execd` 會定期回報一些載入參數與對應值至 `sge_qmaster` 中。這些數值會儲存在 `sge_qmaster` 內部主機物件中（請見第 125 頁的「關於常駐程式和主機」一節）。但是如果已定義聚合體屬性與對應名稱，它們就只能僅供內部使用。這些聚合體屬性包含如何解讀載入數值之定義（請見第 165 頁的「聚合體種類」一節得知詳細資訊）。

在主要安裝後就會回報一組標準載入參數。所有標準載入參數所需之屬性都會定義於主機聚合體中。Sun Grid Engine 之後的版次會提供預設載入參數之延伸設定。因此，預設模式回報之載入參數組會置於 `<sge_root>/doc/load_parameters.asc` 檔中。

注意 – 定義載入屬性的聚合體可決定這些屬性之存取可能性。定義全域聚合體內的載入參數，使其可用於全部叢集與所有主機之中。於主機聚合體定義這些參數，來為所有主機，但不會全域地也於叢集之中，提供屬性。在使用者定義聚合體內定義這些參數後，便可藉由隨附或分離使用者聚合體至主機中以控制載入參數之可見度。

注意 – 載入屬性不應定義於佇列聚合體中，因其將不為任何主機或叢集所用。

新增站點特定載入參數

預設載入參數組也許無法完整描述於叢集內的載入狀況，特別是對於站點特定策略、應用程式、與配置。因此，Sun Grid Engine 軟體提供以仲裁方式延伸載入參數組的方法。為達到此目的，`sge_execd` 提供一介面來回饋載入參數與最近載入數值至 `sge_execd` 中。之後，這些參數就會完全被視為預設載入參數。與預設載入參數一樣（請見第 183 頁的「預設載入參數」一節），對應屬性應定義於載入聚合體中，以使載入參數產生效能。

▼ 如何自行撰寫載入感應器

要以額外的載入資訊輸入 `sge_execd`，您必須提供載入感應器。此載入感應器可以為 `script` 或二進位可執行檔案。無論為哪一種情況，其標準輸入與輸出串流，和其控制流之處理皆須符合以下規定：

載入感應器必須撰寫為無限迴路，從 `STDIN` 於特定点等待輸入。如果字串，`quit`，由 `quit` 讀取，則載入感應器就該退出。只要由 `STDIN` 讀到行列末端 (`end-of-line`)，就可以再度啟動載入資料循環。接著，載入感應器須執行所需之操作以計算理想載入數值。在循環結束前，載入感應器會將結果寫入 `stdout` 中。

規定

格式如下：

- 載入數值回報起始於一行僅有 `begin` 字樣的文字行。
- 個別載入數值會被新行分開。
- 每個載入數值資訊都由冒號 (`:`) 分成三部份，中間沒有空格。
- 載入數值資訊第一部分為載入回報之主機名稱，或特定的名稱，`global`。
- 第二部份為載入數值的象徵名稱，如主機或全域聚合體清單中定義一般（請見《*Sun Grid Engine 5.3* 和 *Sun Grid Engine，企業版 5.3 參考手冊*》裡的 `complex(5)` 項目得知詳細資訊）。如果回報一載入數值於主機或全域聚合體清單中並無任何記錄，就不會使用此回報的載入數值。

- 第三個部分為測量的載入數值。
- 載入數值回報會以 end 字樣做結束。

Script 範例

程式範例 2 為 Bourne shell script 載入感應器範例。

```
#!/bin/sh
myhost='uname -n'
while [ 1 ]; do
    # wait for input
    read input
    result=$?
    if [ $result != 0 ]; then
        exit 1
    fi
    if [ $input = quit ]; then
        exit 0
    fi
    #send users logged in
    logins='who | cut -f1 -d" " | sort | uniq | wc -l' | sed "s/^ *//"
    echo begin
    echo "$myhost:logins:$logins"
    echo end
done
# we never get here
exit 0
```

程式範例 2 Bourne Shell Script 載入感應器

如果此範例已存入 load.sh 檔，而且透過 chmod 指定執行權限，您就可以藉由呼叫 load.sh 檔案，並重覆按下鍵盤上的 Return 鍵，從命令行互動測試此範例。

只要能執行此步驟，透過配置載入感應器路徑，於叢集、全域、或執行 hos- 特定配置中輸入 load_sensor 參數（請見第 139 頁的「基礎叢集配置」一節，或 sge_conf 線上說明手冊），您就可以安裝載入感應器至任何執行主機中。

對應 QMON 螢幕會如圖 8-17 範例所示。

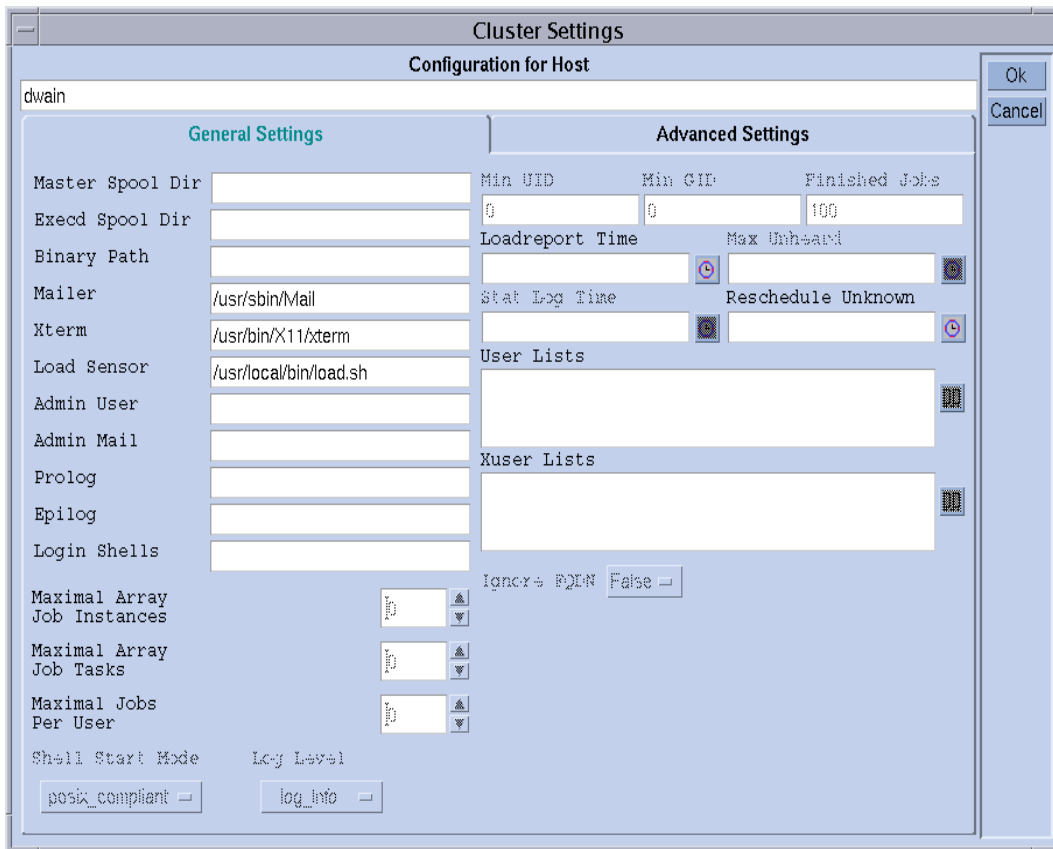


圖 8-17 本端配置與載入感應器

回報之載入參數與登錄都將於新增對應屬性至主機聚合體時立即使用。要求之定義會如圖 8-18 之最後圖表記錄所示，「QMON 聚合體配置」螢幕範例。

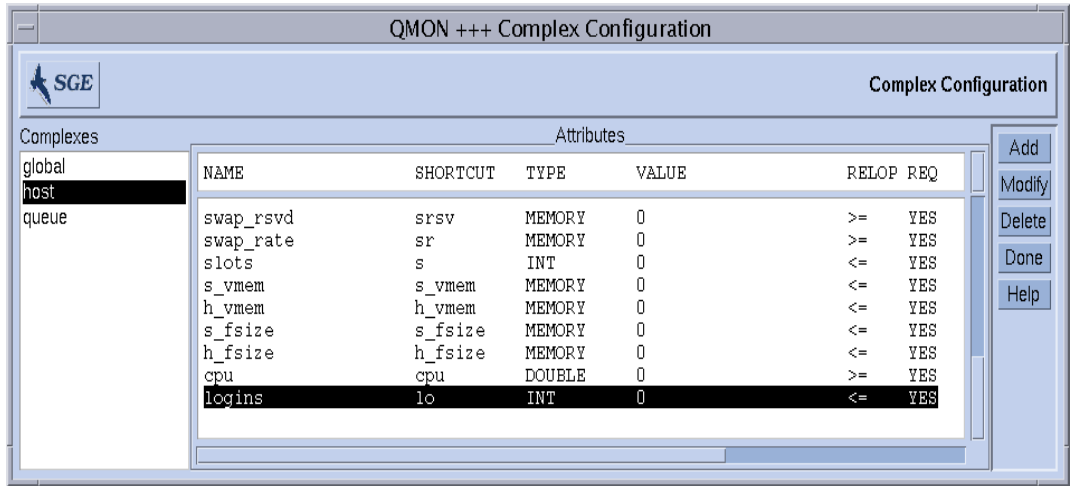


圖 8-18 聚合體配置對話方塊 — 登入

管理使用者存取與策略

本章包含 Sun Grid Engine 系統中與使用者管理、帳號和策略相關之重要資訊。本章主題包含使用者存取、排程、路徑別名、預設要求、運算與利用數據、及檢查點支援。

除了背景資訊外，本章還包涵如何完成下述作業之詳細指示。

- 第 191 頁的「如何以 QMON 配置帳號」
- 第 191 頁的「如何以 QMON 配置管理員帳號」
- 第 192 頁的「如何從指令行配置管理員帳號」
- 第 193 頁的「如何以 QMON 配置操作員帳號」
- 第 194 頁的「如何從指令行配置操作員帳號」
- 第 195 頁的「如何以 QMON 配置使用者存取清單」
- 第 197 頁的「如何從指令行配置使用者存取清單」
- 第 202 頁的「如何以 QMON 配置變更排程器配置」
- 第 210 頁的「如何以 QMON 配置檢查點環境」
- 第 214 頁的「如何從指令行配置檢查點環境」

關於設定使用者

以下清單說明於 Sun Grid Engine 中設定使用者之必要 / 可用作業：

- 必要登入

爲了從主機 *A* 提交作業，並於主機 *B* 執行，使用者必須於主機 *A* 與 *B* 使用相同帳號（即相同之使用者名稱）。執行 `sge_qmaster` 之機器無需做任何登入。

- 設定 Sun Grid Engine 存取權限

Sun Grid Engine 軟體有提供限制使用者存取至全部叢集、佇列、與平行環境的能力。請見第 195 頁的「關於使用者存取權限」一節得知更多詳細說明。

除此之外，Sun Grid Engine 系統使用者也可得到暫停或開啓特定佇列之權限（請見第 156 頁的「如何配置所有人」得知更多資訊）。

- 檔案存取限制

Sun Grid Engine 使用者必須讀到存取至目錄 `<sgc_root>/cell/common`。

在 Sun Grid Engine 作業開始前，Sun Grid Engine 執行常駐程式（以 `root` 執行）會替作業建立一暫時工作目錄，並且改變目錄所有人至作業所有人（當作業結束時就會移除此暫時目錄）。此暫時工作目錄會建立於由佇列配置參數 `tmpdir` 定義之路徑中（請參見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》裡的 `tmpdir` 項目以得知更多資訊）。

確認暫時目錄會建立於 `tmpdir` 底下，先設定 Sun Grid Engine 使用者所有權，如此使用者才可寫入暫時目錄中。

- 站點附屬

依定義，批次作業沒有終端機連結。所以，指令解譯器啟動資源檔（例如：`csch` 的 `cschrc`）中的 UNIX 指令，如 `stty`，也許會導致錯誤。檢查事件，以預防內含如第 42 頁的「如何驗證安裝」裡說明的指令出現。

因 Sun Grid Engine 批次作業通常會離線執行，僅有兩種方法可通知作業所有人錯誤事件與類似事件的發生。第一個方法是將錯誤訊息記錄至檔案中，而另一個則是寄電子郵件（e-mail）。在某些罕見的情況下（例：無法開啓錯誤記錄檔案），就僅能發送電子郵件直接通知使用者（諸如此類的錯誤訊息會記錄至 Sun Grid Engine 系統日誌檔案，但通常使用者並不會去看這個系統日誌檔案）。因此，如果 Sun Grid Engine 使用者有正確安裝電子郵件系統就會很幫助。

- Sun Grid Engine 定義檔案

您可以為 Sun Grid Engine 使用者設定下述定義檔案。

- `qmon`（Sun Grid Engine GUI 的資源檔；請見第 8 頁的「自訂 QMON」一節）
- `sgc_aliases`（最新作業目錄路徑別名；請參見第 205 頁的「關於路徑別名」一節）
- `sgc_request`（預設要求定義檔案；請見第 207 頁的「關於配置預設要求」一節）。

關於使用者存取

Sun Grid Engine 5.3 系統中存有三個使用者目錄。

- 管理員 — 管理員可全權操作 Sun Grid Engine。根據預設值，主要主機與其他管理佇列機器的超級使用者擁有管理員的特權。
- 操作員 — 操作員可執行許多與管理員相同之指令，除了新增、刪除、或修改佇列。
- 所有人 (Owners) — 佇列所有人受限不可暫停 / 繼續，或停用 / 啓動擁有之佇列。這些特權對 `qidle` 使用成功與否是必要的。使用者通常被宣告為位於其桌面工作站佇列的所有人。

以下各節會詳細說明各目錄。

▼ 如何以 QMON 配置帳號

1. 在 QMON 主要功能表上按下「使用者配置」鈕。
2. 根據您要如何配置按以下選擇標籤其中之一。
 - 管理員帳號配置（請見圖 9-1）
 - 操作員帳號配置（請見圖 9-2）
 - 使用者存取清單配置（請參閱圖 9-3）
3. 依以下各節指導進行。

注意 – 當第一次按下「使用者配置」鈕時，預設就會開啓「管理員」帳號配置對話方塊。

▼ 如何以 QMON 配置管理員帳號

當您選擇「管理員」標籤，就會出現「管理員配置」對話方塊（請見圖 9-1）。您可以在此對話方塊中宣告哪些帳號可以執行任何管理 Sun Grid Engine 指令。螢幕下半的選擇清單會顯示已宣告擁有管理權限之帳號。

- 刪除 – 只要按一下名稱，再按下對話方塊右方的「刪除」鈕，即可從清單上刪除一現有管理員帳號。
- 新增 – 只要於選擇清單上方的輸入視窗輸入名稱，按下「新增」鈕，或按鍵盤上的「Return」鈕，即可新增一管理員帳號。

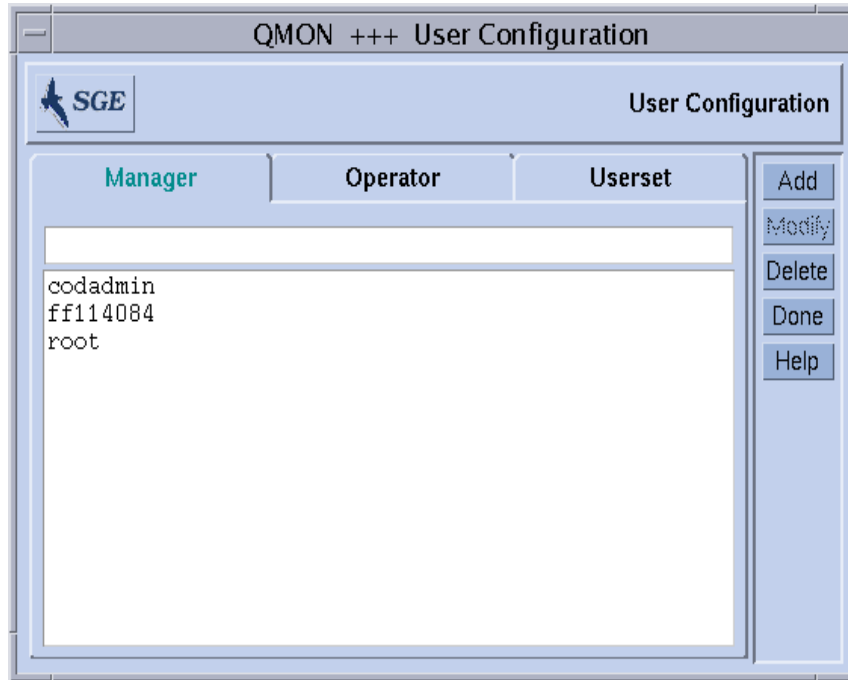


圖 9-1 管理員配置對話方塊

▼ 如何從指令行配置管理員帳號

- 使用合適的開關輸入以下指令。

```
# qconf 開關
```

可用的開關

- `qconf -am 使用者名稱 [...]`

新增管理員 — 此指令會新增一位或多位使用者至 Sun Grid Engine 管理員清單中。根據預設，所有 Sun Grid Engine 信賴的主機的 root 帳號（請見第 125 頁的「關於常駐程式和主機」一節）皆為 Sun Grid Engine 管理員。

- `qconf -dm 使用者名稱 [...]`

刪除管理員 — 此指令可自 Sun Grid Engine 管理員清單上刪除特定使用者。

- `qconf -sm`

顯示管理員 — 此指令會顯示所有 Sun Grid Engine 管理員的清單。

▼ 如何以 QMON 配置操作員帳號

當您選擇「操作員」標籤，就會開啓「操作員」配置對話方塊（請見圖 9-2）。您可以在此對話方塊中宣告哪些帳號允許擁有受限制之管理 Sun Grid Engine 指令權限（除非那些帳號也被宣告為管理員帳號—請參見第 191 頁的「如何以 QMON 配置管理員帳號」）。螢幕下半的選擇清單會顯示已宣告可提供操作員權限之帳號。

- 刪除 — 只要按一下名稱，再按下對話方塊右方的「刪除」鈕，即可從清單上刪除一現有操作員帳號。
- 新增 — 只要於選擇清單上方的輸入視窗輸入名稱，按下「新增」鈕，或按鍵盤上的「Return」鈕，即可新增一操作員帳號。

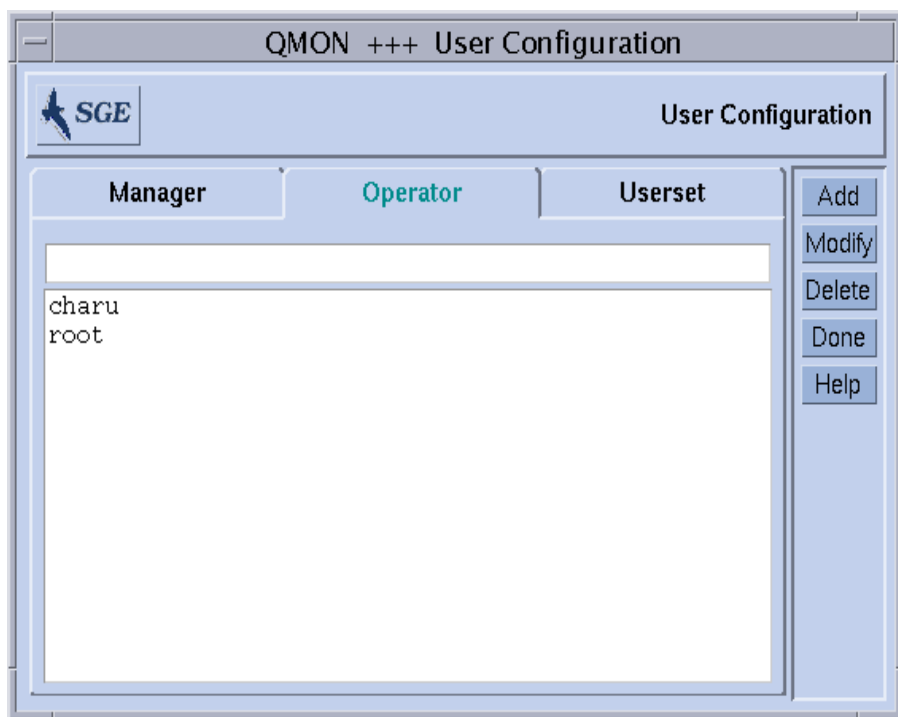


圖 9-2 操作員配置對話方塊

▼ 如何從指令行配置操作員帳號

- 使用合適的開關輸入以下指令。

```
# qconf 開關
```

可用的開關

- `qconf -ao 使用者名稱 [...]`
新增操作員 — 此指令會新增一位或多位使用者至 Sun Grid Engine 操作員清單中。
- `qconf -do 使用者名稱 [...]`
刪除操作員 — 此指令可自 Sun Grid Engine 操作員清單上刪除特定使用者。
- `qconf -so`
顯示操作員 — 此指令會顯示所有 Sun Grid Engine 操作員的清單。

關於佇列所有人帳號

佇列所有人會於配置或修改 Sun Grid Engine 佇列時完成定義。請見第 146 頁的「如何以 QMON 配置佇列」與第 156 頁的「如何從指令行配置佇列」二節。佇列所有人可做下述事項：

- 暫停 — 停止所有執行於佇列或靠近佇列之作業。
- 解除暫停 — 恢復佇列中的執行作業與開啓佇列。
- 停用 — 關閉佇列，但不會影響執行中的作業
- 啓動 — 開啓佇列。

注意 — 當一佇列暫停時，已明確暫停之作業將不會在繼續佇列時解除暫停。這些作業需要明確解除暫停執行。

一般而言，如果使用者有時需要特定機器來執行重要作業，或者他們被執行於背景之 Sun Grid Engine 作業嚴重影響，這些使用者就會被設定為特定佇列所有人。

關於使用者存取權限

任何至少於一個提交主機與執行主機上擁有有效登入之使用者皆有使用 Sun Grid Engine 系統的能力。但是，Sun Grid Engine 管理員仍可限制特定使用者存取至特定或所有佇列，而且也可限制工具的使用，例如特定平行環境（請見第 215 頁的「有關平行環境」一節）。

為定義存取權限，就必須先定義*使用者存取清單*—構成已命名仲裁重疊或未重疊之使用者集。使用者名稱與 UNIX 群組名稱可用來定義這些使用者存取清單。之後，使用者存取清單就會用於叢集配置（請見第 139 頁的「基礎叢集配置」一節）、佇列配置（請見第 154 頁的「如何配置次級佇列」一節）、或配置平行環境介面的作業流程中（請見第 216 頁的「如何以 QMON 配置 PE」一節），來拒絕或允許存取至一特定資源。

▼ 如何以 QMON 配置使用者存取清單

當您選擇「使用者集」標籤時，就會開啓「使用者集」配置對話方塊，如圖 9-3 所示。

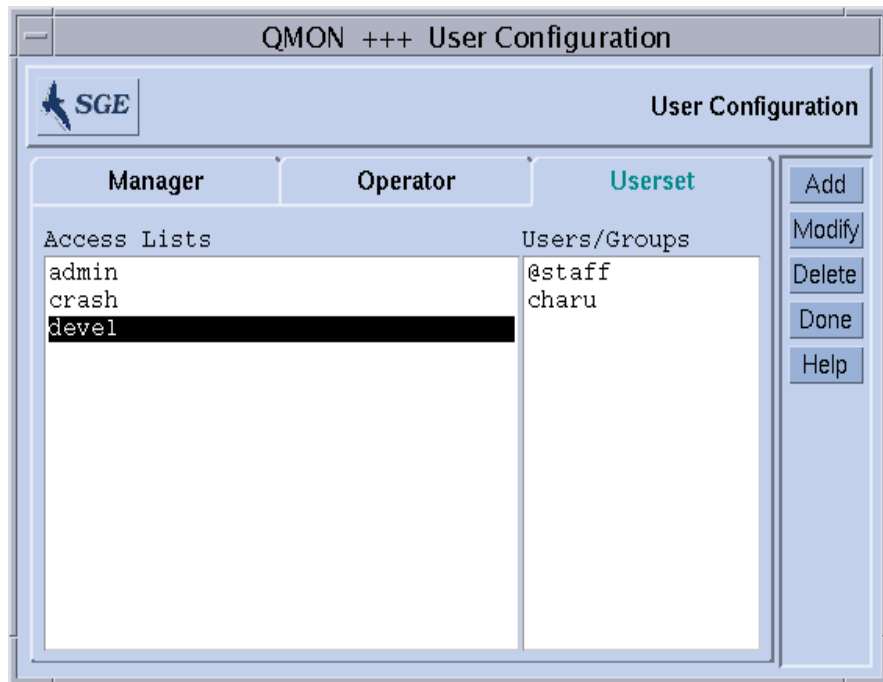


圖 9-3 使用者集配置對話方塊

可用存取清單顯示於螢幕左方的「使用者集」選擇清單中。要於「使用者 / 群組 (Users/Groups)」顯示區內顯示存取清單的內容，按一下「存取清單」選擇清單。

注意 – 「群組」前會以 @ 記號與「使用者」做區別。

您可以使用「使用者集」配置對話方塊來執行下述作業。

- 刪除 – 只要按一下名稱，再按下對話方塊右方的「刪除」鈕，即可從「使用者集」選擇清單上刪除一現有存取清單。
- 新增 – 只要於選擇清單上方的輸入視窗輸入名稱，按下「新增」鈕，或按鍵盤上的「Return」鈕，即可新增存取清單。
- 修改 – 按下「修改」鈕修改選取之存取清單。

要新增與修改時，就會開啓「存取清單」定義對話方塊，如圖 9-4 所示，並提供對應的方法。

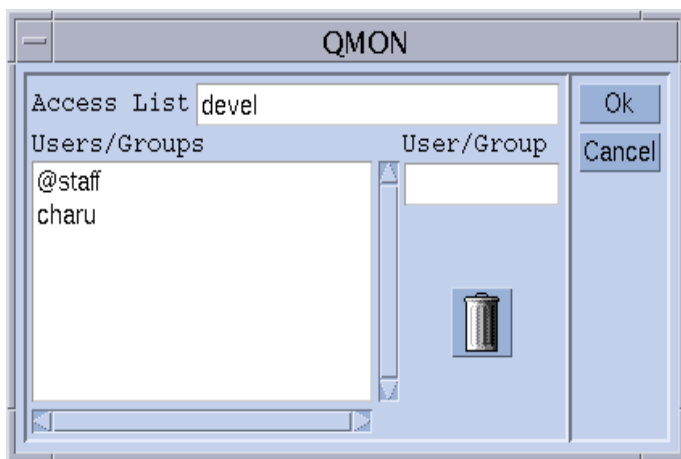


圖 9-4 存取清單定義對話方塊

「存取清單」定義對話方塊視窗解說

- 「使用者集名稱 (UserSet Name)」輸入視窗 – 顯示修改作業中選取之存取清單名稱，或者您也可以自行輸入欲宣告之存取清單名稱。
- 「使用者 / 群組」顯示區 – 包含目前定義的存取清單項目。
- 「使用者 / 群組」輸入視窗 – 必須用以新增新項目至存取清單中。

在您按下鍵盤上的「Return」後，已輸入之使用者或群組名稱（群組前有 @ 記號）就會出現於「使用者 / 群組」顯示區。您可以於顯示區選擇這些項目，然後按下垃圾筒圖示鈕來刪除這些項目。

在您按下 Ok 鈕時就會立刻記錄已修改或新增定義之存取清單，而按下「取消」鈕時就會立即捨棄所做的變更。兩種情況下皆會開啓「存取清單」定義對話方塊。

▼ 如何從指令行配置使用者存取清單

- 輸入以下指令與合適的選項。

```
# qconf 開關
```

可用的選項

- `qconf -au 使用者名稱 [...] 存取清單名稱 [...]`
新增使用者—本指令可新增一或多名使用者至特定存取清單中。
- `qconf -du 使用者名稱 [...] 存取清單名稱 [...]`
刪除使用者—本指令可從特定存取清單中刪除一或多名使用者。
- `qconf -su 存取清單名稱 [...]`
顯示使用者存取清單—本指令可顯示指定的存取清單。
- `qconf -sul`
顯示使用者存取清單—本指令會列出目前定義之所有存取清單列表。

關於排程

Sun Grid Engine 系統之作業排程活動包含以下幾種：

- 預先分派決定—為諸如消除執行佇列之類的活動，因為這些活動不是已滿便是過多，而目前位於等候區中之作業不適合執行。
- 分派—這些活動包含決定所有閒置與執行作業之重要性、感應叢集中所有機器之載入、並遵照配置選擇標準將作業寄至選取機器上的執行佇列。

Sun Grid Engine 軟體會依據以下幾點，跨過電腦不同叢集來排程作業：

- 叢集目前載入
- 作業資源需求（例如：CPU、記憶體、與 I/O 頻寬）

排程決定是依據站點，與叢集內每台電腦之即時載入特性之策略而定。站點之排程策略會透過 Sun Grid Engine 系統之配置參數呈現。載入特性會於系統執行時收集效能資料來確定。

排程策略

管理員會依據以下 Sun Grid Engine 排程作業來設定策略。

- 佇列分類 — 軟體會依據佇列填滿的順序，於叢集中排列佇列。
- 作業分類 — 確認 Sun Grid Engine 系統試圖排程作業之順序。

佇列分類

下述方法是用來確認 Sun Grid Engine 試圖填滿佇列的順序。

- 載入回報 — Sun Grid Engine 管理員可以選擇使用哪個載入參數來比較主機與其佇列之載入狀態。第 183 頁的「載入參數」一節說明可用標準載入參數之廣泛範圍，與利用站點特定載入感應器來延伸此設定之介面。
- 載入比例 — 由不同主機接收之載入回報可使其規範化，並反映一比較情況（請見第 130 頁的「如何以 QMON 配置執行主機」一節）。
- 載入調整 — Sun Grid Engine 軟體可配置為當分派作業至主機時，自動修正最後回報載入。已修正之載入數值會顯示由最新啓動作業所造成之載入情況的預期增加。當這些作業的載入發揮影響時，就會自動減少此人為載入增加。
- 序列編號 — 可依周密的序列分類佇列。

作業分類

在 Sun Grid Engine 開始分派作業前，作業會先以高優先序列排列。Sun Grid Engine 會企圖以優先序列為各作業尋找合適的資源。若沒受到任何管理員的影響，順序會成「先進先出」(FIFO) 排列。管理員會以下述幾種方式控制作業順序。

- 使用者分類 — 如果此排程替代於作用中，則不同使用者的作業會產生交錯。意即，所有使用者提交之第一項作業全受到平等對待，然後才是第二項，依此類推。
- 作業優先權 — 管理員會指定作業優先權編號，從而確認分類順序。使用者可降低指定至其作業之優先權順序。
- 使用者/群組作業最大量 — 同時執行於 Sun Grid Engine 系統的使用者或 UNIX 使用者群組擁有之作業最大量可受到限制。這會影響擱置中作業清單分類順序，因為未超出限制之使用者作業將享有優先權。

在排程器間隔時間內發生的事

排程器會於間隔時間內排程作業。在排程進行期間，Sun Grid Engine 會保留重大事件之資訊，例如作業提交、作業完成、作業取消、叢集配置之更新、或叢集內新機器之記錄。當開始排程時，排程器的工作如下。

- 考慮所有重大事件。
- 依據管理員規格來分類作業與佇列。

- 考慮所有作業資源要求。

然後若有需要，Sun Grid Engine 系統會執行下述工作。

- 分派新作業。
- 暫時執行中的作業。
- 維持現狀。

排程器監視

如果作業尚未開始，而且原因不明，您就可以於作業上執行 `qalter` 與 `-w v` 選項。Sun Grid Engine 軟體會預設一空白之叢集，並檢查是否有適合此作業之可用佇列。

您可執行 `qstat -j 作業id` 以得知更多相關資訊。之後便會列出作業要求資料之摘要，同時也會包含為何作業沒有加入最後排程的原因。不輸入作業 ID 來執行 `qstat -j` 將會摘要所有無法於最後排程間隔時間內列入排程作業的原因。

注意 – 排程原因資訊的收集必須開啓於排程器配置 `sched_conf`。請參閱對應《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `schedd_job_info` 參數，或第 202 頁的「如何以 QMON 配置變更排程器配置」一節。

要得到 Sun Grid Engine 排程器 `sge_schedd` 決定的更詳細資訊，您可使用 `qconf` 指令的 `-tsm` 選項。此指令將強制 `sge_schedd` 將追蹤輸出寫入檔案中。

預設排程

預設的 Sun Grid Engine 排程為先進先出策略，即先提交之作業可先由排程器檢驗分派至佇列中。如果擱置作業清單中的第一個作業找到合適且閒置的佇列，此作業就會先列於排程執行中。但如果第一個作業無法找到一合適可用的資源，第二個作業或之後的作業就會在擱置作業清單中的第一個作業之前，先列入排程執行中。

只要關係到作業佇列選擇，預設的 Sun Grid Engine 策略就會於載入量最少的主機中，選擇能為作業資源要求提供合適服務的佇列。如果有多個合適佇列共享相同載入，就無法預知會選擇哪一個佇列。

排程替代

有許多方法可修改作業排程與佇列選擇策略。

- 改變排程演算法
- 按比例排列系統載入
- 依序列編號選擇佇列
- 依共享選擇佇列

- 限制一使用者或一群組之作業數量

以下幾節將詳細解說這些選擇方法。

改變排程演算法

排程器配置參數 `algorithm`（請見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `sched_conf` 項目得知更多資訊）是設計為使用中的排程演算法提供一個選項。目前為止，`default` 是唯一允許之設定。

作業屬性

Sun Grid Engine 管理會指定一整數為作業優先權順序到於擱置作業清單中多工緩衝處理的作業。作業優先權順序會定義作業於擱置作業清單中的位置。具最高優先權編號的作業將會先由排程器檢驗。作業優先順序的數值範圍是介於 -1024 與 1023 之間，而剛提交作業之優先順序數值則為 0。如果負優先權數值指定到作業，就表示此作業已被分類至剛提交作業之後。如果許多作業擁有相同的優先權編號，預設之先進先出規則也適用於此優先等級中。

藉由下述指令指定一作業之作業優先順序。

```
% qalter -p 優先權作業 id ...
```

在上述指令中，`prio` 會指定作業清單中的優先順序，就像於移動空白分隔的作業 `id` 清單中指定一樣。

注意 – 作業優先順序一詞不應與優先佇列配置參數（定義執行於特定佇列之所有作業的 `nice` 數值）混為一談（請見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `queue_conf` 線上說明手冊）。同時也須注意，`qstat` 輸出中的第二欄會顯示最新指定至提交作業的優先順序。

平等共享分類

上述提及之預設的先進先出排程策略會產生不公平的結果 – 如果使用者於短期間連續提交一系列的作業（例如：使用 `shell script` 程序）。此使用者的作業會長期佔據適合的資源，於是其他使用者便沒有機會分配佇列。

在此情況下，叢集管理就會變更排程策略至所謂的平等共享分類。如果此排程選擇有效，而使用者也於系統中有一執行作業，則此使用者的所有其他作業就會被分類至其他擁有相同優先順序等級的使用者作業之後（請見上一節得知優先等級的詳細資訊）。

如果此排程配置參數 `user_sort` 設為 `TRUE`（請見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `sched_conf` 線上說明手冊），就會啟動平等共享分配。

按比例排列系統載入

Sun Grid Engine 系統使用管理佇列機器上的系統載入資訊為一作業選擇執行佇列。本佇列選擇方案會建立一載入平衡情況，保證能更適當使用叢集之可用資源。

但是系統載入數值並非總是真實的。例如，若比較多重 CPU 機器與單一 CPU 機器系統，多重處理器系統通常會回報較高的載入數值，因其也許會執行更多程序，而且系統載入數值會嚴重受到試圖進入 CPU 存取的作業數量。但是與單一 CPU 機器相比，多重 CPU 系統能滿足較高載入數值。此問題於載入數值處理器的數量調整組（由 `sge_execd` 預設回報）中有詳細論述（請見第 183 頁的「載入參數」一節與 `<sge_root>/doc/load_parameters.asc` 檔得知詳細資訊）。使用這些載入參數，而非原始載入數值，以避免上述提及的問題發生。

另一潛在載入數值錯誤解譯之範例為系統之潛在效能或其 `price` 效能比例有很大的差異，兩者相同的載入數值並不代表可選取仲裁主機以執行作業。在此情況下，**Sun Grid Engine** 管理員應定義執行主機與載入參數之載入比例因素（請參閱第 130 頁的「如何以 `QMON` 配置執行主機」與其相關各節）。

注意 – 依比例載入數值參數也可用於比較兩者間的載入臨界值清單 **載入臨界值** 與 **migr 載入臨界值**（請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `queue_conf` 項目得知詳細資訊）。

與載入參數相關的其他問題為需要解譯數值與其相對重要性之應用程式與從屬站點。CPU 載入會由特定站點常見之特定應用程式類型所主導，而對另一站點與站點運算叢集通常存在之應用程式檔案而言，記憶體載入的重要性又更高。**Sun Grid Engine** 允許管理員於排程配置檔案指定一所謂的 **載入準則** 來描述這問題（請參閱對應的《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》章節以取得詳細資訊）。使用載入準則中的站點定義載入參數（請見第 184 頁的「新增站點特定載入參數」一節）與可耗損資源（請見第 172 頁的「可耗損資源」一節），即可將資源利用與容量規劃的站點特定資訊列入考慮。

最後，也需將載入參數之時間相依也列入考慮。位於執行在系統上的 **Sun Grid Engine** 作業載入數值會隨著時間不同而改變，而且通常 — 以 CPU 載入為例 — 都需要一些時間等待作業系統回報合適的數量。因此，如果最新執行一作業，回報的載入數值就無法提供此作業已於主機上的完整數值。此回報值會隨著時間適應真正的數值，但在一段時間內若回報值若過低，就會導致主機過度耗損。**Sun Grid Engine** 允許管理員指定用於 **Sun Grid Engine** 排程器之 **載入調整要素** 來補償這個問題。請參閱可解決排程器配置檔 `sched_conf` 的《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》以得知如何設定這些載入調整要素之詳細資訊。

依序列編號選擇佇列

另一個改變預設佇列選擇方案的方法為設定全域 Sun Grid Engine 叢集配置參數 `queue_sort_method` 至 `seq_no`，而非預設 `load`（請參見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `sched_conf` 項目）。在此情況下，系統載入就不會為選擇佇列的主要方法。反而會由佇列配置參數 `seq_no`（請見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `queue_conf` 項目）來指定佇列的序列編號，再用來定義選取之佇列之間的固定順序（如果這些序列編號適用於考慮中的作業，而且是閒置的）。

如果您站點提供批次服務之機器是以一作業單一 `price` 的順序排列，此佇列選擇策略也許會有幫助。例如，執行於 A 機器的作業花費一單元的金錢，但它於 B 機器要花費 10 個單元，而 C 機器要花費 100 個單元。因此此排程策略會傾向首先填滿 A 主機，之後 B 主機，到別無選擇後才是 C 主機。

注意 – 如果您已變更佇列選擇方法於 `seq_no`，而且考慮的佇列皆共享相同之順序號碼，則將由預設 `load` 選擇佇列。

限制單一使用者或群組之作業數量

Sun Grid Engine 管理員會指定在任何時間內使用者或 UNIX 群組可執行之作業數量的上限。為了執行此功能，設定《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》內的 `sched_conf` 部分說明之 `maxujobs` 與 `/` 或 `maxgjobs`。

▼ 如何以 QMON 配置變更排程器配置

1. 在 QMON 主要功能表上按一下「排程器配置」。

就會出現「排程器配置」的對話方塊。此對話方塊分隔成「一般參數」一節與「載入調整」一節。您可依據您想達成的目的選擇兩者其中之一。

a. 要改變一般排程參數，請按一下「一般參數」標籤。

圖 9-5 為「一般參數」對話方塊範例。

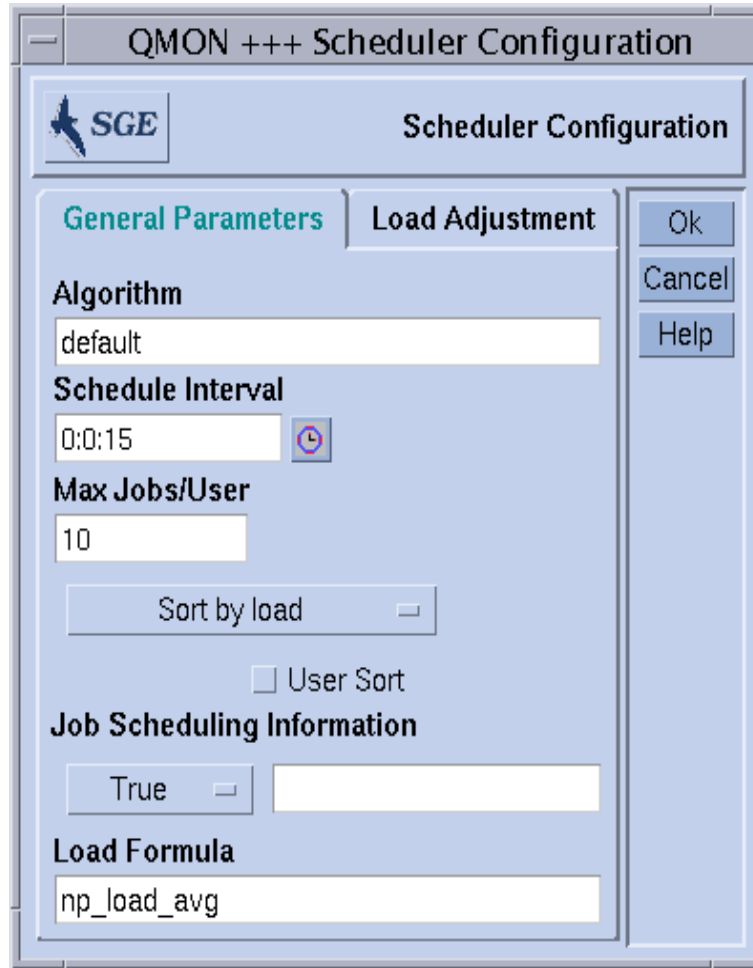


圖 9-5 排程器配置對話方塊 — 一般參數

從「一般參數」對話方塊中，您可以設定以下參數。

- 排程演算法（請見第 200 頁的「改變排程演算法」）
- 排程器執行之規律間隔時間
- 單一使用者與 UNIX 群組允許同時執行之作業最大量（請見第 202 頁的「限制單一使用者或群組之作業數量」）。
- 佇列分類方案—由載入數值或序列編號分類（請見第 202 頁的「依序列編號選擇佇列」）。

- 平等共享分類（User Sort 旗號）是否有啟動（請見第 200 頁的「平等共享分類」）。
 - 作業排程資訊能否透過 `qstat -j` 存取，或此資訊能否應僅由隨附輸入欄位中指定之作業 ID 收集。建議僅暫時開啓作業排程資訊的一般收集，以避免擱置作業量過多。
 - 用以分類主機與佇列之載入準則
- b. 要改變載入調整參數，請選擇「載入調整」標籤。

圖 9-6 為「載入調整參數」對話方塊範例。

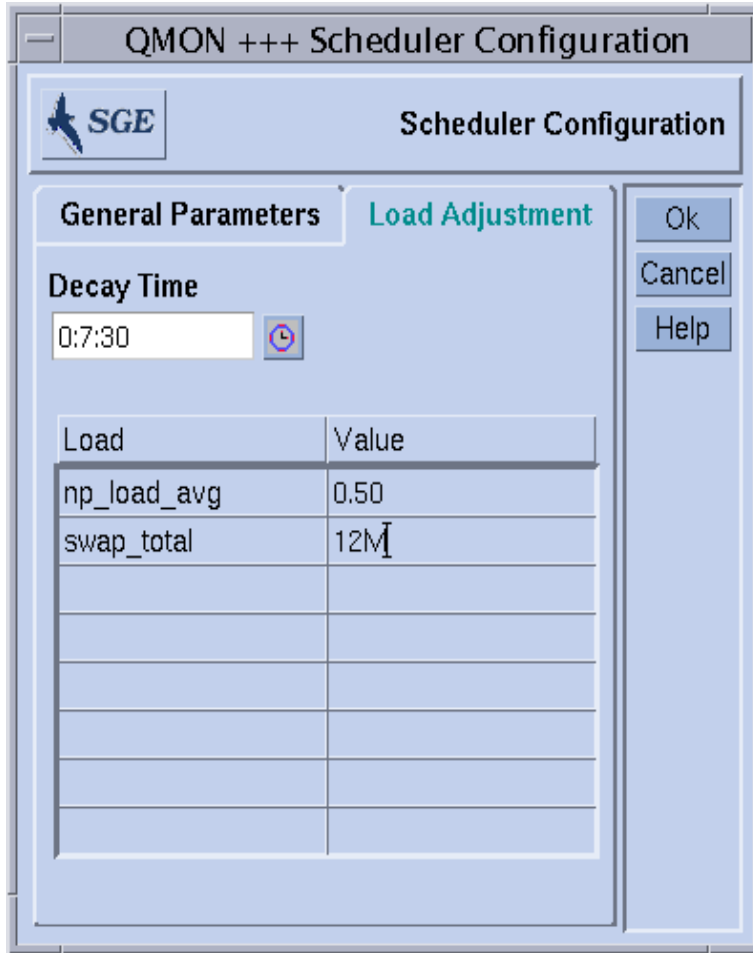


圖 9-6 排程器配置對話方塊 — 載入調整

「載入調整」對話方塊使您可定義以下參數。

- 載入調整衰減時間

- 對話下半的載入調整值圖表會列出所有最近已定義的調整值之載入與可耗損屬性。可按一下上方的「載入」或「數值」鈕來增加清單。之後會開啓一選擇清單，列出所有隨附於主機之屬性（即配置於全域、主機、與管理員定義之聚合體）。「屬性選擇」對話方塊顯示於圖 6-6。選擇其中一個屬性，並按「Ok」鈕確認選項，就會新增此屬性至「可耗損/固定屬性」圖表之「載入」欄，並於「數值」區加入指標。連接二下「數值」欄位以修改現有數值。選擇對應圖表行，然後輸入 CTRL-D 即可刪除屬性 — 或按一下滑鼠右鍵開啓刪除方塊，然後確定刪除動作。

請見第 201 頁的「按比例排列系統載入」以得知相關背景資訊。請參閱《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》裡的 sched_conf 線上說明手冊以得知排程器配置的詳細資訊。

關於路徑別名

在 Solaris 與其他網路之 UNIX 環境中，如果相同的主目錄可跨過 NFS 存取，使用者常於不同機器上使用此相同之主目錄（或部分相同）。但有時候所有機器上的主目錄路徑又非完全相同。

舉例來說，假設使用者主目錄可透過 NFS 與自動裝載器使用。如果使用者在 NFS 伺服器上有一個主目錄，/home/foo，他就可以在所有正確安裝之 NFS 客戶站執行自動裝載器中，以此為路徑存取主目錄。但是請注意，客戶端之 /home/foo 只是到 /tmp_mnt/home/foo 的符號連結，即裝載器實體裝載此目錄至 NFS 伺服器上的真實位址。

如果使用者以 qsub -cwd 旗號（於最新工作目錄中執行作業），從主目錄樹中提交作業至客戶端，Sun Grid Engine 系統連結至執行主機上最新工作目錄就會出現問題 — 如果此主機是 NFS 伺服器的話。這是因為 qsub 指令將到達提交主機上的最新工作目錄，並以 /tmp_mnt/home/foo/ — 為此提交主機上的實體位址。此路徑將會通過到達執行主機，而且如果執行主機為 NFS 伺服器，又有 /home/foo 之實體主目錄路徑，就無法解決此路徑。

導致類似問題之其他情況通常會以不同機器上之不同裝載點路徑（例如於主機上 /usr/people 底下裝載主目錄，並於另一主機上 /usr/users 底下裝載主目錄），或者由以外面進入一可用網路檔案系統之符號連結，來修護（非自動裝載）NFS 裝載。

為避免這類問題發生，Sun Grid Engine 軟體讓管理員與使用者皆可配置路徑別名檔。兩個檔案的位址如下：

- <sge_root>/<儲存格>/common/sge_aliases — 此為叢集全域路徑別名檔。
- \$HOME/.sge_aliases — 此為使用者特定路徑別名檔。

注意 – 只有合格的管理員才應修改叢集全域檔。

檔案格式

兩個檔案共享相同的格式。

- 請跳過第一欄的空白行與有 # 記號的行。
- 每一行—除去空白行與前有 # 之行— 必須包含四個字串，之間分別按下數次空白鍵或 tab 定位格鍵分開。

第一個字串指定資源路徑、第二個是提交主機、第三個為執行主機、而第四個為資源替代路徑。

- 提交與執行主機記錄僅由符合任一主機之 * 記號組合。

如何解譯路徑別名檔

檔案解譯方式如下。

- 在 qsub 取得實體最近作業目錄路徑後，就會讀取叢集全域路徑別名檔 — 如果有的話。之後便會讀取使用者路徑別名檔，有如它是隨附於全域檔案一般。
- 無需忽略的文字行會由檔案上方讀取，一行接著一行，而且如果需要的話，由那些行指定之翻譯也會隨之儲存起來。
- 當提交主機記錄符合執行 qsub 指令之主機，且資源路徑之第一部分為最新作業目錄或已儲存之資源替代路徑時，就會儲存翻譯。
- 只要兩個檔案皆被讀取，儲存之路徑別名資訊就會與提交作業一同傳送出去。
- 在執行主機上會評估別名資訊。如果別名路徑的執行主機記錄符合執行主機，最新作業目錄之第一部分將會由資源替代路徑取代。請注意：最新作業目錄字串在此情況下也會變更，而之後的別名路徑也必須符合套用之替代作業目錄路徑。

路徑別名檔之範例

下例為如何以別名檔案項目解決 NFS/ 自動裝載器 如上描述之問題的範例。

```
# cluster global path aliases file
# src-path   subm-host  exec-host      dest-path
/tmp_mnt/   *              *              /
```

路徑別名檔案範例

關於配置預設要求

批次作業通常是由與特定作業使用者定義的要求檔案相關之 Sun Grid Engine 系統分配至佇列。使用者會收集需要成功執行作業之一連串要求，而 Sun Grid Engine 排程器僅會考慮滿足作業要求之佇列。

如果使用者沒有指定作業之要求，排程器就會考慮使用者會存取之佇列，沒有其他限制。但即使使用者沒有明確指定，Sun Grid Engine 軟體仍允許配置可定義作業資源要求的預設要求。

預設要求可全域配置至 Sun Grid Engine 叢集之所有使用者，也可隱密配置至所有使用者。預設要求配置會顯示於預設要求檔案中。全域要求檔位於 `<sg_e_root>/<cell>/common/sg_e_request` 底下，而使用者指定要求檔，又稱 `.sg_e_request`，則位於使用者主目錄或執行 `qsub` 指令之最新作業目錄底下。

如果這些檔案出現，就會受到所有作業評估。評估的順序如下：

1. 全域預設要求檔案
2. 使用者主目錄內之使用者預設要求檔案
3. 最新工作目錄內之使用者預設要求檔案

注意 – 由作業 `script` 指定，或 `qsub` 指令行提供之要求享有較高之優先權，等同預設要求檔案之要求（請見第四章得知如何明確要求作業資源）。

注意 – 預設要求檔案無意造成之影響可使用 `qsub -clear` 選項加以限制，捨棄之前的任何規格要求。

預設要求檔案之格式

下述清單說明本端與全域預設要求檔案之格式。

- 預設要求檔案會包含行列的仲裁編號。請跳過第一欄的空白行與有 # 記號的行。
- 而其他不可忽略的每一行會包含任何 `qsub` 選項，請見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》說明。一行需超過一個選項。批次 `script` 檔與批次 `script` 之引數選項並未包含至 `qsub` 選項中，所以也不允許出現於預設要求檔案中。
- `qsub -clear` 選項會捨棄先前於最新評估要求檔案或執行之要求檔案內的任何規格要求。

預設要求檔案範例

假設使用者之本端預設要求檔案配置如同程式範例 1 中之 `test.sh` script。

```
# Local Default Request File
# exec job on a sun4 queue offering 5h cpu
-l arch=solaris64,s_cpu=5:0:0
# exec job in current working dir
-cwd
```

程式範例 1 預設要求檔案範例

要執行此 script，使用者需輸入下述指令。

```
% qsub test.sh
```

執行 `test.sh` script 之影響會與使用者直接於指令行中指定 `qsub` 選項一般；如下。

```
% qsub -l arch=solaris64,s_cpu=5:0:0 -cwd test.sh
```

注意 – 像透過 `qsub` 提交之批次作業一般，透過 `qsh` 提交之互動作業也將考慮預設要求檔案。透過 `QMON` 提交之互動或批次作業也將顧及這些要求檔案。

關於收集運算與利用數據

Sun Grid Engine 指令，`qacct`，可用以收集字母與數字構成之運算數據。如果沒有開關，`qacct` 就會顯示 Sun Grid Engine 叢集之所有機器上的利用總合計，即所有已於叢集運算檔案，`<sg_e_root>/<儲存格>/common/accounting`，中完成之作業的總合計。在此情況中，`qacct` 會於短時間內回報三次。

- REAL – wallclock 時間；即作業開始到結束的時間。
- USER – CPU 於執行使用者作業時花費的時間。
- SYSTEM – CPU 於系統呼叫花費的時間。

有些開關可用來回報所有或特定佇列、使用者等諸如此類的運算資訊。特別是也有可能要求所有完成作業之資訊，和符合與 `qsub` 指令相同之 `-l` 語法內之資源指定要求作業的資訊，來提交作業。要取得更多的資訊，請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》裡的 `qacct` 項目。

`qacct` 選項 `-j [作業 id|作業名稱]` 會提供直接存取至完整資源利用資訊（由 Sun Grid Engine 系統儲存），包含 `getrusage` 系統呼叫提供的資訊（請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的對應項目）。

本選項會分別回報作業 `id` 為 `[作業 id]`，或作業名稱為 `[作業名稱]` 之作業資源使用記錄。如果沒有給予任何引數，就會顯示位於參考計算檔案之所有作業。如果有選取一作業 `id`，而且顯示記錄超過一筆，則不是已隱藏作業 `id` 號碼（作業 `id` 的範圍從 1 到 999999），就是顯示已遷移之檢查點作業。

關於檢查點支援

檢查點為一可冷凍執行中作業或應用程式狀態、儲存狀態（即所謂的檢查點）至磁碟、並稍後由檢查點重新執行（如果一作業或應用程式無法完整執行，例：因系統關閉所致）之工具。如果檢查點可至一主機移到另一主機，檢查點就可用來遷移叢集中的應用程式或作業，而不會大量流失運算資源。因此，檢查點工具就需提供動態負載平衡。

Sun Grid Engine 系統支援兩種等級的檢查點。

■ 使用者等級檢查點

在此等級中，使用者或應用程式需全權負責提供檢查點產生機制。使用者等級檢查點範例包含以下幾點。

- 在重要的演算步驟中於應用程式裡編碼並定期撰寫重新啟動檔案，與重新啟動應用程式時這些檔案之正確執行程序。
- 檢查點圖書館之使用，需要連結至應用程式與安裝檢查點機制。

注意 – 各種協力廠商應用程式會以撰寫之重新啟動檔為主，提供一整合之檢查點工具。您可從公共領域（請參閱 Wisconsin 大學的 *Condor* 專案為例）或硬體廠商來使用檢查點圖書館。

■ 核心等級透明檢查點 (*Kernel-level transparent checkpointing*)

此檢查點等級必須由適用潛在仲裁作業之作業系統（或其升級版）提供。無需提供您應用程式之資源碼變更或重新連結，即可使用核心等級檢查點。

核心等級檢查點適用於完成之作業 — 即由作業建立之執行階層 — 而使用者等級檢查點則通常受到單一程式之限制。因此，在作業開始執行時，就必須合適處理受到此程式包圍之作業。

核心等級檢查點與位於檢查點檔案庫之檢查點都會很耗損資源，因為在執行檢查點時，由作業或應用程式使用之全部虛擬位址空間需要被轉儲至磁碟中。相反地，以重新啓動檔爲主之使用者等級檢查點僅會限制寫入重要資訊上的檢查點之資訊。

檢查點環境

要反映檢查點法不同的種類，與其在不同作業系統架構上可能衍生出的種類，Sun Grid Engine 爲所有使用中的檢查點法皆提供一配置屬性說明。

此屬性說明即爲檢查點環境。Sun Grid Engine 分佈會提供預設檢查點環境，並可對應站點需求做修改。

新的檢查點法原則上可以被整合，但整合作業很具有挑戰性，所以應由您 Sun Grid Engine 支援團隊中有經驗之人員來執行。

▼ 如何以 QMON 配置檢查點環境

1. 在 QMON 主要功能表上按下「檢查點配置」圖示。

接著就會出現「檢查點配置」對話方塊，如圖 9-7 所示。

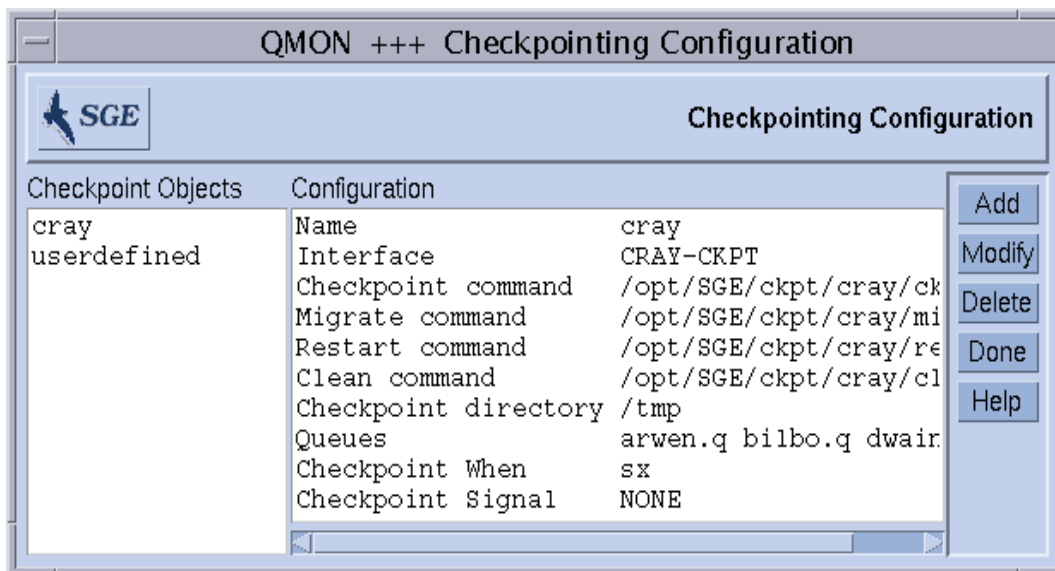


圖 9-7 檢查點配置對話方塊

2. 在檢查點配置對話方塊中，依據您的目的執行以下之一的作業。

檢視配置的檢查點環境

- 要檢視之前配置的檢查點環境，於「檢查點物件」欄選擇一列出的檢查點名稱。其對應配置便會顯示於「配置」欄中。

刪除配置的檢查點環境

- 要刪除配置的檢查點環境，從「檢查點物件」欄中反白欲刪除之名稱，然後按一下「刪除」。

修改配置的檢查點環境

1. 在「檢查點物件」欄中，反白您想修改之已配置的檢查點環境名稱，然後按一下「修改」。
出現與圖 9-8 範例相似的「改變檢查點物件」對話方塊，以及目前選取之檢查點環境的配置。

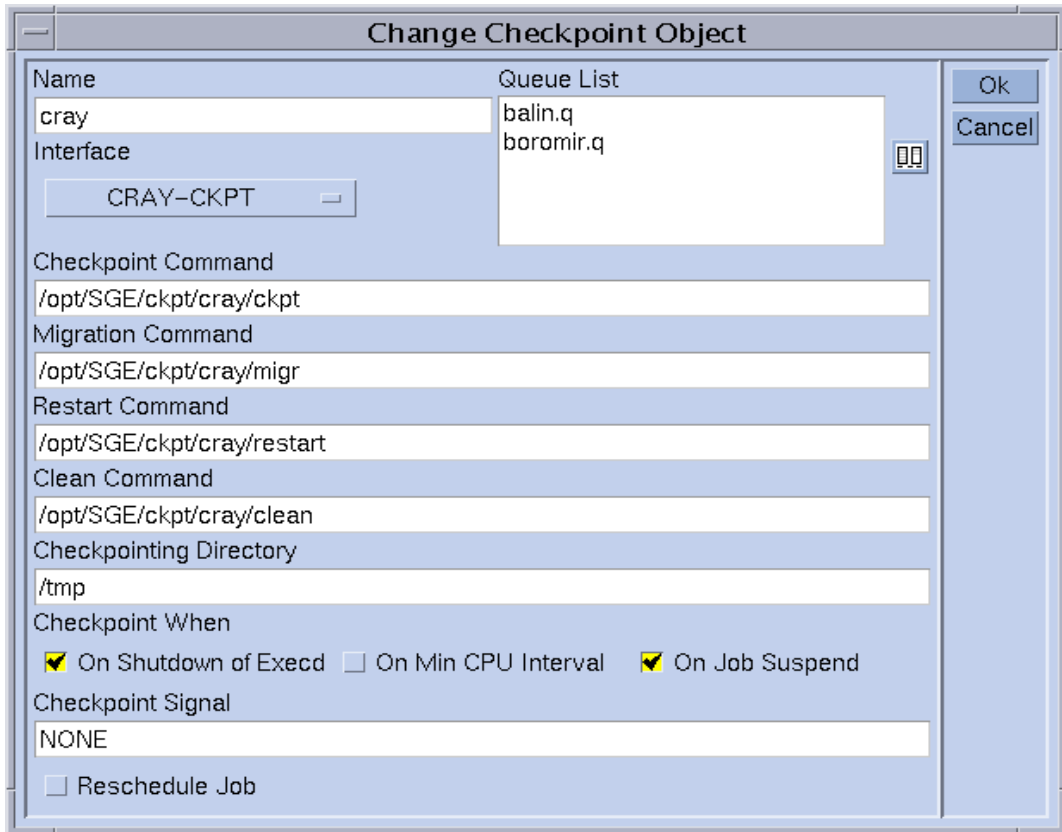


圖 9-8 改變檢查點物件對話方塊

2. 遵照下述指示修改選取之檢查點環境。

「改變檢查點物件」對話方塊可用來變更下述事物。

- 名稱
- Checkpoint、migrate、restart、clean 指令字串
- 檢查點檔儲存之目錄
- 必須起始檢查點之時刻
- 當起始檢查點時必須傳送至作業或應用程式之訊號

注意 – 請參閱《Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊》中的 checkpoint 項目以得知這些參數的詳細資訊。而且您還必須定義使用的介面（又稱檢查點法）。於對應選擇清單中選擇一項，然後至 checkpoint 項目中得知不同介面意義之詳細資訊。

3. **重要** – 在 Sun Grid Engine 分配提供之檢查點環境中，僅變更名稱、檢查點目錄、與佇列清單參數。

要變更「佇列清單」參數，請至「步驟 a」。不然就跳過「步驟 a」，然後至「步驟 4」。

- a. 按下「佇列清單」視窗右方的圖示（請見圖 9-8）。

出現與圖 9-9 範例類似的「選擇佇列」對話方塊。

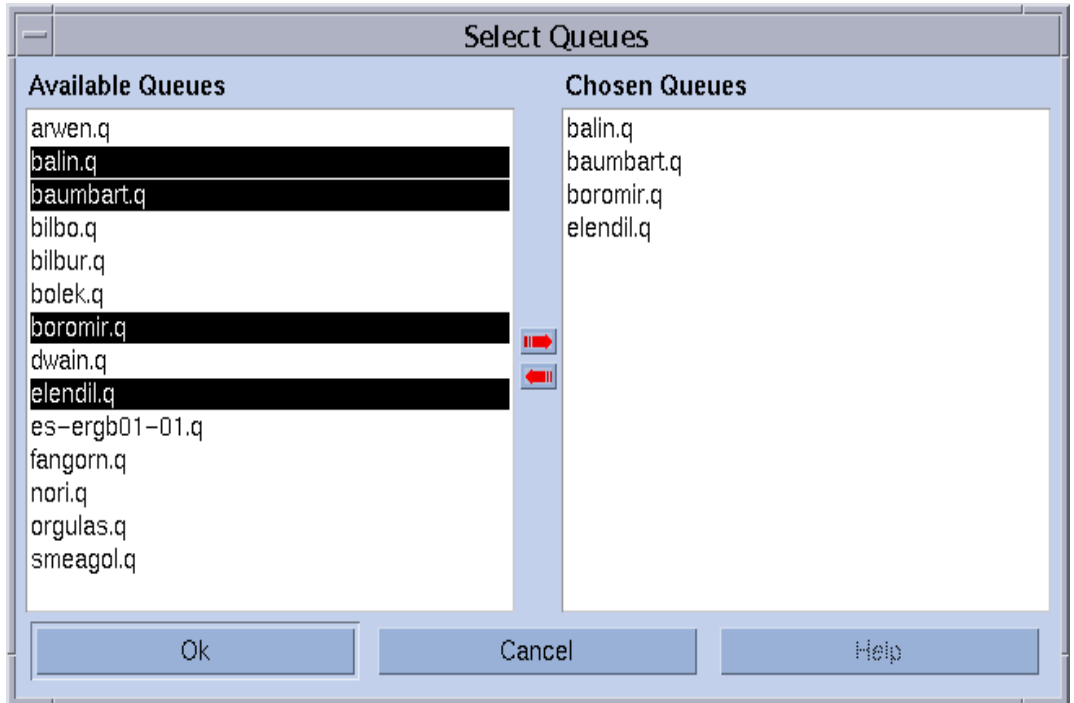


圖 9-9 檢查點佇列選擇對話方塊

- b. 由「可用佇列」清單中選擇您想加入於檢查點環境中的佇列，然後將其新增至「選擇的佇列」清單中。
 - c. 按一下「確定」。
按一下「確定」將佇列新增至「改變檢查點物件」對話方塊之「佇列清單」視窗中。
4. 按一下「確定」以 `sge_qmaster` 來記錄您的變更，或者按一下「取消」捨棄您的變更。

新增檢查點環境

1. 在「檢查點配置」對話方塊中，按一下「新增」。
之後就會出現「改變檢查點物件」對話方塊與您可編輯之平台配置，如圖 9-8 範例所示。

2. 將所需資訊填入平台中。
3. 按一下「確定」以 `sge_qmaster` 來記錄您所做的變更，或按一下「取消」來捨棄變更。

▼ 如何從指令行配置檢查點環境

- 遵照以下幾節指示輸入 `qconf` 指令與適合之選項。

qconf 檢查點選項

- `qconf -ackpt` 檢查點名稱

新增檢查點環境 — 此指令會叫出編輯器（預設 `vi` 或對應 `$EDITOR` 環境變數）與檢查點環境配置平台。此參數 *檢查點名稱* 會指定檢查點環境之名稱，並已填妥對應平台欄位之資訊。變更平台並儲存至磁碟片中就完成配置檢查點環境的工作。請參見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》內的 `checkpoint` 項目以得知改變平台記錄的詳細資訊。

- `qconf -dckpt` 檢查點名稱

刪除檢查點環境 — 此指令會刪除指定之檢查點環境。

- `qconf -mckpt` 檢查點名稱

修改檢查點環境 — 此指令會叫出編輯器（預設 `vi` 或對應 `$EDITOR` 環境變數）與指定之檢查點環境配置平台。變更範本並儲存至磁碟片中就完成修改檢查點環境的工作。請參見《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》內的 `checkpoint` 項目以得知改變平台記錄的詳細資訊。

- `qconf -sckpt` 檢查點名稱

顯示檢查點環境 — 本指令會列出指定的檢查點環境配置於標準輸出上。

- `qconf -sckptl`

顯示檢查點環境清單 — 本指令會顯示所有最新配置之檢查點環境名稱清單。

管理平行環境

本章旨在說明處理與管理平行環境之資訊。

除了這些主題之背景資訊外，本章還包含如何完成下述作業之詳細指示。

- 第 216 頁的「如何以 QMON 配置 PE」
 - 第 216 頁的「顯示 PE 的內容」
 - 第 216 頁的「刪除 PE」
 - 第 216 頁的「修改 PE」
 - 第 217 頁的「新增 PE」
- 第 220 頁的「如何從指令行配置 PE」
- 第 220 頁的「如何從指令行顯示配置的 PE 介面」
- 第 220 頁的「如何以 QMON 顯示配置的 PE 介面」

有關平行環境

平行環境 (PE) 為套用以於網路環境或平行平台同時計算之套裝軟體。過去有許多系統已開發許多可在硬體平台上分配並平行執行之可行科技。現今最常見的二個訊息傳遞環境範例即為 PVM (平行虛擬器 (Parallel Virtual Machine), Oak Ridge National Laboratories) 與 MPI (訊息傳遞介面 (Message Passing Interface), Message Passing Interface Forum)。此二項工具皆包含公共領域與硬體廠商提供之執行。

這些系統皆顯示不同的特性，並有其不同之要求。為了可處理執行於類似系統上之仲裁平行作業，Sun Grid Engine 系統提供一有彈性與強大之介面，可符合眾多的需求。

只要有合適之啟動與結束程序，仲裁 PE 便可由 Sun Grid Engine 來連接介面，詳細情形請見第 223 頁的「終止 PE」的第 222 頁的「PE 啟動程序」一節。

▼ 如何以 QMON 配置 PE

1. 在 QMON 主要功能表上按下「PE 配置」鈕。
接著就會出現「平行環境配置」對話方塊，如圖 10-1 所示。

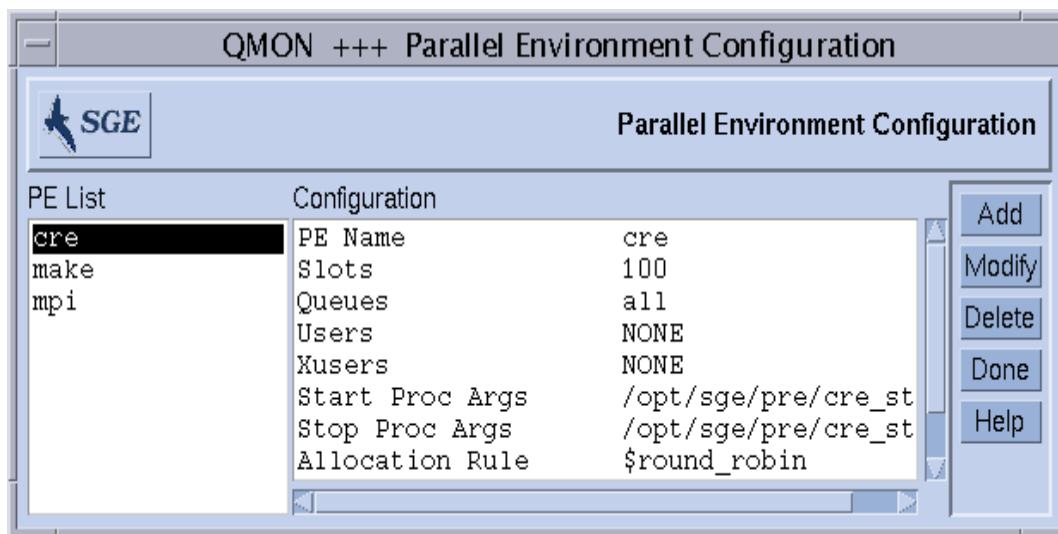


圖 10-1 平行環境配置對話方塊

已配置之 PE 會顯示於螢幕左方的「PE 清單」選擇清單中。

2. 在「平行環境配置」對話方塊中，依據您的目的執行以下之一的作業。

▼ 顯示 PE 的內容

- 要顯示 PE 內容，請按一下「PE 清單」選擇清單中的名稱。
PE 配置的內容就會顯示於「配置」顯示區中。

▼ 刪除 PE

- 要刪除選取的 PE，請反白「PE 清單」選擇清單上的名稱，然後按一下「刪除」（位於視窗右方）。

▼ 修改 PE

1. 要修改 PE，請按一下「修改」鈕。
圖 10-2 為「PE 定義」對話方塊範例。
2. 請遵照第 217 頁的「平行環境定義參數說明」一節指示修改 PE 定義。

3. 按一下「確定」以儲存變更，或「刪除」來捨棄變更。
按一下「確定」或「刪除」來關閉對話方塊。

▼ 新增 PE

1. 要新增 PE，請按一下「新增」鈕。
圖 10-2 為「PE 定義」對話方塊範例。

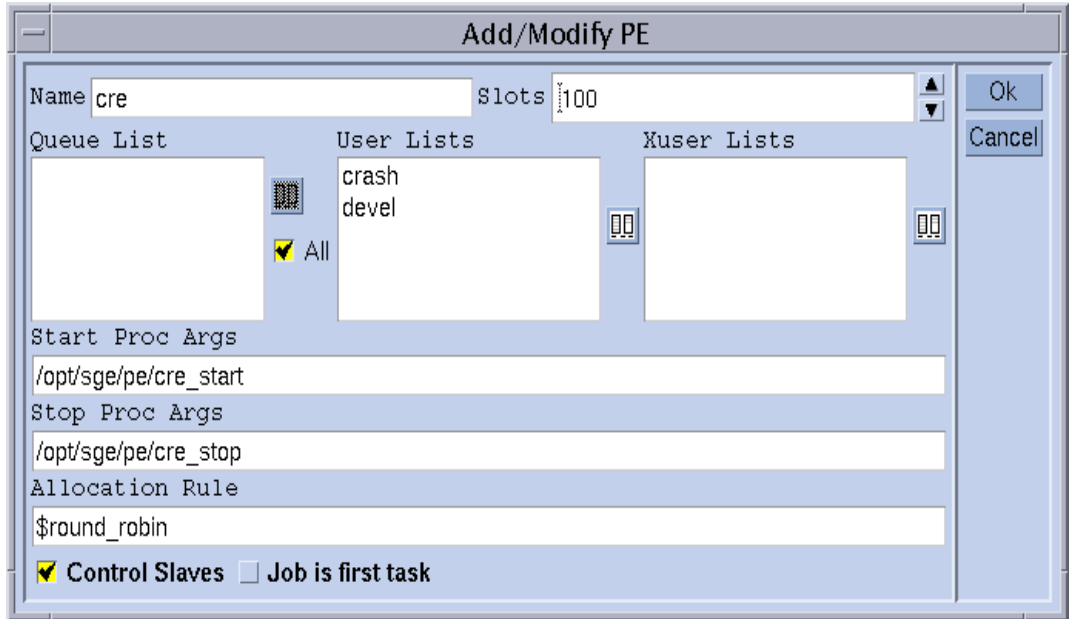


圖 10-2 平行環境定義對話方塊

2. 請遵照第 217 頁的「平行環境定義參數說明」一節指示新增 PE 定義。
3. 按一下「確定」以儲存變更，或「刪除」來捨棄變更。
按一下「確定」或「刪除」來關閉對話方塊。

平行環境定義參數說明

- 名稱輸入視窗會顯示修改作業中選取的 PE 名稱，或者也可自行輸入欲宣告之 PE 名稱。
- 插槽旋轉方塊必須用於輸入所有同時執行中 PE 作業會占據之全部作業插槽的數量。

- *佇列清單* 顯示區會顯示可供 PE 使用之佇列。按一下「佇列清單」顯示區（「選擇佇列」對話方塊）右方的圖示鈕就可修改 PE 佇列清單，如圖 10-3 範例所示。

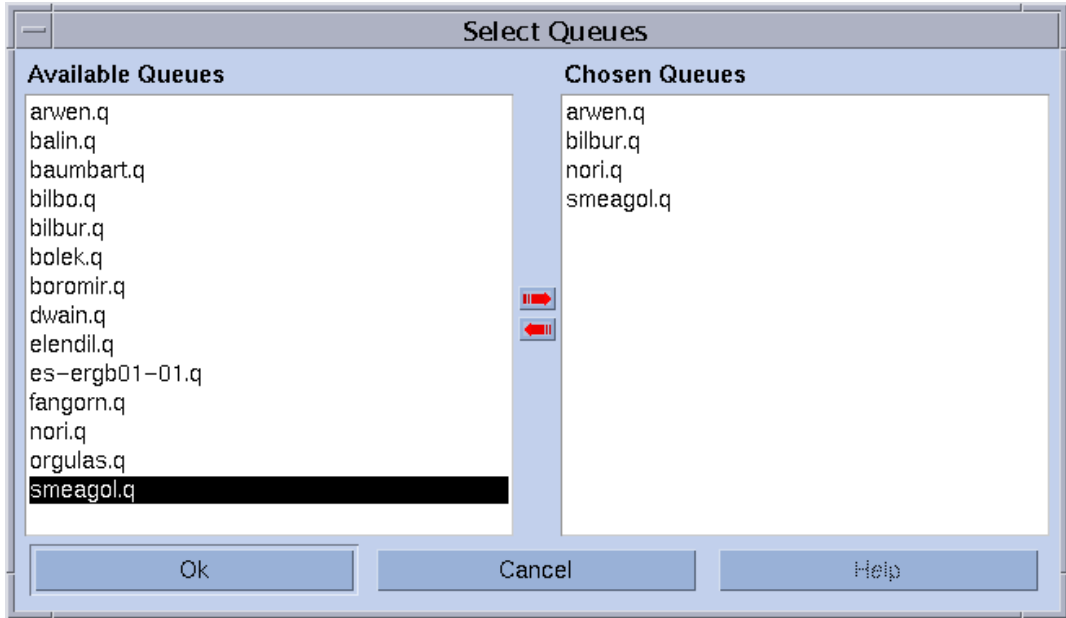


圖 10-3 佇列選擇對話方塊

- 「*使用者清單*」顯示區會顯示允許存取 PE 之使用者存取清單（請見第 195 頁的「關於使用者存取權限」一節）。
- 「*X 使用者清單*」顯示區會顯示被拒絕存取 PE 之存取清單。

按一下「選擇存取清單」對話方塊中兩個顯示區旁的圖示，如圖 10-4 所示。您可使用這些對話方塊來修改兩個存取清單顯示區的內容。

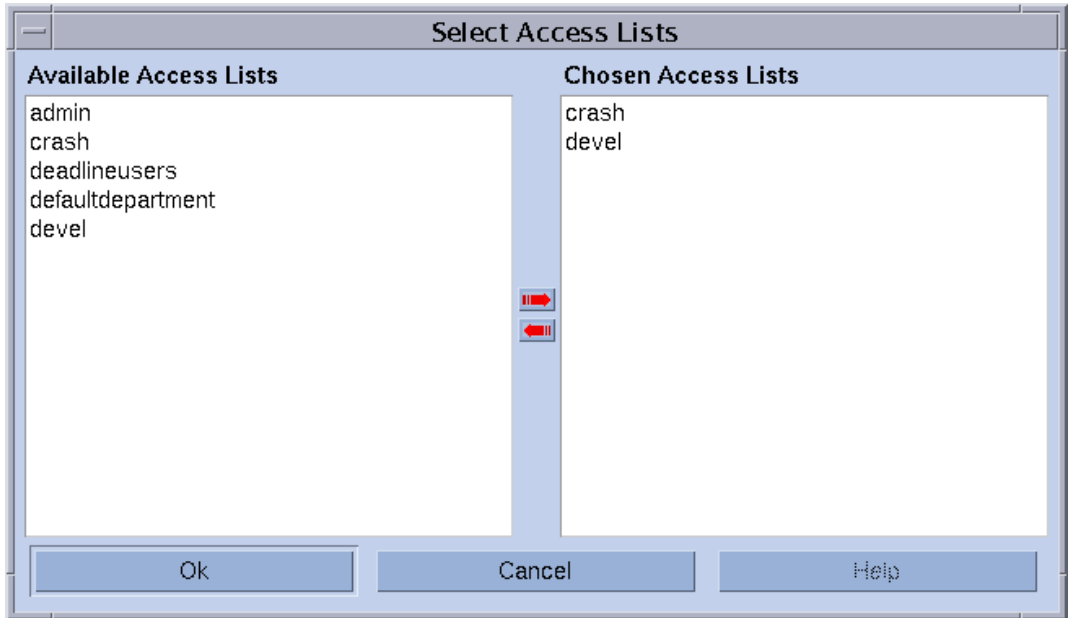


圖 10-4 選擇存取清單對話方塊

- 「*啟動程式引數 (Start Proc Args)*」與「*停止程式引數 (Stop Proc Args)*」輸入視窗可用來輸入 PE 啟動與停止程序之詳細啟用順序（請分別見第 222 頁的「PE 啟動程序」與第 223 頁的「終止 PE」二節）。第一個引數通常為啟動或停止程序本身。其他的參數則為程序的指令行引數。

有許多特別的辨識名稱（以 \$ 開頭），可用來傳遞 Sun Grid Engine 內部執行資訊至程序中。在《*Sun Grid Engine 5.3 和 Sun Grid Engine, 企業版 5.3 參考手冊*》中的 `sge_pe` 項目包含所有可用參數之清單。

- 「*配置規則*」輸入視窗可定義於 PE 使用之機器上分配的平行作業數量。現在僅有支援正整數與特殊數值 `$pe_slots`。`$pe_slots` 代表所有建立的作業必須位於單一主機上。
- 「*控制 Slave (Control Slaves)*」切換鈕會宣告平行作業是否為透過 Sun Grid Engine（即透過 `sge_execd` 與 `sge_shepherd`）而產生，或者對應 PE 是否有執行其作業。如果 Sun Grid Engine 系統可全權控制副 (slave) 作業（修正計算與資源控制）就會很有幫助，但此功能僅可用於 Sun Grid Engine 特殊設定之 PE 介面。請參閱第 223 頁的「PE 與 Sun Grid Engine 軟體之緊密整合」以得知更多詳細資訊。
- 「*作業為首要工作 (Job is first task)*」切換鈕僅在開啓「控制 Slave」時才有效用。這代表作業 script 或其子作業 (child processes) 之一為平行應用程式之平行作業（舉例來說，這通常發生在 PVM 之中）。如果關閉「控制 Slave」，作業 script 就會起始平行應用程式，但不會參與其中（例如：在 MPI 使用 `mpirun`）。

▼ 如何從指令行配置 PE

遵照以下幾節指示輸入 `qconf` 指令與適合之選項。

qconf PE 選項

- `qconf -ap pe 名稱`

新增平行環境 — 於佇列配置平台叫出編輯器（預設 `vi` 或對應至 `$EDITOR` 環境變數）。此參數 `pe 名稱` 會指定 PE 名稱，並已填妥對應平台區之資訊。變更平台並儲存至磁碟片中就完成配置 PE 的工作。請參見《*Sun Grid Engine 5.3 和 Sun Grid Engine, 企業版 5.3 參考手冊*》內的 `sge_pe` 項目以得知改變範本項目的詳細資訊。

- `qconf -dp pe 名稱`

刪除平行環境 — 此指令會刪除指定 PE。

- `qconf -mp pe 名稱`

修改平行環境 — 此指令會叫出編輯器（預設 `vi` 或對應至 `$EDITOR` 環境變數）與指定 PE，以為配置範本。變更範本並儲存至磁碟片中就完成修改 PE 的工作。請參見《*Sun Grid Engine 5.3 和 Sun Grid Engine, 企業版 5.3 參考手冊*》內的 `sge_pe` 項目以得知改變範本項目的詳細資訊。

- `qconf -sp pe 名稱`

顯示平行環境 — 本指令會列出指定之 PE 配置至標準輸出上。

- `qconf -spl`

顯示平行環境清單 — 本指令會顯示所有目前配置之平行環境名稱清單。

▼ 如何從指令行顯示配置的 PE 介面

- 輸入以下指令。

```
% qconf -spl
% qconf -sp pe 名稱
```

第一個指令會列出最新可供 PE 介面使用之名稱清單。第二個指令則會顯示特定 PE 介面的配置。請參閱 `sge_pe` 線上說明手冊以得知 PE 配置相關資訊。

▼ 如何以 QMON 顯示配置的 PE 介面

- 在 QMON 主要功能表上按下「PE 配置」鈕。

接著就會顯示「平行環境配置」對話方塊（請見第 216 頁的「如何以 QMON 配置 PE」一節）。

第 71 頁的「進階範例」一節的範例已定義要求 PE 介面之平行作業，mpi（為訊息傳遞介面所用），至少有四項過程，但最多可至（最好為）十六項過程。「平行環境規格 (PE)」視窗可用來開啓一對話方塊，從可用 PE 清單挑選想要的平行環境（請見圖 10-5）。由作業起始之平行工作的要求數量範圍可新增於「進階提交」螢幕之「PE 規格」視窗的 PE 名稱之後。

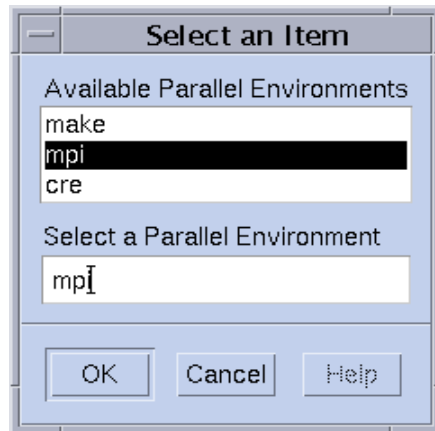


圖 10-5 PE 選擇

請參閱第 82 頁的「如何從指令行提交作業」一節得知對應上述平行作業規格之提令行提交指令，與如何使用 `qsub -pe` 選項來規劃一對等要求。請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `qsub` 項目得知更多 `-pe` 語法的詳細資訊。

於平行作業選擇合適的 PE 介面是相當重要的。PE 介面可以利用無或不同的訊息傳遞系統，於單一或多重主機上分配作業。也許會拒絕特定使用者存取至 PE。僅有特定佇列組可供 PE 介面使用、而 PE 介面也可隨時占據僅有特定佇列插槽數量。因此，您應詢問 Sun Grid Engine 管理以得知適合您平行作業類型使用之最佳 PE 介面。

您可於第 76 頁的「資源需求定義」一節得知如何指定資源需求與 PE 要求。這將會從符合您指定之資源要求的佇列中減少 PE 介面之合格佇列組。舉例來說，假設您已提交下述指令：

```
% qsub -pe mpi 1,2,4,8 -l nastran,arch=osf nastran.par
```

適合此作業之佇列為與 PE 配置之相關 PE 介面 `mpi`，並同時符合 `qsub -l` 選項指定之資源要求規格。

注意 – Sun Grid Engine PE 介面工具屬高配置性。特別是，Sun Grid Engine 管理可用來配置 PE 啟動與停止程序（請見 *sge_pe* 線上說明手冊）以支援站點特定需求。輸出環境變數之 `qsub -v` 與 `-V` 選項可用來將資訊從提交作業之使用者，傳遞到 PE 啟動與停止程序。如果您需要輸出特定環境變數，請詢問 Sun Grid Engine 管理員。

PE 啟動程序

Sun Grid Engine 系統透過 `exec` 系統呼叫，開始啟動程序並開啓 PE。啟動執行檔與傳遞至此執行檔之名稱可於 Sun Grid Engine 系統配置。請見 Sun Grid Engine 分配樹得知 PVM 環境之啟動程序範例。此樹由 shell script 與由 shell script 呼叫之 C 程式組成。shell script 使用 C 程式來乾淨地 (cleanly) 啟動 PVM。所有其他要求的作業都須由 shell script 處理。

shell script 位於 `<sge_root>/pvm/startpvm.sh` 底下，而 C 程式檔案則位於 `<sge_root>/pvm/src/start_pvm.c` 底下。

注意 – 啟動程序也可適用於單一 C 程式。shell script 可用以輕鬆自訂範例啟動程序。

範例 script，`startpvm.sh`，需要下述三個引數。

- 由 Sun Grid Engine 軟體產生之主機檔路徑，包括啟動 PVM 之主機名稱。
- 呼叫 `startpvm.sh` 程序之主機。
- PVM root 目錄的路徑（通常位於 `PVM_ROOT` 環境變數中）

請參閱第 216 頁的「如何以 QMON 配置 PE」內的方法得知可傳遞至啟動 script 之參數。這些參數為由 Sun Grid Engine 於執行期間提供至 PE 啟動與結束 script 之參數。舉例來說，要求之主機檔案是由 Sun Grid Engine 建立，而檔案名稱則會藉由此特殊參數名稱，`$sge_hostfile`，傳遞至 PE 配置之啟動程序中。請參閱《*Sun Grid Engine 5.3 和 Sun Grid Engine，企業版 5.3 參考手冊*》中的 `sge_pe` 項目以得知所有可用參數說明。

主機檔案擁有下述格式。

- 檔案每行代表平行作業欲執行之主機。
- 每行第一個項目為主機名稱。
- 第二筆項目指定於此主機執行之平行作業數量、
- 而第三筆項目則表示於多重處理器上使用之處理器範圍。

此檔案格式是由 Sun Grid Engine 所建立，而且格式已固定。使用不同的檔案格式之 PE（例如 PVM）需要於啟動程序中翻譯此格式（請見 `startpvm.sh` 檔）。

只要 Sun Grid Engine 系統開始啟動 PE 啟動程序，PE 也就隨之啟動。啟動程序應以零退出 (zero exit) 狀態退出。如果啟動程序的退出狀態不是零，Sun Grid Engine 軟體就會回報錯誤訊息，且將不會啟動平行作業。

注意 – 先從指令行測試任何啟動程序是有好處的 — 而非 Sun Grid Engine— 如果此程序已整合於 Sun Grid Engine 架構中，就可移除所有很難追蹤之錯誤。

終止 PE

當平行作業已結束或失敗（透過 `qdel`），就會呼叫中止平行環境之程序。本程序之定義與啟動程式中描述之定義相當類似。此停止程序也可於 PE 配置中定義（請見第 216 頁的「如何以 QMON 配置 PE」）。

此停止程序的目的是用來關閉 PE，然後得到所有相關作業。

注意 – 如果停止程序無法清除 PE 作業，Sun Grid Engine 系統也許是沒有執行於 PE 控制底下之作業資訊，因而導致無法清除。Sun Grid Engine 軟體當然會清除作業 script 執行之直接相關的作業。

Sun Grid Engine 分配樹同時也包含 PVM PE 停止程序之範例，就位於 `<sge_root>/pvm/stoppvm.sh` 底下。共有下述兩個引數。

- 由 Sun Grid Engine 建立之主機檔路徑
- 啟動停止程序之主機名稱

與啟動程序相似，停止程序之退出狀態需為零才算成功，非零退出狀態則為失敗。

注意 – 先從指令行測試任何停止程序是有好處的 — 而非 Sun Grid Engine— 如果此程序已整合於 Sun Grid Engine 架構中，就可移除所有很難追蹤之錯誤。

PE 與 Sun Grid Engine 軟體之緊密整合

於第 216 頁的「如何以 QMON 配置 PE」一節中的「控制副」參數說明有提及由 Sun Grid Engine 元件 `sge_execd` 與 `sge_shepherd` 執行之平行作業創作的 PE 會有益於執行其作業創作之 PE。這是因為 UNIX 作業系統僅允許執行階層之建立者做可靠資源控制，而平行應用程式的功能，如修正計算、資源限制與過程控制只能由所有平行作業之建立者執行。

大多數 PE 不會執行這些功能，因此也就不會提供資源管理系統，如 Sun Grid Engine，整合所需之有效介面。要克服這個問題，Sun Grid Engine 系統有提供緊密整合 PE 之進階 PE 介面，可將作業建立之責任由 PE 傳遞至 Sun Grid Engine 軟體。

Sun Grid Engine 分配包含二個緊密整合範例，一為 PVM 公共領域版，一為由 Argonne National Laboratories 執行之 MPICH MPI。這兩個範例分別位於 `<sg_e_root>/pvm` 與 `<sg_e_root>/mpi` 兩個目錄中。這兩個目錄包含鬆散的介面整合變數，與說明使用與最新限制之 README 檔以供比較。請參閱 README 檔得知更多詳細資訊。

注意 – 執行 PE 緊密整合為一進階作業，所以需要擁有 PE 專業知識與 Sun Grid Engine PE 介面。您可聯絡您的 Sun 支援代表商以獲得協助。

錯誤訊息

本章說明 Sun Grid Engine 5.3 錯誤訊息程序。使用本資訊來幫助您疑難排解使用本軟體會遇到的問題。

Sun Grid Engine 5.3 軟體如何取得錯誤回報

Sun Grid Engine 軟體會藉由特定檔案之登入訊息，與 / 或電子郵件回報錯誤訊息或警告。使用之日誌檔案為：

- 訊息檔案：

`sge_qmaster`、`sge_schedd` 與 `sge_execd` 有許多分隔之訊息檔案。這些檔案都有相同的檔名，`messages`。`sge_qmaster` 日誌檔案位於主多工緩衝處理目錄中，而排程器多工緩衝處理目錄內的 `sge_schedd` 訊息檔案與執行常駐程式的日誌檔案都位於執行常駐程式之多工緩衝處理目錄中（請見第 21 頁的「`root` 目錄下的多工緩衝處理目錄」得知更多多工緩衝處理目錄資訊）。

訊息檔案格式如下：

- 每個訊息皆占據一行。
- 訊息是由一垂直線記號 (|) 分隔成五個部分。
- 第一部分為訊息的時間戳記、
- 第二部分為產生訊息之 Sun Grid Engine 常駐程式、
- 第三部分為常駐程式執行之主機名稱、
- 第四部分則為訊息種類，分為通知用之 N、資訊用之 I（上述兩者皆僅為提供資訊用）、警告用之 W（有地方出問題）、錯誤用之 E（已偵測到錯誤情況）、與緊急用之 C（可能會導致程式執行失敗）。
- 第五個則為訊息文字。

注意 – 若基於某些原因導致錯誤日誌檔案無法存取，Sun Grid Engine 將會試著把錯誤訊息記錄至對應主機的 /tmp/sge_qmaster_messages、
/tmp/sge_schedd_messages 或 /tmp/sge_execd_messages 檔中。

- 作業 STDERR 輸出：

只要啓動作業，就會將作業 script 之標準錯誤 (STDERR) 輸出重新導回至檔案。檔案名稱與位址不是符合預設，便是由特定 qsub 指令行開關指定。請參閱《*Sun Grid Engine 5.3 管理和使用者指南*》與《*Sun Grid Engine 5.3 和 Sun Grid Engine 企業版 5.3 參考手冊*》取得詳細資訊。

在某些情況下，Sun Grid Engine 會透過電子郵件通知使用者與 / 或管理員發生錯誤事件。由 Sun Grid Engine 寄送之郵件訊息並不包含訊息本文。訊息文字全部包含在郵件主旨區中。

不同錯誤或退出代碼之結果

表 11-1 列出不同作業相關之錯誤與退出代碼的結果。這些代碼適用於所有 Sun Grid Engine 作業類型。

表 11-1 作業相關錯誤或退出代碼

Script/ 方法	退出或錯誤代碼	結果
作業 script	0	成功
	99	重新排於佇列中
	其餘	成功：於計算檔案之退出代碼
prolog/epilog	0	成功
	99	重新排於佇列中
	其餘	佇列錯誤狀態：作業重新排於佇列中

表 11-2 列出與平行環境 (PE) 配置相關之作業錯誤或退出代碼的結果。

表 11-2 PE 相關之錯誤或退出代碼

Script/ 方法	退出或錯誤代碼	結果
pe 啓動	0	成功
	其餘	佇列設定至錯誤狀態，作業重新排於佇列中
pe 停止	0	成功
	其餘	佇列設定至錯誤狀態，作業無重新排於佇列中

表 11-3 列出與佇列配置相關之作業錯誤或退出代碼的結果。僅當覆寫對應方法時上述結果才會成立。

表 11-3 佇列相關錯誤或退出代碼

Script/ 方法	退出或錯誤代碼	結果
作業啓動器	0	成功
	其餘	成功，無其他特殊意義
暫停	0	成功
	其餘	成功，無其他特殊意義
恢復	0	成功
	其餘	成功，無其他特殊意義
終止	0	成功
	其餘	成功，無其他特殊意義

表 11-4 列出與檢查點相關之作業錯誤或退出代碼的結果。

表 11-4 檢查點相關之錯誤或退出代碼

Script/ 方法	退出或錯誤代碼	結果
檢查點	0	成功
	其餘	成功 — 但對核心檢查點有特殊意義：檢查點不成功；沒有發生。
遷移	0	成功
	其餘	成功 — 但對核心檢查點有特殊意義：檢查點不成功；沒有發生。將會進行遷移。
重新啓動	0	成功
	其餘	成功，無其他特殊意義
清除	0	成功
	其餘	成功，無其他特殊意義

於除錯模式 (Debug Mode) 中執行 Sun Grid Engine 程式

而針對一些嚴重的錯誤事件，錯誤記錄機制也許不會有足夠的資訊來辨識這些問題。因此，Sun Grid Engine 提供幾乎所有輔助程式與常駐程式皆可執行於除錯模式中的功能。於提供之資訊範圍與深度中有不同的除錯等級。除錯等級範圍由 0 到 10，等級 10 傳達的訊息是最詳細的，而等級 0 則會關閉除錯。

於您的 `.cshrc` 或 `.profile` 資源檔案中設定一除錯等級，並提供 Sun Grid Engine 分配。針對 `csh` 或 `tcsh` 的使用者，就會包含此檔 `<sge_root>/util/dl.csh`。針對 `sh` 或 `ksh` 使用者，對應檔案就會命名為 `<sge_root>/util/dl.sh`。這些檔案必須 "sourced" 進您的標準資源檔案。而 `csh` 或 `tcsh` 的使用者，請包含下述行：

```
source <sge_root>/util/dl.csh
```


進您的 `.cshrc` 檔中。sh 或 ksh 使用者，新增下述行：

```
. <sgc_root>/util/dl.sh
```

進您的 `.profile` 檔是相同的。只要您現在登出再登入，您就可以使用下述指令來設定一除錯等級等級：

```
% dl level
```

如果等級大於 0，隨後啓動 Sun Grid Engine 指令將迫使指令將追蹤輸出寫入 `STDOUT`。依據執行之除錯等級，追蹤輸出也許會內含警告、狀態、錯誤訊息、內部呼叫之程式模組名稱、與資源代碼資訊（有助於錯誤回報）。

注意 – 於視窗中以大量卷軸行緩衝（例：1000 行）來監控除錯追蹤也許會有用。

注意 – 如果您的視窗為 `xterm`，之後您也許會想要使用 `xterm` 記錄機制來檢查追蹤輸出。

執行除錯模式裡 Sun Grid Engine 常駐程式其中之一就會得到常駐程式持續終端連結至寫入追蹤輸出的結果。只要輸入中斷終端模擬之字元就可中止常駐程式。（即 `Control-C`）。

注意 – 要關閉除錯程式，請將除錯等級設為 0。
