



Solaris 一般桌上管理系統環境：進 階使用者和系統管理員指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

元件號碼: 816-4023-10
2002 年 5 月

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

本產品或文件受到版權保護，並在授權限制其使用、複製、分配以及反編譯的情況下分配。未經 Sun 及其授權者（如果有的話）的書面授權，本產品或文件的任何部份皆不得以任何形式、任何方法重新製造。協力廠商的軟體，包括字型技術在內，都受版權的保護，並有來自 SUN 的供應商的授權。

產品的某些部份可能源自 Berkeley BSD 系統，其授權來自加州大學。UNIX 在美國和其它國家都是已登記註冊商標，透過 X/Open 公司獲得獨家授權。

Sun、Sun Microsystems、Sun 的標誌、docs.sun.com、AnswerBook、AnswerBook2，以及 Solaris 都是 Sun Microsystems, Inc 在美國及其它國家的商標、註冊商標或服務標記。所有的 SPARC 商標都是在獲得授權的情況下使用，而且是美國和其它國家 SPARC International, Inc. 的商標和註冊商標。有 SPARC 商標的產品都是以 Sun Microsystems, Inc. DtComboBox 與 DtSpinBox widgets 的代碼與文件是來自於 Interleaf, Inc. 版權所有 1993 年，Interleaf, Inc. 的基本架構所開發。

OPEN LOOK 及 Sun™ 圖形使用者介面是 Sun Microsystems, Inc. 為其使用者和獲得授權者設計出來的。Sun 承認 Xerox 在研究和設計電腦業中視覺化或圖形使用者介面這個觀念上所做的領先努力。Sun 保有 Xerox 對 Xerox 圖形使用者介面非獨佔性的授權，這項授權也涵蓋獲得 Sun 授權使用 OPEN LOOK GUI，或者符合 Sun 的書面軟體授權合約的廠商。

有限權利：由美國政府使用、複製或公佈主要受 FAR 52.227-14(g)(2)(6/87) 及 FAR 52.227-19(6/87) 或 DFAR 252.227-7015(b)(6/95) 及 DFAR 227.7202-3(a) 的限制。

文件以“現狀”提供，所有明示或暗示的條件、陳述或保證，都恕不負責，包括對特定用途的銷售性、適用性或者非侵權行為的任何隱含保證在內，除非這種聲明在法律上被認為是無效的。

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la d'écompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. Le code et la documentation pour les produits DtComboBox et DtSpinBox ont été fournis par Interleaf, Inc. Copyright 1993, Interleaf, Inc

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



020408@3688



內容

前言 15

1 配置登入管理者	19
啟動登入伺服器	20
管理本機與網路顯示器	20
尋找登入伺服器程序 ID	21
在本機顯示器上顯示登入螢幕	22
執行無本機顯示器的登入伺服器	22
存取本機顯示器上的指令行登入	22
使用字元顯示主控台	23
在網路顯示器上顯示登入螢幕	23
控制登入伺服器的存取	24
檢查錯誤	26
停止登入伺服器	26
登入螢幕	27
變更登入螢幕外觀	28
▼ 如果要變更標識圖案	28
▼ 變更歡迎訊息	28
▼ 變更字型	29
▼ 提供替代文字以顯示每種語言	29
變更每個顯示器的登入螢幕行為	30
變更 X 伺服器存取	31
在登入螢幕出現之前發出指令	32
啟動單視窗階段作業	32
在使用者的階段作業結束之後	33
登入伺服器環境	33

變更使用者或系統路徑	33
▼ 如果要變更系統 Shell	34
▼ 如果要變更時區	34
管理登入管理者	34
登入管理者檔案	35
2 配置階段作業管理者	37
什麼是階段作業？	37
初始階段作業	38
目前階段作業	38
主要階段作業	38
顯示器的特定階段作業	38
啓動階段作業	38
當階段作業啓動時	39
找出 .dtprofile 程序檔的來源	39
找出 Xsession.d 程序檔的來源	40
顯示歡迎訊息	40
設定桌上管理系統搜尋路徑	41
收集可供使用的應用程式	41
選擇性找出 .profile 或 .login 程序檔的來源	42
啓動 ToolTalk 傳訊常駐程式	42
啓動階段作業管理者用戶端	42
載入階段作業資源	42
啓動色彩伺服器	43
啓動工作區管理者	44
啓動階段作業應用程式	44
其它的階段作業啓動自訂	45
▼ 設定環境變數	45
▼ 設定資源	45
▼ 設定顯示器特定的資源	46
▼ 變更起始階段作業的應用程式	46
▼ 設定特定顯示器的階段作業	47
在階段作業啓動與登出時執行其它指令	47
▼ 在階段作業啓動時執行其它的指令	47
▼ 在登出時執行其它指令	48
▼ 從備份回復階段作業	48
▼ 調查階段作業啓動問題	48

階段作業管理者檔案及目錄 49

- 3 疑難排解登入與階段作業啟動的問題 51
 - 登入啟動檔案 51
 - 錯誤登錄位置 52
 - 使用者啟動檔案 52
 - Solaris CDE 啟動範例 53

- 4 增加與管理應用程式 55
 - 應用程式管理者的結構 55
 - 應用程式管理者的目錄位置 56
 - 應用程式管理者如何尋找與收集應用程式 56
 - 收集應用程式時的優先順序規則 57
 - 預設桌上管理系統所提供的應用程式群組 57
 - 如何收集應用程式群組的範例 57
 - 增加應用程式至應用程式管理者 58
 - 增加應用程式至應用程式管理者的方法 59
 - ▼ 增加智慧型桌上管理系統應用程式至應用程式管理者 60
 - ▼ 如果要註冊現存或非桌上管理系統的智慧型應用程式 60
 - ▼ 增加應用程式圖示至現存應用程式群組 60
 - 建立及管理一般應用程式群組 61
 - ▼ 建立整個系統的一般應用程式群組 61
 - ▼ 建立個人的一般應用程式群組 61
 - ▼ 自訂內建的應用程式群組 61
 - 修改用於尋找應用程式的搜尋路徑 62
 - 預設搜尋路徑 62
 - 增加應用程式伺服器至應用程式搜尋路徑 62
 - 一般應用程式管理者管理 63
 - ▼ 移除應用程式 63
 - ▼ 在階段作業期間更新應用程式管理者 63
 - 變更文字編輯程式與終端機模擬器 64
 - ▼ 如果要變更預設文字編輯程式或終端機模擬器 64

- 5 註冊應用程式 67
 - 應用程式註冊的概述 68
 - 應用程式註冊提供的功能 68
 - 應用程式註冊的目的 70

註冊應用程式的一般步驟	70
第 1 步：修改字型與色彩資源	71
第 2 步：建立桌上管理系統應用程式 Root	72
第 3 步：建立註冊套裝軟體目錄	73
第 4 步：建立應用程式的動作與資料類型	75
第 5 步：將輔助說明檔案置於註冊套裝軟體中	77
第 6 步：建立應用程式的圖示	78
第 7 步：建立應用程式群組	79
第 8 步：使用 dtappintegrate 註冊應用程式	84
建立註冊套裝軟體的範例	86
您需要知道的有關 "BestTextEditor" 的資訊	86
註冊 "BestTextEditor" 的步驟	87
6 其它配置	93
Solaris CDE 目錄結構	93
/usr/dt	93
/etc/dt	94
/var/dt	94
\$HomeDirectory	94
主要配置檔案	95
Xconfig	95
Xservers	95
啓動登入伺服器	96
從其它的工作站或網路伺服器安裝位置來裝載已安裝的 CDE	98
▼ 裝載已安裝的 CDE	98
▼ 卸載已裝載的 CDE 目錄	98
配置您的桌上管理系統以使用多重螢幕	99
▼ 使桌上管理系統在多重螢幕上啓動	99
網路的桌上管理系統	100
使用 XTerminals	102
登入語言環境與字型路徑	103
使用工作站作為 XTerminals	103
▼ 如果要使用選擇程式以選擇主機 CDE 登入	104
▼ 如果要使用特定的主機 CDE 登入	104
▼ 如果要使用第一可供使用的主機登入	104
特殊 CDE 配置	105
自訂郵件列印	105

	轉換日曆為新的資料格式	106
	從網路增加 AnswerBook 套裝軟體	106
	在 CDE 桌上管理系統之外設定 CDE 環境	106
	桌面環境檔案	107
	使用具有 Apple Macintosh 應用程式環境的磁片與 CD 媒體	107
7	在網路中配置桌上管理系統	109
	桌上管理系統網路的概述	109
	連上網路的桌上管理系統服務類型	110
	典型的網路情況	110
	其它的網路情況	113
	摘要—伺服器的類型	114
	配置桌上管理系統網路的一般步驟	115
	配置桌上管理系統的基本作業系統網路	115
	提供登入帳戶給使用者	115
	配置分散式檔案系統存取	116
	配置遠端印表機的存取	117
	配置電子郵件	117
	配置 X 授權	117
	配置桌上管理系統用戶端與伺服器	118
	配置登入與階段作業服務	118
	配置其它與應用程式有關的服務	118
	管理應用程式服務	122
	搜尋路徑環境變數	122
	配置應用程式伺服器及其用戶端	122
	配置資料庫、圖示及輔助說明服務	124
	特殊網路的應用程式配置	125
8	從桌上管理系統配置與管理列印	129
	增加與刪除印表機	129
	▼ 增加印表機到桌上管理系統	129
	▼ 從桌上管理系統刪除印表機	130
	修改工作更新間隔	130
	印表機圖示影像	130
	圖示檔案名稱與大小	131
	▼ 如果要整體變更圖示、印表機標籤或印表機的說明	131
	配置預設印表機	132

▼ 變更預設列印的目標	132
列印觀念	133
9 桌上管理系統搜尋路徑	135
桌上管理系統搜尋路徑與其環境變數	136
設定搜尋路徑的值	136
▼ 查看搜尋路徑目前的值（輸出變數）	136
▼ 如果要作搜尋路徑的個人修改	137
▼ 如果要作整個系統的搜尋路徑修改	137
應用程式搜尋路徑	137
預設應用程式搜尋路徑	138
應用程式搜尋路徑環境變數	138
應用程式搜尋路徑輸入變數的語法	138
如何組合應用程式搜尋路徑的值	138
變更整個系統的本機位置優先順序	139
應用程式搜尋路徑如何影響資料庫、圖示與輔助說明搜尋路徑	139
資料庫（動作/資料類型）搜尋路徑	140
預設資料庫搜尋路徑	140
應用程式搜尋路徑如何影響資料庫搜尋路徑	141
資料庫搜尋路徑環境變數	141
資料庫搜尋路徑輸入變數的語法	141
如何組合資料庫搜尋路徑	141
圖示搜尋路徑	142
預設圖示搜尋路徑	142
應用程式搜尋路徑如何影響圖示搜尋路徑	142
圖示搜尋路徑環境變數	142
圖示搜尋路徑輸入變數的語法	143
如何組合圖示搜尋路徑	143
輔助說明搜尋路徑	144
預設輔助說明搜尋路徑	144
應用程式搜尋路徑如何影響輔助說明搜尋路徑	144
輔助說明搜尋路徑環境變數	144
輔助說明搜尋路徑輸入變數的語法	144
如何組合輔助說明搜尋路徑	145
本土化搜尋路徑	145

10	動作與資料類型簡介	147
	動作的介紹	147
	動作如何建立應用程式的圖示	149
	動作如何使用資料檔案作為引數	151
	動作的其他用途	152
	資料類型的介紹	152
	什麼是資料類型？	152
	資料類型如何將資料檔案連接至動作	153
	建立資料類型的桌上管理系統列印	155
11	使用建立動作來建立動作與資料類型	157
	建立動作會做什麼	157
	建立動作的限制	158
	動作限制	158
	資料類型限制	158
	以建立動作來建立應用程式的動作及資料類型	159
	▼ 如果要建立應用程式的動作	159
	▼ 建立一個或以上的應用程式資料類型	161
	使用尋找集對話方塊來指定一個圖示	166
12	人工建立動作	169
	您必須人工建立動作的理由	169
	COMMAND 動作	170
	MAP 動作	170
	TT_MSG (ToolTalk 訊息) 動作	170
	人工建立動作：一般步驟	171
	動作的配置檔案	171
	▼ 人工建立動作	171
	建立 COMMAND 動作的範例	172
	建立 MAP 動作的範例	173
	▼ 重新載入動作/資料類型資料庫	173
	建立動作的動作檔案 (圖示)	174
	指定由動作使用的圖示影像	175
	▼ 修改現存動作定義	176
	動作定義中的優先順序	177
	建立 COMMAND 動作的執行字串	178
	執行字串的一般功能	178

建立不使用引數的動作	179
建立接受定位檔案的動作	179
建立一個提示檔案引數的動作	180
建立接受定位檔案或提示符號的動作	180
建立一個提示非檔案引數的動作	181
解譯檔案引數為字串	181
在動作中提供 Shell 能力	181
建立多重檔案引數的 COMMAND 動作	182
COMMAND 動作的視窗支援與終端機模擬器	184
指定動作的視窗支援	184
指定終端機模擬器的指令行選項	184
指定不同的預設終端機模擬器	185
限制動作以確定引數	185
為指定的資料類型限制動作	186
以引數數目為基礎限制動作	186
▼ 提供不同的連按兩下與定位行為	186
以引數模式為基礎限制動作	187
建立在遠端系統上執行應用程式的動作	187
建立執行遠端系統應用程式的動作	188
使用動作與資料類型定義中的變數	188
使用動作中的字串變數	189
使用動作與資料類型中的環境變數	189
從指令行呼叫動作	190
語法 dtaction	190
建立執行其它動作的動作	190
建立以不同使用者執行的動作	190
建立本土化動作	191
本土化動作的位置	191
▼ 將現存動作本土化	191
建立 ToolTalk 應用程式的動作	192
定址與配置欄位	192
不支援的訊息	192

13 人工建立資料類型 195

您必須人工建立資料類型的理由	195
資料類型定義的元件：條件與屬性	196
人工建立資料類型：一般步驟	196

	資料類型的配置檔案	196
	▼ 建立資料類型定義	197
	建立個人動作與資料類型的範例	198
	定義資料類型的資料屬性	199
	指定用於資料類型的圖示影像	199
	使資料類型與動作產生關聯	199
	依據資料類型隱藏檔案	200
	處理檔案時，指定行為	200
	定義資料類型的資料條件	201
	名稱型資料類型	202
	位置型資料類型	202
	以名稱與位置為基礎的資料類型	203
	使用檔案模式作為類型條件	204
	內容型資料類型	205
	▼ 建立具有數項獨立條件的資料類型	206
	建立本土化資料類型	206
	本土化資料類型的位置	207
	▼ 如果要本土化資料類型	207
14	建立桌上管理系統的圖示	209
	圖示影像檔案	209
	圖示檔案格式	210
	圖示檔案名稱	210
	圖示大小慣例	210
	圖示搜尋路徑	211
	經由網路存取圖示	211
	圖示關聯	211
	指定圖示檔案	212
	▼ 如果要使圖示與動作或資料類型關聯	212
	▼ 如果要顯示前方面板控制中的圖示	213
	▼ 如果要關聯圖示與應用程式視窗	213
	▼ 使用檔案管理者作為圖示瀏覽器	214
	圖示設計建議	214
	色彩用法	214
15	進階前方面板自訂	217
	前方面板配置檔案	217

預設前方面板配置檔案	218
前方面板配置檔案的搜尋路徑	218
如何組合前方面板：優先順序規則	218
動態建立的前方面板檔案	219
管理使用者介面自訂	219
▼ 防止個人自訂	220
▼ 回復已刪除的控制或輔助面板	220
前方面板定義的組織	220
前方面板元件	220
前方面板定義的一般語法	221
修改主面板	223
▼ 增加控制至主面板	224
▼ 移除控制	225
▼ 修改控制	225
▼ 交換控制的位置	226
▼ 取代前方面板控制	226
指定控制所使用的圖示	227
建立及修改輔助面板	228
▼ 建立新的整個系統的輔助面板	228
自訂內建的輔助面板	229
▼ 變更輔助面板的自動關閉行為	230
定義前方面板控制	231
前方面板控制定義	231
控制類型	232
▼ 建立新的控制	232
自訂工作區切換	237
▼ 如果要變更預設的工作區數目	238
▼ 變更切換列的數目	238
▼ 在工作區切換中變更或增加控制	238
一般前方面板配置	239
一般步驟	239
▼ 變更預設前方面板位置	239
▼ 標示主要面板中的控制	240
▼ 變更控制的按一下行為	240
▼ 建立全新的前方面板	240
建立具有三列的個人前方面板的範例	240

- 16 自訂工作區管理者 243
 - 工作區管理者配置檔案 244
 - ▼ 建立或修改個人的配置檔案 244
 - ▼ 建立整個系統的配置檔案 244
 - ▼ 包含 (併入資源) 其他檔案 245
 - ▼ 重新啟動工作區管理者 245
 - 自訂工作區 246
 - ▼ 以整個系統的基礎來變更工作區的數目 246
 - ▼ 提供整個系統的工作區名稱 246
 - ▼ 建立其他背景定位 247
 - ▼ 用圖形影像取代背景定位 247
 - 工作區管理者功能表 248
 - 工作區管理者功能表語法 248
 - ▼ 新增功能表項目到工作區功能表 249
 - ▼ 修改工作區功能表 249
 - ▼ 建立新的工作區 (根) 功能表 250
 - ▼ 建立新的視窗功能表 251
 - 自訂按鈕組合 252
 - 按鈕組合語法 252
 - ▼ 增加按鈕組合 253
 - ▼ 建立新的按鈕組合集 253
 - 自訂按鍵組合 254
 - 預設桌上管理系統按鍵組合 254
 - 按鍵組合語法 254
 - ▼ 建立自訂的按鍵組合集 255
 - 在預設與自訂行為之間切換 255

- 17 管理應用程式資源、字型與色彩 257
 - 設定應用程式資源 257
 - ▼ 設定整個系統的資源 257
 - ▼ 設定個人資源 258
 - 桌上管理系統如何載入資源 258
 - 處理管理者資源 258
 - 定義 UNIX 連結 258
 - ▼ 指定 EMACS 樣式轉換 258
 - ▼ 如果要修改 EMACS 樣式轉換 259
 - 由 UNIXbindings 檔案所提供的 UNIX 連結 259

管理字型	262
設定桌上管理系統字型資源	263
▼ 列出可用字型	264
▼ 如果要指定指令行上的字型	264
X 邏輯字型說明 (XLFD)	264
使用者的字型群組檔案系統儲存	266
系統管理者字型群組建立	266
管理色彩	267
色彩調色板	267
色彩集	267
以樣式管理者控制色彩	270
由樣式管理者所使用的色彩數目	271
設定應用程式視窗的陰影濃度	274
18 配置本土化桌面階段作業	275
管理 LANG 環境變數	275
設定多個使用者的語言	276
設定一個階段作業的語言	276
設定一個使用者的語言	277
LANG 環境變數與階段作業配置	277
設定其他 NLS 環境變數	277
尋找字型	278
將 app- 預設資源檔案本土化	278
本土化動作與資料類型	278
將圖示及點陣圖本土化	279
將背景定位名稱本土化	279
將調色板名稱本土化	280
將輔助說明卷次本土化	280
將訊息目錄本土化	281
在遠端執行本土化桌上管理系統應用程式	281
重設您的鍵盤對映	281
A dtconfig(1) Man Page	283
索引	285

前言

本手冊涵蓋自訂 Solaris 一般桌上管理系統環境（CDE）的外觀與行為的進階工作。它包括下列幾章：

- 自訂系統初始化、登入及階段作業起始
- 增加應用程式並提供應用程式及其資料的介面表示法
- 跨網路配置桌上管理系統程序、應用程式及資料
- 自訂桌上管理系統，如視窗管理、列印、色彩及字型

註解 – 在本文件中，名詞“IA”代表 Intel 32 位元處理器架構，它包括 Pentium，Pentium Pro，Pentium II，Pentium II Xeon，Celeron，Pentium III 及 Pentium III Xeon 處理器，以及 AMD 與 Cyrix 製造的相容微處理器晶片。

本書適用對象

本書的讀者包括：

- 系統管理員。本書中有許多工作需要根許可權。
- 想要執行自訂但無法使用桌上管理系統使用者介面來完成的進階使用者。桌上管理系統提供使用者專用的位置，存放它的許多配置檔案。

開始閱讀本書之前

使用者應該熟悉下列書籍：

- *Solaris* 一般桌上管理系統環境：使用者指南
- *Solaris* 一般桌上管理系統環境：使用者轉換指南

本書的編排方式

本手冊包括下列幾章：

第 1 章 涵蓋如何配置桌上管理系統登入管理者的外觀及行為。

第 2 章 涵蓋桌上管理系統如何儲存及取回階段作業，以及如何自訂階段作業啟動。

第 3 章 描述 *Solaris* CDE 啟動檔案、可能的 *Solaris* CDE 啟動問題，以及啟動問題的建議解決方案。

第 4 章 涵蓋應用程式管理員如何收集應用程式，以及解釋如何新增應用程式。

第 5 章 涵蓋如何建立應用程式的註冊套裝軟體。

第 6 章 描述進階的配置主題，如自訂登入配置、設定多重螢幕、上網的桌面管理系統及 X 終端機、修改使用者點檔案、郵件列印自訂、桌面環境設定，以錯誤登錄的類型。

第 7 章 涵蓋如何跨網路分配桌上管理系統服務、應用程式及資料。

第 8 章 涵蓋如何新增及移除桌上管理系統印表機，以及如何指定預設印表機。

第 9 章 涵蓋桌上管理系統如何跨網路尋找應用程式、輔助說明檔案、圖示及其他桌上管理系統資料。

第 10 章 介紹動作及資料類型的概念，以及解釋如何使用它們來提供應用程式的使用者介面。

第 11 章 涵蓋如何使用建立動作應用程式來建立動作及資料類型。

第 12 章 涵蓋如何經由編輯資料庫配置檔案來建立動作定義。

第 13 章 涵蓋如何經由編輯資料庫配置檔案來建立資料類型定義。

第 14 章 涵蓋如何使用圖示編輯程式，以及桌上管理系統圖示的命名慣例、大小及搜尋路徑。

第 15 章 涵蓋如何建立新的整個系統控制及輔助面板，以及其他面板自訂。

第 16 章 涵蓋如何自訂視窗、滑鼠按鈕組合、鍵盤連結及工作區管理員功能表。

第 17 章 涵蓋如何設定應用程式資源，以及桌上管理系統如何使用字型及色彩。

第 18 章 涵蓋執行國際階段作業的系統的系統管理工作。

附錄 A 是 dtconfig(1) 線上援助頁的副本。

訂購 Sun 文件

Fatbrain.com 是網際網路上的專業書店，存有來自 Sun Microsystems 公司的產品文件。

若要取得文件清單及如何訂購它們的資訊，請造訪 Fatbrain.com 上的 Sun Documentation Center：<http://www1.fatbrain.com/documentation/sun>。

線上存取 Sun 文件

docs.sun.comSM Web 網站可讓您線上存取 Sun 技術文件。您可以瀏覽 docs.sun.com 歸檔或搜尋特定書名或主題。URL 為 <http://docs.sun.com>。

印刷慣例

下表描述本書使用之印刷慣例。

表 P-1 印刷慣例

字體或符號	意義	範例
AaBbCc123	指令、檔案及目錄的名稱；螢幕上的電腦輸出	編輯您的 .login 檔案。 使用 <code>ls -a</code> 列出所有檔案。 machine_name% you have mail.

表 P-1 印刷慣例 (繼續)

字體或符號	意義	範例
AaBbCc123	當您鍵入時，請對照螢幕上的電腦輸出	machine_name% su Password:
<i>AaBbCc123</i>	指令行取代符號；換成真正名稱或值	若要刪除檔案，請鍵入 rm 檔案名稱。
<i>AaBbCc123</i>	書名、新詞、名詞或要強調的字	請閱讀 使用者指南 中的第 6 章。 這些稱為 類別 選項。 您必須是 <i>root</i> 使用者方可執行這個動作。

Shell 提示符號範例

下表顯示 C shell、Bourne shell 及 Korn shell 的預設系統提示符號與超級使用者提示符號。

表 P-2 Shell 提示

Shell	提示
C shell 提示	machine_name%
C shell 超級使用者提示	machine_name#
Bourne shell 及 Korn shell 提示	\$
Bourne shell 及 Korn shell 超級使用者提示	#

第 1 章

配置登入管理者

登入管理者是一個負責顯示登入螢幕、認證使用者以及啟動使用者階段作業的伺服器。與傳統的字元模式登入相比，圖形化的登入螢幕是一項有吸引力的替代。由登入伺服器所管理的顯示器可以直接附加至登入伺服器，或附加至網路上的 X 終端機或工作站。

註解 – 您必須是 root 使用者才能啟動、停止或自訂登入伺服器。

- 第 20 頁“啟動登入伺服器”
- 第 20 頁“管理本機與網路顯示器”
- 第 26 頁“檢查錯誤”
- 第 26 頁“停止登入伺服器”
- 第 28 頁“變更登入螢幕外觀”
- 第 34 頁“管理登入管理者”
- 第 35 頁“登入管理者檔案”

登入伺服器：

- 可以無條件地顯示點陣圖顯示器上的登入螢幕，或是依照本機與網路點陣圖顯示器上的請求
- 供應直接附加的字元主控台顯示
- 可以顯示讓使用者能夠從網路上其它登入伺服器顯示登入螢幕的選擇程式螢幕
- 允許登入伺服器的控制存取
- 提供傳統字元模式登入的存取

由登入管理者所管理的顯示器可以直接附加至登入管理者伺服器，或附加至網路上的 X 終端機或工作站。對於本機顯示器而言，登入伺服器將會自動啟動 X 伺服器並顯示登入螢幕。對於網路顯示器而言，例如 X 終端機，登入伺服器支援 X 顯示管理者通信協定 (XDMCP) 1.0，其允許顯示器要求登入伺服器在顯示器上顯示登入螢幕。

啓動登入伺服器

登入伺服器通常是在啓動系統時啓動。您也可以從指令行啓動登入伺服器。

- 如果要設定登入伺服器以在啓動系統時啓動，鍵入 `/usr/dt/bin/dtconfig -e` 然後登入伺服器將會在您重新啓動時自動啓動。

有關桌上管理系統配置公用程式 `dtconfig` 的更多資訊，請參閱 附錄 A。它提供 `dtconfig.1` 線上援助頁的副本。

- 如果要從指令行啓動登入伺服器，鍵入 `/usr/dt/bin/dtlogin -daemon; exit`

註解 – 雖然從指令行啓動登入伺服器可供暫存配置測試使用，但您在啓動系統時應正常地啓動登入伺服器。

管理本機與網路顯示器

圖 1-1 顯示可能的登入伺服器配置。

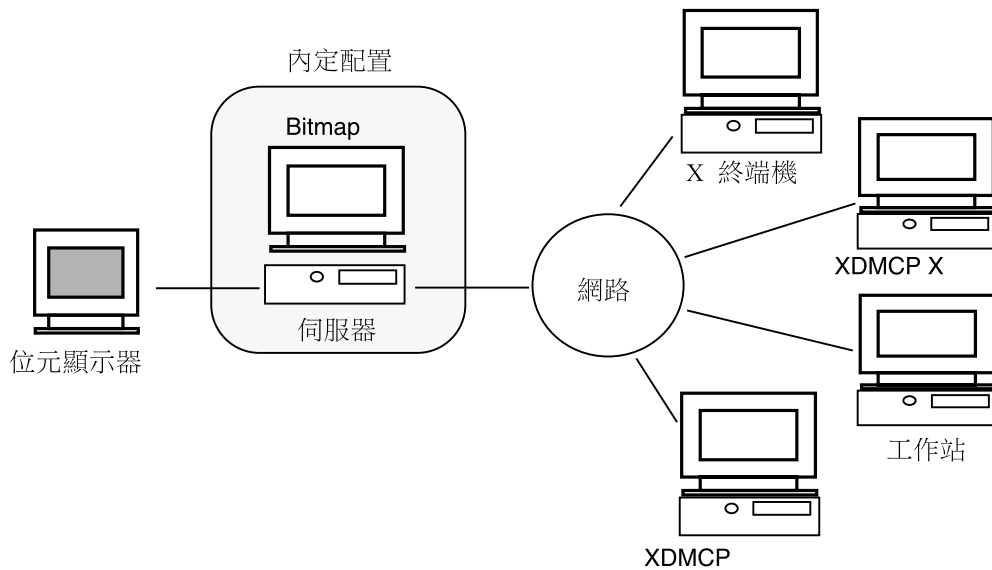


圖 1-1 可能的登入伺服器配置

尋找登入伺服器程序 ID

按預設，登入伺服器將其程序儲存於 `/var/dt/Xpid` 中。

如果要變更此一情況，您可以在 `Xconfig` 檔案中設定 `Dtlogin.pidFile` 資源。如果變更，指定的目錄必須在啟動登入伺服器時存在。

如果要修改 `Xconfig`，從 `/usr/dt/config` 將 `Xconfig` 複製到 `/etc/dt/config`。在修改 `/etc/dt/config/Xconfig` 之後，藉由鍵入下行告訴登入伺服器重新讀取 `Xconfig`：

```
/usr/dt/bin/dtconfig -reset
```

這將發出指令 `kill -HUP login_server_process_ID`。

例如，如果要在 `/var/mysevers/Dtpid` 中儲存登入伺服器程序 ID，請在 `Xconfig` 檔案中設定如下：

```
Dtlogin.pidFile: /var/mysevers/Dtpid
```

當登入伺服器重新啟動時，登入伺服器會將其程序 ID 儲存在 `/var/mysevers/Dtpid` 中。當登入伺服器啟動時，`/var/mysevers` 目錄必須存在。

在本機顯示器上顯示登入螢幕

在啟動時，登入伺服器會檢查 `Xservers` 檔案以決定是否需要啟動 X 伺服器並決定登入螢幕是否應該顯示於本機或網路顯示器上及如何顯示。

如果要修改 `Xservers`，請從 `/usr/dt/config` 將 `Xservers` 複製到 `/etc/dt/config`。在修改 `/etc/dt/config/Xservers` 之後，藉由鍵入下行告訴登入伺服器重新讀取 `Xservers`：

```
/usr/dt/bin/dtconfig -reset
```

這將發出指令 `kill -HUP login_server_process_ID`

`Xservers` 行的格式為：

```
display_name display_class display_type X_server_command
```

其

display_name—告訴登入伺服器在連接至 X 伺服器時使用的連接名稱（在以下範例中的 `:0`）。*（星號）的值擴充至 **主機名稱:0**。指定的數目必須符合 *X_server_command* 連接數目中指定的數目。

display_class—識別專用於此顯示器的資源（`Local`在以下範例中）。

display_type—告訴登入伺服器，此顯示器是本機還是網路顯示器，以及如何管理登入螢幕上的指令行登入選項（在以下範例中的 `local@console`）。

X_server_command—識別指令行、連接數目以及登入伺服器將會用以啟動 X 伺服器（在以下範例中的 `/usr/bin/X11/X: 0`）的其它選項。指定的連接數目必須符合 *display_name*。

預設 `Xservers` 行類似於：

```
:0 Local local@console /usr/bin/X11/X :0
```

執行無本機顯示器的登入伺服器

如果您的登入伺服器系統沒有點陣圖顯示器，藉由註釋使用 #（磅的符號）本機顯示器的 `Xservers` 行來執行不具本機顯示器的登入伺服器。例如，

```
# :0 Local local@console /usr/bin/X11/X :0
```

當登入伺服器啟動時，它會在背景中執行，等待來自網路顯示器的要求。

存取本機顯示器上的指令行登入

當使用者選擇登入螢幕上的指令行登入時，登入伺服器會暫時終止 X 伺服器，允許存取在點陣圖顯示器終端機裝置上執行的傳統指令行登入。在使用者登入然後登出之後，或是在指定逾時之後，登入伺服器會重新啟動 X 伺服器。

註解 – 指令行登入選項在網路顯示器上是不供使用的。

display_type 控制指令行登入的行為。*display_type* 的格式為：

- `local@display_terminal_device`
- `local`
- `foreign`

設定 `local@display_terminal_device` 時，登入伺服器假定 X 伺服器與 `/dev/display_terminal_device` 都在相同的實體裝置上，而且指令行登入（通常 `getty`）正在裝置上執行。當使用者選擇指令行登入時，就會終止 X 伺服器，允許執行指令行登入（`getty`）的存取在 `/dev/display_terminal_device`。

如果要在顯示器上取消指令行登入選項，指定 “none” 作為 *display_terminal_device*。預設的 *display_terminal_device* 為 `console`。當指定 `local` 時，*display_terminal_device* 將預設為 `console`。當指定 `foreign` 時，就會取消指令行登入。

註解 – 當登入伺服器是從指令行啟動時，將會在本機顯示器上取消指令行登入選項。

使用字元顯示主控台

如果您的登入伺服器系統有直接附加的字元顯示器作為主控台，您可能也要設定 *display_terminal_device* 為 `none` 以取消點陣圖顯示器登入螢幕上的指令行登入。

另一種選擇是，如果指令行登入 (`getty`) 正在字元顯示主控台與點陣圖顯示器上執行，您可以在點陣圖顯示器上將 *display_terminal_device* 變更為指令行登入 (`getty`) 裝置。

例如，如果點陣圖顯示器指令行登入 (`getty`) 是在裝置 `/dev/tty01` 上，變更 *display_type* 為 `local@tty01`。

在網路顯示器上顯示登入螢幕

登入伺服器可以接受來自網路顯示器的要求，以在該特定顯示器上顯示登入螢幕。網路顯示器通常是 X 終端機，但也可以是工作站。

如果要管理來自網路顯示器的要求，登入伺服器支援 X 顯示管理者通信協定 (XDMCP) 1.0。此通信協定使登入伺服器能夠協調以及接受或拒絕來自網路顯示器的要求。大多數 X 終端機具有內建的 XDMCP。

來自網路顯示器的 XDMCP 直接要求

當您配置您的 X 終端機以使用 XDMCP 直接（查詢模式）時，請告訴您的 X 終端機登入伺服器主機的主機名稱。當 X 終端機啟動時，其會自動連繫登入伺服器，而登入伺服器在 X 終端機上顯示登入螢幕。有關描述如何替您的 X 終端機配置 XDMCP 直接模式的資訊，請參閱您的 X 終端機文件。

大部分的 X 伺服器也支援 `-查詢` 選項。在此模式中，您的 X 伺服器表現得就好像它是 X 終端機，直接連繫登入伺服器主機並要求其在 X 伺服器上顯示登入螢幕。例如，在工作站 `bridget` 上的點陣圖顯示器上啟動 X 伺服器將會使登入伺服器 `anita` 在 X 伺服器上顯示登入螢幕：

```
X -query anita
```

來自網路顯示器的 XDMCP 間接要求

當您配置您的 X 終端機以使用 XDMCP 間接模式時，請告訴您的 X 終端機登入伺服器主機的主機名稱。當 X 終端機啟動時，它會連繫登入伺服器，而登入伺服器將會經由選擇程式螢幕呈現一張網路上其它登入伺服器主機的清單。從這張清單，使用者可以選擇主機，而該主機將會在使用者的 X 終端機上顯示登入螢幕。有關描述如何替您的 X 終端機配置 XDMCP 間接模式的資訊，請參閱您的 X 終端機文件。

如同直接模式，大部分的 X 伺服器支援 `-間接` 選項，它會使您的 X 伺服器以 XDMCP 間接模式連繫登入伺服器。

管理非 XDMCP 網路顯示器

舊的 X 終端可能不支援 XDMCP。對於登入伺服器而言，如果要在這種類型的 X 終端機上顯示登入螢幕，請在 `Xservers` 檔案中列出 X 終端機名稱。

既然顯示器是在網路上，`display_name` 會包括主機名稱，作為名稱的一部份。此**顯示器類別**可以用於指定對特定類別的 X 終端機的特定資源。此外來的 `display_type` 告訴登入伺服器要連接現存 X 伺服器而非啟動它。在這種情況中，則不指定 `X_server_command`。

範例

`Xservers` 檔案中的下列各行指示登入伺服器要在兩個非 XDMCP X 終端機，`ruby` 與 `wolfie` 上顯示登入螢幕：

```
ruby.blackdog.com:0 AcmeXsta foreign wolfie:0 PandaCo foreign
```

控制登入伺服器的存取

按預設，在您的網路上，任何可以存取您的登入伺服器主機的主機都可以要求顯示登入螢幕。您可以藉由修改 `Xaccess` 檔案來限制登入伺服器的存取。

如果要修改 Xaccess，從 /usr/dt/config 將 Xaccess 複製到 /etc/dt/config。在修改 /etc/dt/config/Xaccess 之後，請藉由鍵入下行告訴登入伺服器重新讀取 Xaccess：

```
/usr/dt/bin/dtconfig -reset
```

這將發出指令 `kill -HUP login_server_process_ID`。

XDMCP 直接

當主機嘗試經由 XDMCP 直接來連接至登入伺服器時，要比較主機名稱與 Xaccess 登錄以決定是否允許主機存取登錄伺服器。每一個 Xaccess 登錄是一個主機名稱，包括通配字元 * (星號) 及 ? (問號)。* (星號) 符合零或更多字元，而 ? (問號) 符合任何一個字元。位在登錄前的 ! (驚歎號) 在沒有前言允許存取時不允許存取。

例如，假設 Xaccess 包含下列三項登錄：

```
amazon.waterloo.com      *.dept5.waterloo.com      !*
```

第一項登錄允許從主機 `amazon.waterloo.com` 存取登入伺服器，第二項登錄允許從任何完整領域名稱以 `dept5.waterloo.com` 結尾的主機的存取，而最後的登錄取消來自任何其它主機的存取。

XDMCP 間接

當主機嘗試經由 XDMCP-間接來連接至登入伺服器時，要比較主機名稱與 Xaccess 登錄以決定是否允許主機存取登錄伺服器。每項 Xaccess 登錄都與 XDMCP-直接登錄類似，包括通配字元，除了每項登錄都以 CHOOSER 字串標明之外。例如：

```
amazon.waterloo.com      CHOOSER BROADCAST      *.dept5.waterloo.com  
CHOOSER BROADCAST      !*                      CHOOSER BROADCAST
```

再一次，第一項登錄允許從主機 `amazon.waterloo.com` 存取登入伺服器，第二項登錄允許從任何完整領域名稱以 `dept5.waterloo.com` 結尾的主機的存取，而最後的登錄取消來自任何其它主機的存取。

在 CHOOSER 之後可以列出下列之一。

- BROADCAST
- 主機名稱的清單

BROADCAST 告訴登錄伺服器要廣播至登入伺服器子網路，以產生可供使用登入伺服器主機的清單。主機名稱的清單告訴登入伺服器要使用可供使用的登入主機清單的該清單。例如：

```
amazon.waterloo.com      CHOOSER shoal.waterloo.com alum.waterloo.com  
*.dept5.waterloo.com    CHOOSER BROADCAST      !*                      CHOOSER BROADCAST
```

如果 `amazon.waterloo.com` 經由 XDMCP-間接來連接，會有包含 `shoal` 與 `alum` 的清單呈現出來。如果 `alice.dept5.waterloo.com` 連接，會有所有登入伺服器子網路上可供使用的登入伺服器主機的清單呈現出來。其它的 XDMCP-間接要求都會被否認。

指定主機名稱清單的另一種選擇是定義一個或一個以上包含主機名稱清單的巨集。例如：

```
%list1    shoal.waterloo.com alum.waterloo.com    amazon.waterloo.com
CHOOSE%list1
```

檢查錯誤

按預設，登入伺服器會在 `/var/dt/Xerrors` 檔案中登錄錯誤。如果要變更，您可以在 `Xconfig` 檔案中設定 `Dtlogin.errorLogFile` 資源。當登入伺服器啟動時，指定的目錄必須存在。

例如，如果要使登入伺服器在 `/var/mylogs/Derrors` 檔案中登錄錯誤，請在 `Xconfig` 檔案中設定如下：

```
Dtlogin.errorLogFile: /var/mylogs/Derrors
```

當登入伺服器重新啟動時，登入伺服器會登錄錯誤至 `/var/mylogs/Derrors` 檔案。當登入伺服器啟動時，`/var/mylogs` 目錄必須存在。

停止登入伺服器

- 當系統啟動時，如果要停用登入伺服器啟動，請鍵入：

```
/usr/dt/bin/dtconfig -d
```

此將會告訴系統在您下次重新啟動時，不要啟動登入伺服器。

- 如果要藉由取消程序 ID 來停止登入伺服器，請鍵入：

```
/usr/dt/bin/dtconfig -kill
```

這將發出指令 `kill login_server_process_ID`

註解 – 取消登入伺服器程序會終止所有由登入伺服器管理的使用者階段作業。

您也可以藉由取消程序 ID 來停止登入伺服器。登入伺服器程序 ID 是藉由 Dtlogin.pidFile 資源儲存於 /var/dt/Xpid 或 Xconfig 中所指定的檔案內。

如果您在取消登入伺服器之時登入桌上管理系統，您的桌上管理系統階段作業會立即終止。

登入螢幕

登入伺服器所顯示的登入螢幕對於傳統字元模式的登入螢幕是一項很有吸引力的選擇，並且提供遠超過由字元模式登入所提供的功能。

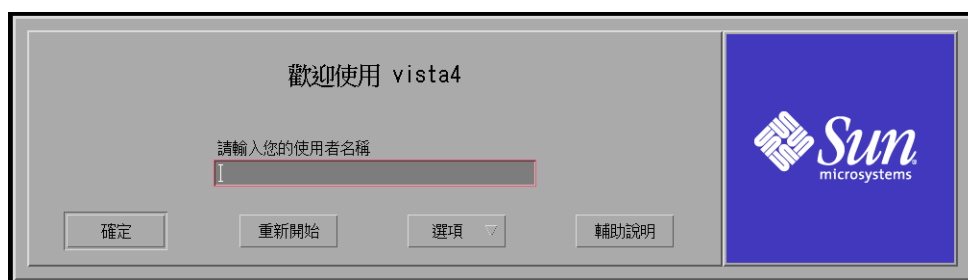


圖 1-2 桌上管理系統登入螢幕

如同字元模式登入，使用者輸入跟隨著密碼的使用者名稱。如果認證過，登入伺服器會為使用者啟動桌上管理系統階段作業。當使用者結束桌上管理系統階段作業時，登入伺服器會顯示新的登入螢幕，並且程序再次開始。

如果要自訂登入螢幕，您可以：

- 變更登入螢幕外觀
- 配置 X 伺服器權限
- 變更預設語言
- 在顯示登入螢幕之前發出指令
- 變更登入螢幕語言功能表的內容
- 指定指令以啟動使用者的階段作業
- 在啟動使用者桌上管理系統階段作業之前發出指令
- 在使用者的階段作業結束之後發出指令

這些每一項都可以為所有顯示器或在每個顯示器的基礎上完成。

變更登入螢幕外觀

如果要自訂登入螢幕外觀，您可以變更標識圖案或圖形、歡迎訊息與字型。

如果要修改 `Xresources`，請從 `/usr/dt/config/language` 將 `language` 複製到 `/etc/dt/config/language`。登入螢幕會在登入螢幕下次顯示時反映出任何變更。如果要強迫登入螢幕重新顯示，請從登入螢幕選項功能表選擇重設登入螢幕。

可以由 `Xresources` 檔案中的資源規格決定的登入螢幕屬性包括：

`Dtlogin*logo*bitmapFile`—點陣圖或圖素映射檔案以顯示為標識圖案影像

`Dtlogin*greeting*persLabelString`—個人化的歡迎訊息

`Dtlogin*greeting*labelString`—歡迎訊息

`Dtlogin*greeting*fontList` 歡迎訊息的字型

`Dtlogin*labelFont` 按鈕及標籤的字型

`Dtlogin*textFont` 輔助說明及錯誤訊息的字型

`Dtlogin*language*languageName` 語言環境名稱 `language` 的替代文字

▼ 如果要變更標識圖案

- 在 `Xresources` 中設定 `Dtlogin*logo*bitmapFile` 資源。

標識圖案可以是彩色圖素映射或點陣圖檔案。

以下範例使用 `Mylogo` 點陣圖作為標識圖案：

```
Dtlogin*logo*bitmapFile: /usr/local/lib/X11/dt/bitmaps/Mylogo.bm
```

▼ 變更歡迎訊息

依據預設值，登入伺服器會在登入螢幕上顯示訊息 `Welcome to 主機名稱`。若要變更這個訊息：

- 在 `Xresources` 中設定 `Dtlogin*greeting*labelString` 資源。

此 `labelString` 資源的值可以包含兩個值，其一為 `%LocalHost%`，此值將由登入伺服器的主機名稱所取代，而其二為 `%DisplayName%`，此值則由 X 伺服器的顯示名稱所取代。

以下範例將歡迎訊息變更為 `Here's 主機名稱!`：

```
Dtlogin*greeting*labelString: Here's %LocalHost%!
```

一旦輸入使用者名稱，登入伺服器會依照預設值顯示訊息 `Welcome 使用者名稱`。您可以藉由在 `Xresources` 中設定 `Dtlogin*greeting*persLabelString` 資源來變更此訊息。此 `persLabelString` 的值可以包含 `%s`，它將會由 `使用者名稱` 所取代。

以下範例將個人化的歡迎訊息變更為 `Hello 使用者名稱`。

```
Dtlogin*greeting*persLabelString: Hello%s
```

▼ 變更字型

您可以藉由在 `Xresources` 中設定下列字型資源之一來變更登入螢幕上的字型：

`Dtlogin*greeting*fontList`—歡迎訊息的字型

`Dtlogin*labelFont`—按鈕及標籤的字型

`Dtlogin*textFont`—輔助說明及錯誤訊息的字型

如果要列出可供使用的字型，請鍵入：

```
xlsfonts [-選項] [-fn 型樣]
```

以下範例使用歡迎訊息的大字型（您所指定的值必須包含在一行上）：

```
Dtlogin*greeting*fontList: -dt-interface \  
system-medium-r-normal-xxl*-***-***-***-***:
```

▼ 提供替代文字以顯示每種語言

如果要在登入螢幕語言功能表上顯示每個語言環境的文字，以代替語言環境名稱的預設顯示的話，請修改 `Xresources` 中的 `Dtlogin*語言 *languageName` 資源名稱資源：

```
Dtlogin*En_US*languageName: American
```

現在將會顯示文字 `American` 而非語言環境名稱 `En_US`。

變更登入螢幕行爲

如果要自訂登入螢幕行爲，您可以修改 `Xconfig` 檔案中所指定的資源。

如果要修改 `Xconfig`，從 `/usr/dt/config` 將 `Xconfig` 複製到 `/etc/dt/config`。在修改 `/etc/dt/config/Xconfig` 之後，藉由鍵入下行告訴登入伺服器重新讀取 `Xconfig`：

```
/usr/dt/bin/dtconfig -reset
```

此發出指令 `kill -HUP 登入伺服器程序 ID`)

Xconfig 檔案中所指定的資源包括：

Dtlogin*authorize—Xaccess 檔案規格

Dtlogin*environment—X 伺服器環境

Dtlogin*language—預設語言

Dtlogin*languageList—登入螢幕語言功能表的語言清單

Dtlogin*resources—Xresources 規格

Dtlogin*setup—Xsetup 檔案規格

Dtlogin*startup—Xstartup 檔案規格

Dtlogin*session—Xsession 檔案規格

Dtlogin*failsafeClient—Xfailsafe 程序檔規格

Dtlogin*reset—Xreset 程序檔規格

Dtlogin*userPath—Xsession 及 Xfailsafe 的 PATH

Dtlogin*systemPath—Xsetup, Xstartup 及 Xfailsafe 的 PATH

Dtlogin*systemShell—Xsetup, Xstartup 及 Xfailsafe 的 SHELL

Dtlogin.timeZone—全部程序檔的 TZ

變更每個顯示器的登入螢幕行爲

在下面的範例中，變更 Xconfig 資源變更了所有顯示器的登入螢幕行爲。在每個顯示器的基礎上可以指定用 * (星號) 列出的資源。這使您能夠指定特定顯示器的自訂登入螢幕行爲。如果要指定特定顯示器的資源，將資源指定為 `Dtlogin* displayName*資源`。例如，如果您想要關閉顯示器 `expo:0` 以使用者為基礎的存取控制，但為其它顯示器保持開啓，您會指定：

```
Dtlogin*expo_0*authorize: False
```

註解 – 顯示器名稱中的任何特殊字元，例如：(冒號) 或。(句號)，都由 _ (底線) 所取代。

變更 X 伺服器存取

按預設，登入伺服器允許 X 伺服器在每個使用者基礎上存取控制，並以 *HomeDirectory/.Xauthority* 檔案中儲存與保護的授權資料為基礎。只允許可以讀取此檔案的使用者連接到 X 伺服器。一般而言，此為 X 伺服器存取控制喜好的方法。

以使用者為基礎的存取控制的另一種選擇是以主機為基礎的存取控制。使用此種方法，如果一部主機有 X 伺服器所允許的存取權，該主機上任何使用者都可連接至 X 伺服器。使用以主機為基礎的控制的原因包括：

- 較舊的 R2 與 R3 X 用戶端將不能夠使用以使用者為基礎的存取控制，連接到 X 伺服器。
- 在不安全的網路上，窺探者可能可以在網路上竊取 X 用戶端與 X 伺服器之間傳送的授權資料。

`Xconfig Dtlogin*authorize` 資源告訴登入伺服器要使用以使用者為基礎的 X 伺服器存取控制。如果要使用以主機為基礎的存取控制，將授權資源值變更為 `False`，例如：

```
Dtlogin*authorize: False
```

▼ 變更 X 伺服器環境

當您由登入伺服器啟動時，如果您想要提供一個或多個環境變數和值給 X 伺服器，您可以在 `Xconfig` 中使用 `Dtlogin*environment` 資源來指定它們。例如：

```
Dtlogin*environment: VAR1=foo VAR2=bar
```

將會使變數 `VAR1` 與 `VAR2` 可用於局部 X 伺服器程序。這些變數也將會匯出至 `Xsession` 與 `Xfailsafe` 程序檔。

▼ 變更預設語言

當使用者由登入螢幕登入桌上管理系統時，使用者階段作業是在由選項功能表的語言子功能表所選擇的語言環境下執行的。如果使用者未選擇語言，則使用登入伺服器預設語言。您可以藉由在 `Xconfig` 中設定 `Dtlogin*language` 資源來控制預設語言的值。例如：

```
Dtlogin*language: Ja_JP
```

檢查您的系統文件以決定安裝在您系統上的語言。

▼ 變更登入螢幕語言功能表的內容

按預設，登入伺服器建立包含系統上所安裝的所有語言環境清單的登入螢幕語言功能表。當使用者從登入螢幕語言清單選擇語言環境時，登入伺服器將會在選擇的區域環境中重新顯示登入螢幕。當使用者隨後登入時，登入伺服器將會在該語言環境中為使用者啟動桌上管理系統階段作業。

您可以藉由在 Xconfig 中修改 Dtlogin*languageList 資源來指定您自己的語言清單：

```
Dtlogin*languageList: En_US De_DE
```

現在登入伺服器在登入螢幕語言功能表中只顯示 En_US 與 De_DE。

在登入螢幕出現之前發出指令

在啟動 X 伺服器之後但在登入螢幕出現之前，登入伺服器執行 Xsetup 程序檔。Xsetup 以 root 授權執行並發出需要在登入螢幕顯示之前執行的指令。

如果要修改 Xsetup，請從 /usr/dt/config 將 Xsetup 複製到 /etc/dt/config。下一次顯示登入螢幕時，將會執行修改的 Xsetup。

在啟動使用者階段作業之前發出指令

在使用者輸入使用者名稱與密碼以及為其認證之後，但在啟動使用者階段作業之前，登入伺服器執行 Xstartup 程序檔。Xstartup 以 root 授權來執行並發出需要在使用者階段作業啟動之前以 root 執行的指令。

如果要修改 Xstartup，請從 /usr/dt/config 將 Xstartup 複製到 /etc/dt/config。下次使用者登入時，將會執行修改的 Xstartup。

啟動桌上管理系統階段作業

按預設，登入伺服器藉由執行 Xsession 程序檔來啟動使用者階段作業。Xsession 以使用者的授權來執行並發出需要啟動桌上管理系統的指令。

註解 – 請勿直接更新 Xsession 程序檔。

有關如何自訂使用者的桌上管理系統階段作業啟動的資訊，請參閱 第 2 章。

啟動單視窗階段作業

如果使用者從登入螢幕選項功能表的階段作業子功能表選擇單視窗階段作業，登入伺服器會執行 Xfailsafe 程序檔。Xfailsafe 以使用者的授權來執行並發出需要啟動最小視窗環境的指令，通常是終端機視窗與選擇性的視窗管理者。

如果要修改 Xfailsafe，請從 /usr/dt/config 將 Xfailsafe 複製到 /etc/dt/config。下次使用者登入時，將會執行修改的 Xfailsafe。

在使用者的階段作業結束之後

在使用者結束桌上管理系統或單視窗階段作業之後，登入伺服器會執行 Xreset 程序檔。Xreset 以 root 授權執行並發出需要在使用者的階段作業結束之後以 root 執行的指令。

如果您希望修改 Xreset，請從 /usr/dt/config 將 Xreset 複製到 /etc/dt/config。下次使用者登入時，將會執行修改的 Xreset。

登入伺服器環境

登入伺服器提供匯出至 Xsetup、Xstartup、Xsession、Xfailsafe 與 Xreset 程序檔的環境。此環境描述於下表 表 1-1 中。其它的變數也可以由登入伺服器匯出。

表 1-1 登入伺服器環境

環境變數	Xsetup	Xstartup	Xsession	Xreset	說明
LANG	X	X	X	X	預設或選擇的語言
XAUTHORITY	X	X	X	X	替代 X 授權檔案 (選擇性的)
PATH	X	X	X	X	Dtlogin*userPath 資源 (Xsession、Xfailsafe) 或 Dtlogin*systemPath 資源 (Xsetup、Xstartup、Xreset) 的值
DISPLAY	X	X	X	X	X 伺服器連接編號
SHELL	X	X	X	X	/etc/passwd (Xsession、Xfailsafe) 或 Dtlogin*systemShell 資源 (Xsetup、Xstartup、Xreset) 中所指定的 Shell
TZ	X	X	X	X	Dtlogin.timeZone 資源或由系統決定的時區值
USER		X	X	X	使用者名稱
HOME		X	X	X	/etc/passwd 中所指定的主目錄
LOGNAME		X	X	X	使用者名稱

變更使用者或系統路徑

登入伺服器在其執行 Xsession 與 Xfailsafe 程序檔時，設定路徑環境變數。您可以提供替代路徑給這些程序檔

▼ 變更使用者路徑

- 在 Xconfig 中設定 Dtlogin*userPath 資源。例如：

```
Dtlogin*userPath: /usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11
```

▼ 變更系統路徑

- 在 Xconfig 中設定 Dtlogin*systemPath 資源。例如：

```
Dtlogin*systemPath: /usr/bin/X11:/etc:/bin:/usr/bin:/usr/ucb
```

▼ 如果要變更系統 Shell

登入伺服器在其執行 Xsetup、Xstartup 與 Xfailsafe 程序檔時，設定 SHELL 環境變數。預設值為 /bin/sh。如果您希望提供替代 shell 給這些程序檔，您可以在 Xconfig 中設定 Dtlogin*systemShell 資源。例如：

```
Dtlogin*systemShell: /bin/ksh
```

▼ 如果要變更時區

登入伺服器在其執行 Xsetup、Xstartup、Xsession、Xfailsafe 與 Xreset 程序檔時設定 TZ 環境變數。預設值是得自系統，因此通常您將不需要變更此行為。如果要提供替代時區給這些程序檔，請在 Xconfig 中設定 Dtlogin.timeZone 資源。例如：

```
Dtlogin.timeZone: CST6CDT
```

管理登入管理者

當登入伺服器啟動時，就會啟動一項 dtlogin 程序。dtlogin 程序讀取 Xconfig 檔案以決定起始登入伺服器配置，並尋找其它的登入伺服器配置檔案。然後登入伺服器會讀取 Xservers 檔案來看看它是否有任何可明確管理的顯示器，並且也會讀取 Xaccess 檔案以控制登入伺服器的存取。

如果登入伺服器由需要管理區域顯示器的 Xservers 檔案尋找，它將會如 Xservers 檔案中所指示的那樣啟動 X 伺服器，然後在該顯示器上顯示登入螢幕。

如果登入伺服器由需要管理網路顯示器的 Xservers 檔案尋找，它將會假定 X 伺服器已經正以指定的顯示器名稱來執行，並且在該顯示器上顯示登入螢幕。

然後登入伺服器將會等待來自網路的 XDMCP 要求。

對於管理的每部顯示器而言，登入伺服器會先建立該顯示器的新 `dtlogin` 程序。這表示如果登入伺服器正在管理 n 顯示器，就會有 $n+1$ `dtlogin` 個程序。登入伺服器將會執行 `Xsetup` 程序檔，載入 `Xresources` 檔案，然後執行 `dtgreet` 以顯示登入螢幕。只要使用者一輸入使用者名稱與密碼並已認證，登入伺服器便會執行 `Xstartup` 程序檔，然後執行 `Xsession` 或 `Xfailsafe` 程序檔。當使用者結束階段作業時，登入伺服器將會執行 `Xreset` 程序檔。

如果登入伺服器取得 XDMCP-間接請求，它將會執行 `dtchooser` 以呈現顯示器上登入伺服器主機的清單。當使用者從清單選擇主機時，主機上的登入伺服器將會管理顯示器。

對於 `Xaccess`、`Xconfig`、`Xfailsafe`、`Xreset`、`language`、`Xresources`、`Xservers`、`Xsetup` 與 `Xstartup` 配置檔案而言，登入伺服器將會依照預設先查看 `/etc/dt/config` 中，然後才查看 `/usr/dt/config`，並且使用找到的第一個檔案。

登入管理者檔案

登入管理者檔案的預設位置為：

`/usr/dt/bin/dtlogin`—登入伺服器與顯示器管理者

`/usr/dt/bin/dtgreet`—顯示顯示器的選擇器螢幕

`/usr/dt/bin/dtchooser`—顯示選擇程式螢幕

`/usr/dt/bin/Xsession`—啟動桌上管理系統階段作業

`/usr/dt/config/Xfailsafe`—啟動單視窗階段作業

`/usr/dt/config/Xconfig`—登入伺服器配置檔案

`/usr/dt/config/Xservers`—登入伺服器顯示說明檔案

`/usr/dt/config/Xaccess`—登入伺服器存取說明檔案

`/usr/dt/config/language` / `Xresources`—顯示佈局資源

`/usr/dt/config/Xsetup`—顯示設定檔

`/usr/dt/config/Xstartup`—前階段作業啟動檔案

`/usr/dt/config/Xreset`—後階段作業重設檔

登入伺服器的 `/var/dt/Xpid`—處理 ID

`/var/dt/Xerrors`—登入伺服器的錯誤登錄檔案

第 2 章

配置階段作業管理者

階段作業管理者負責啓動桌上管理系統並自動儲存與回復執行應用程式、色彩、字型、滑鼠行爲、音量與鍵盤按鍵。

- 第 37 頁“什麼是階段作業？”
- 第 38 頁“啓動階段作業”
- 第 39 頁“當階段作業啓動時”
- 第 45 頁“其它的階段作業啓動自訂 ”
- 第 49 頁“階段作業管理者檔案及目錄”

使用階段作業管理者，您可以：

- 自訂所有桌上管理系統使用者的起始階段作業
- 自訂所有桌上管理系統使用者的環境與資源
- 變更階段作業啓動訊息
- 變更階段作業啓動工具與常駐程式的參數
- 自訂所有使用者的桌上管理系統色彩用法

什麼是階段作業？

階段作業是呈現在使用者的桌上管理系統上的應用程式、設定值與資源的集合。階段作業管理是使階段作業管理者能夠儲存與回復使用者的階段作業的一組慣例與協定。使用者能夠登入系統，並於使用者登出時以執行應用程式、設定值與資源的相同組別來呈現。當使用者第一次登入桌上管理系統時，會載入預設起始階段作業。之後，階段作業管理者會支援目前與主要階段作業的觀念。

初始階段作業

當使用者第一次登入桌上管理系統時，階段作業管理者將會使用系統預設值來產生使用者的初始階段作業。按預設，將會啟動檔案管理者與桌上管理系統簡介、輔助說明內容。

目前階段作業

使用者正在執行的階段作業永遠都會被視為目前階段作業，不論是登入時從儲存的主階段作業回復、儲存的目前階段作業或是系統預設初始階段作業。根據使用者的樣式管理者啟動設定值，當使用者結束階段作業時，階段作業管理者會自動儲存目前階段作業。當使用者下次登入桌上管理系統時，階段作業管理者會重新啟動之前儲存的目前階段作業，表示桌上管理系統將會回復到與使用者上次登出時相同的狀態。

主要階段作業

您也可以讓桌上管理系統每次在使用者登入時就回復到相同的狀態，不論使用者登出時的狀態如何。使用者可以儲存目前階段作業的狀態，然後使用樣式管理者啟動設定值，讓階段作業管理者每次在使用者登入時就啟動該階段作業。

顯示器的特定階段作業

如果要執行特定顯示器的特定階段作業，使用者可以建立顯示特定的階段作業。如果要如此作，使用者可以將 *HomeDirectory/.dt/sessions* 目錄複製到 *HomeDirectory/.dt/顯示器*，其中**顯示器**是實際不限定的主機名稱 (例如，*pablo:0* 是有效的，*pablo.gato.com:0* 或 *unix:0* 則是無效的)。當使用者登入顯示器 *pablo:0* 時，階段作業管理者將會啟動該顯示器特定的階段作業。

啟動階段作業

階段作業管理者是經由 */usr/dt/bin/Xsession* 啟動的。當使用者使用登入管理者登入時，*Xsession* 會因預設而啟動。

使用者可以選擇性地使用傳統的字元模式 (*getty*) 登入來登入，並且使用啟動 X 伺服器的工具人工啟動階段作業管理者，例如 *xinit*。例如：*xinit /usr/dt/bin/Xsession*。

當階段作業啓動時

當階段作業管理者啓動時，要經過下列步驟以啓動使用者的階段作業：

1. 搜尋 *HomeDirectory/.dtprofile* 程序檔
2. 找出 *xsession.d* 程序檔的來源
3. 顯示歡迎訊息
4. 設定桌上管理系統搜尋路徑
5. 收集可供使用的應用程式
6. 選擇性地尋源 *HomeDirectory/.profile* 或 *HomeDirectory/.login*
7. 啓動 ToolTalk® 傳訊常駐程式
8. 載入階段作業資源
9. 啓動色彩伺服器
10. 啓動工作區管理者
11. 啓動階段作業應用程式

下列各段描述以上所列出的步驟。

找出 *.dtprofile* 程序檔的來源

在階段作業啓動時，*xsession* 程序檔會搜尋使用者的 *HomeDirectory/.dtprofile* 程序檔。此 *HomeDirectory/.dtprofile* 程序檔是讓使用者能夠設定其階段作業環境變數的 */bin/sh* 或 */bin/ksh* 程序檔。有關設定環境變數更多的資訊，請參閱第 45 頁“其它的階段作業啓動自訂”。

如果 *HomeDirectory/.dtprofile* 程序檔不存在，例如當使用者正第一次登入桌上管理系統時，*xsession* 將會複製桌上管理系統預設 *sys.dtprofile* 爲 *HomeDirectory/.dtprofile*。

桌上管理系統預設爲 */usr/dt/config/sys.dtprofile*。如果要自訂 *sys.dtprofile* 程序檔，請從 */usr/dt/config* 將 *sys.dtprofile* 複製到 */etc/dt/config* 並編輯新檔案。

找出 Xsession.d 程序檔的來源

在搜尋 *HomeDirectory/.dtprofile* 程序檔之後，Xsession 程序檔會搜尋 Xsession.d 程序檔。這些程序檔是用於設定其它的環境變數並啟動使用者的階段作業選擇性常駐程式。預設的 Xsession.d 程序檔為：

0010.dtpaths—文件可自訂的桌上管理系統搜尋路徑

0020.dtimes—啟動選擇性的輸入方法伺服器

0030.dttmpdir—建立每個使用者、每個階段作業的暫存目錄

0040.xmbind—設定 \$XMBINDDIR 為桌上管理系統預設

在 Xsession.d 中可能有其它廠商特定的程序檔。

Xsession 會先搜尋 */etc/dt/config/Xsession.d* 目錄中的所有檔案，接著才搜尋 */usr/dt/config/Xsession.d* 目錄中的檔案。

桌上管理系統預設 Xsession.d 程序檔是位於 */usr/dt/config/Xsession.d* 目錄中。如果要自訂 Xsession.d 程序檔，請從 */usr/dt/config/Xsession.d* 將程序檔複製到 */etc/dt/config/Xsession.d* 並編輯新檔案。您必須有執行許可權才能執行此工作。

同時，如果要使 Xsession 自動搜尋您自己的程序檔，請將其複製到 */etc/dt/config/Xsession.d*。

註解—當您修改或建立 Xsession.d 程序檔時，請確定您發出的任何前景指令持續時間是短期的，因為指令所花費的時間將會直接影響階段作業啟動時間。如果前景指令未結束，階段作業啟動將會停頓。您想要繼續執行階段作業持續時間的 Xsession.d 程序檔中執行的指令應在背景中執行。

顯示歡迎訊息

在搜尋 *HomeDirectory/.dtprofile* 與 Xsession.d 程序檔之後，Xsession 顯示蓋住螢幕的歡迎訊息。您可以自訂顯示的歡迎訊息，或完全關閉此訊息。dthello 用戶端是用於顯示此訊息的。

如果要改變訊息文字，請藉由修改 `dtstart_hello[0]` 變數來變更 dthello 選項。

如果要變更 `dtstart_hello[0]`，請建立設定新值的 */etc/dt/config/Xsession.d* 程序檔。如果要為所有使用者顯示這一天的訊息，請建立可執行的 *sh* 或 *ksh* 程序檔，例如，*/etc/dt/config/Xsession.d/myvars*，並設定如下的 `tstart_hello[0]`：

```
dtstart_hello[0]="/usr/dt/bin/dthello -file /etc/motd&"
```


同樣地，使用者可以藉由在 *HomeDirectory/.dtprofile* 中設定 `dtstart_hello[0]` 來變更他們的階段作業的歡迎訊息。

如果要關閉歡迎訊息，請設定 `dtstart_hello[0]=" "`。

有關 `dthello` 更多的資訊，請參閱 `dthello` 線上援助頁。

設定桌上管理系統搜尋路徑

桌上管理系統搜尋路徑是在登入時由 `dtsearchpath` 所建立的。有兩大類由 `dtsearchpath` 所使用的環境變數：

輸入變數—數值是由系統管理或一般使用者所設定的整個系統的變數以及個人環境變數。

輸出變數—由 `dtsearchpath` 建立與指派值的變數。每個變數的值都是桌上管理系統階段作業的搜尋路徑。

如果要改變 `dtsearchpath` 的指令行選項，請修改 `dtstart_searchpath` 變數。如果要變更所有使用者的 `dtstart_searchpath` 變數，請建立可執行的 `sh` 或 `ksh` 程序檔（例如 `/etc/dt/config/Xsession.d/myvars`），並設定 `dtstart_searchpath` 如下：

```
dtstart_searchpath="/usr/dt/bin/dtsearchpath"
```

使用者同樣地可以僅為他們自己的階段作業變更 `dtsearchpath`，方法是在 *HomeDirectory/.dtprofile* 中設定 `dtstart_searchpath`。

有關 `dtsearchpath` 的更多資訊，請參閱 第 9 章。有關 `dtsearchpath` 選項的更多資訊，請參閱 `dtsearchpath` 線上援助頁。

收集可供使用的應用程式

設定桌上管理系統搜尋路徑之後的下一步就是使用 `dtappgather` 收集可供使用的應用程式。如果要改變 `dtappgather` 的指令行選項，請修改 `dtstart_appgather` 變數。如果要變更所有使用者的 `dtstart_appgather` 變數，請建立可執行的 `sh` 或 `ksh` 程序檔（例如 `/etc/dt/config/Xsession.d/myvars`），並設定 `dtstart_appgather` 如下：

```
dtstart_appgather="/usr/dt/bin/dtappgather &"
```

使用者同樣地可以僅為他們自己的階段作業變更 `dtappgather`，方法是在 *HomeDirectory/.dtprofile* 中設定 `dtstart_appgather`。

有關 `dtappgather` 選項的更多資訊，請參閱 `dtappgather(4)` 線上援助頁。

選擇性找出 .profile 或 .login 程序檔的來源

Xsession 能夠搜尋使用者的傳統 *HomeDirectory* / .profile 或 *HomeDirectory* / .login 程序檔。依照預設值，此能力是取消的。如果要告訴 Xsession 去搜尋 .profile 或 .login 程序檔，請設定 DTSOURCEPROFILE 為 true。

如果要變更所有使用者的 DTSOURCEPROFILE，請建立設定新值的 /etc/dt/config/Xsession.d 程序檔。如果要設定所有使用者的 DTSOURCEPROFILE 為 true，請建立可執行的 sh 或 ksh 程序檔，例如，/etc/dt/config/Xsession.d/myvars，並設定如下的 DTSOURCEPROFILE：

```
DTSOURCEPROFILE=true
```

使用者可以同樣地經由在 *HomeDirectory* / .dtprofile 中將 DTSOURCEPROFILE 設定為 true，變更他們自己的階段作業的 DTSOURCEPROFILE。

啓動 ToolTalk 傳訊常駐程式

ToolTalk 訊息常駐程式 ttsession 使獨立的應用程式能夠彼此通信而不需對彼此有直接的認識。應用程式會建立並發送 ToolTalk 訊息以彼此通信。ttsession 在網路上通信以發送訊息。

如果要改變 ttsession 的命令行選項，請修改 dtstart_ttsession 變數。如果要變更所有使用者的 dtstart_ttsession 變數，請建立可執行的 sh 或 ksh 程序檔（例如 /etc/dt/config/Xsession.d/myvars），並設定如下的 dtstart_ttsession：

```
dtstart_ttsession="/usr/dt/bin/ttsession -s"
```

使用者同樣地可以為他們自己的階段作業變更 ttsession，方法是在 *HomeDirectory* / .dtprofile 中設定 dtstart_ttsession。

有關 ttsession 選項的更多資訊，請參閱 ttsession 線上援助頁。有關 ttsession 的更多資訊，請參閱 *Common Desktop Environment: ToolTalk Messaging Overview*。

啓動階段作業管理者用戶端

此刻，Xsession 會啓動 /usr/dt/bin/dtsession，此啓動會使階段作業啓動程序繼續進行。

載入階段作業資源

階段作業管理者使用 X 伺服器 RESOURCE_MANAGER 屬性以使桌上管理系統資源可供所有的應用程式使用。階段作業管理者載入 RESOURCE_MANAGER 由：

- 載入系統預設值資源

- 合併任何由管理者指定的整個系統之資源
- 合併任何使用者指定的資源

桌上管理系統預設值資源可以在 `/usr/dt/config/language/sys.resources` 中找到。這些資源將會經由 `RESOURCE_MANAGER` 屬性供每個使用者的階段作業使用。此檔案不應編輯，因為在後續的桌上管理系統安裝時會將它覆寫。

您可以藉由建立 `/etc/dt/config/language/sys.resources` 來增加系統預設資源。在此檔案中，您可以為所有的桌上管理系統使用者置換預設值資源或指定其它的資源。既然此檔案是在階段作業啟動期間合併到桌上管理系統預設值資源內的，此檔案中應只放入新的或更新的資源規格。此檔案中所指定的資源將會經由 `RESOURCE_MANAGER` 屬性供每個使用者的階段作業使用。此檔案中所指定的資源，對於桌上管理系統預設值資源檔案中所指定的那些資源具有優先順序。

使用者可以使用其 `HomeDirectory/.Xdefaults` 檔案來增加桌上管理系統預設值與整個系統的資源。此檔案中所指定的資源將可經由 `RESOURCE_MANAGER` 屬性供每個使用者的階段作業使用。此檔案中所指定的資源對於桌上管理系統預設值或系統管理者資源檔案中所指定的那些具有優先順序。

註解 – X 工具箱本身的公用程式指定它將會從 `RESOURCE_MANAGER` 或從 `HomeDirectory/.Xdefaults` 其中之一而非兩者載入應用程式的資源。一般而言，這表示會略過使用者的 `HomeDirectory/.Xdefaults` 檔案。然而，階段作業管理者藉由在如上述的階段作業啟動時將其合併至 `HomeDirectory` 內以供應 `HomeDirectory/RESOURCE_MANAGER`。如果使用者變更 `HomeDirectory/.Xdefaults`，對新的應用程式而言，要到使用者呼叫重新載入資源動作時才看得到變更。重新載入資源動作將會指示階段作業管理者重新載入具有預設值、整個系統以及使用者指定的資源之 `RESOURCE_MANAGER`。此動作會使對整個系統與個人資源檔案所做的變更可供應用程式使用。

有關更多的資訊，請參閱：

- 第 257 頁“設定應用程式資源”
- `dtresourcesfile(4)` 線上援助頁

啓動色彩伺服器

階段作業管理者作為桌上管理系統的色彩伺服器，並提供以下可用於配置它的 `dtsession` 資源集。

`foregroundColor`—控制是否配置前景色彩的像素

`dynamicColor`—指定是否配置唯讀色彩

`shadowPixmap`—指定是要配置頂端陰影或底部陰影的色彩

`colorUse`—限制色彩配置

`writeXrdbColors`—指定是否在資源資料庫中放入 `*background` 與 `*foreground` 資源

您可以藉由在該檔案中建立 `/etc/dt/config/language/sys.resources` 與指定色彩伺服器資源來設定所有使用者的色彩伺服器資源。

使用者同樣地可以藉由在 `HomeDirectory/.Xdefaults` 中指定色彩資源來為自己的階段作業設定色彩伺服器資源。

有關設定色彩伺服器資源的更多資訊，請參閱 第 267 頁“管理色彩”。

啓動工作區管理者

階段作業管理者負責啓動工作區管理者。依照預設值 `/usr/dt/bin/dtwm` 會啓動。替代視窗管理者可以用 `wmStartupCommand` 資源來指定。

您可以藉由建立 `/etc/dt/config/language/sys.resources` 與指定該檔案中具有 `Dtsession*wmStartupCommand` 資源的視窗管理者之完整路徑名稱及選項來指定替代視窗管理者。

使用者同樣地可以僅為他們自己的階段作業指定替代視窗管理者，方法是在 `HomeDirectory/.Xdefaults` 中指定 `Dtsession*wmStartupCommand`。

有關視窗管理者的更多資訊，請參閱 第 16 章。

啓動階段作業應用程式

在啓動階段作業時，階段作業管理者將會重新啓動作為部分階段作業儲存的任何應用程式。被視為使用者起始階段作業的應用程式所回復之系統預設集，可以在 `/usr/dt/config/language/sys.session` 中找到。此檔案不應編輯，因為它將會在後續的桌上管理系統安裝時無條件地覆寫。

有關更多的資訊，請參閱 `dtsessionfile (4)` 線上援助頁。

系統管理者可以藉由將 `/usr/dt/config/language/sys.session` 複製到 `/etc/dt/config/language/sys.session` 並且修改後面那個檔案，來將啓動為使用者初始階段作業一部份的應用程式集取代掉。不像資源檔案，此檔案將會用作桌上管理系統預設檔案的完整取代，因此您可以複製系統預設檔案並作必要的修改。

其它的階段作業啟動自訂

本節涵蓋：

- 設定環境變數
- 設定資源
- 使用顯示器相依的階段作業
- 於登入時執行程序檔
- 回復備份階段作業

▼ 設定環境變數

- 如果要設定整個系統的環境變數，請在設定並匯出變數的 `/etc/dt/config/Xsession.d` 目錄中建立檔案。

例如，如果您想要建立可執行的 ksh 程序檔，`/etc/dt/config/Xsession.d/myvars`，包含：

```
export MYVARIABLE="value"
```

然後，在下次登入時，變數 `MYVARIABLE` 將會設定在每個使用者的環境中。

- 如果要設定個人環境變數，請在 `HomeDirectory/.dtprofile` 中設定變數。

例如：

```
export MYVARIABLE="value"
```

下次登入時，在每個使用者的環境中設定變數 `MYVARIABLE`。

註解 – 階段作業管理者不會自動讀取 `.profile` 或 `.login` 檔案。然而，可以配置它以使用這些檔案；請參閱 第 42 頁“選擇性找出 `.profile` 或 `.login` 程序檔的來源”。

▼ 設定資源

- 如果要設定整個系統的資源，請將資源增加至檔案 `/etc/dt/config/language/sys.resources`。（您可能必須建立此檔案。）

註解 – `.dtprofile` 只支援 `/bin/sh` 或 `/bin/ksh` 語法。

例如，如果您在 `/etc/dt/config/C/sys.resources` 中，指定：

```
AnApplication*resource: value
```

那麼資源 `AnApplication*resource` 將在下次登入時設定在每一個使用者的 `RESOURCE_MANAGER` 屬性中。

- 如果要設定個人資源，請將資源增加至檔案 `HomeDirectory/.Xdefaults` 中。

▼ 設定顯示器特定的資源

您可以在系統上設定所有桌上管理系統使用者的特定顯示器資源。而且，使用者可以設定限制為其自己的階段作業之特定顯示器資源。此項設定使您能夠視使用者使用何種顯示器以登入桌上管理系統來指定特定顯示器。

- 如果要在系統上設定所有桌上管理系統使用者的特定顯示器資源，請建立指定特定顯示器資源的檔案 `/etc/dt/config/language/sys.resources`。
- 如果要設定個人特定顯示器資源，請在 `HomeDirectory/.Xdefaults` 中指定該資源。

您藉由將這些資源附於 `cpp` 條件式敘述中來定義它們。A `DISPLAY displayname` 巨集是根據 `$DISPLAY` 變數的值來定義的。這是藉由轉換所有的 `.` (句點) 與 `:` (冒號) 字元為 `_` (底線)、去除任何螢幕說明以及最後將字首 `DISPLAY_` 加於結果所完成的。

例如，`$DISPLAY` 的 `:0` 將是 `DISPLAY_0`，而 `$DISPLAY` 的 `blanco.gato.com:0.0` 將是 `DISPLAY_blanco_gato_com_0`。結果值可作為階段作業資源檔中的 `cpp` 測試的一部份。例如，如果您在 `/etc/dt/config/C/sys.resources` 中，可以指定：

```
Myapp*resource: 值
#ifdef DISPLAY_blanco_gato_com_0 Myapp*resource: specialvalue1
#endif
#ifdef DISPLAY_pablo_gato_com_0 Myapp*resource: specialvalue2
#endif
```

資源 `MyApp*resource` 將在 `MyApp*resource` 中設定為 `specialvalue1`，若使用者在顯示器 `blanco.gato.com:0` 登入的話；設定為 `specialvalue2`，若使用者在 `pablo.gato.com:0`；登入的話，以及設定為 `value`，若使用者在另一個顯示器上登入的話。

▼ 變更起始階段作業的應用程式

您可以指定替代應用程式，將其視為使用者起始階段作業的一部份啟動。

1. 將 `/usr/dt/config/language/sys.session` 複製到 `/etc/dt/config/language/sys.session` 中。
2. 修改新的 `sys.session` 檔案。
`sys.session` 中每個項目的出現如：

```
dtsmcmd -cmd command_and_options
```

如果要將其它的應用程式視為使用者起始階段作業的一部份啟動，請以完整的路徑名稱指定新的 `sys.session` 項目。例如，如果要將 `/usr/bin/X11/xclock` 視為使用者起始階段作業的一部份啟動，請增加 `xclock` 項目至

```
/etc/dt/config/C/sys.session :  
  
#  
# Start up xclock...  
#  
dtsmcmd -cmd "/usr/bin/X11/xclock  
-digital"
```

▼ 設定特定顯示器的階段作業

使用者可以設定特定顯示器的階段作業以將階段作業調整到特定的顯示器。

- 將 `HomeDirectory/.dt/sessions` 目錄複製到 `HomeDirectory/.dt/` 此處的顯示器是真正的非限定的主機名稱 (`pablo:0` 是有效的，`pablo.gato.com:0` 或 `unix:0` 則是無效的)。

例如，如果要建立顯示器 `pablo.gato.com:0` 的特定顯示器階段作業：

```
cp -r HomeDirectory/.dt/sessions HomeDirectory/.dt/pablo:0
```

當使用者下次登入顯示器 `pablo.gato.com:0` 時，階段作業管理者將會啟動該特定顯示器的階段作業。

在階段作業啟動與登出時執行其它指令

當使用者登入其桌上管理系統階段作業時，他們可以指定啟動其它指令。對於設定未由階段作業管理者儲存的 X 設定值，這是很有用的。例如，使用者可以使用 `xsetroot` 來自訂 `root` (工作區) 指標。另一種用途是啟動無法由階段作業管理者儲存與回復的應用程式。

▼ 在階段作業啟動時執行其它的指令

- 建立包含指令的檔案 `HomeDirectory/.dt/sessions/sessionetc`。

一般而言，此檔案為程序檔且必須有執行許可權。`sessionetc` 中啟動的程序應於背景中執行。

註解 – 請勿使用 `sessionetc` 啟動將透過階段作業管理者自動回復的用戶端。這樣做可能會啟動多個應用程式副本。您可能無法立刻看到副本，因為視窗可能相互堆疊。

▼ 在登出時執行其它指令

`sessionetc` 的姐妹檔案為 `sessionexit`。使用 `sessionexit` 以執行不是由階段作業管理者所處理的階段作業結束時的一些作業。

- 建立包含指令的檔案 `HomeDirectory/.dt/sessions/sessionexit`。

如同 `sessionetc`，此檔案通常是具有執行許可權的程序檔。

▼ 從備份回復階段作業

當階段作業管理者儲存階段作業時，如果使用的是特定顯示器階段作業，階段作業資訊會儲存於 `HomeDirectory/.dt/sessions` 目錄中或 `HomeDirectory/.dt/顯示器目錄` 中。在這些目錄中，階段作業管理者會建立名為 `目前` 或 `主` 的子目錄以儲存個別在目前或主階段作業的資訊。在儲存階段作業資訊之前，階段作業管理者會以該名稱製作前階段作業的備份並將其儲存於 `current.old` 或 `home.old` 中。

1. 從登入螢幕使用單視窗階段作業或指令行登入來登入。
2. 將備份階段作業目錄複製到作用中的名稱。例如，如果要回復備份主階段作業：

```
cp -r HomeDirectory/.dt/sessions/home.old \  
HomeDirectory/.dt/sessions/home
```

可以用相同的方式回復特定顯示器的階段作業。

▼ 調查階段作業啟動問題

- 檢查檔案 `HomeDirectory/.dt/startlog` 中。

階段作業管理者在此檔案中登錄每個使用者的階段作業啟動進度。

階段作業管理者檔案及目錄

- `/usr/dt/bin/Xsession`
- `/usr/dt/config/Xsession.d/*`
- `/usr/dt/bin/dtsession`
- `/usr/dt/bin/dtsession_res`
- *HomeDirectory* `/.dt/sessions/current`
- *HomeDirectory* `/.dt/sessions/home`
- *HomeDirectory* `/.dt/display/current`
- *HomeDirectory* `/.dt/display/home`

第 3 章

疑難排解登入與階段作業啟動的問題

本章描述 Solaris CDE 啟動檔案與可能的 Solaris CDE 啟動問題，並建議啟動問題的解決方案。

- 第 51 頁“登入啟動檔案”
- 第 52 頁“錯誤登錄位置”
- 第 52 頁“使用者啟動檔案”
- 第 53 頁“Solaris CDE 啟動範例”

登入啟動檔案

當 Solaris CDE 登入管理者認證使用者時，會呼叫以下程序檔以啟動桌上管理系統：

```
/usr/dt/bin/Xsession
```

Xsession 呼叫的第一個特定的使用者檔案為 *HomeDirectory/.dtprofile*。

新的使用者第一次登入 Solaris CDE 時，*.dtprofile* 檔案會被複製到使用者的主目錄中。依照預設值，此檔案不執行任何動作。然而，它的確包含許多有關可能會如何編輯它的註釋。使用者可以編輯此檔案以增加特定的使用者環境變數。

註解 – 如同呼叫 *.dtprofile* 的 Xsession 程序檔，此檔案使用 ksh 語法。

一種有用的編輯為取消 *.dtprofile* 檔案最後一行的註釋：

```
DTSOURCEPROFILE=true
```

此行允許使用者的 *HomeDirectory/.login* (適用於 csh 使用者) 或 *HomeDirectory/.profile* (適用於其它 shell 使用者) 被視為啟動程序的一部份搜尋。

錯誤登錄位置

從 Solaris CDE 登入螢幕的選項功能表選擇單視窗階段作業，而不用正常的 Xsession 啟動來查看錯誤登錄及定出可能的使用者點檔案問題。表 3-1 顯示錯誤登錄以及其位置。

表 3-1 錯誤登錄位置

位置	錯誤登錄
<code>/var/dt/Xerrors</code>	在使用者登入之前的 Solaris CDE 登入視窗系統錯誤
<i>HomeDirectory</i> <code>/.dt/startlog</code>	在 X 階段作業、 <code>.dtprofile</code> 、 <code>.login</code> 或 <code>.profile</code> 期間的 Solaris CDE 啟動錯誤
<i>HomeDirectory</i> <code>/.dt/errorlog</code>	在啟動 Xsession 後發生 Solaris CDE 錯誤
<i>HomeDirectory</i> <code>/.dt/sessionlogs</code>	階段作業管理者與視窗管理者錯誤的階段作業登錄目錄

系統主控台視窗中也會顯示一些錯誤。如果系統主控台視窗沒有執行，備用主控台登錄檔案名稱爲 `wscon` 且連接著時間/顯示碼。例如：

```
/usr/tmp/wsconAAAA004EE:0.0
```

使用者啟動檔案

如果要搜尋 `.login`（對於 `csch` 使用者）或 `.profile`（對於 `sh` 或 `ksh` 使用者）檔案，請取消 `.dtprofile` 檔案中最後一行以讀取：

```
DTSOURCEPROFILE=true
```

在大部分的情況中這就是必備的條件了。然而，在某些情況中，您需要修改 `.login` 或 `.profile` 檔案，因為它可能會包含將不適用於 Solaris CDE 登入管理者的指令。如果指令之一有問題，通常是與預期終端機鍵盤輸入資源例如 `stty`、`tset` 或任何“等候輸入”指令有關。



小心 – 如果設定 `.dtprofile` 來搜尋其中有使 shell 故障的問題指令的 `.login` 或 `.profile` 檔案，桌上管理系統啟動將會失敗。因此，沒有桌上管理系統會出現。取而代之的是您將會看到 Solaris CDE 登入螢幕重新顯示。從 `.login` 或 `.profile` 啟動錯誤通常將會記錄在 `HomeDirectory/.dt/startlog` 中。使用單視窗登入階段作業或指令行登入以除錯在 `.login` 或 `.profile` 中的問題指令。

有關問題與其可能的解答之更多資訊，請復習 `.dtprofile` 檔案。一般而言，問題指令與終端機資訊以及控制有關。

Solaris CDE 啟動範例

本節提供您可能會如何編輯下列使用者啟動檔案的範例：

- `.login` (適用於 `cs` 使用者)
- `.profile` (適用於 `sh` 或 `ksh` 使用者)
- `.Xdefaults`

Solaris CDE 啟動程序定義 `.login` 或 `.profile` 程序檔中名為 `DT` 的 shell 變數，因此可以在 Solaris CDE 啟動程序期間完成檢查。此防止執行與終端機有關的指令，例如來自執行的 `tty` 與 `stty`。請參照下列程序檔範例：

```
.login (C shell)
    if ( ! ${?DT} ) then
        stty erase `^h`
    endif

.profile (sh or ksh)
    if [ ! "$DT" ]; then
        stty erase `^h`
    fi
```

當 `DT` 未定義時，其餘有關 `DT` 的語法會防止來自 C shell 的警告。當使用者從習慣用法的以文字為基礎的主控台登入提示符號登入時，並未定義 `DT`。有關設定點檔案的更多資訊，請參閱 `HomeDirectory/.dtprofile` 檔案。

註解 – 您在 `.Xdefaults` 中指定 Solaris CDE `tty` 設定值。

在 Solaris CDE 啓動期間，爲特定的使用者資源搜尋使用者的 .Xdefaults 檔案。例如，ttyModes 確保使用者最喜愛的 tty 設定值是用於終端機模擬視窗中，如 dtterm 與 xterm。下行顯示在 .Xdefaults 檔案中的典型 ttyModes 設定值：

```
*ttyModes: erase ^H intr ^C kill ^U start ^Q stop ^S susp ^Z`
```

註解 – 最喜愛的 Solaris CDE 資源與預設設定值不同。

桌上管理系統視窗管理者的圖示位置是預設設定值的範例之一。在這種情況下，.Xdefaults 檔案中的下行顯示預設圖示位置設定值：

```
Dtwm*iconPlacement: 右上方
```

第 4 章

增加與管理應用程式

應用程式管理者是可供使用者使用的應用程式之桌上管理系統儲存區。

- 第 55 頁“應用程式管理者的結構”
- 第 58 頁“增加應用程式至應用程式管理者”
- 第 61 頁“建立及管理一般應用程式群組”
- 第 62 頁“修改用於尋找應用程式的搜尋路徑”
- 第 63 頁“一般應用程式管理者管理”
- 第 64 頁“變更文字編輯程式與終端機模擬器”

應用程式管理者的結構

應用程式管理者的頂層通常包含目錄。每一個這些目錄及其內容稱為**應用程式群組**。

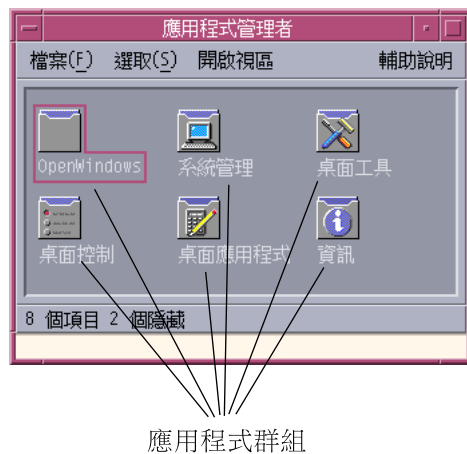


圖 4-1 應用程式管理者中的應用程式群組

應用程式群組與其內容是由局部與整個網路的多重位置所收集的。

應用程式管理者的目錄位置

在檔案系統中，應用程式管理者是目錄 `/var/dt/appconfig/appmanager/login-hostname-display`。每次使用者登入時就會動態地建立目錄。

例如，如果使用者 `ronv` 從顯示器 `wxyz:0` 登入，就會建立應用程式管理者目錄 `/var/dt/appconfig/appmanager/ronv-wxyz-0`。

應用程式管理者如何尋找與收集應用程式

應用程式管理者是藉由收集局部與遠端應用程式群組所建立的。應用程式群組是從位於沿著應用程式搜尋路徑的目錄所收集來的。

預設應用程式搜尋路徑是由位置組成，顯示如：表 4-1

表 4-1 預設應用程式搜尋路徑位置

範圍	位置
內建	<code>/usr/dt/appconfig/appmanager/language</code>
整個系統的	<code>/etc/dt/appconfig/appmanager/language</code>
個人	<code>HomeDirectory / .dt/appmanager</code>

如果要建立應用程式管理者的頂層，在登入時間，建立由位於應用程式搜尋路徑到應用程式管理者目錄 `/var/dt/appconfig/appmanager/login-hostname-display` 的連結。收集作業是由桌上管理系統公用程式 `dtappgather` 所完成的，此公用程式是在使用者成功地登入之後由登入管理者所自動執行的。

例如，桌上管理系統提供內建應用程式群組：

```
/usr/dt/appconfig/appmanager/language/Desktop_Tools
```

在登入時，建立符號連結到：

```
/var/dt/appconfig/appmanager/登入-主機名稱-顯示器/Desktop_Tools
```

應用程式搜尋路徑可以包括遠端目錄。這就提供一個從位於整個網路的系統來收集應用程式群組的方法。有關更多的資訊，請參閱第 62 頁“增加應用程式伺服器至應用程式搜尋路徑”。

收集應用程式時的優先順序規則

複製隨著搜尋路徑而存在之處，個人應用程式群組對於整個系統的群組有優先順序，而整個系統的群組對於內建群組有優先順序。例如，如果

`/usr/dt/appconfig/appmanager/C/Desktop_Tools` 與
`/etc/dt/appconfig/appmanager/C/Desktop_Tools` 均存在，將會使用 `/etc` 下的應用程式群組。

預設桌上管理系統所提供的應用程式群組

未自訂的桌上管理系統提供四種應用程式群組。

- 桌面應用程式
- 桌面工具
- 桌面控制
- 資訊
- 系統管理

如何收集應用程式群組的範例

圖 4-2 顯示包含各種應用程式群組的應用程式管理者視窗。表 4-2 顯示收集應用程式群組的目錄。



圖 4-2 典型的應用程式管理者視窗

表 4-2 應用程式群組的來源：

名稱	收集的目錄
CAD_App	/net/ApServA/etc/dt/appconfig/appmanager/C/CAD_App
DrawingApp	/etc/dt/appconfig/appmanager/C/DrawingApp
Desktop_Apps	/usr/dt/appconfig/appmanager/C/Desktop_Apps
Desktop_Tools	/usr/dt/appconfig/appmanager/C/Desktop_Tools
Information	/usr/dt/appconfig/appmanager/C/Information
OpenWindows	/usr/dt/appconfig/appmanager/C/Information
System_Admin	/etc/dt/appconfig/appmanager/C/System_Admin
MySpreadSheet	/users/anna/.dt/appmanager/MySpreadSheet
Media_Tools	/etc/dt/appconfig/appmanager/C/Media_Tools

如果已自訂資訊或系統管理應用程式群組，它們將會從 /etc/dt/appconfig/appmanager/C 收集。

因為已將名為 ApServA 的系統增加至應用程式搜尋路徑，所以 CAD_App 群組已被收集 (請參閱。第 62 頁“增加應用程式伺服器至應用程式搜尋路徑”) MySpreadSheet 是個人應用程式群組，只供使用者 anna 使用。

增加應用程式至應用程式管理者

如果應用程式已被增加至應用程式管理者，應用程式群組中會有一個啟動該應用程式的圖示。

許多應用程式都提供應用程式群組。應用程式群組是應用程式管理者頂層的目錄，它包含應用程式圖示以及其它與應用程式有關的檔案。

有些應用程式可能沒有自己的應用程式群組。相反地，啟動應用程式的圖示會位於一般應用程式群組中。例如，您可以建立稱為“遊戲”的空白應用程式群組，這樣一來，您就將它視為一個安裝在系統上所有遊戲的儲存區來使用。

增加應用程式至應用程式管理者的方法

要增加應用程式至應用程式管理者有兩種方法：

- 註冊應用程式
- 不需註冊應用程式而增加應用程式圖示

註冊應用程式

應用程式註冊提供完整的應用程式整合。

已註冊的應用程式：

- 有自己的應用程式群組。
- 在單一位置下收集其桌上管理系統配置檔案。桌上管理系統配置檔案的此群組稱為**註冊套裝軟體**。
- 可能有已經註冊的輔助說明卷次。

要使應用程式成為已註冊的有兩種方法：

- 當您安裝智慧型桌上管理系統應用程式時，註冊會自動發生。請參閱 第 60 頁“增加智慧型桌上管理系統應用程式至應用程式管理者”。
- 現存應用程式可藉由建立註冊套裝軟體來註冊。請參閱 第 60 頁“如果要註冊現存或非桌上管理系統的智慧型應用程式”。

註冊套裝軟體的使用使得在桌上管理系統上管理應用程式更加地容易。註冊套裝軟體是建立於檔案系統內某處，而非桌上管理系統配置檔案所用的位置。

不須使用註冊套裝軟體來增加應用程式

當您想要應用程式管理者只包含一個啟動應用程式的圖示時，此為增加應用程式較好的方法。

不需使用註冊套裝軟體來增加應用程式：

- 可以有自己的應用程式群組，但通常其圖示是放在現存應用程式群組中
- 讓其桌上管理系統配置檔案直接放在隨著桌上管理系統搜尋路徑的位置中。

請參閱 第 60 頁“增加應用程式圖示至現存應用程式群組”。

▼ 增加智慧型桌上管理系統應用程式至應用程式管理者

智慧型桌上管理系統應用程式是當安裝應用程式時，自動註冊至應用程式管理者中的應用程式。應用程式的檔案集包括桌上管理系統所需要的註冊套裝軟體。

1. 使用應用程式所提供的指令來安裝應用程式。
2. 當安裝完成時，連按兩下桌面工具應用程式群組中的重新載入應用程式。
3. 驗證安裝已完成：
 - a. 開啓應用程式管理者並檢查新應用程式群組的存在。
 - b. 如果要開啓應用程式，開啓應用程式群組並連按兩下應用程式的圖示。

▼ 如果要註冊現存或非桌上管理系統的智慧型應用程式

這是將應用程式與桌上管理系統完全整合的較好方法。

桌上管理系統提供一種工具 `dtappintegrate`，以建立註冊套裝軟體檔案與桌上管理系統搜尋路徑上的目錄之間的連結。

桌上管理系統註冊在中講述 第 5 章。

▼ 增加應用程式圖示至現存應用程式群組

此程序解釋如何增加應用程式圖示至現存應用程式群組。

例如，桌上管理系統提供名為系統管理的應用程式群組，其已保留給與管理系統有關的各種應用程式以及程序檔。如果您有使用者經常執行的程序檔，您可能會想要讓使用者能夠藉由連按兩下系統管理應用程式群組中的圖示來執执行程序檔。

1. 使用建立動作以建立應用程式的動作定義。
有關建立動作的更多資訊，請參閱 第 11 章。
2. 在目錄中，為應用程式群組建立與動作名稱相同的執行檔案。檔案的內容是不相干的。
例如，如果您已建立了執行系統管理工具，名為“清除”的動作，您會建立可執行的檔案：

```
/etc/dt/appconfig/appmanager/language /System_Admin/Cleanup
```

建立及管理一般應用程式群組

一般應用程式是與特定的應用程式產品無關的應用程式群組（目錄）。例如，內建的桌面工具應用程式群組是一般群組，它包含大量應用程式圖示，雖然與單一產品相關但並非產品的一部分。

您可以建立其它的一般應用程式群組。例如，您可能會想要建立名為遊戲的群組，來將系統上各種可供使用的遊戲聚集在一起。

一般應用程式群組可以是整個系統的或是個人範圍的。

▼ 建立整個系統的一般應用程式群組

1. 登入為 **root**。
2. 在 `/etc/dt/appconfig/appmanager/language` 下建立目錄。
目錄名稱會變成應用程式群組的名稱。
3. 在桌面工具應用程式群組中連按兩下重新載入應用程式。

▼ 建立個人的一般應用程式群組

1. 在 `HomeDirectory /.dt/appmanager` 中建立目錄。
目錄名稱變成應用程式群組的名稱。
2. 在桌面工具應用程式群組中連按兩下重新載入應用程式。

▼ 自訂內建的應用程式群組

1. 登入為 **root**。
2. 如果應用程式群組位於 `/usr/dt/appconfig/appmanager/language` 中，請將應用程式群組複製到 `/etc/dt/appconfig/appmanager/language`。
例如，以下指令複製桌面工具應用程式群組：

```
cp -r /usr/dt/appconfig/appmanager/C/Desktop_Tools \  
/etc/dt/appconfig/appmanager/C
```

應用程式群組的新副本對於內建的版本將會有優先順序。
3. 修改應用程式群組的副本。例如，您可以增加新的動作檔案（與動作名稱相同的可執行的檔案）。

4. 如果要看到變更，請登出後再登入。

修改用於尋找應用程式的搜尋路徑

修改應用程式搜尋路徑的主要理由是增加應用程式伺服器。當您增加應用程式伺服器至搜尋路徑時，應用程式管理者會收集所有伺服器的整個系統之應用程式群組。

有關應用程式搜尋路徑的更多資訊，請參閱 第 137 頁“應用程式搜尋路徑”。

預設搜尋路徑

預設應用程式搜尋路徑顯示在 表 4-3。

表 4-3 預設應用程式搜尋路徑目錄

範圍	搜尋路徑目錄
個人	<i>HomeDirectory</i> / .dt / appmanager
整個系統的	<i>/etc/dt/appconfig/appmanager/language</i>
內建	<i>/usr/dt/appconfig/appmanager/language</i>

增加應用程式伺服器至應用程式搜尋路徑

除了修改應用程式搜尋路徑之外，您可能需要執行其它的配置工作以啟動與應用程式伺服器的通信。請參閱 第 122 頁“管理應用程式服務”。

▼ 設定整個系統的應用程式搜尋路徑

1. 登入為 **root**。
2. 如果檔案 */etc/dt/config/Xsession.d/0010.dtpaths* 不存在，請藉由複製 */usr/dt/config/Xsession.d/0010.dtpaths* 來建立它。
3. 開啓 */etc/dt/Xsession.d/0010.paths* 來做編輯。增加或編輯設定與匯出 **DTSPSYSAPPHOSTS** 變數：

```
export DTSPSYSAPPHOSTS=主機名稱 : [,hostname]
```

例如，以下這一行將增加系統 ApServA 至應用程式搜尋路徑：

```
export DTSPSYSAPPHOSTS=ApServA:
```

4. 通知系統上所有使用者，他們必須登出然後再登入以使變更生效。

▼ 設定個人的應用程式搜尋路徑

1. 開啓 `HomeDirectory / .dtprofile` 以編輯。
2. 增加或編輯設定與匯出 `DTSPUSERAPPHOSTS` 變數：

```
export DTSPUSERAPPHOSTS=主機名稱 :[,hostname]
```

例如，以下這一行將增加系統 `ApServB` 與 `ApServC` 至應用程式搜尋路徑：

```
export DTSPUSERAPPHOSTS=ApServB:,ApServC:
```

3. 登出然後再登入。

一般應用程式管理者管理

一般的應用程式管理者管理工作包括：

- 移除應用程式
- 在階段作業期間重新讀取應用程式的資料庫

▼ 移除應用程式

如果已使用 `dtappintegrate` 工具來註冊應用程式，您也可以使用 `dtappintegrate` 來反向程序。當應用程式被取消註冊時，其應用程式群組會從應用程式管理者移除，而且其動作、資料類型、圖示與輔助說明也不再供人使用。

1. 登入為 `root`。
2. 執行指令：

```
dtappintegrate -s app_root -u
```

▼ 在階段作業期間更新應用程式管理者

如果您增加應用程式並且希望這些變更立刻生效，您必須重新建立應用程式管理者。

- 開啓桌面工具應用程式群組並連按兩下重新載入應用程式。

當應用程式增加至應用程式伺服器時，重新載入應用程式對於更新應用程式管理者是很有用的。但是，重新載入應用程式無法檢查出已從應用程式伺服器移除的應用程式，或是已從一個位置移動到另一位置的應用程式。當使用者登出並再登入時，這些變更就會生效。

變更文字編輯程式與終端機模擬器

文字編輯程式與終端機模擬器應用程式均可藉由在前方面板中選擇控制，或是連按兩下應用程式管理者中的圖示來啟動。

這些應用程式也可由其它的桌上管理系統活動來啟動。

- 當使用者在檔案管理者中選擇文字檔案並從「所選項目」功能表選擇開啓時，文字編輯程式應用程式即開啓。預設文字編輯程式為 `dtpad`。
- 當使用者從檔案管理者的檔案功能表選擇開啓終端機，或是當動作開啓終端機模擬器視窗時，終端機模擬器即執行。預設終端機模擬器為 `dtterm`。

您可以配置桌上管理系統以在這些情況中使用不同的文字編輯程式或終端機模擬器應用程式。

▼ 如果要變更預設文字編輯程式或終端機模擬器

1. 如果變更是整個系統的，則登入為 `root`。
2. 為新的文字編輯程式或終端機模擬器應用程式建立動作。
 - 您可以使用建立動作應用程式。圖 4-3 顯示填入了名為 `TextPad` 的應用程式的建立動作視窗。有關建立動作的更多資訊，請參閱 第 11 章。

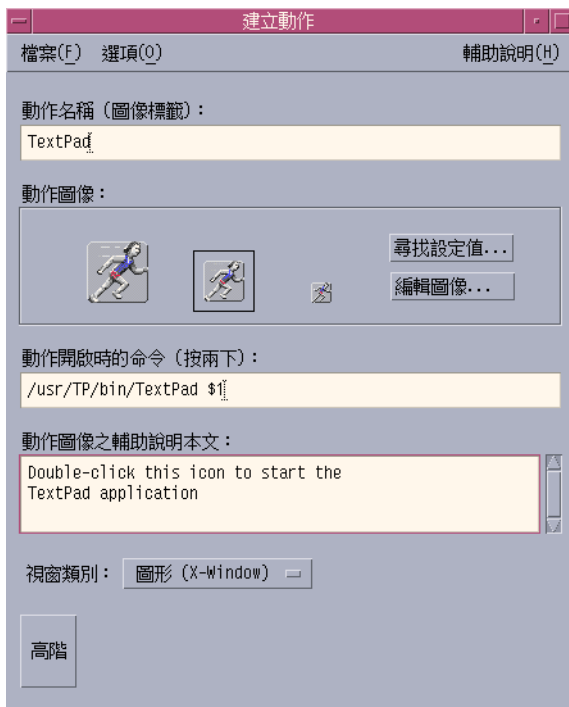


圖 4-3 建立動作視窗

- 或，您可以人工建立動作定義；例如：.

```

ACTION TextPad
{
    LABEL                TextPad
    TYPE                 COMMAND
    WINDOW_TYPE         NO_STDIO
    EXEC_STRING          /usr/TP/bin/TextPad %(File)Arg_1%
    DESCRIPTION         Double-click this icon to start the \
                        TextPad application.
}

```

有關人工建立動作定義的資訊，請參閱 第 12 章。

3. 請將包含新動作的配置檔案放在正確的目錄中：

- 整個系統的： /etc/dt/appconfig/types/language
- 個人的： *HomeDirectory* / .dt/types

4. 如果它並非存在，建立合適的 *user-prefs.dt* 檔案之方式為藉由將 /usr/dt/appconfig/types/language /user-prefs.dt 複製到：

- 整個系統的： /etc/dt/appconfig/types/language directory
- 個人的： *HomeDirectory* / .dt/types 目錄

5. 在整個系統的或個人的 `user-prefs.dt` 檔案中編輯文字編輯程式或終端機動作。修改 **MAP_ACTION** 用來對應該動作至新動作行。

例如，變更此行：

```
MAP_ACTION Dtpad
```

至：

```
MAP_ACTION TxtPd
```

6. 儲存 `user-prefs.dt` 檔案。
7. 連按兩下桌面工具應用程式群組中的重新載入動作以重新載入動作資料庫。

第 5 章

註冊應用程式

本章描述如何建立應用程式的註冊套裝軟體，以及如何註冊桌上管理系統上的應用程式。

- 第 68 頁“應用程式註冊的概述”
- 第 70 頁“註冊應用程式的一般步驟”
- 第 71 頁“第 1 步：修改字型與色彩資源”
- 第 72 頁“第 2 步：建立桌上管理系統應用程式 Root”
- 第 73 頁“第 3 步：建立註冊套裝軟體目錄”
- 第 75 頁“第 4 步：建立應用程式的動作與資料類型”
- 第 77 頁“第 5 步：將輔助說明檔案置於註冊套裝軟體中”
- 第 78 頁“第 6 步：建立應用程式的圖示”
- 第 79 頁“第 7 步：建立應用程式群組”
- 第 84 頁“第 8 步：使用 `dtappintegrate` 註冊應用程式”
- 第 86 頁“建立註冊套裝軟體的範例”

當應用程式完整註冊於桌上管理系統上時，它有：

- 它自己在應用程式管理者頂層的應用程式群組
- 啟動應用程式的動作。該動作是由應用程式群組中的圖示所代表
- 其資料檔案的選擇性資料類型

應用程式註冊對應用程式而言是非侵入性的作業：

- 它本身不包含可執行的應用程式的修改。因此，您可以在系統上註冊現存應用程式。
- 它不需要將任何應用程式的發送檔案（例如可執行的與 `app-defaults`）移動到其它的檔案位置。
- 它可以很輕易地還原。用於註冊應用程式的 `dtappintegrate` 工具提供反向程序的指令行選項。

您將會想要建立註冊套裝軟體的，如果您是：

- 想要在桌上管理系統註冊現存應用程式的系統管理者
- 想要為智慧型桌上管理系統應用程式建立安裝套裝軟體的軟體程式設計者

應用程式註冊的概述

本節解釋：

- 應用程式註冊的目的
- 由應用程式註冊所提供給您的應用程式功能

註解 – 有關顯示如何註冊現存應用程式的詳細範例，請參閱 第 86 頁“建立註冊套裝軟體的範例”。

應用程式註冊提供的功能

應用程式註冊提供圖形的方法給使用者以：

- 尋找您的應用程式。
安裝時，您的應用程式“註冊”至應用程式管理者，並有其自己的應用程式群組。



圖 5-1 在應用程式管理者頂層的應用程式群組

- 啓動您的應用程式。
您應用程式的應用程式群組包含一個，使用者可以連按兩下以啓動您的應用程式的圖示。

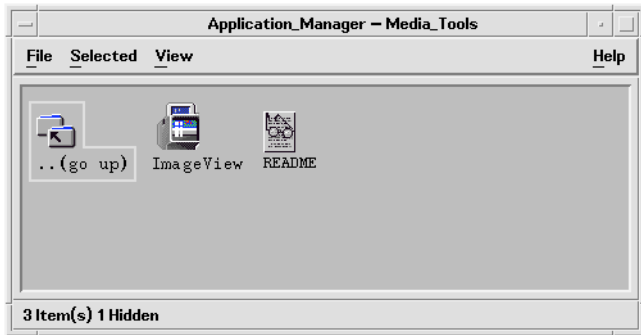


圖 5-2 包含可啟動應用程式的圖示之應用程式群組

- 識別與處理資料檔案。應用程式的資料檔案在檔案管理者中將會有一個唯一的圖示。使用者可以使用資料檔案圖示以：
 - 啟動（開啓）應用程式
 - 列印資料檔案

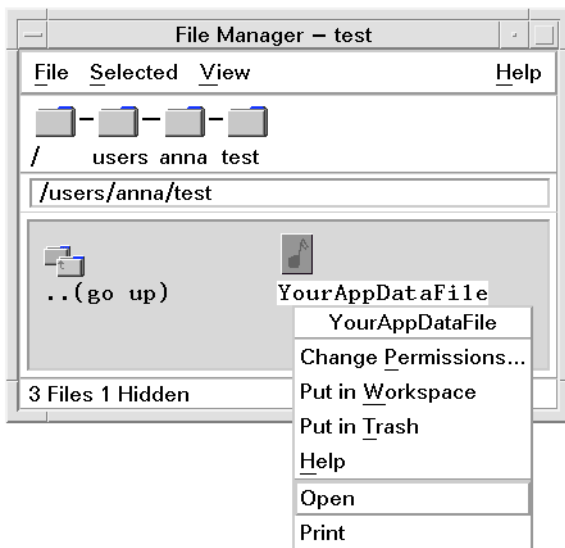


圖 5-3 資料檔案的即現式功能表包含“開啓”與“列印”

- 執行其它的作業，例如郵寄、壓縮、查看或播放（聲訊）資料

應用程式註冊的目的

已註冊的桌上管理系統應用程式有由桌上管理系統所使用的特定配置檔案，以提供應用程式的使用者介面：

- 動作與資料類型定義檔案
- 圖示影像（像素對映或點陣圖）檔案
- 建立應用程式群組的目錄與檔案
- 選擇性的桌上管理系統輔助說明檔案與前方面板定義檔案

爲了認出這些檔案並由桌上管理系統所使用，它們必須在由桌上管理系統搜尋路徑所指定的特定目錄中。

當其配置檔案分散在眾多的目錄之中時，要管理應用程式是很困難的。因此，桌上管理系統允許應用程式在單一目錄之下，收集其所有的桌上管理系統配置檔案。此檔案群組稱爲 **註冊套裝軟體**。

如果應用程式是智慧型桌上管理系統，它會供應註冊套裝軟體作爲其安裝套裝軟體的一部分。如果您是自己建立配置檔案的系統管理者，您便可以自己建立註冊套裝軟體。

註冊套裝軟體中的配置檔案不供桌上管理系統使用，因爲它們並非位於正確的搜尋路徑目錄中。將這些檔案放在正確位置中的程序稱爲 **註冊或整合** 應用程式。

桌上管理系統提供一種工具 `dtappintegrate`，其藉由建立正確的搜尋路徑目錄中檔案的象符號連結表示方式來執行註冊。

許多智慧型桌上管理系統應用程式將會在安裝程序期間自動執行 `dtappintegrate`。如果您是整合現存應用程式的系統管理者，您可以在建立註冊套裝軟體之後自己執行它。

只要應用程式一在系統的桌上管理系統上註冊，應用程式就可供系統上所有的使用者使用。如果系統配置爲桌上管理系統應用程式伺服器，應用程式也可供整個網路上的其它系統使用。

`dtappintegrate` 工具有藉由中斷連結以反向程序的指令行選項。這使得要從應用程式管理者移除應用程式變得更加容易，因此它可以移動至不同的應用程式伺服器或更新。

註冊應用程式的一般步驟

註解 – 有關使用這些步驟以建立應用程式套裝軟體的詳細範例，請參閱 第 86 頁“建立註冊套裝軟體的範例”。

1. 修改任何設定字型與色彩的應用程式資源。否則，桌上管理系統的動態字型與色彩將不會正確地運作。

請參閱 第 71 頁“第 1 步：修改字型與色彩資源”。

2. **建立應用程式 root 位置。**
請參閱 第 72 頁“第 2 步：建立桌上管理系統應用程式 Root”。
3. **建立在應用程式 root 之下的目錄結構。**
請參閱 第 73 頁“第 3 步：建立註冊套裝軟體目錄”。
4. **建立應用程式的動作與資料類型。**
請參閱 第 75 頁“第 4 步：建立應用程式的動作與資料類型”。
5. **將輔助說明檔案放在合適的目錄中。**
請參閱 第 77 頁“第 5 步：將輔助說明檔案置於註冊套裝軟體中”。
6. **建立應用程式的圖示**
請參閱 第 78 頁“第 6 步：建立應用程式的圖示”。
7. **建立應用程式的應用程式群組。**
請參閱 第 79 頁“第 7 步：建立應用程式群組”。
8. **使用 dtappintegrate 來註冊應用程式。**
請參閱 第 84 頁“第 8 步：使用 dtappintegrate 註冊應用程式”。

第 1 步：修改字型與色彩資源

註解 – 有關修改應用程式資源的範例，請參閱 第 86 頁“建立註冊套裝軟體的範例”。

桌上管理系統提供設定與處理介面字型和視窗色彩的機制。為了使應用程式正確地使用這些機制，您可能需要修改應用程式的 `app-defaults` 檔案。

修改字型資源

註解 – 本節應用於使用 OSF/Motif 1.2™（或較新的版本）建立的應用程式。樣式管理者無法設定使用較早版本的 OSF/Motif 寫入的應用程式介面字型。

如果應用程式沒有指定特定的應用程式字型，桌上管理系統樣式管理者將會設定使用 OSF/Motif 1.2（或較新的版本）建立的應用程式介面字型。

樣式管理者提供兩種字型：

系統字型—由像標籤、功能表與按鈕之類的系統區域使用

使用者字型—用於像文字欄位之類的可編輯區域

每種字型都提供有七種大小，在字型對話框中從 1 標到 7。樣式管理者字型經由 `/usr/dt/app-defaults/language/Dtstyle` 中所設定的樣式管理者資源連接至系統上的實際字型。

如果您希望應用程式使用樣式管理者字型，您應該刪除任何介面指定字型的應用程式資源。桌上管理系統將會自動適當地設定應用程式的資源：

`FontList`—設定為系統字型

`XmText*FontList`—設定為使用者字型

`XmTextField*FontList`—設定為使用者字型

修改色彩資源

樣式管理者提供動態變更應用程式色彩的能力。應用程式必須是 OSF/Motif 1.1 或 1.2 用戶端。以其它工具箱所寫入的用戶端無法動態變更色彩；用戶端重新啟動時，色彩變更會生效。

使用由桌上管理系統所提供的動態色彩的最簡單方法就是移除任何應用程式背景與前景色彩的色彩資源。

第 2 步：建立桌上管理系統應用程式 Root

註解 – 關於為應用程式建立桌上管理系統應用程式根目錄的范例，請參閱第 2 步。。

應用程式的註冊套裝軟體檔案被聚集於稱為應用程式 `root` 或 `app_root`。用於桌上管理系統配置檔案的 `app_root` 目錄可以是應用程式的安裝 `app_root` 的相同目錄，或某其它的位置。

例如，假設應用程式安裝於目錄 `/usr/BTE` 之下。此相同的目錄可用作桌上管理系統配置檔案的 `app_root`。然而，如果您正在整合現存非桌上管理系統智慧型應用程式，請建立不同的桌上管理系統 `app_root` 目錄。此將會防止您所建立的配置檔案在您更新應用程式時被重寫。

例如，系統管理者可能會想要建立目錄 `/etc/desktop_approots/BTE` 做為桌上管理系統 `app_root` 目錄。

第 3 步：建立註冊套裝軟體目錄

註解 – 關於為應用程式建立註冊套裝軟體目錄的范例，請參閱第 3 步。

註冊套裝軟體是由桌上管理系統所使用的桌上管理系統配置檔案的群組，用來提供應用程式的圖形介面。

註冊套裝軟體內容

桌上管理系統配置檔案包括：

- 動作與資料類型定義檔案
- 圖示影像檔案
- 應用程式群組目錄與其內容
- 選擇性的：輔助說明資料檔案與前方面板配置檔案

註冊套裝軟體是在稱為應用程式 `root` 或 `app_root` 的頂層目錄下聚集而成的。

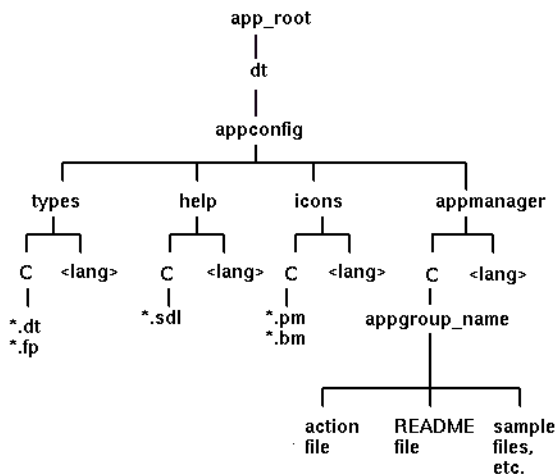


圖 5-4 在應用程式根目錄之下的註冊套裝軟體

在 `app_root/dt/appconfig` 目錄之下的配置欄位之主要種類如下表 5-1。

表 5-1 配置欄位主要種類

子目錄	內容
類型	動作與資料類型定義檔案
輔助說明	桌上管理系統輔助說明檔案
圖示	由應用程式的動作與資料類型所使用的點陣圖與像素對映影像檔案
appmanager	建立應用程式群組的目錄與內容

每一種主要種類中都有語言相依的檔案子目錄。預設語言檔案放在 c 目錄中。

▼ 如果要建立註冊套裝軟體

- 建立這些目錄。如果您正提供語言相依的配置檔案，請為每種語言建立分開的目錄。如果您只供應一種語言，請將檔案放在 c 目錄中。

- `app_root / dt / appconfig / types / language`
- `app_root / dt / appconfig / help / language`
- `app_root / dt / appconfig / icons / language`
- `app_root / dt / appconfig / appmanager / language / appgroup_name`，其中 `appgroup_name` 是應用程式群組的名稱。

例如，圖 5-5 顯示包含其 `appgroup_name` 為 “Media_Tools” 的群組的應用程式管理者



圖 5-5 應用程式管理者頂層的應用程式群組

`dtappintegrate` 工具只在 類型、輔助說明、圖示 與 `appmanager` 目錄中的桌上管理系統配置檔案進行操作。應用程式的可執行二進位、`app-defaults` 與訊息目錄檔案是分開管理的。

第 4 步：建立應用程式的動作與資料類型

註解 – 關於為應用程式建立動作和資料類型的范例，請參閱第 4 步。

動作與資料類型提供應用程式的使用者介面。

- 動作提供使用者介面指令以啟動應用程式。
- 資料類型提供應用程式資料檔案的自訂外觀與行爲。

應用程式所需的動作與資料類型

典型的應用程式會要求下列的動作與資料類型定義：

- 開啓應用程式的動作。
- 您的應用程式的資料檔案的資料類型。如果您建立一種資料類型，您也將會想要建立：
 - 您的應用程式的資料檔案開啓動作
 - 您的應用程式的資料檔案列印動作
- 應用程式群組的資料類型（請參閱第 80 頁“配置應用程式群組以使用唯一的圖示”）。

有關如何在桌上管理系統中使用動作與資料類型的介紹，請參閱第 10 章。

動作與資料類型定義配置檔案的位置

動作與資料類型定義於配置檔案中。包含動作與資料類型定義的檔案的唯一命名要求就是它們必須是 `.dt` 字尾。按慣例，您可能會想要命名檔案為 `action_name .dt` 或 `application_name .dt`。

將包含動作與資料類型的檔案放在目錄 `app_root /dt/appconfig/types/language` 中的應用程式 `root` 之下。預設語言是 `C`。

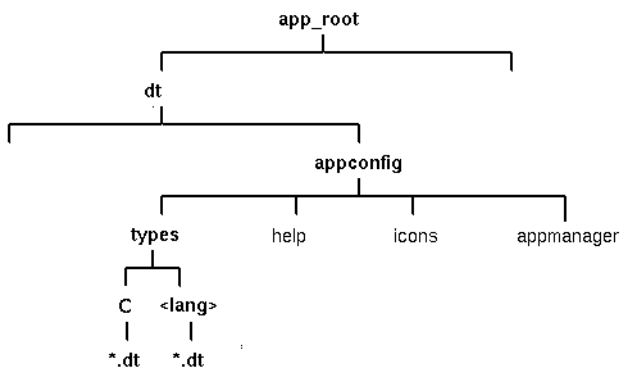


圖 5-6 動作與資料類型定義檔案

建立動作與資料類型的方法

您可以用下列兩種方法任一來建立應用程式的動作與資料類型定義：

- 使用建立動作工具。
建立動作會用您填入的文字欄位提供簡單易用的介面。然而，此工具有特定的限制。
- 人工建立定義。
此需要您學習建立定義的語法，但提供完整功能範圍的存取。

▼ 如果要使用建立動作來建立動作與資料類型

此程序使用建立動作公用程式以建立應用程式的動作與資料類型。

有關建立動作的更多資訊，請使用其線上輔助說明或參閱 第 11 章。

1. 開啓桌面應用程式群組並連按兩下建立動作。
2. 使用建立動作來建立應用程式與其資料類型的動作與資料類型定義。
由建立動作所建立的配置檔案將會寫入至 `HomeDirectory/.dt/type/action_name.dt`。動作檔案（具有與動作相同名稱的可執行檔案）是放在您的主目錄中。
3. 使用您的主目錄中所建立的動作檔案來測試動作。
4. 將動作定義檔案 `HomeDirectory/.dt/type/action_name.dt` 複製到 `app_root/dt/appconfig/types/language` 目錄。
5. 在建立應用程式群組目錄之後（請參閱 第 79 頁“第 7 步：建立應用程式群組”），將動作檔案 `HomeDirectory/action_name` 複製到 `app_root/dt/appconfig/appmanager/language/appgroup_name` 目錄中。

▼ 如果要人工建立動作與資料類型

- 建立包含應用程式的動作與資料類型定義的配置檔案。

動作與資料類型定義檔案必須遵循命名慣例 *name .dt*。

您可以將您所有的動作與資料類型定義放在一個檔案中，或是將它們分散在多重檔案之中。對於每個檔案，請使用系統管理者將會輕易地與您的應用程式連接的檔案名稱。

動作與資料類型名稱必須是一個字（無內含的空間）。您可以使用底線字元。按慣例，動作或資料類型名稱的第一個字母是大寫的。請勿使用現存動作名稱或檔案名稱。使用進階使用者與系統管理者將會輕易地與您的應用程式連接的名稱。

如果您想要讓應用程式的圖示標上與動作名稱不同的名稱，請將 LABEL 欄位包含在動作定義中。

有關建立動作與資料類型的更多資訊，請參閱：

- 第 10 章
- 第 11 章
- 第 12 章
- 第 13 章

第 5 步：將輔助說明檔案置於註冊套裝軟體中

註解 – 關於在註冊套裝軟體中加入輔助說明檔案的范例，請參閱第 5 步。。

如果應用程式包括桌上管理系統輔助說明卷次（以桌上管理系統輔助說明設計者工具箱建立的輔助說明卷次），輔助說明卷次主檔案（*.sdl）應放在目錄 *app_root* /*appconfig/help/language* 中。

由輔助說明檔案所使用的圖形通常是放在 圖形 子目錄中。圖形必須是位於在建立輔助說明卷次時與主輔助說明卷次（*.sdl）檔案相對的相同目錄中。

如果應用程式不提供輔助說明卷次，若是您有輔助說明設計者工具箱，您就可以建立一個。

有兩個輔助說明卷次整合的層次：

- 全部整合。
當桌上管理系統輔助說明全部整合時，輔助說明卷次可以由應用程式存取—例如，由關於項目輔助說明與輔助說明功能表。全部整合包含可執行應用程式的修改。
- 部分整合。
當桌上管理系統部分整合時，它可以從輔助說明管理員的頂層提供使用。然而，您無法從應用程式的視窗存取輔助說明卷次。您也可以提供動作以從應用程式群組存取輔助說明。以下範例動作顯示位於輔助說明主檔案 *MyApp.sdl* 中的輔助說明卷次：

```

ACTION OpenMyAppHelp
{
    LABEL           MyAppHelp
    ARG_COUNT       0
    TYPE            COMMAND
    WINDOW_TYPE     NO_STDIO
    EXEC_STRING     /usr/dt/bin/dthelpview -helpVolume MyApp
    DESCRIPTION     Displays help for the MyApp application.
}

```

第 6 步：建立應用程式的圖示

註解 – 關於建立應用程式圖示檔案的范例，請參閱第 6 步。。

桌上管理系統提供動作、資料類型與應用程式群組的預設圖示。然而，您或許將會想要建立應用程式的唯一圖示。

將圖示放在目錄 `app_root /dt/appconfig/icons/language` 中。

桌上管理系統所需的圖示

應用程式在桌上管理系統上使用這些圖示影像：

- **動作圖示**。這是使用者連按兩下以啟動您的應用程式（動作）的圖示。參考其於啟動應用程式的動作 `ICON` 欄位中。
供應三種大小：小型、中型與大型。
- **資料類型圖示**。此圖示是用於代表檔案管理者中應用程式的資料檔案。參考其於資料類型定義的 `ICON` 欄位中。
如果您的應用程式支援多重資料類型，您應該提供不同的圖示給每種資料類型。
供應兩種大小：小型與中型。
- **應用程式群組圖示**。這是代表在應用程式管理者頂層的目錄圖示。參考其於應用程式群組的資料類型定義 `ICON` 欄位中。（請參閱第 79 頁“第 7 步：建立應用程式群組”）
供應兩種大小：小型與中型。
您可能需要供應每個圖示的像素對映與點陣圖兩種版本以支援彩色（八位元的與更大的）以及單色（少於八位元）顯示器。

表 5-2 圖示檔案的命名慣例

大小	像素空間	點陣圖名稱	像素對映名稱
小型	16 乘 16	基本名稱 .t.bm	基本名稱 .t.pm
中型	32 乘 32	基本名稱 .m.bm	基本名稱 .m.pm
大型	48 乘 48	基本名稱 .l.bm	基本名稱 .l.pm

如果您不提點陣圖檔案，桌上管理系統會對應圖素檔案的色彩規格至黑色與白色。然而，此對應可能無法製造您所想要的外觀。

有關圖示的更多資訊，請參閱 第 209 頁“圖示影像檔案”。

第 7 步：建立應用程式群組

註解 – 關於建立應用程式群組的范例，請參閱第 7 步。。

只要您一建立應用程式的動作與資料類型定義，您就必須建立負責建立使用者實際上看到應用程式群組與其內容的配置檔案。

應用程式群組是在應用程式管理者頂層的目錄（請參閱 圖 5-1）。

建立應用程式群組有三個步驟：

- 在註冊套裝軟體中建立應用程式群組目錄。
- 選擇性的：配置應用程式群組以使用唯一的圖示。此包含建立應用程式群組目錄的資料類型定義。
- 建立應用程式群組的內容。

建立應用程式群組目錄

如果要建立應用程式群組，請在 `appmanager` 之下的註冊套裝軟體中建立目錄，如 圖 5-7。

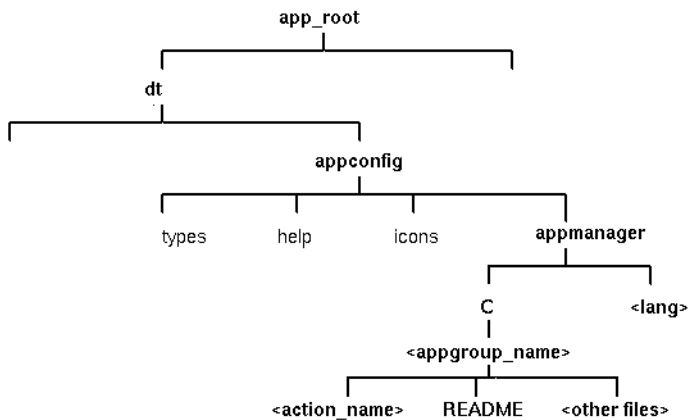


圖 5-7 appmanager 目錄

應用程式群組名稱

圖 5-7 中的 `<appgroup_name>` 是應用程式群組的名稱。



圖 5-8 應用程式群組名稱 (`<appgroup_name>`)

此名稱可以是任何可允許的檔案（目錄）名稱。使用描述應用程式的名稱。

配置應用程式群組以使用唯一的圖示

桌上管理系統提供預設應用程式群組的圖示。然而，您可能會想要提供自訂圖示。

如果您想要提供唯一的圖示給應用程式群組，您必須建立：

- 在應用程式管理者頂層出現的目錄的資料類型。
- 開啓並列印資料類型的動作。

例如，假設您想要建立稱為 `Media_Tools` 的應用程式群組。以下放在檔案 `app_root/dt/appconfig/types/language/name.dt` 中的資料類型定義，指派一個唯一的圖示給應用程式群組圖示。

```
DATA_ATTRIBUTES Media_ToolsAppgroup
{
    ACTIONS          OpenInPlace,OpenNewView
    ICON             MediaTools
    DESCRIPTION      Double-click to open the Media_Tools \
                    application group
}

DATA_CRITERIA      Media_ToolsAppgroupCriterial
{
    DATA_ATTRIBUTES_NAME Media_ToolsAppgroup
    MODE             d
    PATH_PATTERN     */appmanager/*/Media_Tools
}
```

定義的屬性區段指定使用的圖示。定義的條件區段指定將資料類型定義為任何名為 `Media_Tools` 的目錄，而且此目錄是名為 `appmanager` 的目錄之子目錄。

圖 5-9 顯示應用程式群組名稱與資料類型定義之間的關係。資料類型定義中的 `PATH_PATTERN` 欄位將唯一的圖示連接至應用程式群組。



圖 5-9 應用程式群組如何取得唯一的圖示

您也應該建立應用程式群組資料類型的開啓與列印動作：

```
ACTION Open
{
    ARG_TYPE          Media_ToolsAppGroup
    TYPE              MAP
    MAP_ACTION        OpenAppGroup
}

ACTION Print
{
    ARG_TYPE          Media_ToolsAppGroup
    TYPE              MAP
    MAP_ACTION        PrintAppGroup
}
```

OpenAppGroup 與 PrintAppGroup 動作是 `/usr/dt/appconfig/types/language/dtappman.dt` 中所定義的內建動作。

建立應用程式群組的內容

應用程式群組中最重要的項目就是啟動應用程式（動作圖示）的圖示。如果應用程式群組包含一組應用程式，通常每組應用程式都會有一個圖示。

除了一個或多個動作圖示以外，應用程式群組還可包含：

- 一個或多個 README 檔案
 - 一個或多個範例資料檔案
 - 範本
 - 使用者可以連按兩下以查看輔助說明資訊的圖示
 - 線上援助頁
 - 專用的前方面板控制
- 應用程式群組可以包含子目錄。

建立動作檔案（應用程式圖示）

應用程式群組應包含啟動應用程式的圖示。如果群組供應一組應用程式，每個應用程式應該都有一個圖示。既然它們代表隱藏的動作，這些圖示就稱為應用程式圖示或動作圖示。

動作圖示是藉由建立具有與其將會執行的動作相同名稱的可執行檔案所建立的：

```
app_root /dt/appconfig/appmanager/appgroup_name/action_name
```

此檔案稱為 **動作檔案**，因為它的目的是建立隱藏動作的視覺表示方式。

例如，如果您已建立執行 `BestTextEditor` 應用程式的名為 `BestTextEditor` 的動作，您就會執行名為 `BestTextEditor` 的可執行的檔案。在檔案管理者與應用程式管理者中，動作檔案將會使用動作定義中所指定的圖示影像。

圖 5-10 舉例說明動作定義、動作檔案與應用程式管理者視窗實際登錄之間的關係。

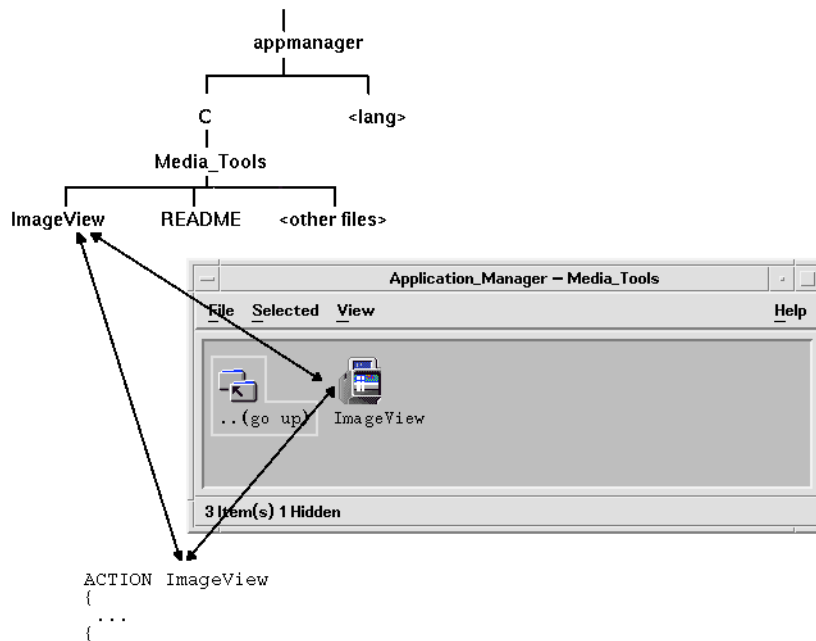


圖 5-10 應用程式圖示是應用程式群組中的一個檔案

README 檔案

桌上管理系統提供 README 資料類型，您可以將其用於您的應用程式的 README 檔案。使用這些命名慣例之一：

- README
- README
- README.*
- Read.*.Me
- read.*.me
- READ.*.ME

建立專用化的前方面板控制

在大部分的情況中，您不需要提供前方面板的控制定義；使用者可以藉由將動作圖示定位於子面板中的安裝圖示控制來將應用程式增加至前方面板。

如果您想要使用者能夠安裝行為不同於動作圖示的控制，您可能需要為您的應用程式建立一個包含控制定義的前方面板配置檔案—例如，假設控制在被監視的檔案變更的同時監視一個檔案並變更外型。

前方面板配置檔案是放在 `app_root /dt/appconfig/types/language` 目錄中。命名慣例為 `name .fp`。

如果您供應包含控制的配置檔案，使用者就可以藉由將 *.fp 檔案定位於子面板中的安裝圖示控制來將控制增加至子面板。

例如，以下定義可以放在應用程式群組中的前方面板配置檔案中。如果使用者將此檔案定位於子面板中的安裝圖示控制上，即在執行 BestTextEditor 應用程式的單一實例的子面板中建立控制。如果 BestTextEditor 已經在執行，視窗會移動至目前工作區中的視窗堆疊頂端。

```
CONTROL BestTextEditorControl
{
    TYPE          icon
    ICON          BTEFPanel
    PUSH_RECALL   True
    CLIENT_NAME   BTEd
    PUSH_ACTION   BTEditor
    DROP_ACTION   BTEditor
    HELP_STRING   SStart BestTextEditor application.
}
```

有關建立前方面板配置檔案的其它資訊，請參閱：

- 第 15 章
- dtfpfile (4) 線上援助頁

第 8 步：使用 dtappintegrate 註冊應用程式

註解 – 有關註冊應用程式的範例，請參閱第 8 步 第 86 頁“建立註冊套裝軟體的範例”。

只要您已在應用程式 root 之下建立註冊套裝軟體，您就已準備好了執行實際的應用程式註冊。

應用程式註冊建立註冊套裝軟體與位於沿著桌上管理系統搜尋路徑的目錄之間的連結（請參閱 第 85 頁“dtappintegrate 如何整合應用程式”）。

▼ 如果要註冊具有 dtappintegrate 的應用程式

如果應用程式是智慧型桌上管理系統，dtappintegrate 通常會自動以初始程序中的最後步驟來執行。如果它不是自動執行，或是如果您已建立配置檔案以整合非桌上管理系統智慧型應用程式，那麼您就可以人工執行 dtappintegrate。

1. 登入為 root。
2. 執行指令：

```
/usr/dt/bin/dtappintegrate -s app_root
```

其中 *app_root* 為桌上管理系統應用程式的根目錄。有關更多的資訊，請參閱 dtappintegrate(1) 線上援助頁。

3. 開啓桌面工具應用程式群組並連按兩下重新載入應用程式。
4. 檢驗已正確地註冊應用程式：
 - a. 顯示應用程式管理者的頂層。新的應用程式群組應出現在應用程式管理者中。
 - b. 開啓應用程式群組並連按兩下動作圖示。

語法與選項的 dtappintegrate

```
dtappintegrate -s app_root [-t target_path] [-l language] [-u]
```

<code>-s <i>app_root</i></code>	所需的參數，指定應用程式在其下安裝的應用程式 <i>root</i> 。
<code>-t <i>target_path</i></code>	選擇性的參數，預設成系統位置 <code>/etc/dt/appconfig</code> 。指定桌上管理系統配置檔案連接的位置。您必須使用在應用程式搜尋路徑上的位置。
<code>-l <i>language</i></code>	選擇性的參數，預設成所有語言。指定語言相依的桌上管理系統配置檔案來整合。
<code>-u</code>	選擇性的參數，取消整合應用程式，刪除所有在整合期間所設定的連結。

dtappintegrate 如何整合應用程式

dtappintegrate 的功能是在安裝的檔案與桌上管理系統尋找配置檔案的位置之間設定連結。

動作及資料類型

dtappintegrate 從註冊套裝軟體中的動作與資料類型定義檔案起，建立象符號連結至沿著動作資料庫輔助說明搜尋路徑的整個系統的目錄。其完成是藉由建立連結自

```
app_root/dt/appconfig/types/language/*.dt
```

至

```
/etc/dt/appconfig/types/language  
/*.dt
```

輔助說明資訊檔

dtappintegrate 從註冊套裝軟體中的輔助說明檔案起，建立象符號連結至沿著輔助說明搜尋路徑的整個系統的目錄。其完成是藉由建立連結自

```
app_root/dt/appconfig/help/language/help_file.sdl
```

至

```
/etc/dt/appconfig/help/language/help_file.sdl
```

圖示檔案

`dtappintegrate` 從註冊套裝軟體中的圖示檔案起，建立象符號連結至沿著圖示搜尋路徑的整個系統的目錄。其完成是藉由建立連結自

```
app_root/dt/appconfig/icons/language/icon_files
```

至

```
/etc/dt/appconfig/icons/language/icon_files
```

應用程式群組

如果要將應用程式的應用程式群組放在應用程式管理者的頂層中，`dtappintegrate` 會建立在註冊套裝軟體中的應用程式群組與沿著應用程式搜尋路徑的整個系統的位置之間的連結。其完成是藉由建立連結自目錄

```
app_root /dt/appconfig/appmanager/language/appgroup_name
```

至

```
/etc/dt/appconfig/appmanager/language/appgroup_name
```

建立註冊套裝軟體的範例

下列步驟建立現存名為 `BestTextEditor` 的非桌上管理系統的智慧型應用程式註冊套裝軟體。

您需要知道的有關 “BestTextEditor” 的資訊

範例假定下列有關 `BestTextEditor` 應用程式的事實：

- 其安裝在目錄 `/usr/BTE` 中。
- 使用者的階段作業語言為預設值，`C`。
- 啟動 `BestTextEditor` 的指令行是：

```
BTEd {檔案名稱}
```

其中 **檔案名稱** 是新視窗中要開啓的資料檔案名稱。`BestTextEditor` 建立其自己的視窗——也就是說，它不在終端機模擬器視窗內執行。

- BestTextEditor 建立並使用兩種類型的資料檔案：
 - 文件檔案。其使用命名慣例 *.bte。BestTextEditor 提供列印其 .bte 資料檔案的指令行。此指令的語法為：

```
BTEPrint [-d 目標] [-s] 檔案名稱
```

其中：

- d 目標—指定目標印表機。
- s—指定靜音列印。不顯示應用程式的列印對話框。

檔案名稱—指定要列印的檔案。

- 範本檔案。其使用命名慣例 *.tpl。無法列印範本檔案。
- BestTextEditor 的現存非桌上管理系統 app-defaults 檔案包含介面字型以及前景與背景色彩的資源。
- BestTextEditor 的線上輔助說明卷次是使用桌上管理系統輔助說明設計者工具箱來建立的。當建立線上輔助說明卷次時，其使用下列的資源檔案：

```
.../BTEHelp.htg
.../graphics/BTE1.xwd
.../graphics/BTE2.xwd
```

並產生檔案 .../BTEHelp.sdl。

註冊 "BestTextEditor" 的步驟

以下逐步的程序註冊了 BestTextEditor。

1. 修改字型與色彩資源。

在 BestTextEditor 的 app-defaults 檔案中，刪除所設定的如下資源：

- 文字的字型
- 前景與背景的色彩

2. 建立應用程式 root。

建立目錄：

```
/Desktop_approots/BTE
```

如果您正在整合現存應用程式，您應該在應用程式的安裝位置以外之處建立應用程式根目錄；否則，當您更新應用程式時，您所建立的配置檔案可能會被刪除。

3. 建立註冊套裝軟體目錄。

建立這些目錄：

```
/desktop_approots/BTE/dt/appconfig/types/C
/desktop_approots/BTE/dt/appconfig/help/C
/desktop_approots/BTE/dt/appconfig/icons/C
```

```
/desktop_approuts/BTE/dt/appconfig/appmanager/C/ BestTextEditor
```

4. 建立應用程式的動作與資料類型。

a. 建立動作與資料類型定義的配置檔案：

```
/desktop_approuts/BTE/dt/appconfig/types/C/BTE.dt
```

b. 建立執行 BestTextEditor 的動作定義：

```
ACTION BTEditor
{
    WINDOW_TYPE          NO_STUDIO
    ICON                 BTERun
    DESCRIPTION          Double-click this icon or drop \
                        a BTE data file on it to run \
                        BestTextEditor.
    EXEC_STRING          /usr/BTE/BTEd%Arg_1%
}
```

c. 建立 *.bte 檔案的資料類型：

```
DATA_ATTRIBUTES      BTEDataFile
{
    DESCRIPTION        BestTextEditor data file.
    ICON               BTEData
    ACTION             Open, Print
}

DATA_CRITERIA        BTEDataFileCriteria1
{
    DATA_ATTRIBUTES_NAME  BTEDataFile
    NAME_PATTERN          *.bte
    MODE                  f
}
```

d. 建立 *.tpl 檔案的資料類型：

```
DATA_ATTRIBUTES      BTETemplateFile
{
    DESCRIPTION        BestTextEditor template file.
    ICON               BTETempl
    ACTION             Open
}

DATA_CRITERIAL       BTETemplateFileCriteria1
{
    DATA_ATTRIBUTES_NAME  BTETemplateFile
    NAME_PATTERN          *.tpl
    MODE                  f
}
```

e. 建立 *.bte 檔案的開啓動作。

```
ACTION Open
{
```



```

ARG_TYPE          BTEDataFile
TYPE              MAP
MAP_ACTION        BTEditor
}

```

f. 建立 *.bte 檔案的列印動作。

此為將會列印資料檔案的簡單列印動作。這些動作需要 LPDEST 環境變數的值並略過 -s 列印選項。（如果 LPDEST 未設定，動作可能會失敗。）

```

ACTION Print
{
ARG_TYPE          BTEDataFile
TYPE              MAP
MAP_ACTION        BTEPrintData
}

ACTION BTEPrintData
{
WINDOW_TYPE      NO_STDIO
EXEC_STRING       BTEPrint -d $LPDEST%Arg_1%
}

```

此為 BTEPrintData 動作與伴隨的程序檔之另一種版本。它們共同處理未設定的 LPDEST 或是如果要求靜音列印的情況。

```

ACTION BTEPrintData
{
WINDOW_TYPE      NO_STDIO
EXEC_STRING       /usr/BTE/bin/BTEenvprint \
                  % (File) Arg_1%
}

```

/usr/BTE/bin/BTEenvprint 程序檔的內容為：

```

# BTEenvprint
#!/bin/sh
DEST=""
SILENT=""
if [ $LPDEST ] ; then
DEST="-d $LPDEST "
fi
BTEPrint $DEST SILENT $1

```

g. 建立 *.tpl 檔案的開啓動作：

```

ACTION Open
{
ARG_TYPE          BTETemplateFile
TYPE              MAP
MAP_ACTION        BTEditor
}

```

h. 建立 *.tpl 檔案的列印動作：

```

ACTION Print
{
ARG_TYPES         BTETemplateFile
}

```

```

        TYPE          MAP
MAP_ACTION          NoPrint
    }

```

無列印是內建的動作，會顯示對話框告訴使用者無法列印的檔案。

5. 將輔助說明檔案放在註冊套裝軟體中。

a. 將輔助說明檔案放在下列位置中：

```

/desktop_approots/BTE/dt/appconfig/help/C/BTEHelp.sdl
/desktop_approots/BTE/dt/appconfig/help/C/graphics/BTE1.xwd
/desktop_approots/BTE/dt/appconfig/help/C/graphics/BTE2.xwd

```

b. 建立檔案：

```

/desktop_approots/BTE/dt/appconfig/types/C/BTEhelp.dt.

```

將以下動作定義放在檔案中：

```

ACTION BTEHelp
{
    WINDOW_TYPE    NO_STUDIO
    EXEC_STRING    /usr/dt/bin/dthelpview -helpVolume BTEHelp.sdl
    DESCRIPTION    Opens the BestTextEditor help volume.
}

```

6. 建立應用程式的圖示。

使用圖示編輯程式來建立圖示。表 5-3 顯示使用的大小指導。

表 5-3 圖示大小指導

名稱	大小
基本名稱 .t.pm	16 乘 16
基本名稱 .m.pm	32 乘 32
基本名稱 .l.pm	48 乘 48

在目錄 `/desktop_approots/BTE/dt/appconfig/icons/C` 中建立這些圖示檔案：

- 代表執行應用程式的動作的圖示：`BTERun.t.pm`、`BTERun.m.pm`、`BTERun.l.pm`
- 代表 `*.bte` 檔案的圖示：`BTEData.t.pm`、`BTEData.m.pm`、
- 代表 `*.tpl` 檔案的圖示：`BTETempl.t.pm`、`BTETempl.m.pm`
- 代表應用程式群組（用於步驟 7 中）的圖示：`BTEApp.t.pm`、`BTEApp.m.pm`

7. 建立應用程式群組。

a. 如果您尚未如此作，請建立此目錄。

```

/desktop_approots/BTE/dt/appconfig/appmanager/C/BestTextEditor

```

- b. 此步驟是選擇性的。其藉由建立資料類型以及相關的應用程式群組動作來提供應用程式群組圖示的唯一圖示。如果您省略此步驟，應用程式群組將會使用預設圖示。

將下列資料類型與動作定義增加至檔案 `/desktop`

`_approots/BTE/dt/appconfig/types/C/BTE.dt`。資料類型指定由 `BestTextEditor` 應用程式群組所使用的圖示。動作會提供和內建應用程式群組相同的開啓與列印行爲。

```
DATA_ATTRIBUTES BestTextEditorAppGroup
{
    ACTIONS      OpenInPlace,OpenNewView
    ICON         BTEApp
}

DATA_CRITERIA BestTextEditorAppGroupCriterial
{
    DATA_ATTRIBUTES_NAME      BestTextEditorAppGroup
    MODE                       d
    PATH_PATTERN                */appmanager/*/BestTextEditor
}

ACTION Open
{
    ARG_TYPE          BestTextEditorAppGroup
    TYPE              MAP
    MAP_ACTION        OpenAppGroup
}

ACTION Print
{
    ARG_TYPE          BestTextEditorAppGroup
    TYPE              MAP
    MAP_ACTION        PrintAppGroup
}
```

- c. 在將會啓動應用程式的應用程式群組中建立圖示。如果要如此作，請建立此檔案：

```
/desktop_approots/BTE/dt/appconfig/appmanager/C \
/BestTextEditor/BTEditor
```

並使檔案成爲可執行的。

- d. 在將會開啓輔助說明卷次的應用程式群組中建立動作檔案。如果要如此作，請建立此檔案：

```
/desktop_approots/BTE/dt/appconfig/appmanager/C \
/BestTextEditor/BTEHelp
```

並使檔案成爲可執行的。

- e. 將其它檔案放在應用程式群組中；例如，“README”檔案、範例資料與範本檔案。

8. 註冊應用程式。

在終端機模擬器視窗中：

a. 登入為 **root** 。

b. 執行指令：

```
/usr/dt/bin/dtappintegrate -s /desktop_approots/BTE
```

c. 開啓桌面工具應用程式群組並連按兩下重新載入應用程式。

第 6 章

其它配置

本章提出設定與系統管理主題。

- 第 93 頁“Solaris CDE 目錄結構”
- 第 95 頁“主要配置檔案”
- 第 96 頁“啓動登入伺服器 ”
- 第 98 頁“從其它的工作站或網路伺服器安裝位置來裝載已安裝的 CDE ”
- 第 99 頁“配置您的桌上管理系統以使用多重螢幕”
- 第 100 頁“網路的桌上管理系統”
- 第 102 頁“使用 XTerminals”
- 第 103 頁“登入語言環境與字型路徑”
- 第 103 頁“使用工作站作為 XTerminals”
- 第 105 頁“特殊 CDE 配置”

Solaris CDE 目錄結構

本節描述您的桌上管理系統環境中所包括的主目錄。

`/usr/dt`

此目錄是 Solaris CDE 的安裝位置。它也可以是來自遠端檔案伺服器的安裝點。表 6-1 描述 `/usr/dt` 子目錄。

表 6-1 `/usr/dt` 子目錄

子目錄	說明
<code>/bin</code>	SCDE 應用程式與公用程式

表 6-1 /usr/dt 子目錄 (繼續)

子目錄	說明
/lib	SCDE 執行時間共用程式庫
/config	預設系統配置檔案
/man	[選擇性的] 線上援助頁
/app-defaults	預設應用程式資源
/appconfig	預設應用程式圖示、類型與動作
/examples	[選擇性的] CDE 碼/程式範例
/include	[選擇性的] 開發者包括檔案
/palettes	色彩調色盤
/share	CDE AnswerBook 文件與預設背景定位

/etc/dt

此目錄包含自訂的特定工作站配置檔案。這些檔案使您能夠以下列方法自訂您的環境：

- 設定 X 伺服器的配置選項
- 使用多重螢幕
- 自訂工作站動作檔案、資料類型、圖示與字型

/var/dt

此目錄是用於儲存 Solaris CDE 應用程式的暫存檔，例如登入管理者與應用程式管理者。

\$HomeDirectory

此目錄包含與使用者的桌上管理系統設定相關的使用者特定的檔案。它們包括應用程式、色彩計畫、工作區功能表與前方面板修改、以及錯誤登錄。

主要配置檔案

您的桌上管理系統環境的大部分自訂都可以包含許多檔案。下列檔案便是兩個範例：

- `/usr/dt/config/Xconfig`
- `/usr/dt/config/Xservers`

Xconfig

Xconfig 是由 dtlogin 所使用的主配置檔案。其設定登入資源並指定其它 dtlogin 所需的檔案位置。以下範例顯示在預設 Xconfig 檔案中所找到的幾行：

```
Dtlogin.errorLogFile:      /var/dt/Xerrors
Dtlogin.servers:          /usr/dt/config/Xservers
Dtlogin*session:         /usr/dt/bin/Xsession
```

有關此檔案的更多資訊，請參照 附錄 A。您也可以復習檔案本身中的說明註釋。

Xservers

以 XDM (X 顯示器管理者) 為基礎，dtlogin 就與 XDM 一樣使用 Xservers 檔案以指定如何啟動局部 Xserver。按預設，此檔案中的最後一行為：

```
:0 Local local@console /usr/openwin/bin/Xsun :0
```

- `:0` 表示 Xserver 顯示器為 `<localhost:0>`。
- `local` 指示局部性啟動新的 X11 伺服器。

註解 – 如果您想要附加至執行的 X11 伺服器，請以 `foreign` 來取代 `local`。

- `console` 表示指令行登入轉義為 `/dev/console`。
- `/usr/openwin/bin/Xsun` 是至 X11 伺服器的路徑。

註解 – Solaris CDE 及 OpenWindows™ 環境執行相同的 X11 伺服器。

有關此檔案的更多資訊，請參照 `dtlogin(1)` 線上援助頁。您也可以復習檔案本身中的說明註釋。

啓動登入伺服器

當系統啓動時，登入伺服器通常是自動啓動的。您也可以從指令行啓動登入伺服器，但是您必須先登入爲 root 使用者。

當系統啓動時，如果要設定登入伺服器來啓動：

- 鍵入以下指令之後按 **Return**：

```
# /usr/dt/bin/dtconfig -e
```

此將會將 S99dtlogin 檔案增加至您的 /etc/rc2.d 目錄。當您重新啓動時，登入伺服器會自動啓動。

如果要在系統啓動時取消登入伺服器的自動啓動：

- 鍵入以下指令之後按 **Return**：

```
# /usr/dt/bin/dtconfig -d
```

如果要從指令行啓動登入伺服器：

- 鍵入以下指令之後按 **Return**：

```
# /usr/dt/bin/dtlogin -daemon; exit
```

註解 – 雖然臨時配置測試可以從指令行啓動登入伺服器，但是當系統啓動時，仍應正常啓動登入伺服器。

如果要清除登入伺服器、Xserver 與整個 Solaris CDE 桌上管理系統：

- 鍵入以下指令之後按 **Return**：

```
# /usr/dt/bin/dtconfig -kill
```

有關桌上管理系統配置公用程式 dtconfig 的更多資訊，請參閱附錄 A。它提供 dtconfig.1 線上援助頁的副本。

如果要使用指令行登入選項來結束 Solaris CDE 登入螢幕：

- 從 Solaris CDE 登入螢幕上的選項功能表來選擇指令行登入。在螢幕清除之後，請按 **Return** 以顯示登入提示。



註解 – 桌上管理系統登入常駐程式將會在您完成指令行登入之後，自動啟動桌上管理系統登入螢幕。

如果要啟動單個終端機模擬視窗：

- 從 Solaris CDE 登入螢幕上選項功能表的階段作業子功能表來選擇單視窗階段作業。



如果您想要讓 Xserver 繼續執行，請從階段作業子功能表選擇單視窗階段作業。此將會啟動單個 xterm 視窗。單視窗階段作業選項永遠可供使用，即使是當指令行登入選項不供使用時。

從其它的工作站或網路伺服器安裝位置來裝載已安裝的 CDE

既然此程序不會使用您的局部磁碟空間，如果您的工作站上沒有所需的磁碟空間，您可以從其它的工作站或網路伺服器安裝位置來裝載已安裝的 CDE。

註解 – 由於 Solaris 版次與用戶端工作站之間的 `/usr/dt` 目錄結構不同（SPARC 目錄結構與 Intel 目錄結構不同），用戶端工作站必須裝載合適的 NFS 伺服器 `/usr/dt` 影像。例如，具有 Solaris 2.5 的 SPARC 系統會從其它具有 `/usr/dt` 安裝的 CDE 的 Solaris 2.5 系統來裝載 `/usr/dt`。

▼ 裝載已安裝的 CDE

1. 從先前安裝的工作站或網路伺服器將 `/usr/dt` 目錄裝載至您的工作站上的 `/usr/dt` 目錄。
2. 鍵入 `/usr/dt/bin/dtconfig -inetd`
3. 藉由鍵入下行來啓動 Solaris 桌上管理系統登入：

```
/usr/dt/bin/dtconfig -e
```

4. 重新啓動您的工作站。
有關桌上管理系統配置公用程式 `dtconfig` 的更多資訊，請參閱附錄 A。它提供 `dtconfig (1)` 線上援助頁的副本。

▼ 卸載已裝載的 CDE 目錄

1. 藉由鍵入下行停用 Solaris 桌上管理系統登入：

```
/usr/dt/bin/dtconfig -d
```

2. 鍵入 `/usr/dt/bin/dtconfig -inetd.ow`
3. 卸載 `/usr/dt`。
4. 重新啓動您的工作站。

有關桌上管理系統配置公用程式 `dtconfig` 的更多資訊，請參閱附錄 A。它提供 `dtconfig (1)` 線上援助頁的副本。

配置您的桌上管理系統以使用多重螢幕

標準登入在單個螢幕上啟動桌上管理系統。您可以編輯 `Xconfig` 檔案以使桌上管理系統在多重螢幕上啟動。您必須登入為 `root` 使用者以變更此檔案。

註解 – 有關登入配置資訊的較快速編輯，請使用單視窗階段作業選項以執行單個 `xterm` 視窗而非整個桌上管理系統。

▼ 使桌上管理系統在多重螢幕上啟動

1. 藉由鍵入以下指令之後按 **Return** 來建立 `Xserver` 檔案的副本：

```
# cp /usr/dt/config/Xservers /etc/dt/config/Xservers
```

註解 – `/etc/dt/config/Xservers` 置換 `/usr/dt/config/Xservers`。

2. 編輯 `/etc/dt/config/Xservers` 檔案以設定兩個框架緩衝區（螢幕）：

- a. 尋找此行：

```
:0 Local local_uid@console root /usr/openwin/bin/Xsun :0
```

- b. 在此行結束處，增加兩個 `-dev` 選項。此範例假定它們名為 `/dev/fb0` 與 `/dev/fb1`。在下行之前增加一個空間：

```
-dev /dev/fb0 -dev /dev/fb1
```

3. 從選項功能表來選擇重設登入螢幕。



註解 – 在多重螢幕上執行 Solaris CDE 時，前方面板會顯示在每個螢幕上。這些前方面板彼此都是獨立的，且有分開的配置檔案。

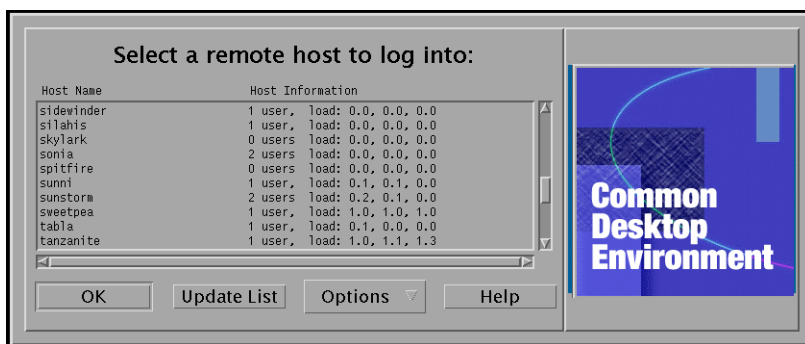
網路的桌上管理系統

Solaris CDE 登入管理者為網路察覺。按預設值，登入螢幕會由 Solaris CDE 選擇程式來回應查詢。

如果要顯示遠端主機的清單：

- 從 Solaris CDE 登入螢幕上的遠端登入功能表來選擇從清單選擇主機。

如果要獨立執行登入螢幕的選擇程式，另外的替代方式為參照區段 第 103 頁“使用工作站作為 XTerminals”。以下螢幕顯示典型可供使用的伺服器的選擇程式清單。



您可以從清單選擇閒置伺服器，以經由選擇程式使用其資源來登入 Solaris CDE。

任何執行 Solaris CDE 登入管理者的網路工作站都可以支援區域桌上管理系統使用者與多重遠端使用者，包括使用工作站作為 Solaris CDE 桌上管理系統伺服器的 Xterminal 使用者。（請參閱圖 6-1。）

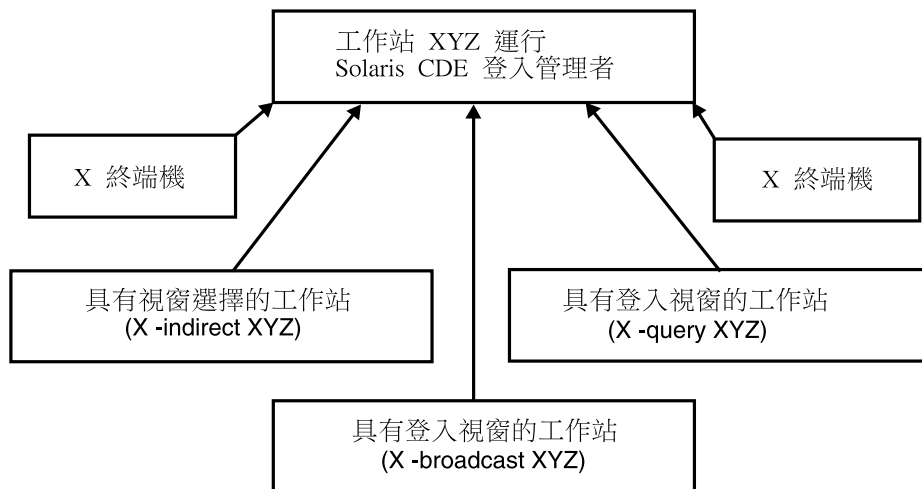


圖 6-1 網路的桌上管理系統

註解 – 圖 6-1 可包括混合式硬體環境。

使用 XTerminals

任何支援 XDM 通信協定的 Xterminal 都可以使用選擇程式來登入 Solaris CDE。如果要執行選擇程式，請參照區段 第 103 頁“使用工作站作為 XTerminals”。SPARC Xterminal™ 軟體版本 2.0 (或更新版本) 與 Solaris CDE 可運作良好。在 X 終端機的控制視窗上按一下屬性，並選擇階段作業為種類以顯示 圖 6-2 所示的設定螢幕。

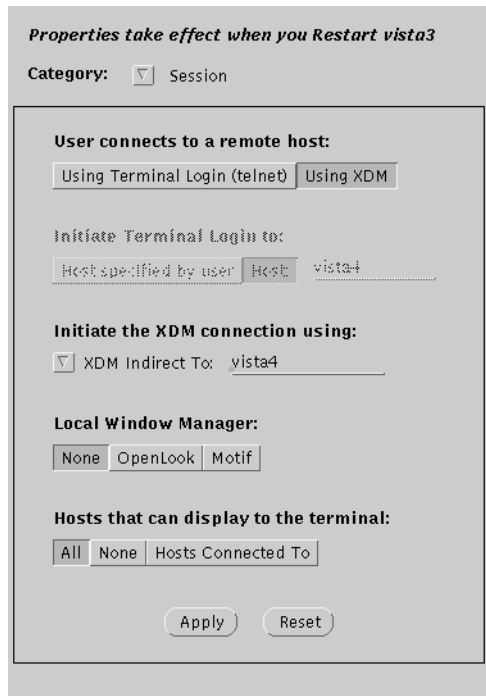
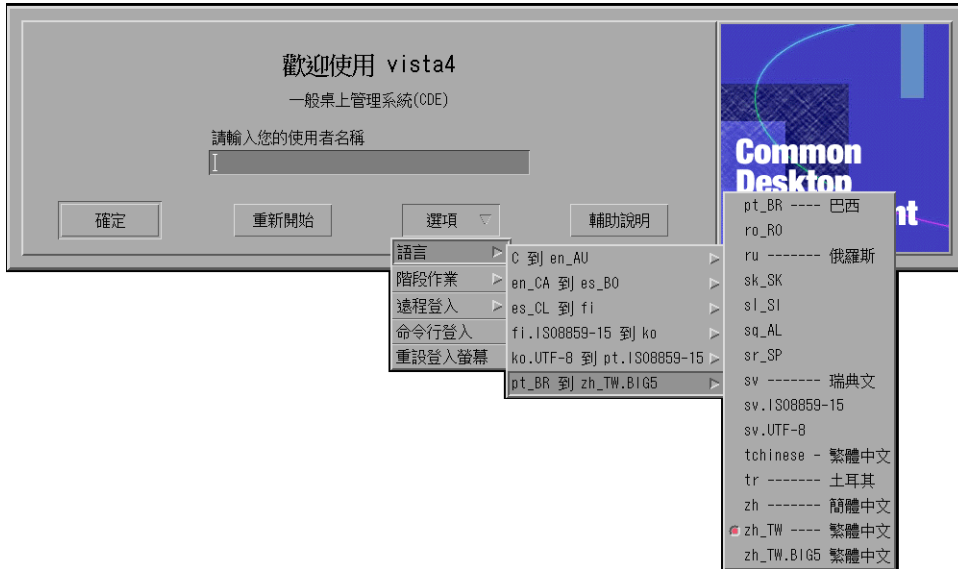


圖 6-2 階段作業設定螢幕

選擇局部視窗管理者的無 (None)，以及選擇 XDM 連接 (XDM Connection) 的 XDM 間接 (XDM Indirect) 至執行 Solaris CDE 的遠端系統。

登入語言環境與字型路徑

當您登入至 Solaris CDE 時，您選擇喜好的語言。以下的 Solaris CDE 登入螢幕顯示選項功能表上的語言選擇。



在工作站上，字型路徑自動設定為包括與喜好語言以及系統的基本“C”語言環境有關的字型（與別名）。對於 Xterminals 而言，這些字型是由 X11 字型伺服器自動供應的。

使用工作站作為 XTerminals

如果您有具有舊型工作站或是具有 16 MB 或較少記憶體的工作站，當其以 Solaris 2.4 軟體或較新的（或具有 Xserver 巨大修補的 Solaris 2.3 軟體）載入時，您可以使用它們作為 Xterminals。

註解 – 當遵循以下程序時，如果您的工作站螢幕上的字型顯示不正常，Solaris CDE 就可能沒有安裝在主機工作站上。使用 `pkgadd` 公用程式以從 Solaris CDE 1.0.x 解除附加的版次分別增加使用 `pkgadd` 公用程式，個別從 Solaris CDE 1.0.x 非隨附發行版本新增 `SUNWdtft()` Solaris CDE 字型套裝軟體。

▼ 如果要使用選擇程式以選擇主機 CDE 登入

1. 結束任何執行中的視窗系統。
2. 啟動選擇程式。從主控台指令行，鍵入下列指令（使用 `csch`）：

```
setenv OPENWINHOME /usr/openwin  
  
/usr/openwin/bin/X -indirect CDE_login_host
```

Xserver 從主機啟動並顯示選擇程式視窗。

3. 從選擇程式視窗選擇想要的登入主機。

▼ 如果要使用特定的主機 CDE 登入

- 鍵入以下指令之後按 **Return**：

```
/usr/openwin/bin/X -once -query CDE_login_host
```

Xserver 從主機啟動並顯示登入螢幕。此 `-once` 選項將會在一次登入/登出階段作業之後退出伺服器。如果未增加 `-once` 選項，登入螢幕在 Solaris CDE 登出之後會再次出現。

▼ 如果要使用第一可供使用的主機登入

- 鍵入以下指令之後按 **Return**：

```
/usr/openwin/bin/X -broadcast
```

此啟動 Xserver，在區域子網路上，其廣播 XDM（X 顯示器管理者）登入服務的要求。如果子網路上有任何系統正在執行 Solaris CDE 登入管理者（或任何其它以 XDM 為基礎的登入視窗），第一部回應的主機會將其登入視窗放在您的桌上管理系統上。

特殊 CDE 配置

本節提出特殊配置。

自訂郵件列印

CDE 電子郵件使用定義為 DTMAIL_FILE 資料類型的列印動作來列印郵件訊息。它不使
用由 OpenWindows 郵件工具所指定的列印程序檔。如果要變更列印行為，您必須修改
此列印動作。

▼ 如果要修改列印動作

1. 使用您最喜愛的編輯程式，建立以下檔案：

```
HomeDirectory/.dt/types/dtmail.dt
```

2. 將下列各行輸入此檔案中：

```
#  
# Override default Print action for mailboxes  
#  
ACTION Print  
{  
  LABEL          Print  
  ARG_TYPE       DTMAIL_FILE  
  type           command  
  WINDOW_TYPE    NO_STDIO  
  EXEC_STRING    sh -c '\  
                  dtmailpr -p -f% (File)Arg_1% | mp -m -l | \  
                  dtlp -u% (File)Arg_1%;'  
}
```

註解 – dtmailpr 是去除附件的列印過濾器，結果是純文字的串流。dtlp 是至 lp 的
標準 CDE 介面。%(File)Arg_1% 是列印的檔案。

3. 修改 EXEC_STRING 以包括您所想要的列印指令。

4. 重新啓動電子郵件。

轉換日曆爲新的資料格式

版本 4 可延展的資料格式是由 CDE 日曆所支援的新格式。OpenWindows 日曆管理者將不能夠讀取此格式。如果您需要在 OpenWindows 與 CDE 平台之間來回轉換，您不應將您的日曆轉換爲版本 4 的資料格式。然而，您可以使用 `sdtcm_convert` 程序檔以刪減您的日曆。

註解 – 您可以藉由從「日曆輔助說明」功能表選擇「有關日曆」來找出您日曆的資料版本。

否則，請使用 `sdtcm_convert` 程序檔以將您的日曆轉換爲版本 4 的資料格式。

有關此檔案的更多資訊，請參照 `sdtcm_convert(1)` 線上援助頁。

註解 – 如果您只安裝一般使用者 CDE 套裝軟體，則未安裝線上援助頁。如果要參閱線上援助頁，您將會需要以 `pkgadd` 公用程式來安裝它們。

從網路增加 AnswerBook 套裝軟體

當安裝 Solaris CDE 時，安裝功能表給予您安裝 AnswerBook CDE 套裝軟體的選項。如果您選擇「是」，AnswerBook 套裝軟體會安裝在 `/usr/dt/share/answerbooks/language` 目錄中（*language* 是特別的 AnswerBook 轉換的區域名稱）。

既然 AnswerBook 套裝軟體需要 109 MB 的磁碟空間，您可能不想要安裝它。然而，下列兩個選項是可供使用的：

- 您可以裝載 (或從 `/net/. . .` 連結) AnswerBook 套裝軟體至 `/usr/dt/share/answerbooks/language`。
- 您可以增加 `AB_CARDCATALOG` 環境變數至 `HomeDirectory/.dtprofile`。例如，如果您的網路上有匯出的版本，則請將下行增加至 `HomeDirectory/.dtprofile`：

```
export AB_CARDCATALOG=/net/hostname/usr/dt/share/answerbooks \
/language/ab_cardcatalog
```

在 CDE 桌上管理系統之外設定 CDE 環境

由您的 Solaris CDE 應用程式所使用的環境變數可以在 CDE 桌上管理系統之外設定。例如，您可能想要登入遠端工作站，並顯示返回您工作站的 CDE 應用程式。使用 CDE 公用程式 `dtsearchpath` 以設定各種 CDE shell 環境變數。

有關 Bourne 與 Korn shell，請鍵入以下指令：

```
eval `/usr/dt/bin/dtsearchpath`
```

有關 C shell，請鍵入以下指令：

```
eval `/usr/dt/bin/dtsearchpath -c`
```

您可以藉由將「顯示器」設定回局部工作站，然後從遠端執行 CDE 應用程式，來遵循此指令。結果將會顯示在局部工作站上。

註解 – 在此範例中，Bourne 與 Korn shell `dtsearchpath` 語法是在執行 CDE 的所有平台上均可使用的部分 CDE 範例執行。然而，C shell (`-c`) 選項只在 Sun 平台上可供使用。

桌面環境檔案

CDE 桌上管理系統提供一個執行時間環境，供 OpenWindows DeskSet™ 應用程式使用。您不需特別對 CDE 設定做任何事以達成此能力。以下檔案提供一些此環境的設定：

```
/usr/dt/config/Xsession.d/0015.sun.env
```

如果您需要為一特別的工作站從此環境設定增加或做刪減，此檔案可以在目前視窗編輯或是先複製到以下的檔案位置：

```
/etc/dt/config/Xsession.d/0015.sun.env
```

一個附加物的範例將是由 OpenWindows DeskSet 應用程式使用的舊型 OpenWindows 虛擬鍵盤啟動。因為 CDE 桌上管理系統（與大多數的 Sun 使用者）上大部分的應用程式均不使用此虛擬鍵盤公用程式，所以此程式便從預設啟動順序中省略以增進全面的桌上管理系統啟動效能。

有關選擇性的 OpenWindows 虛擬鍵盤 (`vkdb`) 啟動的其它資訊，請參閱 `0015.sun.env` 檔案中的註釋。

使用具有 Apple Macintosh 應用程式環境的磁片與 CD 媒體

如果您已經安裝 Apple Macintosh (MAE) 版本 1.0，並且預期透過 OpenWindows 檔案管理者應用程式來使用抽換式媒體（磁片或 CD-ROM），您必須編輯 `/etc/rmmount.conf` 檔案。這樣做可以確保 MAE 將會與檔案管理者抽換式媒體正確地運作。

▼ 變更 /etc/rmmount.conf 檔案

1. 藉由鍵入 `su` 成為 `root` 使用者，然後鍵入您的密碼。
2. 變更為 `/etc` 目錄。鍵入以下指令之後按 **Return**：

```
# cd /etc
```

3. 以您最喜愛的編輯程式來開啓 `/etc/rmmount.conf` 檔案。
4. 將在 `#Actions` 之下的下行移動至清單的結尾：

```
action floppy action_macfs.so
```

例如：

```
# 更多的 rmmount.conf
# @(#)rmmount.conf 1.2      92/09/23 SMI
#
# 抽換式媒體裝設器配置檔案。
#
# 檔案系統識別
ident hsfs ident_hsfs.so cdrom
ident ufs ident_ufs.so cdrom floppy
ident pcfs ident_pcfs.so floppy
ident macfs ident_macfs.so floppy
# 動作
action cdrom action_filemgr.so
action floppy action_filemgr.so
action floppy action_macfs.so
```

5. 儲存後結束。

安裝 MAE 時，它會替您將 `macfs action` 行放在 `rmmount.conf` 檔案中。如果 MAE 取消安裝，即刪除此行。

此變更使您能夠在 OpenWindows 或 CDE 檔案管理者應用程式中使用抽換式媒體，並且仍能夠在 MAE 中使用 Macintosh 磁片。然而，它確實能防止 MAE 抓取未格式化的、無法讀取的或是 DOS 磁片。使用此設定程序，這些磁片都會使用這個程序到 OpenWindows 檔案管理者應用程式。

如果您想要允許 MAE 能存取無法讀取的、未格式化的、或是 DOS 磁片，則必須把 `action floppy action action_macfs.so` 這一行放在 `action floppy action_filemgr.so` 這一行之前。

第 7 章

在網路中配置桌上管理系統

桌上管理系統是設計在高度網路化的環境中運作的。

- 第 109 頁“桌上管理系統網路的概述”
- 第 115 頁“配置桌上管理系統網路的一般步驟”
- 第 115 頁“配置桌上管理系統的基本作業系統網路”
- 第 118 頁“配置桌上管理系統用戶端與伺服器”
- 第 122 頁“管理應用程式服務”

桌上管理系統的架構讓系統管理者可以分配整個網路上的計算資源，包括：

- 應用程式
- 應用程式的資料檔案
- 桌上管理系統階段作業服務（桌上管理系統應用程式，例如登入管理者與檔案管理者）
- 輔助說明服務（輔助說明資料檔案可以放在中央輔助說明伺服器上）

桌上管理系統網路的概述

作業系統提供各種不同的網路服務，包括分散式檔案系統與遠端執行。X 伺服器提供其它的網路能力，包括遠端顯示器與安全服務的存取。

桌上管理系統將使用者介面分層在這些網路功能的頂端。此介面的目標與其隱藏的架構是要使得網路系統：

- 更容易使用。使用者可以執行應用程式並存取資料檔案，而不需煩惱應用程式與資料在網路中的位置。
- 更容易管理。桌上管理系統提供應用程式整合工具以及網路搜尋路徑，以便使系統更易於尋找遠端資料與應用程式。此外，桌上管理系統的檔案名稱對應處理，可使得管理包含眾多伺服器的複雜網路更加容易。

- 有彈性。當桌上管理系統的管理功能已爲了特定共用網路情況而設計之時，桌上管理系統便可以容納許多其它的自訂網路配置。

連上網路的桌上管理系統服務類型

網路讓使用者能夠存取各種分配在其它系統之間的計算服務，例如：

- 桌上管理系統階段作業與其應用程式—例如，工作區管理者與檔案管理者
- 其它的應用程式
- 資料檔案

網路術語使用名詞**伺服器**來描述提供計算服務給一個或多個其它系統的系統。當系統接收來自伺服器的服務時，就稱爲該伺服器的**用戶端**。

在一個複雜的網路中，系統可以使用位於整個網路中許多系統上的服務。此外，系統可以作爲特定類型的伺服器（例如，階段作業伺服器）並且也可以作爲用戶端（例如，應用程式伺服器）。

典型的網路情況

從桌上管理系統的觀點，典型的網路配置可包含這些主要元件的組合：

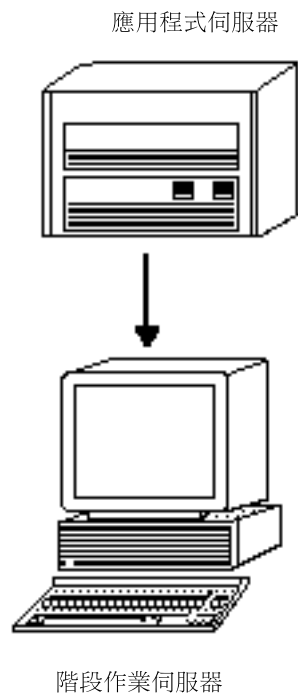
顯示器—X 伺服器正在執行之處

登入/階段作業伺服器—桌上管理系統應用程式（登入管理者、工作區管理者等等。）執行

應用程式伺服器—其它的應用程式執行之處

檔案伺服器—應用程式所使用的資料所在之處

最常見的網路配置之一是包含存取應用程式伺服器的系統。圖 7-1 舉例說明使用應用程式伺服器的工作站。X 伺服器與桌上管理系統階段作業正在工作站上執行。



■ 7-1 應用程式伺服器提供服務給桌上管理系統階段作業

網路也經常使用檔案伺服器來儲存大量的資料。此資料可以由在應用程式伺服器上執行的應用程式或是桌上管理系統應用程式（例如，檔案管理者需要存取資料檔案以在檔案管理者視窗中顯示它們）來使用。

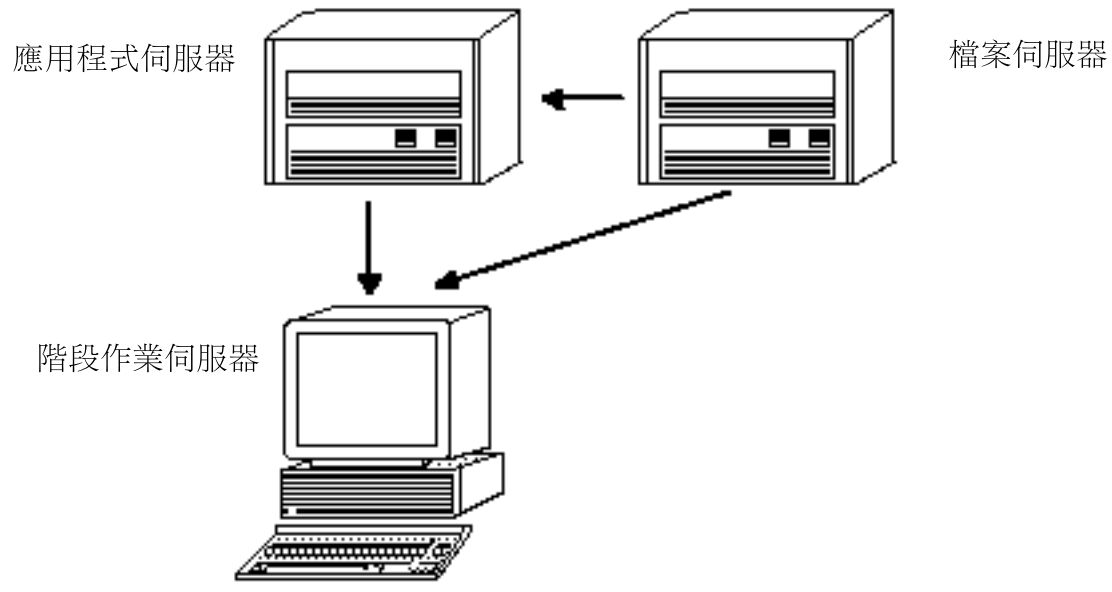


圖 7-2 檔案伺服器提供資料給應用程式伺服器與階段作業伺服器

X 終端機執行 X 伺服器並從其它系統取得桌上管理系統階段作業服務。

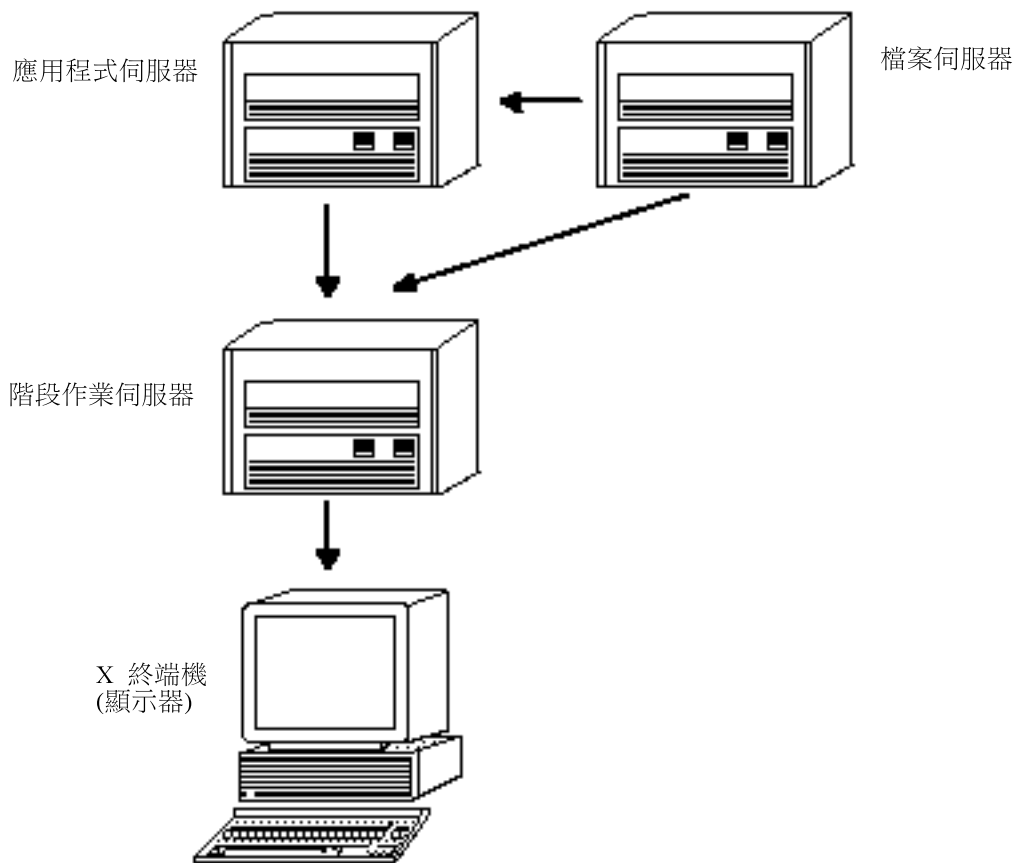


圖 7-3 X 終端機從階段作業伺服器取得階段作業服務

其它的網路情況

桌上管理系統是有彈性的，而且可以支援更複雜的網路配置。這種情況通常包含各種服務，除檔案伺服器外，可供應用程式伺服器使用。

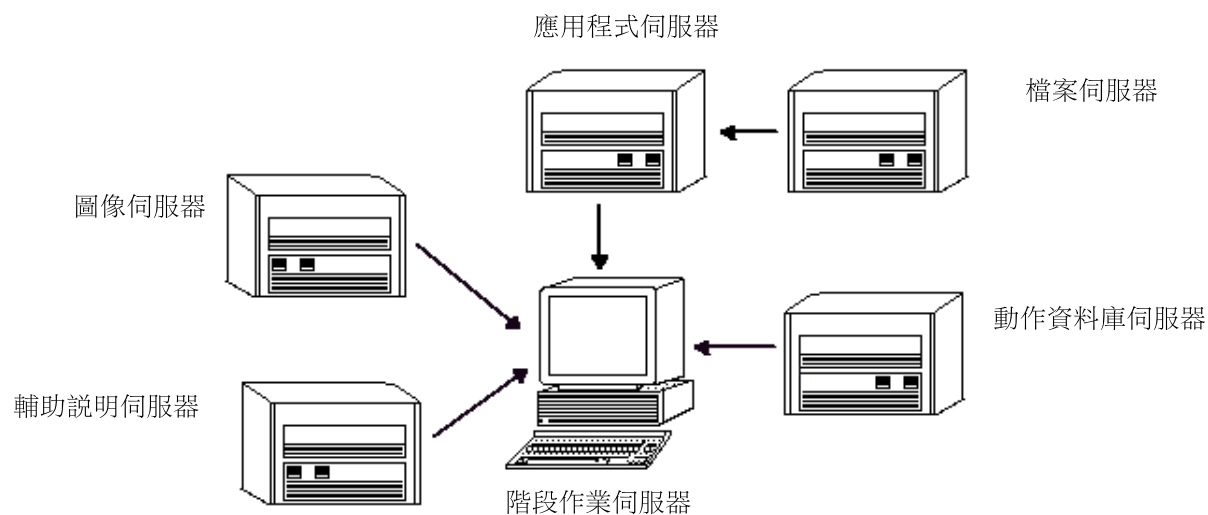


圖 7-4 桌上管理系統應用程式伺服器所需要的服務可以做分配

摘要—伺服器的類型

顯示器—執行 X 伺服器的系統。

登入與階段作業伺服器—執行桌上管理系統階段作業的系統（登入管理者、階段作業管理者、視窗管理者、檔案管理者等等）。

應用程式伺服器—應用程式在其上執行的系統。也稱為 **執行主機**。

檔案伺服器—於其上儲存應用程式的資料檔案的系統。

輔助說明伺服器—儲存輔助說明資料檔的系統。

（動作）資料庫伺服器—儲存包含動作與資料類型定義檔案的系統。

圖示伺服器—於其上儲存圖示檔案的系統。

網路可包括其它的伺服器，例如密碼伺服器、郵件伺服器、影像伺服器等等。

配置桌上管理系統網路的一般步驟

有三個配置桌上管理系統網路的一般步驟：

1. 配置基本作業系統網路服務。

這些是您作業系統所提供的桌上管理系統所依據的網路服務。請參閱 第 115 頁“配置桌上管理系統的基本作業系統網路”。

2. 安裝與配置桌上管理系統網路軟體與服務。

這些是桌上管理系統所需的服務，不論設定的用戶端或伺服器系統的類型為何。請參閱 第 118 頁“配置桌上管理系統用戶端與伺服器”。

3. 配置伺服器或用戶端的特別類型。

例如，配置應用程式伺服器與配置檔案伺服器需要不同的步驟。請參閱 第 122 頁“管理應用程式服務”。

配置桌上管理系統的基本作業系統網路

桌上管理系統需要以下的基本網路配置：

- 使用者必須在階段作業伺服器上，以及在提供桌上管理系統服務給階段作業伺服器的每個系統上有登入帳戶。使用者在所有的用戶端與伺服器系統上都必須要有相同的使用者 ID 與群組 ID。
- 系統必須有包含由階段作業以及其它應用程式所使用的資料的遠端檔案系統的存取。
- 1p 列印排存器的配置必須能存取遠端印表機。
- 發送郵件 必須配置，方可進行郵件程式服務。
- 必須設定 X 授權。

提供登入帳戶給使用者

本節描述桌上管理系統網路的登入帳戶需求。

提供登入帳戶

使用者必須有登入帳戶於：

- 提供服務至桌上管理系統的所有系統，包括應用程式伺服器、檔案伺服器，以及提供網路印表機的系統。

- 使用者可存取的所有階段作業伺服器。通常階段作業伺服器是由 X 終端機使用的。

提供一致的使用者與群組 ID

UNIX 使用者是由登入名稱與數值使用者 ID (UID) 來識別的。在桌上管理系統網路中，使用者在所有用戶端與伺服器系統上應有相同的登入名稱與 UID。

UNIX 使用者還會指派給一個或多個登入群組。每個群組都有一個群組名稱及一個數值群組 ID (GID)。在桌上管理系統網路中，所有系統都應使用一致的群組名稱與群組 ID。

有關更多資訊，請參閱 `id(1)` 或 `id(1M)` 線上援助頁。

配置分散式檔案系統存取

桌上管理系統使用 NFS 來共用系統之間的檔案。您必須識別在您的網路中包含共用檔案的所有檔案系統，並確定它們都正確地裝載於所有合適的系統上。

一般而言，您必須提供以下的遠端檔案存取：

- 使用者的主目錄必須由所有桌上管理系統用戶端與伺服器系統共用。這是必要的，因為：
 - 主目錄包含必須由遠端系統上的應用程式存取的資料檔案。例如，使用資料檔案的應用程式經常使用主目錄作為預設資料檔案位置。
 - 主目錄是預設 `dtspcd` 認證目錄。有關 `dtspcd` 的更多資訊，請參閱第 120 頁“配置子程序控制常駐程式”。
- 如果使用者需要存取不在其主目錄中的資料檔案，這些資料檔案必須由所有的桌上管理系統用戶端與在資料檔案上作業的伺服器系統所共用。
- 桌上管理系統安裝與配置目錄 (`/usr/dt` 與 `/etc/dt`) 必須由所有的桌上管理系統用戶端與伺服器系統所共用，這樣才能使所有使用者的應用程式存取相同的桌上管理系統配置檔案。

提供網路的主目錄

當使用者有單個主目錄供網路上所有用戶端與伺服器系統共用時，桌上管理系統網路的運作最有效。

網路的主目錄讓使用者能夠使用網路中不同的系統，而不會失去個人的自訂與配置。這是因為要回復先前階段作業所需的個人的自訂與資訊都儲存於主目錄的子目錄中。

也需要共用主目錄的有：

- 預設 X 授權機制（請參閱第 117 頁“配置 X 授權”）。
- 桌上管理系統子程序控制常駐程式。此常駐程式參與遠端應用程式的啟動，且必須要能夠寫入使用者的主目錄。

檔案名稱一致性

您應該配置網路，才能讓使用者使用相同名稱，從所有的系統存取其資料檔案。此即為提供 **檔案名稱一致性**，且通常是由建立合適的象徵性連結所達成的。例如您可以配置每個系統，以使每個使用者的主目錄都可以藉由建立象徵性連結至目錄的實際裝載位置例如 `/users/login_name` 般可供使用。

配置遠端印表機的存取

桌上管理系統使用 `lp` 列印排存器來存取局部或遠端印表機。有關配置 `lp` 排存器的資訊，請參閱 `lpadmin(1M)` 線上援助頁。

在嘗試使用桌上管理系統圖形介面列印之前，您應該測試您能否使用 `lp` 指令正確地列印至所有的印表機。

確定使用一致的印表機裝置名稱。例如，如果直接連接到系統上的特別印表機已知為 `Postscript1`，所有其它遠端存取印表機的系統也應使用 `Postscript1` 的名稱。

配置電子郵件

桌上管理系統郵件程式使用 `發送郵件` 在系統之間發送郵件。有關如何配置電子郵件連接的更多資訊，請參閱 `傳送郵件 (1M)` 線上援助頁。

在嘗試從桌上管理系統傳送或接收郵件之前，您應該測試您能否使用 `mailx` 指令正確地傳送與接收郵件。

配置 X 授權

桌上管理系統使用預設 X 機制來授權遠端應用程式 (X 用戶端) 存取局部顯示器。配置此種授權最簡單的方法就是提供每個使用者一個網路主目錄。這樣做可以確定下列的需求符合：

- 使用者對檔案 `HomeDirectory/.Xauthority` 必須有讀取與寫入許可權。
- 應用程式伺服器上的 `.Xauthority` 檔案必須包含其上將會執行應用程式的顯示器的“magic cookie”。

有關更多的資訊，請參閱 `X(1)` 或 `xauth(1)` 線上援助頁。

配置桌上管理系統用戶端與伺服器

本節涵蓋管理系統的特定網路配置需求——也就是說，這些能力是由桌上管理系統而非基本作業系統所提供的。

本節分為兩部分：

- 配置登入與階段作業服務。
- 應用程式與其資料所需的配置服務。此包括應用程式、資料庫、圖示、檔案與輔助說明伺服器以及其用戶端。

配置登入與階段作業服務

登入/階段作業伺服器是供應桌上管理系統服務（登入管理者、階段作業管理者、檔案管理者、視窗管理者等等）給顯示器與 X 伺服器的系統。

一般而言，階段作業伺服器供應服務給 X 終端機。然而，網路配置可以設定集中階段作業服務於一個或多個伺服器，而這些伺服器是由 X 終端機與工作站兩者所存取。

登入管理者負責供應登入服務給其它顯示器的桌上管理系統元件。只要使用者一登入，階段作業管理者就會為使用者啟動。

有關配置登入/階段作業伺服器與 X 終端機的資訊，請參閱第 23 頁“在網路顯示器上顯示登入螢幕”。

配置其它與應用程式有關的服務

本節涵蓋桌上管理系統所共用的網路需求：

- 應用程式伺服器
- 資料庫伺服器
- 圖示伺服器
- 輔助說明伺服器

▼ 如果要配置桌上管理系統用戶端與伺服器

1. 提供桌上管理系統所需的作業系統網路配置。
請參閱第 115 頁“配置桌上管理系統的基本作業系統網路”。
2. 安裝桌上管理系統或最小檔案集。
您必須安裝：

- 整個共用桌上管理系統環境執行時間檔案集
- 或是, 這些檔案集：CDE-MIN 與 CDE-TT

註解 – 廠商之間的安裝與檔案集可能會有所不同。

3. 配置 ToolTalk 檔案名稱資料庫伺服器常駐程式 `rpc.ttdbserver` 的系統。

當桌上管理系統安裝時，此配置應會自動發生。若需要更進一步的資訊，請參閱 第 121 頁“配置 ToolTalk 資料庫伺服器”。

4. 安裝與配置子程序控制常駐程式 (`dtspcd`)。

當桌上管理系統安裝時，此配置應會自動發生。有關更多的資訊，請參閱 第 120 頁“配置子程序控制常駐程式”。

5. 裝載所有所需的遠端資料。

當資料不是位於使用它的應用程式所執行的系統上時，這份資料就視為“遠端”。

例如：

- 如果應用程式使用位於檔案伺服器上的資料，它就必須裝載這些檔案。
- 如果檔案管理者圖示位於圖示伺服器上，階段作業伺服器必須裝載這些檔案。
- 如果網路須使用輔助說明伺服器取得桌上管理系統輔助說明檔案時，階段作業伺服器與所有的應用程式伺服器都必須裝載輔助說明資料。

有關裝載點的更多資訊，請參閱下一節，第 119 頁“配置遠端檔案系統的裝載點”。

配置遠端檔案系統的裝載點

當桌上管理系統從一個系統傳送檔案名稱到另一個系統時，它必須變換或對映，使這些檔案名稱成為對目標系統有意義的名稱。因為檔案可以裝載於不同系統上的不同位置中，所以此對映是必須的，也因此必須使用不同的名稱來存取。例如 `sysA` 上的檔案 `/projects/big` 可以於 `sysB` 上，以 `/net/sysA/projects/big` 來進行存取。例如，`sysA` 上的 `/projects/big` 可以當作 `sysB` 上的 `/net/sysA/projects/big` 來存取。

檔案名稱對映的要求

如果要正確地執行此檔案名稱對映，下列之一必須是正確的：

- 裝載指令是用於靜態地裝載檔案系統。這些靜態裝載的類型一般是配置於如 `/etc/checklist`、`/etc/mnttab` 或 `/etc/filesystems` 的檔案中。
如果要讓檔案名稱對映在系統之間正確運作，檔案系統裝載必須使用一致的主機名稱。如果主機具有數個名稱（例如別名，或是如果主機有一個以上具有不同名稱的 LAN 位址），您必須使用所有裝載的相同名稱與名稱形式。
- 或是，使用自動裝載程式在預設 `/net` 裝載點裝載檔案系統。

- 或是，使用自動裝載程式在 `/net` 以外的位置裝載檔案系統，而且設定 `DTMOUNTPOINT` 環境變數以指示裝載點。請參閱下一節。第 120 頁“設定 `DTMOUNTPOINT` 的值”
- 有關自動裝載程式的資訊，請參閱自動裝載程式 (1M) 線上援助頁。

▼ 設定 `DTMOUNTPOINT` 的值

如果下列兩項條件均是正確的，您必須設定 `DTMOUNTPOINT` 環境變數：

- 使用自動裝載程式以裝載檔案系統。
 - 而且，遠端檔案系統裝載於 `/net` 以外的位置。
`DTMOUNTPOINT` 必須為程序而設定，包括：
 - 在使用者登入時會自動啟動使用者桌上管理系統程序，例如工作區管理者 (`dtwm`) 與檔案管理者 (`dtfile`)
 - 像是 `rpc.ttdbserver` 以及 `dtspcd` 的系統程序是由類似 `inetd` 的機制所啟動的
 - 由局部或遠端系統上的桌上管理系統所啟動的應用程式
 - 由使用者從 `shell` 指令行所啟動的應用程式
- 如果要設定這些程序的 `DTMOUNTPOINT`：

1. 編輯檔案 `/etc/inetd.conf`：

a. 尋找 `dtspcd` 登錄並增加：

```
-mount_point mount_point
```

b. 尋找 `rpc.ttdbserver` 登錄並增加：

```
-m mount_point
```

例如，如果自動裝載程式與 `/nfs` 的裝載點一起使用，`/etc/inetd.conf` 中的登錄為：

```
dtspcd stream tcp nowait root /usr/dt/bin/dtspcd \  
    /usr/dt/bin/dtspcd -mount_point /nfs  
rpc stream tcp wait root /usr/dt/bin/rpc.ttdbserver \  
100083 1 rpc.ttdbserver -m /nfs
```

2. 執行您的系統上重新讀取 `/etc/inetd.conf` 的程序。有關更多的資訊，請參閱 `inetd (1M)` 線上援助頁。

3. 設定 `DTMOUNTPOINT`，以使其值由使用者登入來繼承。

可以藉由在 `/etc/dt/config/Xsession.d` 中設定變數來完成。有關設定環境變數的更多資訊，請參閱第 45 頁“設定環境變數”。

配置子程序控制常駐程式

桌上管理系統子程序控制 (SPC) 服務提供用戶端/伺服器指令的執行。

桌上管理系統子程序控制常駐程式 (dtspcd) 是由桌上管理系統所使用，以啓動遠端應用程式。其為接受來自遠端用戶端執行指令的要求的 inet 常駐程式。有關如何配置 inet 常駐程式的更多資訊，請參閱 inetd.conf (1M) 線上援助頁。

桌上管理系統動作呼叫程式庫使用 SPC 服務以呼叫遠端動作。

▼ 配置 dtspcd

- 確認 dtspc 在 /etc/services 與 /etc/inetd.conf 中正確地註冊。

請參閱 dtspcd (1M) 線上援助頁。

SPC 安全性

子程序控制服務的認證是以檔案系統認證為基礎的。dtspcd 必須具備也是由所有 SPC 用戶端系統所裝載的 認證目錄 的存取。

按預設值，dtspcd 認證目錄是使用者的主目錄。但是，您可以藉由在 /etc/inetd.conf 目錄中設定 - auth_dir 選項來配置 dtspcd 以使用不同的位置。有關更多的資訊，請參閱 dtspcd (1M) 線上援助頁。

因為 SPC 認證是以檔案系統認證為基礎的，所以 SPC 服務的安全只與您的分散式檔案系統一樣。如果您正在您不信任的分散式檔案系統網路中使用桌上管理系統，您可能希望取消 dtspcd。如果要取消 dtspcd，請在 /etc/services 中將 dtspc 登錄註釋掉。

配置遠端執行的環境變數

當桌上管理系統使用動作以啓動遠端系統上的應用程式時，就會將使用者的環境變數複製到遠端系統，並放在應用程式的環境中。

按預設值，一些環境變數在複製到遠端系統之前就已經改變了。您可以配置動作呼叫元件與桌上管理系統的子程序控制服務，以便在將變數放入應用程式的環境前，執行其它的环境變數程序。

有關預設配置與如何修改的更多資訊，請參閱 dtactionfile(4) 與 dtspcdenv(4) 線上援助頁。

配置 ToolTalk 資料庫伺服器

ToolTalk 的一個元件是 ToolTalk 資料庫伺服器， /usr/dt/bin/rpc.ttdbserver。

ToolTalk 資料庫伺服器是由 ToolTalk 訊息服務所使用，同時也用於檔案名稱對映。當安裝桌上管理系統且不需要其它配置時，它通常是註冊於 /etc/inetd.conf 中。

有關 ToolTalk 資料庫伺服器與其配置選項的更多資訊，請參閱 rpc.ttdbserver (1M) 線上援助頁。

配置 ToolTalk 訊息伺服器

ToolTalk 訊息伺服器為 `ttsession`。按預設值，它不需要任何配置；而是於登入期間由 `Xsession` 程序檔所啟動的。

有關 ToolTalk 訊息伺服器與其配置選項的更多資訊，請參閱 `ttsession` 線上援助頁。

配置日曆常駐程式

日曆應用程式的一個元件為日曆常駐程式 `rpc.cmsd`。當安裝桌上管理系統且不需要其它配置時，它通常是註冊於 `/etc/inetd.conf` 中。

有關日曆常駐程式與其配置選項的更多資訊，請參閱 `rpc.cmsd(1)` 線上援助頁。

管理應用程式服務

本節涵蓋特定配置需求給：

- 應用程式伺服器與其用戶端
- 提供特殊服務的桌上管理系統伺服器—資料庫伺服器、圖示伺服器與輔助說明伺服器
它也涵蓋網路應用程式兩種特殊配置的網路需求：
- 遠端執行主機
- 經由檔案系統裝載執行的應用程式

搜尋路徑環境變數

桌上管理系統使用一組環境變數以指定用於尋找應用程式桌上管理系統配置檔案的搜尋路徑，例如動作與資料類型資料庫、輔助說明檔案與圖示檔案。

有關如何使用搜尋路徑環境變數的資訊，請參閱 第 136 頁“桌上管理系統搜尋路徑與其環境變數”或 `dtenvvar(5)` 線上援助頁。

配置應用程式伺服器及其用戶端

在標準應用程式伺服器配置中，應用程式伺服器包含所有二進位以及與應用程式有關的配置檔案，包括：

- 可執行的應用程式
- 標準應用程式配置檔案，例如 `app-defaults`、訊息目錄與該應用程式的共用程式庫。

- 桌上管理系統配置檔案：
 - 動作與資料類型定義檔案
 - 圖示影像檔案
 - 桌上管理系統輔助說明資料檔案

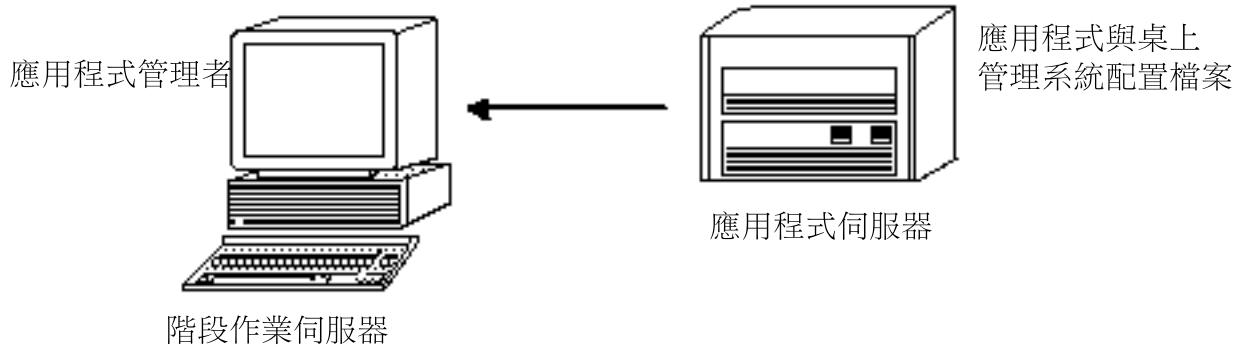


圖 7-5 標準應用程式伺服器配置

▼ 配置應用程式伺服器

1. 提供桌上管理系統所需的作業系統網路配置。
請參閱 第 115 頁“配置桌上管理系統的基本作業系統網路”。
2. 提供伺服器所需的一般桌上管理系統配置。
請參閱 第 118 頁“如果要配置桌上管理系統用戶端與伺服器”。
3. 安裝應用程式。
4. 如果應用程式未自己自動註冊，您必須執行註冊程序。
請參閱 第 5 章。

▼ 配置應用程式伺服器的用戶端

1. 提供桌上管理系統所需的作業系統網路配置。
請參閱 第 115 頁“配置桌上管理系統的基本作業系統網路”。
2. 提供用戶端所需的一般桌上管理系統配置。
請參閱 第 118 頁“如果要配置桌上管理系統用戶端與伺服器”。
3. 將應用程式伺服器增加至整個系統或個人基礎上的應用程式搜尋路徑：
整個系統的—設定並匯出 DTSPSYSAPPHOSTS 變數於
/etc/dt/config/Xsession.d/0010.dtpaths 中

個人—設定並匯出 DTSPUSERAPPHOSTS 變數於 *HomeDirectory* / *.dtprofile* 中
例如，*/etc/dt/config/Xsession.d/0010.dtpaths* 中的下行將具有主機名稱
SysAAA 和 SysBBB 的系統增加至應用程式搜尋路徑：

```
export DTSPSYSAPPHOSTS=SysAAA:;SysBBB:
```

有關設定應用程式搜尋路徑的更多資訊，請參閱：

- 第 137 頁“應用程式搜尋路徑”
- 第 136 頁“設定搜尋路徑的值”

配置資料庫、圖示及輔助說明服務

一般而言，與應用程式有關的動作與資料類型定義、圖示以及輔助說明資料檔案，均如應用程式一般安裝於相同的系統上。

例如，試想典型的輔助說明資料檔案的配置：

- 檔案管理者的輔助說明檔案通常位於階段作業伺服器上。桌上管理系統可以找到它們，是因為輔助說明搜尋路徑會自動搜尋階段作業伺服器上正確的位置。
- 其它應用程式的輔助說明檔案通常是和應用程式位於相同的應用程式伺服器上。階段作業伺服器可以找到它們是因為，修改應用程式搜尋路徑會自動修改輔助說明搜尋路徑。

在某些情況之下，您可能會想要將資料庫（動作與資料類型）、輔助說明或圖示資料放在網路上的其它位置。例如，如果您的網路使用多重階段作業伺服器，您可能會想要建立輔助說明伺服器，在其上儲存所有桌上管理系統應用程式（檔案管理者、樣式管理者等等）的輔助說明資料檔案。這個動作可以保存磁碟空間，因為輔助說明檔案不需要在每個階段作業伺服器上複製。

▼ 建立資料庫、輔助說明或圖示伺服器

1. 提供桌上管理系統所需的作業系統網路配置。

請參閱 第 115 頁“配置桌上管理系統的基本作業系統網路”。

2. 提供用戶端所需的一般桌上管理系統配置。

請參閱 第 118 頁“如果要配置桌上管理系統用戶端與伺服器”。

3. 安裝資料庫、輔助說明或圖示檔案。

檔案可位於系統上的任何地方。然而，既然下列位置是當指派應用程式伺服器給系統時會自動搜尋的目錄，使用它們可能會更比較容易。

- 資料庫檔案：*/etc/dt/appconfig/types/language*
- 輔助說明檔案：*/etc/dt/appconfig/help/language*
- 圖示檔案：*/etc/dt/appconfig/icons/language*

如果您正在設定資料庫伺服器，必須寫入動作以指定其指令（EXEC_STRINGs）將執行之處。請參閱 第 125 頁“指定遠端執行主機”。

▼ 配置階段作業伺服器以尋找資料庫、圖示或輔助說明伺服器

1. 提供桌上管理系統所需的作業系統網路配置。
請參閱 第 115 頁“配置桌上管理系統的基本作業系統網路”。
2. 提供用戶端所需的一般桌上管理系統配置。
請參閱 第 118 頁“如果要配置桌上管理系統用戶端與伺服器”。
3. 增加資料庫、圖示或輔助說明伺服器至合適的搜尋路徑。
 - 如果已將資料檔置於 第 124 頁“建立資料庫、輔助說明或圖示伺服器”的 步驟 3 中指定的位置，則您便可以修改應用程式搜尋路徑。
 - 如果您將資料檔案放在其它位置中，您必須修改特定的搜尋路徑。
例如，如果您將輔助說明檔案放在系統 SysCCC 上的目錄 /etc/dt/help 中，您會增加下行至 /etc/dt/config/Xsession.d/0010.dtpaths：

```
export DTSPSYSHELP=/net/SysCCC/etc/dt/help
```


有關設定搜尋路徑的更多資訊，請參閱：
 - 第 140 頁“資料庫 (動作/資料類型) 搜尋路徑”
 - 第 142 頁“圖示搜尋路徑”
 - 第 144 頁“輔助說明搜尋路徑”
 - 第 136 頁“設定搜尋路徑的值”

特殊網路的應用程式配置

本節描述如何配置系統以執行應用程式：

- 在遠端執行主機上除包含動作—的系統以外之處
- 局部性地經過檔案系統裝載

指定遠端執行主機

在典型的應用程式伺服器配置中，動作定義與可執行的應用程式位於相同系統上。然而，此動作可被寫入以在其他系統上執行指令。在這個配置中，含有應用程式的系統稱為**執行主機**。

動作定義可位於階段作業伺服器或提供動作與資料類型給階段作業伺服器上——稱為**資料庫伺服器**或**資料庫主機**的系統上。

動作定義使用 EXEC_HOST 欄位以指定其指令 (EXEC_STRINGS) 應執行之處。例如，以下動作定義指定 xload 用戶端以主機名稱 SysDDD 在系統上執行：

```
ACTION XloadSysDDD
{
    TYPE                COMMAND
```

```

EXEC_HOST      SysDDD
EXEC_STRING    /usr/bin/X11/xload -label SysDDD
}

```

如果 EXEC_HOST 欄位指定一個以上的主機名稱，桌上管理系統會試著在每台主機上順序執行 EXEC_STRING，直到其找出可以執行動作者為止。例如，以下 EXEC_HOST 欄位指定動作應先嘗試執行在 SysDDD 上執行 EXEC_STRING，如果失敗，再嘗試 SysEEE。

```

EXEC_HOST      SysDDD, SYSEEE

```

如果 EXEC_HOST 欄位不是為動作而設定，其預設為此值 %DatabaseHost%。%DatabaseHost% 的值是由資料庫搜尋路徑所獲得的。

例如，假定資料庫搜尋路徑已藉由增加下行至 /etc/dt/config/Xsession.d/0010.dtpaths 而修改：

```

DTSPSYSDATABASEHOSTS=SysAAA: ,/net/SysBBB/etc/dt/appconfig/types/C

```

SysAAA 是使用符合主機的語法— SysAAA: 來指定的。發現一使用此搜尋路徑的元素的動作定義將資料庫主機設定為 SysAAA。然而，發現一使用搜尋路徑的 /net/SysBBB... 部分的動作將資料庫主機設定為局部系統，因為語法不包括主機定義式。

▼ 配置遠端執行主機

1. 提供桌上管理系統所需的作業系統網路配置。
請參閱 第 115 頁“配置桌上管理系統的基本作業系統網路”。
2. 提供伺服器所需的一般桌上管理系統配置。
請參閱 第 118 頁“如果要配置桌上管理系統用戶端與伺服器”。
3. 確定應用程式已正確地安裝並為局部執行而配置。

▼ 如果要配置包含動作定義的系統

1. 提供桌上管理系統所需的作業系統網路配置。
請參閱 第 115 頁“配置桌上管理系統的基本作業系統網路”。
2. 提供伺服器所需的一般桌上管理系統配置。
請參閱 第 118 頁“如果要配置桌上管理系統用戶端與伺服器”。
3. 建立與安裝動作定義與應用程式群組。
請參閱 第 187 頁“建立在遠端系統上執行應用程式的動作”與 第 61 頁“建立及管理一般應用程式群組”。

▼ 如果要配置階段作業伺服器

1. 提供桌上管理系統所需的作業系統網路配置。
請參閱 第 115 頁“配置桌上管理系統的基本作業系統網路”。
2. 提供用戶端所需的一般桌上管理系統配置。
請參閱 第 118 頁“如果要配置桌上管理系統用戶端與伺服器”。
3. 修改動作搜尋路徑以包含資料庫主機。
請參閱 第 140 頁“資料庫 (動作/資料類型) 搜尋路徑”。
4. 修改應用程式搜尋路徑以包含執行主機。
請參閱 第 137 頁“應用程式搜尋路徑”。

在本端執行應用程式

標準的應用程式伺服器配置在應用程式伺服器上執行應用程式。有時將應用程式安裝在遠端系統上，但在階段作業伺服器上做局部執行是比較受喜愛的。

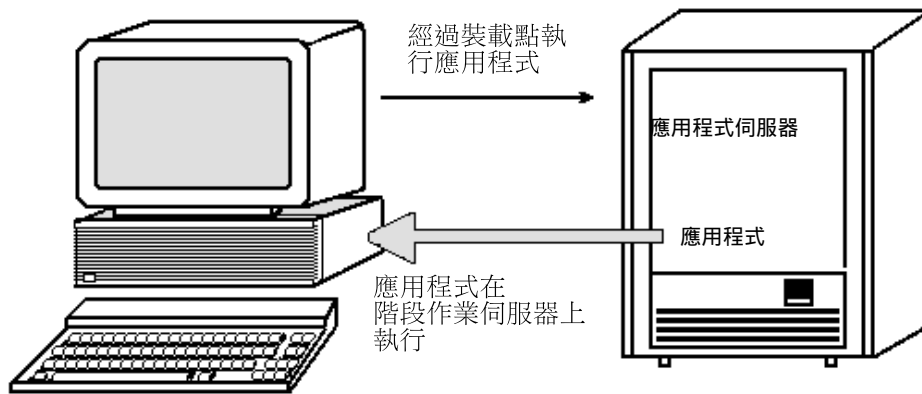


圖 7-6 經過裝載點執行

▼ 如果要配置應用程式伺服器

不需要特殊配置。

▼ 如果要配置階段作業伺服器

- 修改應用程式搜尋路徑。使用局部絕對路徑至應用程式。

例如，您可能會使用以下變數定義以尋找在 `sysAAA` 上註冊的應用程式：

```
DTSPSYSAPPHOSTS=/net/SysAAA/etc/dt/appconfig/appmanager/C
```

階段作業伺服器必須能夠存取應用程式的配置檔案，例如 `app-defaults`、訊息目錄與共用程式庫。

第 8 章

從桌上管理系統配置與管理列印

桌上管理系統使用者有各種列印檔案的方法。方法分為兩大類：從桌上管理系統列印以及從應用程式列印。

從桌上管理系統列印的方法包括：

- 在檔案管理者中選擇檔案，並從「所選項目」功能表或圖示的即現式功能表選擇列印
 - 從檔案管理者將檔案拖曳至前方面板印表機控制或個人印表機輔助面板
 - 從檔案管理者將檔案拖曳至列印管理者主視窗中的印表機
- 如果要從應用程式列印，請使用列印指令，通常是由功能表或應用程式視窗內的其它控制來存取。
- 第 129 頁“增加與刪除印表機”
 - 第 130 頁“印表機圖示影像”
 - 第 132 頁“配置預設印表機”
 - 第 133 頁“列印觀念”

增加與刪除印表機

本節包含從桌上管理系統增加與刪除印表機的程序。

▼ 增加印表機到桌上管理系統

1. 增加印表機至您的系統配置。
遵循您作業系統的系統管理文件中的指令。

2. 執行指令：

```
env LANG=language /usr/dt/bin/dtprintinfo  
-populate
```

3. 從應用程式管理者中的桌面工具應用程式群組來重新啟動列印管理者或連按兩下重新登入動作。驗證印表機已出現。
4. 傳送郵件給您的使用者，以使他們知道他們也應重新啟動列印管理者或執行重新載入動作。

每次呼叫它時，列印管理者就讀取系統印表機配置清單。如果其檢查到新的印表機，其會自動為該印表機建立新的桌上管理系統印表機動作以及圖示。您不需要做任何事來使印表機出現於桌上管理系統上。

▼ 從桌上管理系統刪除印表機

1. 從您的系統配置刪除印表機。
遵循您作業系統的系統管理文件中的指令。
2. 從應用程式管理者中的桌面工具應用程式群組來重新啟動列印管理者或連按兩下重新登入動作。驗證印表機已消失。
3. 傳送郵件給您的使用者，以使他們知道他們也應重新啟動列印管理者或執行重新載入動作。

每次呼叫它時，列印管理者就讀取系統印表機配置清單。如果其看到印表機已從清單中刪除，就會自動從列印管理者與檔案管理者刪除該印表機的動作與圖示。您不需要作任何其它的事來從桌上管理系統刪除印表機。

註解 – 列印管理者無法從前方面板刪除印表機。因此，無論何時，只要您從您的配置刪除印表機，您就應傳送郵件給系統上所有的使用者，告訴他們從前方面板刪除任何已刪除的印表機圖示。

修改工作更新間隔

如果要變更列印管理者中顯示資訊更新的頻率，請修改工作更新間隔。按預設值，列印管理者每三十秒就查詢印表機有關其列印工作的資訊。您可以藉由使用設定選項對話框中的更新間隔調整器（藉由從查看功能表選擇設定選項來顯示）來變更列印管理者查詢印表機的頻率。

印表機圖示影像

當您增加印表機時，它會自動指派預設印表機圖示。如果您有另一個想要使其可供此印表機使用的圖示，請將圖示檔案放在 `/etc/dt/appconfig/icons/language` 中，或是圖示搜尋路徑沿途的一些其它目錄中。然後使用者可以選擇此圖示來取代印表機的預設圖示。

您必須建立完整的圖示集（大型、中型與小型），否則它們將不會出現在列印管理者中的圖示選擇元中。

圖示搜尋路徑的詳細資訊，請參閱 第 142 頁“圖示搜尋路徑”。

圖示檔案名稱與大小

圖示檔案命名需求為：

base_name.size.類型

其中：

大小—l (大), m (中), t (微小)。有關圖示大小的更多資訊，請參閱 第 210 頁“圖示大小慣例”。

類型—pm (色彩像素對映), bm (點陣圖)。

例如，彩色印表機的中型與小型像素對映圖示的檔案名稱可能是 ColorPrinter.m.pm 與 ColorPrinter.t.pm。

有關建立圖示的更多資訊，請參照 第 14 章。

▼ 如果要整體變更圖示、印表機標籤或印表機的說明

您應該在使用者使用列印管理者修改之前，一增加印表機就立刻變更整體印表機的屬性。一旦使用者使用列印管理者修改印表機的屬性，他們會看不到您所做的變更。

以希望的圖示、印表機標籤或說明資訊來編輯檔案
/etc/dt/appconfig/types/language/ printer_queue_name.dt :

1. 在 ICON 欄位,更新基本名稱 為新的圖示基本名稱。
2. 在 LABEL 欄位, 更新標籤名稱 為新的印表機標籤。
3. 更新 DESCRIPTION 欄位中的文字。

這是放置印表機位置、印表機類型與印表機連繫的好地方。如果要增加一行以上，請將 \ 放在行尾處。例如：

```
說明    此為建築物 1 中的 PostScript 印表機 \  
        123 室。如有問題請打 555-5555。
```

配置預設印表機

預設印表機的存取是當使用者：

- 在前方面板印表機控制上定位物件時
- 在檔案管理者中選擇物件，並從「所選項目」功能表或圖示的即現式功能表選擇列印時
- 從使用預設印表機的應用程式列印時

▼ 變更預設列印的目標

如果要變更所有使用者的預設印表機：

1. **開啓檔案** `/etc/dt/config/Xsession.d/0010.dtpaths`。
如果 `/etc/dt/config/Xsession.d/0010.dtpaths` 不存在，請從 `/usr/dt/config/Xsession.d/0010.dtpaths` 複製它。
2. 在 `LPDEST=printer` 行中，更新 `printer` 為預設列印的新目標。
如果此行不存在，請增加一行 `LPDEST= printer`，其中 `printer` 是您想要使其成為您的預設印表機的印表機名稱。
3. 使用者需要登出再登入。

如果要變更單一使用者的預設印表機，該使用者應：

- 從個人印表機輔助面板將另一部印表機複製到前方面板。

如果要指派不同的印表機作為您的預設印表機：

1. 移至您的主檔案夾並開啓檔案 `.dtprofile`。
2. 增加或編輯設定 `LPDEST` 環境變數值的一行：

```
LPDEST=printer_device; export LPDEST
```

如果您正在使用 `csch` 語法為：

```
setenv LPDEST printer_device
```

例如，下行會將預設印表機變更為其裝置名稱為 `laser3d` 的印表機。

```
LPDEST=laser3d; export LPDEST
```

如果您正在使用 `csch` 語法為：

```
setenv LPDEST laser3d
```

列印觀念

當列印要求是藉由將檔案定位於印表機控制上來起始時，系統進行如下：

1. 系統搜尋定位物件定義的資料類型資料庫。
2. 如果有一個資料類型（使用列印動作中的 `ARG_TYPE` 欄位來指定）的唯一列印動作，就是使用它；否則就是使用預設列印動作（`dtlp`）。例如，如果檔案是 PostScript® 檔案，系統使用 PostScript 檔案的列印動作。（此動作定義於 `/usr/dt/appconfig/types/language/dt.dt` 中。）如果您使用建立動作工具於資料類型，您所輸入的列印指令是將會用於列印具有此資料類型的檔案的唯一列印動作。
3. 檔案是使用一般的 UNIX `lp` 列印子系統發送至印表機的。

第 9 章

桌上管理系統搜尋路徑

桌上管理系統使用搜尋路徑來尋找應用程式與其相關的桌上管理系統檔案。

- 第 136 頁“桌上管理系統搜尋路徑與其環境變數”
- 第 136 頁“設定搜尋路徑的值”
- 第 137 頁“應用程式搜尋路徑”
- 第 140 頁“資料庫 (動作/資料類型) 搜尋路徑”
- 第 142 頁“圖示搜尋路徑”
- 第 144 頁“輔助說明搜尋路徑”
- 第 145 頁“本土化搜尋路徑”

桌上管理系統提供四種搜尋路徑，描述於表 9-1 中。

表 9-1 桌上管理系統搜尋路徑

搜尋路徑	說明
應用程式	用於尋找應用程式。當使用者登入時，應用程式管理者使用應用程式搜尋路徑以動態地總裝其頂層。
資料庫	用於指定動作與資料類型定義檔案 (*.dt 檔案) 與前方面板檔案 (*.fp 檔案) 的其它位置。
圖示	用於指定圖示的其它位置。
輔助說明資料	用於指定桌上管理系統輔助說明資料的其它位置。

搜尋路徑可以包括本機與遠端目錄。因此，搜尋路徑在桌上管理系統的網路架構方面扮演了重要的角色。例如，系統在應用程式伺服器上尋找應用程式是因為，應用程式伺服器列在應用程式搜尋路徑中。

當搜尋路徑包括遠端位置時，您必須配置至此位置的遠端檔案存取。有關更多的資訊，請參閱第 116 頁“配置分散式檔案系統存取”。

桌上管理系統搜尋路徑與其環境變數

桌上管理系統搜尋路徑是由桌上管理系統公用程式 `dtsearchpath` 於登入時所建立的。此 `dtsearchpath` 公用程式使用環境變數與內建位置的組合來建立搜尋路徑。

`dtsearchpath` 所讀取的环境變數稱為**輸入變數**。這些是由系統管理者或一般使用者所設定的變數。輸入變數使用命名慣例 `DTSP*`。

當 `dtsearchpath` 於登入時間執行時，其組合指派給這些變數的值、增加內建位置以及建立**輸出變數**的值。每個搜尋路徑都有一個輸出變數。

表 9-2 桌上管理系統搜尋路徑環境變數

搜尋路徑：	輸出環境變數	整個系統的輸入變數	個人輸入變數
應用程式	<code>DTAPPSEARCHPATH</code>	<code>DTSPSYSAPPHOSTS</code>	<code>DTSPUSERAPPHOSTS</code>
資料庫 ¹	<code>DTDATABASESEARCHPATH</code>	<code>DTSPSYSDATABASEHOSTS</code>	<code>DTSPUSERDATABASEHOSTS</code>
圖示	<code>XMICONSEARCHPATH</code> , <code>XMICONBMSEARCHPATH</code>	<code>DTSPSYSICON</code>	<code>DTSPUSERICON</code>
輔助說明資料	<code>DTHELPSEARCHPATH</code>	<code>DTSPSYSHELP</code>	<code>DTSPUSERHELP</code>

¹Actions, data types, and Front Panel definitions

使用元件輸出變數值。例如，應用程式管理者使用應用程式搜尋路徑（`DTAPPSEARCHPATH`）的值來尋找應用程式群組。

設定搜尋路徑的值

您可以用整個系統或個人基礎來修改搜尋路徑。修改是藉由設定整個系統或個人的輸入變數來完成的。您所作的任何修改會增加至內建的搜尋路徑位置。

▼ 查看搜尋路徑目前的值（輸出變數）

- 使用 `dtsearchpath` 指令來顯示搜尋路徑目前的值：
 - 如果要獲得目前（登入）使用者的值，請鍵入指令：

```
dtsearchpath -v
```

- 如果要獲得不同使用者的值，請鍵入指令：

```
dtsearchpath -u user
```


搜尋路徑值包括這些變數：

%H—用於 DTHELPSEARCHPATH. 中。輔助說明內容名稱。

%B—用於 XMICONSEARCHPATH. 中。圖示檔案的基本名稱。

%M —圖示檔案的大小 (XMICONSEARCHPATH.

%L — LANG 環境變數。

▼ 如果要作搜尋路徑的個人修改

1. 開啓 *HomeDirectory/.dtprofile* 以編輯。
2. 增加或編輯定義與匯出個人輸入變數的一行。
例如，下行增加位置至使用者的個人應用程式搜尋路徑：

```
export DTSPUSERAPPHOSTS=/projects1/editors
```
3. 如果要使變更生效，請登出再登入。

▼ 如果要作整個系統的搜尋路徑修改

1. 登入為 **root**。
2. 如果檔案 */etc/dt/config/Xsession.d/0010.dtpaths* 不存在，請藉由複製 */usr/dt/config/Xsession.d/0010.dtpaths* 來建立。
3. 開啓 */etc/dt/config/Xsession.d/0010.paths* 以編輯。增加或編輯定義與匯出整個系統的輸入變數的一行。
例如，下行增加位置至整個系統的輔助說明搜尋路徑：

```
export DTSPSYSHELP=/applications/helpdata
```
4. 通知系統上所有的使用者，他們必須登出再登入以使變更生效。

應用程式搜尋路徑

應用程式搜尋路徑是由桌上管理系統所使用的主要的搜尋路徑，以尋找整個網路上的本機系統與應用程式伺服器上的應用程式。

當增加位置至應用程式搜尋路徑時，其它的搜尋路徑（資料庫、圖示與輔助說明）會自動更新以反映該資料的相對位置；因此，應用程式搜尋路徑提供相當簡易的管理給應用程式及其桌上管理系統配置檔案。請參閱 第 139 頁“應用程式搜尋路徑如何影響資料庫、圖示與輔助說明搜尋路徑”。

預設應用程式搜尋路徑

預設應用程式搜尋路徑包括個人、整個系統以及內建的位置。預設語言是 c。

個人的位置—*HomeDirectory* / .dt/appmanager

整個系統的位置—*etc/dt/appconfig/appmanager/language*

內建位置—*usr/dt/appconfig/appmanager/language*

應用程式搜尋路徑環境變數

應用程式搜尋路徑是由內建位置與下列輸入變數所組合的：

DTSPSYSAPPHOSTS—整個系統的應用程式搜尋路徑輸入變數

DTSPUSERAPPHOSTS—個人的應用程式搜尋路徑輸入變數

組合的搜尋路徑是由輸出變數 DTAPPSEARCHPATH 來指定的。

應用程式搜尋路徑輸入變數的語法

變數的語法 DTSPSYSAPPHOSTS 與 DTSPUSERAPPHOSTS 的格式為：

變數=位置 [, 位置...]

其 位置 可以有語法：

/路徑—在本機（階段作業伺服器）系統上指定目錄。使用此語法來增加本機目錄。

主機名稱:—指定整個系統目錄 */etc/dt/appconfig/appmanager/language* 系統上主機名稱。使用此語法來增加應用程式伺服器。

主機名稱:/路徑—指定遠端系統 主機名稱。

本機主機:—本機系統的位置。此關鍵字是用於改變本機整個系統位置的優先順序。請參閱 第 139 頁“變更整個系統的本機位置優先順序”。

如何組合應用程式搜尋路徑的值

應用程式搜尋路徑（DTAPPSEARCHPATH）的值是藉由組合下列位置所建立的，按照優先順序列出：

- 使用 DTSPUSERAPPHOSTS 變數所指定的位置
 - 預設個人位置：*language* / .dt/appmanager
 - 預設位置：*/etc/dt/appconfig/appmanager/language*
 - 使用 DTSPSYSAPPHOSTS 變數所指定的位置
 - */usr/dt/appconfig/appmanager/language*
- 語法：
- 主機名稱：
- 擴展至在系統 主機名稱 上指定目錄 */etc/dt/appconfig/appmanager* 。

變更整個系統的本機位置優先順序

按預設值，整個系統的本機位置 (*/etc/dt/appconfig/appmanager/language*) 對於遠端位置有優先順序。因此，本機應用程式群組對於具有相同名稱的遠端群組有優先順序。例如，如果本機與遠端系統都有印表機應用程式群組 (*/etc/dt/appconfig/appmanager/language /Printers*)，則是使用本機群組。

應用程式搜尋路徑輸入變數提供指定本機整個系統的應用程式群組優先順序的語法：

本機主機：

例如，假定您的系統必須存取應用程式伺服器 *SysA*、*SysB* 與 *SysC*，而您想要 *SysB* 上的整個系統應用程式群組對於具有相同名稱的任何本機群組有優先順序。

以下 DTSPSYSAPPHOSTS 的值建立此行為：

```
DTSPSYSAPPHOSTS=SysB: , localhost: , SysA: , SysC:
```

應用程式搜尋路徑如何影響資料庫、圖示與輔助說明搜尋路徑

應用程式搜尋路徑的附加物自動將同等位置增加至資料庫、圖示與輔助說明搜尋路徑。這個動作會藉由只設定應用程式搜尋路徑輸入變數，來提供將應用程式伺服器增加至搜尋路徑的能力。

例如，如果您設定 DTSPSYSAPPHOSTS 如下：

```
export DTSPSYSAPPHOSTS=servera:
```

那麼搜尋路徑所受到的影響如表 9-3 中所示。

表 9-3 受影響的搜尋路徑

搜尋路徑	增加至搜尋路徑的目錄
應用程式	<code>servera:/etc/dt/appconfig/appmanager/language</code>
資料庫	<code>servera:/etc/dt/appconfig/types/ language</code>
圖示	<code>servera:/etc/dt/appconfig/icons/language</code>
輔助說明	<code>servera:/etc/dt/appconfig/help/language</code>

同樣地，如果您設定 DTSPSYSAPPHOSTS 如下：

```
export DTSPSYSAPPHOSTS=/projects1/apps
```

那麼搜尋路徑所受到的影響如表 9-4 中所示。

表 9-4 受影響的搜尋路徑

搜尋路徑	增加至搜尋路徑的目錄
應用程式	<code>/projects1/apps/appmanager/language</code>
資料庫	<code>/projects1/apps/types/ language</code>
圖示	<code>/projects1/apps/icons/language</code>
輔助說明	<code>/projects1/apps/help/language</code>

資料庫 (動作/資料類型) 搜尋路徑

資料庫搜尋路徑指示桌上管理系統搜尋指定的檔案位置包含：

- 動作與資料類型定義 (*.dt 檔案)
- 前方面板定義 (*.fp 檔案)。

當您建立資料庫伺服器，或當您增加資料庫檔案的本機位置時，您可能會需要修改資料庫搜尋路徑。

預設資料庫搜尋路徑

預設資料庫搜尋路徑包括個人的、整個系統的以及內建的位置。預設 語言 為 C。

個人的位置—`HomeDirectory / .dt/types`

整個系統的位置—`/etc/dt/appconfig/types/language`

內建的位置—`/usr/dt/appconfig/types/language`

應用程式搜尋路徑如何影響資料庫搜尋路徑

當位置增加至應用程式搜尋路徑時，合適的資料庫子目錄會自動增加至資料庫搜尋路徑（請參閱第 139 頁“應用程式搜尋路徑如何影響資料庫、圖示與輔助說明搜尋路徑”）。

例如，如果將應用程式伺服器 `hosta:` 增加至應用程式搜尋路徑，目錄 `hosta:/etc/dt/appconfig/types/language` 會自動增加至資料庫搜尋路徑。

資料庫搜尋路徑環境變數

資料庫搜尋路徑是由內建位置與下列輸入變數所組合的：

`DTSPSYSDATABASEHOSTS`—整個系統的資料庫搜尋路徑輸入變數

`DTSPUSERDATABASEHOSTS`—個人資料庫搜尋路徑輸入變數

使用這些輸入變數來指定應用程式搜尋路徑之外的位置。

組合的資料庫搜尋路徑是由輸出變數 `DTDATABASESEARCHPATH` 來指定的。

資料庫搜尋路徑輸入變數的語法

變數的語法 `DTSPSYSDATABASEHOSTS` 與 `DTSPUSERDATABASEHOSTS` 的格式為：

`變數=位置 [, 位置...]`

其 `language` 可以有語法：

`/路徑`—在本機（階段作業伺服器）系統上指定目錄。使用此語法來增加本機目錄。

主機名稱：—在系統主機名稱上的，指定整個系統目錄
`/etc/dt/appconfig/types/language`。

主機名稱：`/路徑`—指定遠端系統主機名稱上的目錄。

如何組合資料庫搜尋路徑

資料庫搜尋路徑（`DTDATABASESEARCHPATH`）的值是藉由組合下列位置所建立的，按照優先順序列出：

- 使用 `DTSPUSERDATABASEHOSTS` 變數所指定的位置

- 由 DTSPUSERAPPHOSTS 變數所指定的位置
- 預設個人位置：*HomeDirectory* / *.dt/types*
- 預設位置：*/etc/dt/appconfig/types/language*
- 使用 DTSPSYSDATABASEHOSTS 變數所指定的位置
- 由 DTSPSYSAPPHOSTS 變數所指定的位置
- */usr/dt/appconfig/types/language*

語法：

主機名稱：

擴展為指定系統主機名稱上的目錄 */etc/dt/appconfig/types*。

圖示搜尋路徑

圖示搜尋路徑指示桌上管理系統搜尋包含由桌上管理系統所使用的點陣圖與像素對映影像檔案的檔案指定位置。

預設圖示搜尋路徑

預設圖示搜尋路徑包括個人、整個系統與內建的位置。預設語言為 C。

個人的位置—*HomeDirectory* / *.dt/icons*

整個系統位置—*/etc/dt/appconfig/icons/language*

內建位置—*/usr/dt/appconfig/icons/language*

應用程式搜尋路徑如何影響圖示搜尋路徑

當位置增加至應用程式搜尋路徑時，合適的圖示子目錄會自動增加至圖示搜尋路徑（請參閱第 139 頁“應用程式搜尋路徑如何影響資料庫、圖示與輔助說明搜尋路徑”）。

例如，如果將應用程式伺服器 *hosta:* 增加至應用程式搜尋路徑，目錄 *hosta:/etc/dt/appconfig/icons/language* 會自動增加至圖示搜尋路徑。

圖示搜尋路徑環境變數

資料庫搜尋路徑是由內建位置與下列輸入變數所組合的：

DTSPSYSICON—整個系統的圖示搜尋路徑輸入變數

DTSPUSERICON—個人圖示搜尋路徑輸入變數

使用這些輸入變數來指定應用程式搜尋路徑之外的位置。

組合的資料庫搜尋路徑是由兩個輸出變數所指定的：

XMICONSEARCHPATH—由彩色顯示器所使用

XMICONBMSEARCHPATH—由單色顯示器所使用

圖示搜尋路徑輸入變數的語法

變數的語法 DTSPSYSICON 與 DTSPUSERICON 的格式為：

變數=位置 [, 位置...]

其 位置 可以有語法：

/路徑—在本機（階段作業伺服器）系統上指定目錄。使用此語法來增加本機目錄。

如果要在其它系統上指定位置，請使用其網路檔案名稱—例如，
/nfs/servera/projects/icons。

如何組合圖示搜尋路徑

圖示搜尋路徑（XMICONSEARCHPATH 與 XMICONBMSEARCHPATH）的值是藉由組合下列位置所建立的，按照優先順序列出：

- 使用 DTSPUSERICON 變數所指定的位置
- 由 DTSPUSERAPPHOSTS 變數所指定的位置
- 預設個人位置：*HomeDirectory* / .dt/icons
- 預設位置：*/etc/dt/appconfig/icons/language*
- 使用 DTSPSYSICON 變數所指定的位置
- 由 DTSPSYSAPPHOSTS 變數所指定的位置
- */usr/dt/appconfig/icons/language*

彩色與單色搜尋路徑只在給予像素對映與點陣圖的優先順序上有所不同。此 XMICONSEARCHPATH 變數將像素對映列在點陣圖之前；XMICONBMSEARCHPATH 將點陣圖列在像素對映之前。

輔助說明搜尋路徑

輔助說明搜尋路徑指示桌上管理系統搜尋，包含將會註冊於您系統上的輔助說明資訊檔案的指定位置。

預設輔助說明搜尋路徑

預設輔助說明搜尋路徑包括個人、整個系統與內建的位置。預設 **語言** 為 C。

個人的位置—*HomeDirectory* / .dt/help

整個系統的位置—*/etc/dt/appconfig/help/language*

內建位置—*/usr/dt/appconfig/help/language*

應用程式搜尋路徑如何影響輔助說明搜尋路徑

將位置增加至應用程式搜尋路徑時，合適的輔助說明子目錄會自動增加至輔助說明搜尋路徑（請參閱 第 139 頁“應用程式搜尋路徑如何影響資料庫、圖示與輔助說明搜尋路徑”）。

例如，如果將應用程式伺服器 `hosta:` 增加至應用程式搜尋路徑，目錄 `hosta:/etc/dt/appconfig/help/language` 會自動增加至輔助說明搜尋路徑。

輔助說明搜尋路徑環境變數

輔助說明搜尋路徑是由內建位置與下列輸入變數所組合的：

`DTSPSYSHELP`—整個系統的輔助說明搜尋路徑輸入變數

`DTSPUSERHELP`—個人輔助說明搜尋路徑輸入變數

使用這些輸入變數來指定應用程式搜尋路徑之外的位置。

組合的資料庫搜尋路徑是由輸出變數 `DTHELPPSEARCHPATH` 來指定的。

輔助說明搜尋路徑輸入變數的語法

變數的語法 `DTSPSYSHELP` 與 `DTSPUSERHELP` 的格式為：

`變數=位置 [, 位置...]`

其 `位置` 可以有語法：

/路徑—在本機（階段作業伺服器）系統上指定目錄。使用此語法來增加本機目錄。

如果要在其它系統上指定位置，請使用其網路檔案名稱—例如，`/nfs/servera/projects/help`。

如何組合輔助說明搜尋路徑

輔助說明搜尋路徑 (DTHELPSEARCHPATH) 的值是藉由組合下列位置所建立的，按照優先順序列出：

- 使用 DTSPUSERHELP 變數所指定的位置
- 由 DTSPUSERAPPHOSTS 變數所指定的位置
- 預設個人位置：*HomeDirectory* / *.dt/help*
- 預設位置：`/etc/dt/appconfig/help/language`
- 使用 DTSPSYSHELP 變數所指定的位置
- 由 DTSPSYSAPPHOSTS 變數所指定的位置
- `/usr/dt/appconfig/help/language`

本土化搜尋路徑

輸出變數包括本土化的與預設的 (C) 位置兩者的登錄。

例如，預設應用程式搜尋路徑為：

```
HomeDirectory/.dt/appmanager  
/etc/dt/appconfig/appmanager/language /etc/dt/appconfig/appmanager/C  
/usr/dt/appconfig/appmanager/language /usr/dt/appconfig/appmanager/C
```

其 *language* 是 LANG 環境變數的值。

對於每個範圍（整個系統與內建的）而言，特定的語言位置對於預設位置有優先順序。

第 10 章

動作與資料類型簡介

動作與資料類型是整合應用程式與桌上管理系統功能強大的元件。它們提供建立啟動應用程式與操縱其資料檔案的使用者介面方法。

- 第 147 頁“動作的介紹”
- 第 152 頁“資料類型的介紹”

本章介紹動作與資料類型的觀念。其描述：

- 您為何會想要建立應用程式的動作與資料類型。
- 動作與資料類型彼此有何關聯。
- 動作與資料類型如何與桌上管理系統列印有關聯。

本手冊中有三章涵蓋了建立動作與資料類型的程序與規則。

- 第 11 章 解釋如何使用桌上管理系統應用程式建立動作，來建立動作與資料類型。
您可以使用建立動作來建立大部分應用程式的動作與資料類型，而不需學習其定義的語法規則。
- 第 12 章 及 第 13 章 解釋如何藉由建立與編輯配置檔案來人工建立動作與資料類型。
當您想要使用建立動作不支援的進階功能時，您必須人工建立動作與資料類型。

動作的介紹

動作是所寫的指令使桌上管理系統工作自動化，例如執行應用程式與開啓資料檔案的指令。動作的運作與應用程式巨集或程式設計功能很相像。每個動作都有一個用以執行動作的名稱。

一旦您定義了動作，就可以使用它來配接桌上管理系統使用者介面，以使得工作更容易執行。桌上管理系統提供附加使用者介面元件，例如圖示、前方面板控制，以及動作之功能表項目的能力。

例如，應用程式管理者中的桌面工具應用程式群組，包含啟動各種公用程式的圖示。



圖 10-1 桌面工具應用程式群組中的動作圖示

連按兩下時，每一個這些圖示都會執行一個動作。例如，此為當使用者連按兩下標示 Xwd 顯示器的圖示時，所執行的動作定義的一部分。配置檔案 `/usr/dt/appconfig/types/language/xclients.dt` 中：

```
ACTION Xwd
{
  LABEL          Xwd Display
  TYPE           COMMAND
  EXEC_STRING    /usr/bin/X11/xwud -noclick -in \
                %(File)Arg_1"Xwd File To Display:"%
  ...
}
```

當使用者連按兩下圖示時，會執行動作的 `EXEC_STRING` 中的指令。

前方面板也使用動作。例如，此為個人應用程式輔助面板中標為終端機的控制定義部分。配置檔案 `/usr/dt/appconfig/types/language/dtwm.fp`:

```
CONTROL Term
{
  ICON          Fpterm
  LABEL         Terminal
  PUSH_ACTION   Dtterm
  ...
}
```

當使用者按一下控制時，`PUSH_ACTION`欄位會指定執行的動作—在此情況中，動作名為 `Dtterm`。

其它動作常見的用法在功能表中。資料檔案在其檔案管理者中的「所選項目」功能表中通常會有動作。例如，XWD 檔案（具有名稱以 .xwd 或 .wd 結尾的檔案）有藉由執行 Xwud 動作來顯示螢幕影像的開啓動作。

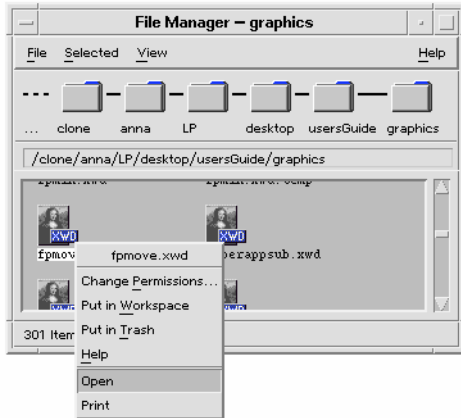


圖 10-2 開啓資料類型 XWD 檔案的動作

XWD 檔案的資料類型定義中指定了「所選項目」功能表中的動作。定義位於配置檔案 `/usr/dt/appconfig/types/language/xclients.dt` 中。

```
DATA_ATTRIBUTES XWD
{
    ACTIONS      Open, Print
    ICON         Dtxwd
    ...
}
```

XWD 資料類型和與其有關聯的開啓與列印動作在 第 153 頁“資料類型如何將資料檔案連接至動作”中解釋。

動作如何建立應用程式的圖示

試想桌面工具應用程式群組中的 Xwd 顯示器圖示。連按兩下這個圖示將執行 X 用戶端 `xwud`。然而，此圖示並不直接代表實際的 `xwud` 可執行 `/usr/bin/X11/xwud`。

標示 Xwd 顯示器的圖示出現在應用程式群組中，是因為該目錄中有一個名為 `Xwud` 的檔案（請參閱 圖 10-3）。此檔案代表具有相同名稱—`Xwud` 的隱藏的動作。在動作定義中，動作名稱是跟隨著動作 關鍵字的名稱：

```
ACTION Xwud
{
    LABEL          Xwd Display
    TYPE           COMMAND
    WINDOW_TYPE   NO_STDIO
}
```

```

EXEC_STRING          /usr/bin/X11/xwud -noclick -in \
                    %(File)Arg_1"Xwd File To Display:"%
DESCRIPTION          The Xwd Display (Xwud) XwdDisplay action \
                    displays an xwd file that was created using the \
                    Xwd Capture (Xwd) action. It uses \
                    the xwud command.
}

```

此檔案稱為**動作檔案**是因為其代表動作。當檔案是具有與動作相同名稱的可執行檔案時，則為動作檔案。其圖示在應用程式管理者（或檔案管理者）中稱為**動作圖示**或**應用程式圖示**，因為連按兩下就會啟動應用程式。

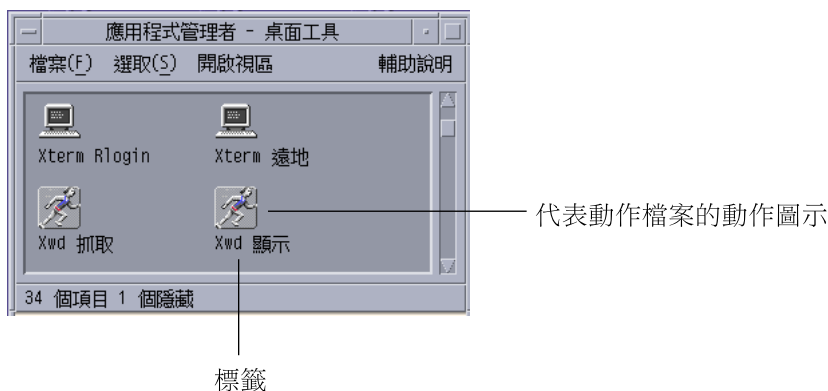


圖 10-3 代表動作檔案的應用程式（動作）圖示

當應用程式管理者檢查到可執行的檔案時，它會仔細檢查動作資料庫來看看是否有任何名稱符合檔案名稱的動作。如果找到符合的，應用程式管理者就知道此檔案是動作檔案。

動作檔案的內容是無關的；動作檔案通常包含描述其桌上管理系統功能的註釋。

註解 – 此**動作檔案**與**動作定義檔案**是不同的。此**動作檔案**是具有與動作名稱相同的檔案。它是用於建立檔案管理者或應用程式管理者中的**應用程式圖示**。**動作定義檔案**是包含動作定義的名為 *name.dt* 的檔案。

一旦桌上管理系統決定一個檔案是動作檔案，隱藏的動作定義是用於定義動作檔案的外觀與行為。

- 此 EXEC_STRING 欄位指定應用程式圖示的行為。在 Xwd 顯示器圖示的情況中，EXEC_STRING 會指定動作圖示執行具有特定指令行引數的 xwud X 用戶端。
- 此標籤欄位指定應用程式圖示的標籤。
- 此說明欄位描述於使用者要求「關於項目」輔助說明時所顯示的文字。

- Xwud 應用程式圖示使用動作的預設圖示影像，因為其動作定義不包含 圖示 欄位以指定不同的影像。
相對之下，標示為壓縮檔案的圖示使用不同的圖示影像，因為其隱藏的動作定義包含 圖示 欄位：

例如：

```
ACTION Compress
{
    LABEL           Compress File
    ICON            Dtmpmrs
    ...
}
```

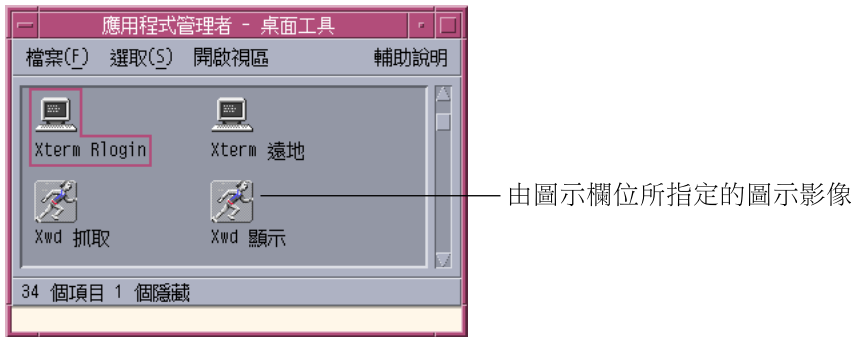


圖 10-4 在動作定義中由圖示欄位所指定的圖示影像

Xwud 動作稱為**指令動作**，因為其定義包含要執行的指令（EXEC_STRING）。動作定義中的類型欄位定義動作類型。

剛開始時，Xwd 顯示器圖示出現在桌面工具應用程式群組中。但是，您可以在任何您有寫入許可權的目錄中建立動作圖示的其它副本。只要 Xwud 動作定義是資料庫的一部分，任何您所建立名為 xwud 的可執行檔案將會是代表該動作的動作檔案，而且其在檔案管理者或應用程式管理者中的圖示可以用以執行動作。

動作如何使用資料檔案作為引數

指令的**引數**通常是指令在其上發生作用的東西，通常是一個檔案。可以寫入動作以接受檔案引數。

例如，Xwud 動作的 EXEC_STRING 指定檔案引數是必需的：

```
EXEC_STRING      /usr/bin/X11/xwud -noclick -in \
                 %(File)Arg_1"Xwd File To Display:"%
```

名詞 Arg 代表**引數**這個字。語法 Arg_1 表示第一個引數，而 (File) 表示動作將引數當作是檔案。

對使用者而言，要提供檔案引數最容易的方法就是在應用程式圖示上定位資料檔案。桌上管理系統決定定位檔案的路徑，並將它放在 % symbols 間的文字以取代這部份的指令行 (%(檔案)Arg_1"Xwd 欲顯示的檔案："%)。因此，執行的指令為：

```
/usr/bin/X11/xwud -noclick -in file_path
```

當使用者連按兩下應用程式圖示時，桌上管理系統會從 EXEC_STRING 決定檔案引數是需要的，並顯示提示使用者輸入檔案名稱或路徑的對話框。在 Xwud 動作的例子中，提示符號為：

欲做顯示的 Xwd 檔案：

由使用者所供應的檔案名稱或路徑作為檔案引數使用。

動作的其他用途

除了啟動應用程式之外，動作還用於整個桌上管理系統以建立功能於：

- 前方面板
前方面板控制的定義包括，指定當使用者按一下控制或在其上定位檔案時，所執行的動作欄位。有關更多的資訊，請參閱 第 231 頁“定義前方面板控制”。
- 功能表
視窗與工作區功能表定義的語法，允許您指定由功能表項目執行的動作。有關更多的資訊，請參閱 第 248 頁“工作區管理者功能表”與 dtwmrc(4) 線上援助頁。
- 應用程式之間的通信
應用程式可以設計來使用稱為 ToolTalk 訊息 (TT_MSG) 的特殊類型動作以傳送與接收資訊。TT_MSG 動作將在桌上管理系統的開發者環境文件中加以描述。

資料類型的介紹

當使用者建立新的資料檔案時，檔案管理者中的檔案圖示外觀與行為會依據使用者所建立的資料檔案類型而改變。此種建立檔案與目錄的自訂外觀與行為的能力是由桌上管理系統的資料鍵入機制所提供的。

什麼是資料類型？

資料類型是在桌上管理系統資料庫內定義的一個結構。例如，此為 XWD 資料類型的定義。定義是在配置檔案 /usr/dt/appconfig/types/language/xclients.dt 中：

```
DATA_ATTRIBUTES XWD
{
    ACTIONS          Open, Print
```



```

        ICON                Dtxwd
        NAME_TEMPLATE       %s.xwd
        MIME_TYPE           application/octet-stream
        SUNV3_TYPE          xwd-file
        DESCRIPTION         This file contains a graphics image in the XWD \
                            format. These files are typically created by \
                            taking snapshots of windows using the XwdCapture \
                            action. Its data type is named XWD. XWD files \
                            have names ending with '.xwd' or '.wd'.
    }

DATA_CRITERIA XWD1
{
    DATA_ATTRIBUTES_NAME   XWD
    MODE                    f
    NAME_PATTERN            *.xwd
}

DATA_CRITERIA XWD2
{
    DATA_ATTRIBUTES_NAME   XWD
    MODE                    f
    NAME_PATTERN            *.wd
}

```

每種資料類型定義都有兩部分：

DATA_ATTRIBUTES—描述資料類型的外觀與行為。

DATA_CRITERIA—指定將檔案分類為屬於該資料類型的規則（命名或內容）。

此 DATA_ATTRIBUTES_NAME 欄位將條件連接至屬性。

可以有 DATA_ATTRIBUTE 的多重 DATA_CRITERIA。例如，XWD 資料類型有兩項指定兩項不同命名條件（NAME_PATTERN）——名稱以 .xwd 或 .wd 結尾。

資料類型如何將資料檔案連接至動作

試想 XWD 資料類型。使用者藉由給予檔案一個或兩個檔案名稱的字尾（副檔名）來建立 XWD 類型的檔案：.xwd 或 .wd。使用者可以建立 XWD 類型檔案，方法是給與檔案這兩個檔案名稱字尾（延伸名稱）之一：.xwd 或 .wd。桌上管理系統使用該檔案名稱為指派檔案為該類型的條件。

XWD 資料類型供應該資料類型的每個檔案：

- 輔助使用者認出資料檔案的唯一圖示影像。
- 告訴您有關資料類型的關於項目輔助說明。
- 包含動作開啓與列印的檔案管理者中自訂的「所選項目」功能表。XWD 檔案的開啓動作執行 Xwud 動作。

從選定的功能表執行動作

檔案管理者中的「所選項目」功能表只有在選擇檔案或目錄時才是作用中的。「所選項目」功能表底部的指令視資料類型而定。例如，如果選擇 XWD 檔案，「所選項目」功能表就包括開啓與列印項目。

資料類型定義中的動作欄位指定增加至資料類型的「所選項目」功能表底部的指令。

```
DATA_ATTRIBUTES XWD { ACTIONS Open,Print ... }
```

「所選項目」功能表的內容視資料類型而定。然而，許多不同的資料類型都提供開啓動作——也就是說，當您選擇檔案管理者中該特殊資料類型的檔案並顯示「所選項目」功能表時，您會看到開啓指令。

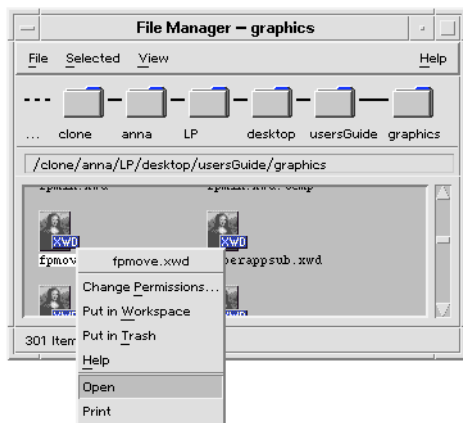


圖 10-5 XWD 檔案的「所選項目」功能表

開啓動作通常執行與資料檔案有關聯的應用程式。例如，開啓 XWD 檔案執行 Xwud 動作，這個動作會換過來執行 xwud X 用戶端以顯示螢幕影像。換言之，對於 XWD 資料類型而言，開啓動作與 Xwud 動作是同義的。同樣地，開啓資料類型文字檔案（TEXTFILE）的檔案會執行文字編輯程式，而開啓 BM（點陣圖）或 PM（像素對映）檔案則會執行圖示編輯程式。

建立各種執行不同事物的開啓動作能力，會使用動作定義的兩項功能：

- 動作對映
動作對映讓您建立執行其它動作的動作，而非直接執行指令。例如，您可以建立對映至（執行）Xwud 動作的開啓動作。
- 動作上的資料類型限制
動作定義可以包括限制特定資料類型的動作的 ARG_TYPE 欄位。例如，您可以指定對映至 Xwud 動作的開啓動作只應用於資料類型 XWD 的檔案。

這是為 XWD 資料類型對映開啓動作至 Xwud 動作的動作定義。其位於資料庫配置檔案 `/usr/dt/appconfig/types/C/xclients.dt` 中。

```
ACTION Open
{
    LABEL                Open
    ARG_TYPE              XWD
    TYPE                  MAP
    MAP_ACTION            Xwud
}
```

此類型欄位指定此為對映動作；MAP_ACTION 欄位指定此動作執行 Xwud 動作。此 ARG_TYPE 欄位指定此動作只應用於其資料類型是 XWD 的檔案。

比較開啓動作的前定義與出現在資料庫檔案 `/usr/dt/appconfig/types/C/dt.dt` 中的下一個定義。

```
ACTION Open
{
    LABEL                Open
    ARG_TYPE              BM
    TYPE                  MAP
    MAP_ACTION            Dticon
}
```

此定義應用於資料類型 (ARG_TYPE) BM (點陣圖檔案) 的檔案。定義對映開啓動作至執行圖示編輯程式的 Dticon 動作。

定義資料類型的連按兩下的行為

資料類型的連按兩下行為是由動作欄位中第一項登錄所定義的。例如，對於 XWD 資料類型而言，連按兩下的行為就是執行開啓動作，此動作會反過來執行 Xwud 動作。

在動作圖示上定位資料檔案

當使用者在動作圖示上定位資料檔案時，系統會使用該資料檔案作為動作的引數來執行動作（請參閱第 151 頁“動作如何使用資料檔案作為引數”）。

例如，當 XWD 資料檔案定位於 Xwd 顯示器圖示上時，Xwud 動作會使用資料檔案引數來執行。此動作會執行具有該資料檔案的 `xwud X` 用戶端。

建立資料類型的桌上管理系統列印

桌上管理系統列印提供這些列印資料檔案的方法：

- 如果有的話，使用檔案管理者「所選項目」功能表中的列印指令。
- 在桌上管理系統印表機定位區域（列印管理者中的前方面板印表機控制或印表機圖示）上定位資料檔案。

除了桌上管理系統列印之外，許多應用程式都提供從應用程式內列印的方法。

桌上管理系統列印使用名為列印的動作。就像開啓一樣，列印是用於許多不同類型的資料的動作名稱。因此，列印動作使用動作對映與 ARG_TYPE 欄位來自訂每種資料類型的列印。

例如，此為 XWD 資料類型的列印動作。定義位於 /usr/dt/appconfig/types/language/xclients.dt 中：

```
ACTION Print
{
    LABEL          Print
    ARG_TYPE       PCL
    TYPE           MAP
    MAP_ACTION     PrintRaw
}
```

此專用於 XWD 檔案的列印動作對映至無列印動作。無列印是 /usr/dt/appconfig/types/language/dt.dt 中所定義的特殊動作。無列印動作顯示對話框，告訴使用者此資料類型無法列印。

比較 XWD 列印動作與以下 PCL 檔案的列印動作：

```
ACTION Print
{
    LABEL          Print
    ARG_TYPE       PCL
    TYPE           MAP
    MAP_ACTION     PrintRaw
}
```

配置檔案 /usr/dt/appconfig/types/language/print.dt 中所定義的列印原始動作包含列印 PCL 檔案的指令行。

```
ACTION PrintRaw
{
    TYPE           COMMAND
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING     /usr/dt/bin/dt1p -w% (File
```

第 11 章

使用建立動作來建立動作與資料類型

建立動作是建立的工具：

- 啟動應用程式的動作
- 應用程式資料檔案的一種或以上的資料類型
- 開啓與列印應用程式的資料檔案的動作

建立動作對於建立執行作業系統指令與 shell 程序檔也是很有用的。

- 第 157 頁“建立動作會做什麼”
- 第 158 頁“建立動作的限制”
- 第 159 頁“以建立動作來建立應用程式的動作及資料類型”

有關參考資訊，請參閱 dtcreate (1X) 線上援助頁。

建立動作會做什麼

建立動作包括主視窗與一組建立動作和與其相關資料類型的對話框。

建立動作執行如下：

- 建立執行指令的動作定義。
- 建立檔案 `HomeDirectory/.dt/types/action_name.dt`。此檔案儲存為應用程式所建立的動作與資料類型定義。
- 在使用者的主目錄中建立**動作檔案**。動作檔案是具有與動作相同名稱的可執行檔案。檔案管理者中的動作檔案表示方式稱為**應用程式圖示**，因為連按兩下會啟動應用程式。
當您建立動作時，您可以選擇性地藉由指定可定位的資料類型來使動作圖示成為定位區域。
- 建立應用程式資料檔案的一個或以上的資料類型（選擇性的）。

- 建立每種資料類型的開啓動作。
- 建立每種資料類型的列印動作（選擇性的）。
- 重新載入動作與資料類型的資料庫。這樣會使得動作與資料類型立即生效。

建立動作的限制

建立動作是設計來建立執行應用程式的動作與資料類型。不過，動作與資料類型非常有彈性，而且包括唯有當您人工建立定義時才可存取的其他功能。

有關更多的資訊，請參閱：

- 第 12 章
- 第 13 章

動作限制

如果下列條件有任何一項是正確的，您就無法使用建立動作來建立應用程式的動作：

- 指令行需要非檔案的引數（參數）。

例如，您無法使用建立動作來寫入指令的動作：

```
lp -d 裝置檔案名稱
```

其中每次執行指令時，使用者都必須供應 **裝置**。

- 應用程式圖示必須有與動作名稱不同的標籤。
例如，您無法使用建立動作來提供現存動作的本地語言版本。
- 動作需要動作資料庫的任何一種進階功能。
這些進階功能的範例是如下的動作：
 - 從動作定義啓動遠端系統上的指令
 - 呼叫其它動作
 - 必須以不同的使用者（例如，以高階使用者）來執行
 - “對映”功能的延伸利用
 - 視供應至動作的檔案引數的數目而定，而會有相當不同的行爲

資料類型限制

如果下列條件任何一項是正確的，您就無法使用建立動作來建立應用程式的資料類型：

- 資料類型必須有開啓與列印以外的與其有關聯的其他動作。
- 資料類型的開啓動作不是動作的指令。

例如，您無法使用建立動作來建立提供代表應用程式群組目錄的唯一圖示的資料類型。

以建立動作來建立應用程式的動作及資料類型

在您執行建立動作之前，您需要知道有關應用程式的一些事情。

- 啟動應用程式的指令行。
您需要知道指令行是包括所需的檔案引數、選擇性的檔案引數或是沒有檔案引數。如果應用程式需要非檔案的引數，您就無法使用建立動作來建立動作。
- 應用程式可以接受的資料檔案類型。
有些應用程式只可接受一種資料類型。其它的（例如，ASCII 編輯程式或圖形編輯程式）可以接受多重資料類型。
- 應用程式識別其資料檔案的方法。
此可能是命名慣例（例如，以 .doc 結尾的檔案名稱），及/或可能取決於檔案的內容。如果應用程式不使用檔案名稱慣例，您仍然可以為動作圖示設定一個。
- 選擇性的：列印檔案的指令行。

▼ 如果要建立應用程式的動作

1. 在桌面應用程式群組中連按兩下建立動作。



圖 11-1 在應用程式管理者中建立動作圖示

這會顯示主要的「建立動作」視窗。



圖 11-2 建立動作主視窗

2. 鍵入將會將動作圖示標示在動作名稱文字欄位中的名稱。
3. 使用動作圖示控制來指定應用程式的圖示。剛開始時，會顯示預設圖示。
 - 如果要選擇不同的現存圖示，請按一下找出組別以顯示找出組別對話框。請參閱 第 166 頁“使用尋找集對話方塊來指定一個圖示”。
 - 如果要建立新圖示，請選擇編輯圖示來執行圖示編輯程式。
4. 在動作圖示開啓時的指令文字欄位中，鍵入指令以啓動應用程式。

使用語法 $\$n$ 檔案引數的; 例如：

```
emacs
  bitmap $1
  diff $1 $2
  lp -oraw $1
```

如果指令行包括檔案引數 ($\$n$)，那麼動作圖示將會是檔案的定位區域。

除非您明確地指定 shell 的使用，否則指令行不會傳送至 shell。例如，這些行使用 shell 處理：

```
/bin/sh -c 'ps | lp'
/bin/sh -c 'spell $1 | more'
```


5. 將動作圖示的「關於項目」輔助說明文字鍵入動作圖示的輔助說明文字欄位中。
文字將會在文字欄位中自動換行。但是，這些斷行不會保留在線上。如果您想要指定強制的斷行，請使用 \n。
6. 從視窗類型選項功能表選擇動作所需的視窗支援。
圖形 (X 視窗) — 應用程式建立其自己的視窗
終端機 (自動關閉) — 應用程式會在使用者結束應用程式時自動關閉的終端機模擬器視窗中執行
終端機 (人工關閉) — 應用程式會在終端機模擬器視窗中執行，此視窗會保持開啓直到使用者確實地關閉它為止
無輸出 — 應用程式不產生輸出至顯示器
7. 進行如下：
 - 如果您的應用程式有資料檔案，而您想要為其建立一個或以上的資料類型，請參閱下節。第 161 頁“建立一個或以上的應用程式資料類型”
 - 如果您不需要建立資料類型，請藉由從檔案功能表選擇儲存來儲存動作。然後，藉由在您的主目錄中連按兩下其圖示來測試新動作。

▼ 建立一個或以上的應用程式資料類型

1. 使用上一節中的程序來定義應用程式的動作，第 159 頁“如果要建立應用程式的動作”。
2. 按一下建立動作視窗中的進階按鈕以擴展視窗。

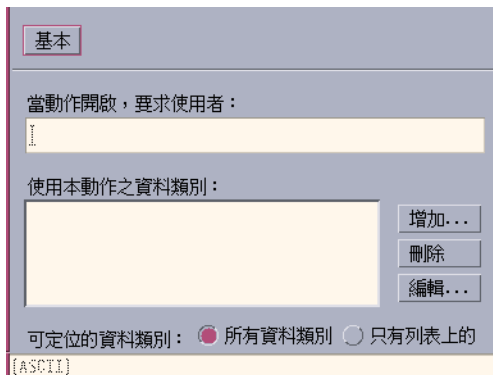


圖 11-3 主建立動作視窗中的進階功能

3. 如果您想要應用程式圖示在連按兩下圖示時提示檔案引數，請將提示符號的文字鍵入“當動作開啓時，要求使用者”的文字欄位中。
使用這些指導於文字欄位：
 - 如果應用程式的指令行有**需要的**檔案引數，您就必須使用此欄位。

- 如果指令行不包括檔案引數，您便必須將此欄位空白。
 - 如果應用程式指令行中的檔案引數是選擇性的，您就可以有選擇。如果您供應提示符號文字，動作圖示將會在連按兩下時為此檔案做提示。如果您沒有供應提示符號文字，動作將會以空字串作為檔案引數來執行。
4. 指定動作將會接受為引數的檔案類型：
- 如果動作可以接受任何資料類型，請選擇「所有資料類型」。
 - 如果動作只能接受您為應用程式所建立的資料類型，請選擇「以上只有的清單」。剛開始時，「使用此動作的資料類型」清單是空白的。當您建立應用程式的資料類型時，它們就會增加至清單中。
5. 按一下「使用此動作的資料類型清單方塊旁邊的「增加」，以顯示「增加資料類型」對話框。



圖 11-4 建立「動作的增加資料類型」對話框

6. 選擇性的：如果您不想使用預設資料類型名稱，將資料類型的新名稱鍵入資料類型種類的名稱文字欄位中。

名稱不得包含空格。應用程式使用者是看不到資料類型名稱的；它是用於動作/資料類型資料庫以識別資料類型定義。

7. 按一下「識別特性」方塊旁邊的「編輯」按鈕，以顯示「識別特性」對話框。

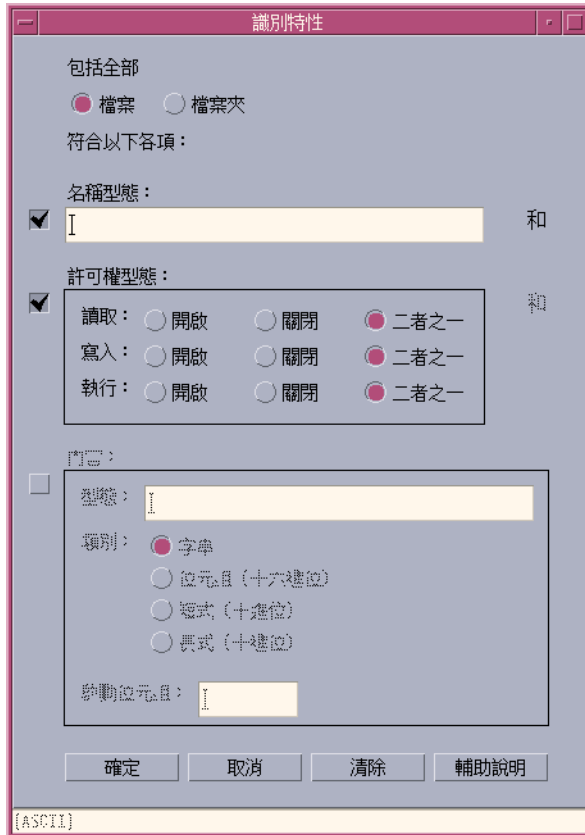


圖 11-5 建立動作的識別特性對話框

資料類型的特性是用於分辨此資料類型與其它資料類型的條件。您可以選擇下列條件中的一項或以上：

檔案或檔案夾—資料類型只應用於檔案或檔案夾

名稱型樣—以檔案名稱為基礎的資料類型

許可權型樣—讀取、寫入、執行許可權

內容—檔案指定部分的內容

8. 選擇資料類型是代表檔案或檔案夾。

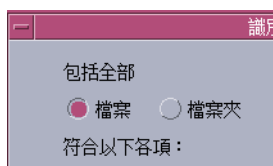


圖 11-6 指定資料類型的檔案或目錄特性。

9. 如果資料類型取決於名稱，請選擇名稱型樣核對方塊並填入文字欄位。

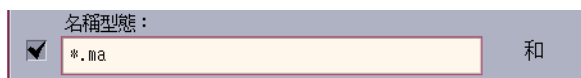


圖 11-7 指定資料類型的檔案名稱特性

您可以使用 * 與 ? 作為通配符號：

*—符合任何順序的字元

?—符合任何單個字元

10. 如果資料類型取決於許可權，選擇許可權型樣核取方塊並選擇資料類型的許可權。

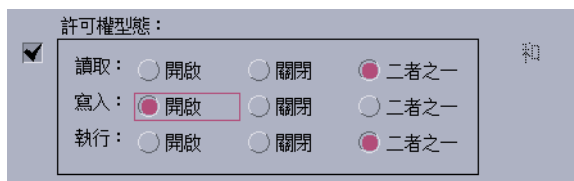


圖 11-8 指定資料類型的許可權特性

開—檔案必須有指定的許可權

關—檔案必須缺乏指定的許可權

兩者任一—指定的許可權沒有關係

11. 如果資料類型取決於內容，請選擇內容核對方塊並供應要求的資訊—搜尋的型樣與內容類型。您可以選擇性地供應開始搜尋的位元組位置。

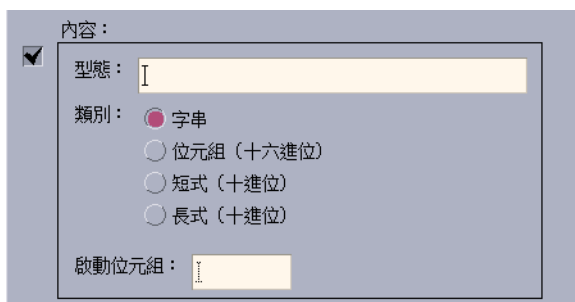


圖 11-9 指定資料類型的內容特性

註解 – 使用以內容為基礎的資料類型可能會影響到系統的效能。

12. 按一下「確定」以關閉「識別特性對話框」。
特性將會以此編碼顯示在識別特性欄位中：
d—目錄
r—檔案有讀取許可權
w—檔案有寫入許可權
x—檔案有執行許可權
!—邏輯運算子否
&—邏輯運算子及
13. 將資料檔案的輔助說明文字鍵入輔助說明文字的文字欄位中。
14. 使用資料類型圖示控制來指定資料檔案的圖示。剛開始時，會顯示預設圖示。
 - 如果要選擇不同的現存圖示，請按一下找出組別以顯示找出組別對話框。請參閱 第 166 頁“使用尋找集對話方塊來指定一個圖示”。
 - 如果要建立新圖示，請按一下編輯圖示以執行圖示編輯程式。
15. 驗證「開啓此資料類型的指令」文字欄位中的指令。這是使用者連按兩下資料檔案時會執行的指令。
16. 選擇性的：如果應用程式供應從指令行列印資料檔案的列印指令，請將指令鍵入「列印此資料類型的指令」文字欄位中，使用檔案引數的語法 $\$n$ 。
17. 執行下列之一以儲存資料類型定義：
 - 按一下「確定」以儲存資料類型並關閉「增加資料類型」對話框。
 - 按一下「應用」以儲存資料類型，而不需關閉「增加資料類型」對話框。這樣做可以讓您立刻進行定義動作的其它資料類型。

使用尋找集對話方塊來指定一個圖示

當您按一下「建立動作」主視窗或「增加資料類型」視窗中的「找出組別」時，「找出組別」對話框就會顯示出來。使用此對話框來指定將會用於動作或資料類型的圖示。

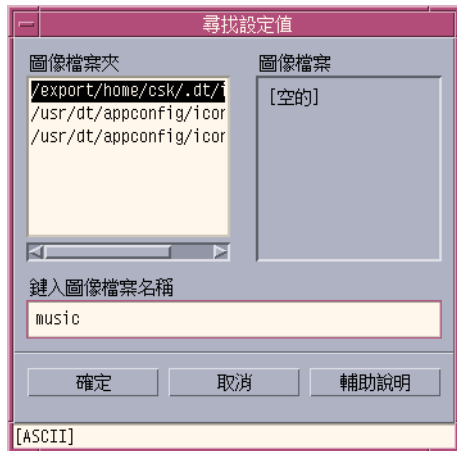


圖 11-10 尋找集對話方塊

「找出組別」對話框可以讓您指定一組圖示影像檔案，其位於：

- 圖示搜尋路徑上的目錄中。「圖示檔案夾」清單包括圖示搜尋路徑上的所有目錄。
- 將使用 dtappintegrate 與桌上管理系統整合的註冊套裝軟體中。這些圖示尚未位於圖示搜尋路徑上的目錄中，但將會由 dtappintegrate 置於該處。

註解 – 使用建立動作寫好圖示檔案的基本名稱（檔案名稱減去檔案名稱大小與類型的字尾）所建立的動作與資料類型定義。以建立動作來建立的動作與資料類型圖示最後必須放在圖示搜尋路徑上的目錄中。

▼ 如果要指定一組位於圖示搜尋路徑上的圖示

1. 在「找出組別」對話框的「圖示檔案夾」清單中，連按兩下包含圖示的檔案夾路徑。
「圖示檔案」清單將會顯示該檔案夾中的所有圖示檔案。
2. 在「圖示檔案」清單中，按一下您想要使用的圖示。
這樣會將圖示檔案的基本名稱放在「輸入圖示檔案」名稱文字欄位中。
3. 按一下「確定」。

▼ 如果要指定註冊套裝軟體中的圖示

如果您是系統管理者或建立註冊套裝軟體的程式設計者，圖示影像檔案剛開始時是位於註冊套裝軟體中的目錄中：

```
app_root/dt/appconfig/icons/language
```

在以 `dtappintegrate` 註冊之後，圖示檔案將會複製到圖示搜尋路徑上的 `/etc/dt/appconfig/icons/language`。

使用此程序來指定為註冊套裝軟體一部分的圖示：

1. 在「找出組別」對話框的「輸入圖示檔案名稱」文字欄位中，鍵入圖示檔案的基本名稱。
2. 按一下「確定」。
建立動作會顯示對話框以通知您在圖示搜尋路徑上的目錄中找不到圖示。
3. 在出現的資訊對話框中，選擇「無變更」。

第 12 章

人工建立動作

建立動作有兩種方法：

- 使用「建立動作」桌上管理系統應用程式
- 人工建立動作定義

人工建立動作需要您編輯資料庫檔案。本章描述如何人工建立動作定義。

- 第 169 頁“您必須人工建立動作的理由”
- 第 171 頁“人工建立動作：一般步驟”
- 第 178 頁“建立 COMMAND 動作的執行字串”
- 第 184 頁“COMMAND 動作的視窗支援與終端機模擬器”
- 第 185 頁“限制動作以確定引數”
- 第 187 頁“建立在遠端系統上執行應用程式的動作”
- 第 188 頁“使用動作與資料類型定義中的變數”
- 第 190 頁“從指令行呼叫動作”
- 第 191 頁“建立本土化動作”
- 第 192 頁“建立 ToolTalk 應用程式的動作”
- 有關動作的簡介，請參閱 第 10 章。
- 有關使用「建立動作」的資訊，請參閱 第 11 章。
- 有關動作定義的參考資訊，請參閱 dtactionfile(4) 線上援助頁。

您必須人工建立動作的理由

有三種基本類型的動作：

- COMMAND
- MAP
- TT_MSG

建立動作工具是設計來建立特定類型的 COMMAND 與 MAP 動作。所有的 TT_MSG 動作都必須以人工建立。

有關更多的資訊，請參閱 第 158 頁“建立動作的限制”。

COMMAND 動作

指令動作執行啟動應用程式的指令、執行 shell 程序檔或執行作業系統指令。動作的定義包括要執行的指令 (EXEC_STRING)。

建立「動作工具」工具可以用於建立最常見的指令動作類型。然而，可能會有一些您必須以人工建立動作的情況；但是，如果動作指定如下，您必須人工建立 指令動作：

- 每個引數都有不同提示符號的多重檔案引數。
- 動作呼叫—動作呼叫其它動作的能力。
- 引數計數的相依行為—為不同數目的檔案引數建立有非常不同行為的動作能力。
- 遠端執行主機—在系統上而非包含動作定義者執行應用程式的能力。
- 使用者的變更—以不同的使用者執行動作的能力（例如，以 root 密碼提示然後以 root 執行）。

MAP 動作

對映動作是“對映”至其它動作而非直接指定指令或 ToolTalk 訊息的動作。

對映提供指定動作的替代名稱的能力。例如，名為 IconEditor 的指令動作啟動圖示編輯程式。資料庫也包括定義中限制為點陣圖與像素對應檔案（由 ARG_TYPE 欄位）的開啓動作，此動作對映至 IconEditor 動作。此讓使用者藉由選擇「檔案管理者」中的點陣圖或像素對應，然後從「所選項目」功能表選擇「開啓」來啟動「圖示編輯程式」。

「建立動作」為「開啓」與「列印」動作提供了有限的對映。其它所有的對映動作都必須以人工建立。

TT_MSG (ToolTalk 訊息) 動作

TT_MSG 動作可發送 ToolTalk 訊息。所有的 TT_MSG 動作都必須以人工建立。

人工建立動作：一般步驟

本節解釋如何建立動作定義的配置檔案。

動作的配置檔案

包含動作定義的配置檔案必須符合這些需求：

- 檔案必須使用命名慣例 *name .dt*
- 檔案必須位於資料庫（動作與資料類型）搜尋路徑上。預設搜尋路徑為：
 - 個人的動作—*HomeDirectory /.dt/types*
 - 整個系統的動作—*/etc/dt/appconfig/types/language*
 - 內建的動作—*/usr/dt/appconfig/types/language*。您不應使用此目錄。

有關修改動作/資料類型搜尋路徑的資訊，請參閱 第 136 頁“設定搜尋路徑的值”。

▼ 人工建立動作

1. 開啓現存資料庫檔案或建立新檔案。
請參閱前一節，第 171 頁“動作的配置檔案”。

2. 使用此語法建立動作定義：

```
ACTION action_name
{
    TYPE          action_type
    action_field
    ...
}
```

其中：

action_name—用於執行動作的名稱。

action_type—指令（預設）、對映或 TT_MSG。

action_field—此類型動作的所需或選擇性的欄位之一。所有欄位都是由一個關鍵字與一個值所組成的。

本章中涵蓋許多的動作欄位。有關更多的資訊，請參閱 `dtactionfile(4)` 線上援助頁。

3. 儲存檔案。
4. 如果您想要動作圖示有個唯一的影像，請為此動作建立圖示。圖示的預設位置為：
 - 個人圖示：*HomeDirectory /.dt/icons*

- 整個系統的圖示：`/etc/dt/appconfig/icons/language`。預設 語言 為 C。
有關更多的資訊，請參閱 第 175 頁“指定由動作使用的圖示影像”。

5. 連按兩下桌面工具應用程式群組中的重新載入動作。
6. 建立動作的動作檔案。動作檔案在「檔案管理者」或代表動作的「應用程式管理者」中建立圖示。（如果動作的寫入目的是為了啟動應用程式，圖示就稱為 應用程式圖示。）如果要建立動作檔案，請建立與 `action_name` 有相同名稱的可執行檔案。您可以將檔案放在任何您有寫入許可權的目錄中。您可以任意地建立動作檔案的數目。

建立 COMMAND 動作的範例

下列步驟建立啟動在遠端系統 `AppServerA` 上的傳真應用程式的個人動作。啟動傳真應用程式的指令為：

```
/usr/fax/bin/faxcompose [檔案名稱]
```

1. 建立包含指令的檔案 `HomeDirectory /.dt/types/Fax.dt`。
2. 將以下動作定義放在檔案中：

```
ACTION FaxComposer
{
    TYPE                COMMAND
    ICON                fax
    WINDOW_TYPE        NO_STDIO
    EXEC_STRING        /usr/fax/bin/faxcompose -c %Arg_1%
    EXEC_HOST          AppServerA
    DESCRIPTION        Runs the fax composer
}
```

`WINDOW_TYPE` 與 `EXEC_STRING` 欄位描述動作的行為。

`WINDOW_TYPE`—`NO_STDIO` 關鍵字指定動作不需在終端機模擬器視窗中執行。請參閱 第 184 頁“指定動作的視窗支援”。

`EXEC_STRING`—語法 `%Arg_1%` 接受定位檔案。如果連按兩下動作圖示，動作會開啓空白的傳真撰寫程式視窗。

請參閱 第 178 頁“建立 COMMAND 動作的執行字串”。

3. 儲存檔案。
4. 使用「圖示編輯程式」在 `HomeDirectory /.dt/icons` 目錄中建立下列的圖示影像檔案：
 - `fax.m.pm`，大小 32 乘 32 像素
 - `fax.t.pm`，大小 16 乘 16 像素
5. 連按兩下桌面工具應用程式群組中的重新載入動作。
6. 建立您有寫入許可權的目錄（例如，您的主目錄）中名為 `FaxComposer` 的可執行的檔案。

建立 MAP 動作的範例

假定您傳真的大部分檔案都是以「文字編輯程式」所建立的而且是資料類型 TEXTFILE（名為 *.txt 的檔案）。

這些步驟將“傳真”功能表項目增加至資料類型的「所選項目」功能表。

1. 開啓之前範例中建立的檔案 *HomeDirectory* / .dt/types/Fax.dt。
2. 增加此對映動作定義至檔案：

```
ACTION Fax
{
    ARG_TYPE          TEXTFILE
    TYPE              MAP
    MAP_ACTION        FaxComposer
}
```

3. 儲存檔案。
4. 從 /usr/dt/appconfig/types/ 取得優先順序的個人語言相依的配置檔案 /dtpad.dt 複製 TEXTFILE 的資料屬性定義至新檔案 *HomeDirectory* / .dt/types/textfile.dt。將傳真動作增加至 ACTIONS 欄位。

```
DATA_ATTRIBUTES TEXTFILE
{
    ACTIONS          Open,Print,Fax
    ICON              Dtpenpd
    ...
}
```

5. 儲存檔案。
6. 開啓「應用程式管理者」並連按兩下桌面工具應用程式群組中的「重新載入動作」。

▼ 重新載入動作/資料類型資料庫

為了使新的或編輯的動作定義生效，桌上管理系統必須重新讀取資料庫。

- 開啓桌面工具應用程式群組並連按兩下「重新載入動作」。
- 或，執行指令：

```
dtaction ReloadActions
```

ReloadActions 是其圖示標為“重新載入動作”的動作的名稱。當使用者如下執行時，也會重新讀取動作資料庫：

- 登入
- 重新啓動工作區管理者
- 藉由從檔案功能表選擇儲存以在建立動作視窗中儲存動作

建立動作的動作檔案 (圖示)

動作檔案 是建立以提供「檔案管理者」或「應用程式管理者」中的動作視覺表示方式的檔案。



圖 12-1 應用程式管理者中的動作檔案 (動作圖示)

既然動作檔案的圖示代表動作，有時它也稱為 **動作圖示**。如果隱藏的動作啟動應用程式，動作檔案圖示就稱為 **應用程式圖示**。

連按兩下動作圖示會執行動作。動作圖示也可以是定位區域。

▼ 如果要建立動作檔案 (動作圖示)

- 建立與動作名稱有相同名稱的可執行檔案。檔案的內容不會影響。

例如，如果動作定義為：

```
ACTION MyFavoriteApp
{
    EXEC_STRING           Mfa -file %Arg_1%
    DESCRIPTION           Runs MyFavoriteApp
    ICON                  Mfapp
}
```

那麼動作檔案將會是名為 MyFavoriteApp 的可執行檔案。在「檔案管理者」與「應用程式管理者」中，MyFavoriteApp 檔案會使用圖示影像 Mfapp.size.type。連按兩下 MyFavoriteApp 的圖示將會執行動作的執行字串，而圖示的「關於項目」輔助說明將會是說明欄位的內容 (“runs MyFavoriteApp”)。

動作標籤

如果動作定義包括 標籤 欄位，將會將動作檔案標示在「檔案管理者」與具有此欄位內容的「應用程式管理者」而非檔案名稱 (*action_name*) 中。例如，如果動作定義包括：

```
動作 MyFavoriteApp
{
    LABEL      Favorite Application
    ...
}
```

那麼動作圖示將會標示為“最喜愛的應用程式”。

指定由動作使用的圖示影像

使用 `ICON` 欄位來指定為動作所建立的動作圖示的「檔案管理者」與「應用程式管理者」所使用的圖示。

如果您未指定圖示，系統則使用預設動作圖示影像檔案 `/usr/dt/appconfig/icons/language/Dtactn.*`。

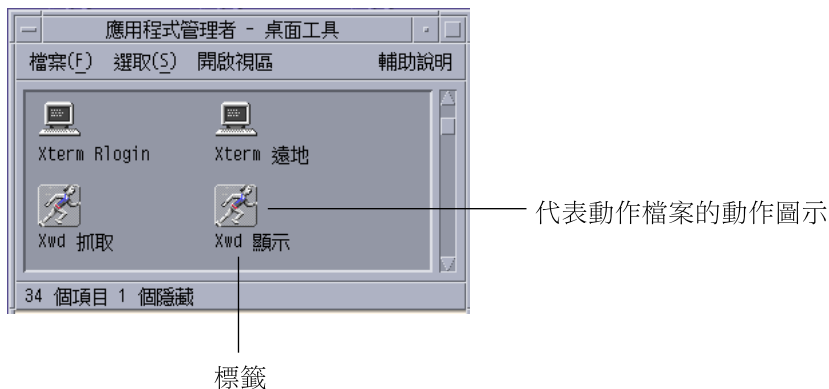


圖 12-2 預設動作圖示影像

預設動作圖示可以使用資源來變更：

```
*actionIcon:    icon_file_name
```

其中 *icon_file_name* 可以是基本名稱或絕對路徑。

`ICON` 欄位的值可以是：

- 基本檔案名稱

基本檔案名稱是包含圖示影像減去檔案名稱的大小 (m 與 t) 與影像類型 (bm and pm) 字尾的檔案名稱。例如，如果檔案名稱為 `GameIcon.m.pm` 與 `GameIcon.t.pm`，請使用 `GameIcon`。

如果您使用基本檔案名稱，圖示檔案必須放在圖示搜尋路徑上的目錄中：

- 個人圖示：*HomeDirectory* / *.dt/icons*
 - 整個系統的圖示：*/etc/dt/appconfig/icons/language*
 - 圖示檔案的絕對路徑，包括完整的檔案名稱
唯有當圖示檔案不在圖示搜尋路徑上時才應使用絕對路徑。例如，如果圖示檔案 *GameIcon.m.pm* 是放在不在圖示搜尋路徑上的目錄 */doc/projects* 中，*ICON* 欄位的值會是 */doc/projects/GameIcon.m.pm*。
- 表 12-1 列出您應該建立的圖示大小與相等的檔案名稱。

表 12-1 動作圖示的圖示名稱與大小

像素的大小	點陣圖名稱	像素對應名稱
48 乘 48	<i>name.l.bm</i>	<i>name.l.pm</i>
32 乘 32	<i>name.m.bm</i>	<i>name.m.pm</i>
16 乘 16	<i>name.t.bm</i>	<i>name.t.pm</i>

▼ 修改現存動作定義

您可以修改您系統上任何可供使用的動作，包括內建動作。

註解 – 當修改內建動作資料庫時，請小心使用。內建動作是設計來與桌上管理系統應用程式共同良好運作的。

1. 尋找您想要修改的動作定義。

動作定義的預設位置為：

- 內建動作：*/usr/dt/appconfig/types/language*
- 整個系統的動作：*/etc/dt/appconfig/types/language*
- 個人動作：*HomeDirectory* / *.dt/types*

您的系統可能會包括其它的位置。如果要看您的系統為動作所使用的位置清單，請鍵入指令：

```
dtsearchpath -v您的系統使用列在 DTDATABASESEARCHPATH 之下的目錄。
```

2. 如果有必要，請將動作定義的文字複製至這些目錄之一中的新的或現存檔案：

- 整個系統的動作：*/etc/dt/appconfig/types/language*
- 個人動作：*HomeDirectory* / *.dt/types*

既然您不應在 */usr/dt/appconfig/types/language* 目錄中編輯檔案，那麼您必須複製內建動作。

3. 編輯動作定義。
4. 當您完成編輯時，請儲存檔案。
5. 連按兩下桌面工具應用程式群組中的重新載入動作。

動作定義中的優先順序

當使用者呼叫動作時，系統會搜尋符合動作名稱的資料庫。當一個以上的動作以該名稱存在時，系統會使用優先順序規則來決定要使用何者。

- 如果未應用其它的優先順序規則，優先順序以定義的位置為基礎。下表的優先順序是由高到低的順序：
 - 個人動作 (*HomeDirectory* / *.dt/types*)
 - 整個系統的本機動作 (*/etc/dt/appconfig/types/language*)
 - 整個系統的遠端動作 (**主機名稱**:*/etc/dt/appconfig/types/language*)。搜尋的遠端主機就是列在應用程式搜尋路徑中的那一些。
 - 內建動作 (*/usr/dt/appconfig/types/language*)
- 在已知的目錄中，*.dt 檔案是以字母順序讀取的。
- 由 ARG_CLASS、ARG_TYPE、ARG_MODE 或 ARG_COUNT 所限制的動作對於未受限制的動作有優先順序。(此四個欄位的預設值為 *。)

其中應用一項以上的限制，由高到低的優先順序為：

- ARG_CLASS
- ARG_TYPE
- ARG_MODE
- ARG_COUNT

其中有一項以上的受限制 ARG_COUNT 存在，由高到低的優先順序為：

- 特定的整數值 *n*
- <*n*
- >*n*
- *

例如，試想下列部分的動作定義：

```
ACTION EditGraphics
# EditGraphics-1
{
    ARG_TYPE          XWD
    ...
}
```

```
ACTION EditGraphics
# EditGraphics-2
```

```

    {
        ARG_COUNT          0
        ...
    }

ACTION EditGraphics
# EditGraphics-3
{
    ARG_TYPE              *
    ...
}

```

連按兩下 EditGraphics 動作圖示會啓動 EditGraphics-2，因為未提供引數，而且 ARG_COUNT 0 有優先順序。當 XWD 類型的檔案引數被提供時，會使用 EditGraphics-1，因為其指定 XWD ARG_TYPE。所有其它檔案引數都使用 EditGraphics-3。

建立 COMMAND 動作的執行字串

指令 動作的最小需求為兩個欄位 — 動作 與 EXEC_STRING。

```

動作 action_name
{
    EXEC_STRING execution_string
}

```

執行字串是 指令 動作定義最重要的部分。它會使用與您將會在終端機視窗中所執行的指令行類似的語法，但還包括處理檔案與字串引數的其它語法。

執行字串的一般功能

執行字串可包括：

- 檔案與非檔案引數
- Shell 語法
- 絕對路徑或可執行的名稱

動作引數

引數是指令或應用程式要正確執行所需要的資訊。例如，試想您可以用以開啓「文字編輯程式」中的檔案之指令行：

```

dtpad 檔案名稱

```

在此指令中，**檔案名稱**是 dtpad 指令的檔案引數。

就像應用程式與指令，動作可以有引數。指令動作可以使用的資料有兩種類型：

- 檔案
- 字串資料

在執行字串中使用 Shell

執行字串是直接執行的，而不是經由 shell。但是，您可以在執行字串中明確地呼叫 shell。

例如：

```
EXEC_STRING      \  
    /bin/sh -c \  
    'tar -tvf% (File) Arg_1% 2>&1 | \${PAGER:-more};\  
    echo "\n*** 從視窗功能表選擇關閉以關閉 ***"'
```

可執行的名稱或絕對路徑

如果您的應用程式是位於路徑變數中列出的目錄中，您可以使用簡單的可執行名稱。如果應用程式是在它處，您必須使用絕對路徑於可執行的檔案。

建立不使用引數的動作

使用您將會用以從指令行啟動應用程式的 EXEC_STRING 的相同語法。

範例

- 此執行字串是啟動 X 用戶端 xcutsel 的部分動作。

```
EXEC_STRING xcutsel
```

- 此執行字串啟動具有數位時鐘的用戶端 xclock。指令行包括指令行的選項，但不需要引數。

```
EXEC_STRING xclock -digital
```

建立接受定位檔案的動作

為檔案引數使用此語法：

```
%Arg_n%
```

或

`%(File) Arg_n%`

(檔案) 是選擇性的，因為提供給 `Arg_n` 的引數被假設為 (依據預設值) 檔案。(請參閱第 181 頁“解譯檔案引數為字串”如何使用 `%(String) Arg_n%` 語法的資訊。)

此語法讓使用者在動作圖示上定位資料檔案物件，以啟動具有該檔案引數的動作。其在指令行中代替 `n`th 引數。檔案可以是本機或遠端檔案。

範例

- 執行字串將使用定位檔案作為 -載入 參數來執行 `wc -w`。

```
EXEC_STRING wc -w%Arg_1%
```

- 此範例顯示只與目錄引數運作的動作的部分定義。當目錄定位於動作圖示上時，動作在具有 讀取及寫入的許可權的目錄中顯示所有檔案的清單。

```
ACTION List_Writable_Files
{
    ARG_TYPE      FOLDER
    EXEC_STRING /bin/sh -c 's -l %Arg_1% | grep rw-'
    ...
}
```

建立一個提示檔案引數的動作

為檔案引數使用此語法：

```
%(File) "提示符號"%
```

當使用者連按兩下動作圖示時，此語法建立顯示檔案名稱的提示符號的動作。

例如，這個執行字串會顯示一個對話方塊，提示 `wc -w` 指令的檔案引數：

```
EXEC_STRING wc -w%(File) "在檔案中計數字數："%
```

建立接受定位檔案或提示符號的動作

為檔案引數使用此語法：

```
%Arg_n"提示符號"%
```

或

```
%(File) Arg_n"提示符號"%
```

此語法產生如下動作：

- 接受定位檔案為檔案引數。

- 當使用者連按兩下動作圖示時，顯示提示檔案名稱的對話框。
例如，這個執行字串會執行 `lp -oraw`。如果動作是由連按兩下圖示所啟動的，會出現對話框以提示檔案名稱。

```
EXEC_STRING lp -oraw %Arg _1"列印的檔案："%
```

建立一個提示非檔案引數的動作

為非檔案參數使用此語法：

```
%"提示符號"%
```

或

```
%(String) "提示符號"%
```

(String) 是選擇性的，因為按預設解譯引號的文字為字串資料。此語法顯示提示非檔案資料的對話框；當提示檔案名稱時，請勿使用此語法。

例如，此執行字串執行 `xwd` 指令與要增加至每個像素的值的提示符號：

```
EXEC_STRING xwd -add% "增加值："% -out %Arg _1" 檔案名稱："%
```

解譯檔案引數為字串

為引數使用此語法：

```
%(String) Arg_n%
```

例如，此執行字串列印具有包含檔案名稱的標題的檔案，方法是使用指令 `lp -t 標題檔案名稱`。

```
EXEC_STRING lp -t%(String) Arg_1%% (File) Arg_1" 列印的檔案："%
```

在動作中提供 Shell 能力

在執行字串中指定 shell：

```
/bin/sh -c '指令'  
/bin/ksh -c '指令'  
/bin/csh -c 'command'
```

範例

- 此執行字串舉例說明使用 shell 管道的動作。

```
EXEC_STRING /bin/sh -c 'ps | lp'
```

- 此為需要 shell 處理並接受檔案引數的更複雜的執行字串。

```
EXEC_STRING /bin/sh -c 'tbl %Arg_1"線上援助頁："% | troff -man'
```

- 此執行字串需要引數成爲壓縮檔案。動作將解壓縮檔案並使用 `lp -oraw` 來列印它。

```
EXEC_STRING /bin/sh -c 'cat%Arg_1 "列印的檔案："% | \  
uncompress | lp -oraw'
```

- 此執行字串啓動 shell 程序檔。

```
EXEC_STRING /usr/local/bin/StartGnuClient
```

建立多重檔案引數的 COMMAND 動作

動作處理多重檔案引數的方法有三種：

- 動作可以重覆地執行，一次一個引數。當 `EXEC_STRING` 包含單個檔案引數與藉由在動作圖示上定位多重檔案所提供的多重檔案引數時，每個檔案引數的動作是分開執行的。

例如，如果供應多重檔案引數至以下動作定義：

```
ACTION DisplayScreenImage  
{  
    EXEC_STRING xwud -in %Arg_1%  
    ...  
}
```

`DisplayScreenImage` 動作是重覆地執行。

- 動作可以使用兩個或以上的不可交換的檔案引數。例如：

```
xsetroot -cursor cursorfile maskfile
```

需要兩個特別順序的唯一檔案。

- 動作可以按照順序地執行每個檔案引數上的相同指令。例如：

```
pr file [file ...]
```

將會在一份列印工作中列印一份或許多檔案。

建立不可交換的引數的動作

使用下列語法慣例之一：

- 如果您想要動作提示檔案名稱，使用此語法於每個檔案引數：

```
% (File) "提示符號"%
```

每個引數使用不同的 **提示符號** 字串。

例如，此執行字串提示兩個檔案。

```
EXEC_STRING xsetroot -cursor%  
(File) "游標點陣圖："% \ % (檔案) "遮罩點陣圖："%
```

- 如果要接受定位檔案，請使用此語法於每個檔案引數：

`%Arg _n%`

每個引數使用不同的 n 值。例如：

`EXEC_STRING diff%Arg _1%%Arg _2%`

建立具有可交換檔案引數的動作

使用下列語法慣例之一：

- 如果要建立接受定位檔案並以格式 **指令檔案 1 檔案 2 ...**，發出指令的動作，請使用此語法於檔案引數：

`%Args%`

- 如果要建立接受多重定位檔案的動作，或在連按兩下時會顯示單一檔案提示符號的動作，請使用此語法於檔案引數：

`%Arg _1"提示符號"% %Args%`

動作將會以此格式發出指令：**指令檔案 1 檔案 2**

範例

- 此執行字串會建立執行如下的動作：

`pr 檔案 1 檔案 2`

以多重檔案引數。

`EXEC_STRING pr%Args%`

- 此執行字串建立與之前的範例類似的動作，除了此動作會在連按兩下時顯示提示符號（無檔案引數）。

`EXEC_STRING pr%Arg _1"列印的檔案："%%Args%`

建立多重定位檔案的動作

如果要接受多重定位檔案引數並執行指令行格式如下：

指令 檔案 1 檔案 2 ...

使用語法：

`%Args%`

範例

- 此執行字串會為多重檔案執行名為 Checkout 的程序檔：

```
EXEC_STRING      /usr/local/bin/Checkout \  
                  %Arg_1" 結束什麼檔案?"%%Args%
```

- 這個執行字串將對多重檔案執行 `lp -oraw` :

```
EXEC_STRING      lp -oraw%Arg_1"列印的檔案:"%%Args%
```

COMMAND 動作的視窗支援與終端機模擬器

指令 動作在桌上管理系統上支援視窗的方法有數種。

- 如果應用程式有其自己的視窗，便可以寫入動作以不提供其它的視窗支援。當動作執行不需要直接的使用者輸入且無輸出的指令時，也使用此選項。
- 如果應用程式必須在終端機模擬器視窗中執行，便可以寫入動作以開啓視窗，然後執行應用程式。有數種終端機選項。

指定動作的視窗支援

使用 `WINDOW_TYPE` 欄位指定如表 12-2 中所示的動作需要的視窗支援類型。

表 12-2 `WINDOW_TYPE` 欄位及視窗支援提供的

<code>WINDOW_TYPE</code>	視窗支援提供的
<code>NO_STDIO</code>	無。如果應用程式有其自己的視窗，或是如果指令沒有可見的輸出，使用 <code>NO_STDIO</code> 。
<code>PERM_TERMINAL</code>	永久終端機模擬器視窗。動作開啓終端機視窗，此視窗會保持開啓直到使用者確實地關閉它。使用者可以將資料輸入視窗中。使用獲得一些輸入、產生一些輸出、然後終止指令 (例如， <code>ls directory</code>)。
<code>TERMINAL</code>	暫存終端機模擬器視窗。動作會開啓指令一完成就關閉的終端機視窗。使用全螢幕的指令 (例如， <code>vi</code>)。

指定終端機模擬器的指令行選項

使用動作定義中的 `TERM_OPTS` 欄位來指定終端機模擬器的指令行選項。

例如，下列執行主機的動作提示符號：

```
ACTION OpenTermOnSystemUserChooses  
{  
    WINDOW_TYPE          PERM_TERMINAL
```



```

EXEC_HOST           %(String)"Remote terminal on: "%
TERM_OPTS           -title %(String)"Window title: "%
EXEC_STRING         $SHELL
}

```

指定不同的預設終端機模擬器

由動作所使用的預設終端機模擬器是 `dtterm`。您可以將此變更為其它的終端機模擬器。預設終端機模擬器是用於當動作未明確地指定使用的終端機模擬器時。

由動作所使用的終端機模擬器必須有這些指令行選項：

- `-標題 window_title`
- `-e 指令`

兩項資源決定預設終端機模擬器由動作所使用：

- `localTerminal` 資源指定由本機應用程式所使用的終端機模擬器。

```
*localTerminal:      終端機
```

例如：

```
*localTerminal:      xterm
```

- `remoteTerminal` 資源指定由遠端應用程式所使用的終端機模擬器。

```
*remoteTerminal:     host:終端機 [,host:終端機...]
```

例如：

```
*remoteTerminal:     sysibm1:/usr/bin/xterm,sysph2:/usr/bin/yterm
```

限制動作以確定引數

限制動作為特殊類型的引數使動作更精確。例如，您應該限制 PostScript 檔案的查看器呼叫為只有 PostScript 檔案引數的動作；有了限制，如果指定非 PostScript 的檔案，動作將會回覆錯誤對話。

您可以限制動作，基於：

- 檔案引數的資料類型。
- 檔案引數的數目—例如，無引數對一個或多個引數。這樣會提供動作圖示的不同定位與連接兩下的行為。
- 引數的讀取/寫入模式。

為指定的資料類型限制動作

使用 ARG_TYPE 欄位來指定其動作為有效的資料類型。使用資料屬性名稱。

您可以輸入資料類型的清單，以逗號分開登錄。

例如，以下動作定義假定 GIF 資料類型已建立。

```
ACTION Open_Gif
{
    TYPE                COMMAND
    LABEL                "Display Gif"
    WINDOW_TYPE         NO_STDIO
    ARG_TYPE             Gif
    ICON                 xgif
    DESCRIPTION         Displays gif files
    EXEC_STRING          xgif
}
```

以引數數目為基礎限制動作

使用 ARG_COUNT 欄位以指定動作可以接受的引數數目。有效值為：

* (預設值) —任何數目的引數。其他值對於 * 有優先順序。

n —任何非負數的整數，包括 0。

$>n$ —多於 n 個引數。

$<n$ —少於 n 個引數。

ARG_COUNT 的用途之一就是提供不同的動作圖示行為，視使用者是否連按兩下圖示或在其上定位檔案而定。請參閱下節，“如果要提供不同的連按兩下與定位行為。”

▼ 提供不同的連按兩下與定位行為

使用此程序以建立接受定位檔案但當連按兩下動作圖示時不提示檔案的動作。

1. 建立連按兩下功能的動作定義。

使用 ARG_COUNT 欄位以指定 0 引數。使用語法於不接受定位引數的 EXEC_STRING。

2. 建立定位功能的第二種動作定義。

使用 ARG_COUNT 欄位以指定 >0 引數。使用語法於接受定位檔案的 EXEC_STRING。

例如，假定下列兩個指令行可以用於啟動名為 vedit 的編輯程式：

- 如果要啟動沒有檔案引數的編輯程式：

```
vedit
```

- 如果要啓動具有開啓爲唯讀文件的檔案引數的編輯程式：

```
vedit -R 檔案名稱
```

下列兩個動作建立名爲 Vedit 動作的定位與連按兩下功能。既然 ARG_COUNT 0 比定位功能定義的暗示 ARG_COUNT * 更特定，第一個動作在資料庫搜尋配對時有優先順序。

```
# Double-click functionality
ACTION Vedit
{
    TYPE                                COMMAND
    ARG_COUNT                            0
    WINDOW_TYPE                          PERM_TERMINAL
    EXEC_STRING                           vedit
}

# Drop functionality
ACTION Vedit
{
    TYPE                                COMMAND
    WINDOW_TYPE                          PERM_TERMINAL
    EXEC_STRING                           vedit -R %Arg_1%
}
```

以引數模式爲基礎限制動作

使用 ARG_MODE 欄位以指定引數的讀取/寫入模式。有效值爲：

* (預設值)—任何模式

!w—不可寫入

w—可寫入

建立在遠端系統上執行應用程式的動作

討論到動作與遠端執行時，有兩個經常使用的名詞：

資料庫主機—包含動作定義的系統

執行主機—執行可執行檔案的系統

在大部分的情況中，動作與其應用程式是位於相同的系統上；既然動作的預設執行主機是資料庫主機，就不需要特殊的語法了。

然而，當執行主機與資料庫主機不同時，動作定義必須指定執行字串應執行之處。

在不同的系統上尋找動作與應用程式的能力是桌上管理系統的用戶端/伺服器架構的一部分。有關網路應用程式更深入的討論，請參閱第 122 頁“管理應用程式服務”。

建立執行遠端系統應用程式的動作

使用動作定義中的 EXEC_HOST 欄位以指定應用程式的位置。

EXEC_HOST 的有效值為：

%DatabaseHost%—定義動作的主機。

%LocalHost%—呼叫動作的主機(*the session server*)。

%DisplayHost%—正在執行 X 伺服器的主機 (不允許 X 終端機)。

%SessionHost%—正在執行控制登入管理器的主機。

主機名稱—命名的主機。使用其中應永遠在一特定主機上呼叫動作的此環境值。

%**提示符號**%—每次呼叫動作時，提示使用者主機名稱。

預設值為 %DatabaseHost%、%LocalHost%。EXEC_HOST 欄位時，動作會先嘗試在包含動作定義的主機上執行指令。如果失敗，動作會嘗試在階段作業伺服器上執行指令。

範例

- 此欄位指定主機 ddsyd：

```
EXEC_HOST ddsyd
```

- 此欄位提示主機名稱：

```
EXEC_HOST%  "包含應用程式的主機："%
```

- 此欄位指定動作將會嘗試執行包含動作定義的主機上的應用程式。如果失敗，動作會嘗試執行主機 ddsyd 上的應用程式。

```
EXEC_HOST%DatabaseHost% , ddsyd
```

使用動作與資料類型定義中的變數

您可以將字串變數與環境變數納入動作與資料類型定義檔案中。

使用動作中的字串變數

字串變數定義從定義的位置到檔案的結尾都一直維持有效。沒有整體的資料庫字串變數。

如果字串變數與環境變數有相同的名稱，字串變數有優先順序。

▼ 如果要定義字串變數

- 使用語法：

```
設定 variable_name=value
```

變數名稱可以包含任何文字字元與底線（_）。每個變數定義都必須在不同的行上。

例如：

```
設定 Remote_Application_Server=sysapp  
設定 Remote_File_Server=sysdata
```

▼ 如果要參考字串變數

- 使用語法：

```
${[variable_name]}
```

例如：

```
EXEC-HOST    $Remote_Application_Server  
CWD          /net/${Remote_File_Server}/doc/project
```

使用動作與資料類型中的環境變數

- 使用語法參考環境變數：

```
${[變數所指定的位置]}。
```

當載入資料庫時，變數會擴展（由其值取代）。如果字串變數與環境變數有相同的名稱，字串變數有優先順序。

例如，此執行字串會列印具有包含登入名稱的標題檔案。

```
EXEC-STRING lp -t$LOGNAME% (檔案) Arg_1%
```

從指令行呼叫動作

桌上管理系統提供從指令行執行動作的 `dtaction` 指令。您可以使用 `dtaction`，執行來自下列的動作：

- 程序檔
- 其它動作
- 終端機模擬器指令行

語法 dtaction

```
dtaction [-user user_name] [-execHost 主機名稱] \  
  action_name [argument ...]
```

`-user user_name`—提供以不同使用者來執行動作的能力。如果 `dtaction` 是由 *user_name* 以外的使用者所呼叫，會有密碼的提示符號顯示出來。

`-execHost 主機名稱`—僅適用於 `COMMAND` 動作而已；指定將在其上執行指令的主機。

argument—動作的引數；通常是檔案引數。

`dtaction` 用戶端有其它的指令行選項。有關更多的資訊，請參閱 `dtaction(1)` 線上援助頁。

建立執行其它動作的動作

使用動作的 `EXEC_STRING` 中的 `dtaction`。

例如，以下動作使用名為 `Spell`（拼字）（在「應用程式管理者」中標為“檢查拼字”的動作）的內建動作。新動作會執行「文字編輯程式」與 `Spell`（拼字）動作，在不同的終端機模擬器視窗中顯示拼字錯誤。

```
ACTION EditAndSpell  
{  
    WINDOW_TYPE          NO_STDIO  
    EXEC_STRING          /bin/sh -c 'dtaction Spell \  
                        %Arg_1"File:"%; dtpad %Arg_1'  
}
```

建立以不同使用者執行的動作

使用以下 `EXEC_STRING` 中的語法：

```
EXEC_STRING  dtaction -user user_name action_name [file_argument]
```

新的使用者 (*user_name*) 必須經由下列機制之一，以到系統的顯示器存取權：

- 登入使用者的 `.Xauthority` 檔案上的讀取許可權
- 或，`xhost` 許可權

例如，下列兩個動作提供成爲 `root` 與編輯 `app-defaults` 檔案的能力。

```
ACTION AppDefaults
{
    WINDOW_TYPE          NO_STDIO
    EXEC_STRING           /usr/dt/bin/dtaction -user root \
                          EditAppDefaults %Arg_1"File:"%
}
ACTION EditAppDefaults
{
    WINDOW_TYPE          TERMINAL
    EXEC_STRING           /bin/sh -c 'chmod +w %Arg_1%; \
                          vi %Arg_1%; chmod -w %Arg_1%'
}
```

建立本土化動作

資料類型的搜尋路徑包括語言相依的位置。桌上管理系統使用 `LANG` 的值來決定搜尋資料類型的位置。

本土化動作的位置

本土化的動作定義必須放在沿著動作搜尋路徑的正確語言相依目錄中。

預設搜尋路徑爲：

- 個人動作：`HomeDirectory/.dt/types`
- 整個系統的動作：`/etc/dt/appconfig/types/language`
- 內建動作：`/usr/dt/appconfig/types/language`

▼ 將現存動作本土化

1. 在合適的語言相依目錄中建立檔案（例如，在 `/etc/dt/appconfig/types/japanese` 中）。
2. 將動作定義複製到語言相依的配置檔案。
例如，您可能會複製動作定義自

```
app_root/dt/appconfig/types/C/file.dt
```

至

```
app_root /dt/appconfig/types/japanese/newfile.dt
```

3. 增加 LABEL (標籤) 欄位或修改現存的 LABEL 欄位。

LABEL 字串

「應用程式管理者」與「檔案管理者」使用標籤字串來識別動作的圖示。

4. 在動作定義中本土化全部的下列欄位：

- 對於本土化圖示：ICON
- 對於本土化的關於項目輔助說明：DESCRIPTION
- 對於本土化的提示符號：任何 EXEC_STRING 中被參照的文字

建立 ToolTalk 應用程式的動作

註解 – 以下資訊只應用於支援 ToolTalk 訊息的應用程式。

使用動作類型 TT_MSG 以建立發出 ToolTalk 訊息的動作。

```
動作 action_name
$
{ $
    TYPE                      TT_MSG $
    ... $
}
```

定址 與 配置 欄位

- ToolTalk 定址 欄位永遠是設定為 TT_PROCEDURE。
- ToolTalk 配置 欄位預設為靜態訊息型樣中的規格。

不支援的訊息

下列為 TT_MSG 類型動作所不支援的：

- ToolTalk 物件導向的訊息
- 訊息中的內容引數

TT_MSG 動作的關鍵字

表 12-3 顯示 TT_MSG 動作的關鍵字與用法。

表 12-3 TT_MSG 動作關鍵字及用途

關鍵字	用途
TT_CLASS	定義 ToolTalk 類別訊息欄位的值
TT_SCOPE	定義 ToolTalk 範圍訊息欄位的值
TT_OPERATION	定義 ToolTalk 作業訊息欄位的值
TT_FILE	定義 ToolTalk 檔案訊息欄位的值
TT_ARGn_MODE	定義第 n 個訊息引數的 ToolTalk 模式屬性的值
TT_ARGn_VTYPE	定義第 n 個訊息引數的 ToolTalk vtype 屬性的值
TT_ARGn_VALUE	定義第 n 個訊息引數的值

第 13 章

人工建立資料類型

建立資料類型定義的方法有兩種：

- 使用「建立動作」工具。使用「建立動作」是涵蓋在 第 11 章。
- 藉由人工建立資料類型定義。

人工建立資料類型需要您編輯資料庫檔案。

本章描述如何人工建立資料類型定義。

- 第 195 頁“您必須人工建立資料類型的理由”
- 第 196 頁“資料類型定義的元件：條件與屬性”
- 第 196 頁“人工建立資料類型：一般步驟”
- 第 198 頁“建立個人動作與資料類型的範例”
- 第 201 頁“定義資料類型的資料條件”
- 有關資料類型的簡介，請參閱 第 10 章。
- 有關資料類型定義的參考資訊，請參閱 the dtddsfile(4) 線上援助頁。

您必須人工建立資料類型的理由

人工建立資料類型可以讓您使用所有建立於資料類型定義語法中的能力。

如果您想要使用這些資料類型的功能，您就必須人工建立資料類型：

- 以位置（路徑）為基礎的資料類型
- 指定與「開啓」及「列印」以外的資料類型有關聯的動作的能力
- 多重名稱、型樣或相同資料類型的內容條件—例如，以名為 *.abc 或 *.def
- 以連結為基礎的資料分類

資料類型定義的元件：條件與屬性

資料類型定義是由兩種不同的資料庫定義所構成的：

- `DATA_ATTRIBUTES` 定義
`DATA_ATTRIBUTES` 定義描述資料類型的名稱以及此類型檔案的外觀與行爲。
- `DATA_ATTRIBUTE` 定義
`DATA_ATTRIBUTES` 定義描述類型條件。每個條件定義會指定該條件所適用的 `DATA_ATTRIBUTES` 定義。

每項 `DATA_ATTRIBUTE` 定義至少要有一項 `DATA_ATTRIBUTES` 定義；`DATA_ATTRIBUTES` 定義可以有與其有關聯的多重 `DATA_CRITERIA`。

例如，您可以建立描述 PostScript 在「檔案管理者」中的外觀與行爲的 PostScript 檔案的屬性定義。然後，您可以為 PostScript 資料類型建立兩項不同的條件——一項以檔案名稱為基礎，而另一項以檔案內容為基礎。

有關更多的資訊，請參閱 第 201 頁“定義資料類型的資料條件”。

人工建立資料類型：一般步驟

本節描述如何建立資料類型配置檔案。

資料類型的配置檔案

包含資料類型定義的配置檔案的要求為：

- 檔案必須使用命名慣例 `name .dt`
- 檔案必須位於資料庫搜尋路徑上。預設搜尋路徑為：
 - 個人資料類型—`language / .dt/types/`
 - 整個系統的資料類型—`/etc/dt/appconfig/types/language`
 - 內建資料類型—`/usr/dt/appconfig/types/language`。您不應使用此目錄。有關修改資料庫搜尋路徑的資訊，請參閱 第 136 頁“設定搜尋路徑的值”。

▼ 建立資料類型定義

1. 開啓現存資料庫檔案或建立新檔案。

有關更多的資訊，請參閱前節 第 196 頁“資料類型的配置檔案”。

2. 使用此語法定義資料類型的資料屬性：

```
DATA_ATTRIBUTES data_type_name
{
    ICON image_name
    DESCRIPTION 字串
    attribute_field
    attribute_field
    ...
}
```

其中：

data_type_name—給予此資料類型的唯一名稱。

image_name—圖示檔案的檔案名稱或路徑。使用基本名稱於此檔案。例如，對於圖示檔案 *myimage.m.pm* 與 *myimage.t.pm*，請使用 *myimage*。

attribute_field—定義資料類型的外觀或行為的欄位。

字串—字元字串。內容將會是資料類型的關於項目輔助說明。

請參閱 第 198 頁“建立個人動作與資料類型的範例”。

3. 使用此語法定義資料類型的資料條件：

```
DATA_CRITERIA criteria_name
{
    DATA_ATTRIBUTES_NAME data_type_name
    criteria_field
    criteria_field
    ...
}
```

其中：

criteria_name—此條件定義的唯一名稱

data_type_name—用於 DATA_ATTRIBUTES 定義中的名稱

criteria_field—用於定義指派檔案至此資料類型的條件的欄位

請參閱 第 201 頁“定義資料類型的資料條件”。

4. 儲存資料庫檔案。

5. 建立資料類型的圖示。

有關更多的資訊，請參閱 第 199 頁“指定用於資料類型的圖示影像”。

6. 如果有需要，請建立列出於屬性定義的 ACTION（動作）欄位中的動作。

7. 連按兩下桌面工具應用程式群組中的「重新載入動作」以重新載入資料庫。

建立個人動作與資料類型的範例

假定您的系統包含名為 `xgif` 的應用程式，此應用程式會顯示 GIF 圖片。一般而言，您藉由執行如下來執行程式：

`xgif` 檔案名稱

您想要能夠以數種方法顯示 GIF 圖片：

- 藉由連按兩下 GIF 資料檔案
- 藉由選擇資料檔案以及從「所選項目」功能表選擇應用程式

1. 開啓新檔案 `language /.dt/types/GifViewer.dt` 以編輯。
2. 鍵入資料類型定義：

```
DATA_ATTRIBUTES Gif
{
    DESCRIPTION          Gif image file.
    ICON                 GifIcon
    ACTIONS              View
}
DATA_CRITERIA Gif_Criteria
{
    DATA_ATTRIBUTES_NAME Gif
    NAME_PATTERN         *.gif
}
```

3. 鍵入 `GifViewer` 動作的動作定義：

```
ACTION GifViewer
{
    EXEC_STRING          xgif %(File)Arg_1"Gif file to view:"
    WINDOW_TYPE         NO_STDIO
    DESCRIPTION         Double-click or drop a file to \
                        start the Gif viewer.
}
```

既然定義不包括 `ICON` 欄位，動作將會使用系統的預設圖示。

4. 鍵入以下對映動作以將 `GifViewer` 動作連接至列於資料類型定義中的「查看」動作。使用 `ARG_TYPE` 欄位以限制此查看動作為 `Gif` 類型的檔案。

```
ACTION View
{
    ARG_TYPE            Gif
    TYPE               MAP
    MAP_ACTION         GifViewer
}
```

5. 儲存檔案。
6. 連按兩下桌面工具應用程式群組中的「重新載入動作」以重新讀取資料庫。

定義資料類型的資料屬性

DATA_ATTRIBUTES 定義定義資料類型的外觀與行為。其指定資料類型的名稱，並提供能力以指定：

- 檔案管理者圖示 (ICON 欄位)
- 「所選項目」功能表的連按兩下行為與內容 (ACTION 欄位)
- 資料類型的關於項目輔助說明 (DESCRIPTION 欄位)

指定用於資料類型的圖示影像

使用 ICON 欄位以指定「檔案管理者」中所使用的圖示。如果您未指定圖示影像，「檔案管理者」便只會顯示標籤。

ICON 欄位的值可以是：

- 基本檔案名稱。
基本檔案名稱是包含圖示影像的檔案名稱，減去檔案名稱大小 (m 與 t) 與影像類型 (bm 與 pm) 的字尾。例如，如果檔案名稱為 GameIcon.m.pm 與 GameIcon.t.pm，請使用 GameIcon。
如果您使用基本檔案名稱，圖示檔案必須放在圖示搜尋路徑上的目錄中：
 - 個人圖示：`language/.dt/icons`
 - 整個系統的圖示：`/etc/dt/appconfig/icons/language`
- 到圖示檔案的絕對路徑，包括完整的檔案名稱。
唯有當圖示檔案不在圖示搜尋路徑上時才應使用絕對路徑。例如，如果圖示檔案 GameIcon.m.pm 是放在不在圖示搜尋路徑上的目錄 `/doc/projects` 中，ICON (圖示) 欄位的值會是 `/doc/projects/GameIcon.m.pm`。
表 13-1 列出您應該建立的圖示大小與相應的檔案名稱。

表 13-1 資料類型圖示的圖示名稱與大小

像素的大小	點陣圖名稱	像素對映名稱
32 乘 32	<code>name.m.bm</code>	<code>name.m.pm</code>
16 乘 16	<code>name.t.bm</code>	<code>name.t.pm</code>

使資料類型與動作產生關聯

使資料類型與動作有關聯的方法有兩種：

- DATA_ATTRIBUTES 定義中的 ACTIONS 欄位會列出將出現在「檔案管理者」的「選定」功能表中的動作。清單中第一個動作是預設（連按兩下）動作。
- 動作可以使用動作定義的 ARG_TYPE 欄位來限制為指定的資料類型。

例如，以下資料類型定義會建立由您的系統管理者使用命名慣例 *.rm 所建立的特殊“讀我”檔案的資料類型。

```
DATA_ATTRIBUTES SysReadmeFile
{
    ICON                      SysReadMe
    ACTIONS                   Open, Respond
}
DATA_CRITERIA SysReadmeFileCriteria
{
    NAME_PATTERN              *.rm
    DATA_ATTRIBUTES_NAME    SysReadmeFile
}
```

檔案的特殊「回應」動作定義於下。其開啓「文字編輯程式」中檔案的可寫入副本。當儲存檔案及結束「文字編輯程式」時，檔案會郵寄給系統管理者（郵件地址 sysadmin@utd）。

```
ACTION Respond
{
    ARG_TYPE                  SysReadmeFile
    EXEC_STRING               /bin/sh -c 'cp%Arg_1% $HOME/readme.temp;\
                             chmod +w $HOME/readme.temp; \
                             dtpad $HOME/readme.temp; \
                             cat $HOME/readme.temp \
                             cat $HOME/readme.temp | \
                             /usr/bin/mailx sysadmin@utd; \
                             rm $HOME/readme.temp'
    WINDOW_TYPE              NO_STDIO
}
```

依據資料類型隱藏檔案

如果檔案是隱形的資料類型，它絕不會出現在「檔案管理者」中。

使用 DATA_ATTRIBUTES 定義中的 PROPERTIES 欄位來指定將隱藏的物件類型：

```
PROPERTIES                  invisible
```

處理檔案時，指定行爲

表 13-2 顯示主要由應用程式設計者所使用的 DATA_ATTRIBUTES 欄位。它們會指定當使用者執行各種不同的桌上管理系統活動時，檔案的行爲要如何。

有關更多的資訊，請參閱 [共用桌上管理系統環境程式設計者指南](#)，其為開發者環境文件的一部分。

表 13-2 DATA_ATTRIBUTES 欄位及說明

欄位	說明
MOVE_TO_ACTION	用於如目錄的容器。當檔案移動至此資料類型的容器時，指定要執行的動作。
COPY_TO_ACTION	用於如目錄的容器。當檔案複製到此資料類型的容器時，指定要執行的動作。
LINK_TO_ACTION	當檔案連結至此資料類型的檔案時，指定要執行的動作。
IS_TEXT	指定此資料類型的檔案包含可以顯示在文字方塊中的文字。
MEDIA	指定相應的 ToolTalk 媒體類型。
MIME_TYPE	指定相應的 MIME 類型。
X400_TYPE	指定相應的 X400 類型。

定義資料類型的資料條件

DATA_CRITERIA 定義會定義用於將物件類型指派給檔案或目錄的條件。

您可以使用表 13-3 中所示的物件類型的條件。

表 13-3 DATA_CRITERIA 條件與說明

條件	說明
檔案名稱	檔案名稱必須符合指定的型樣。使用 NAME_PATTERN 欄位。
檔案位置	路徑必須符合指定的型樣。使用 PATH_PATTERN 欄位。
檔案內容	檔案內容的指定部分必須符合指定的資料。使用 CONTENT 欄位。
檔案模式	檔案必須擁有指定的許可權（讀取、寫入、執行、目錄）。使用 MODE 欄位。
符號連結	類型是以物件連結的檔案為基礎。

您可以使用一項以上的資料類型條件。但是，您不應在相同的資料類型中使用 NAME_PATTERN 與 PATH_PATTERN 條件。

名稱型資料類型

使用 NAME_PATTERN 欄位來指定命名需求。欄位值可包括下列通配符號：

?—符合任何單一字元

*—符合任何順序的字元（包括空字串）

[cc...]—符合括弧中所附的任何字元（*c*）

[c-c]—符合範圍中的任一字元 *c* 到 *c*

範例

- 以下資料類型定義建立了以檔案名稱為基礎的資料類型。檔案名稱必須以 QS 開始，以 .doc 結尾。

```
DATA_ATTRIBUTES QS_Doc
{
    DESCRIPTION      This file contains a document for the QS
project.
    ICON             Word_Doc
    ACTIONS          Open
}
DATA_CRITERIA QS_Doc_Criteria
{
    NAME_PATTERN     QS*.doc
    DATA_ATTRIBUTES_NAME QS_Doc
}
```

- 以下的定義建立名為 Demo_*n* 其 *n* 為 0 到 9。

```
DATA_ATTRIBUTES Demo_directory
{
    DESCRIPTION      This is a directory. Double-click to open it.
    ICON             Demo
    ACTIONS          OpenInPlace,OpenNewView
}

DATA_CRITERIA Demo_directory_criteria
{
    NAME_PATTERN     Demo_[0-9]
    MODE             d
    DATA_ATTRIBUTES_NAME Demo_directory
}
```

位置型資料類型

使用 PATH_PATTERN 欄位來指定路徑。您可以使用與 NAME_PATTERN 相同的通配符號。

例如，以下的資料類型使用以路徑為基礎的條件。

```

DATA_ATTRIBUTES Project_Graphics
{
    DESCRIPTION      Graphics file for the QS project. Double-click the \
                    icon to see the graphic.
    ICON              QSgraphics
}
DATA_CRITERIA Project_Graphics_Criteria
{
    DATA_ATTRIBUTES_NAME    Project_Graphics
    PATH_PATTERN              */projects/QS/graphics/*
}

```

以名稱與位置為基礎的資料類型

如果要建立同時以檔案名稱與位置兩者為基礎的資料類型，請將名稱包含於 `PATH_PATTERN` 值中。您無法在相同的條件定義中同時使用 `NAME_PATTERN` 與 `PATH_PATTERN`。

範例

- 定義如下的 `QS_Source_Files` 資料類型應用於所有名為 `appn.c` 的檔案，其中 $n=1$ 到 9，位於 `*/projects/QS` 的子目錄中。

```

DATA_ATTRIBUTES QS_Source_Files
{
    ...
}
DATA_CRITERIA QS_Source_Files_Criteria
{
    PATH_PATTERN          */projects/QS/*/app[1-9].c
    DATA_ATTRIBUTES_NAME    QS_Source_Files
}

```

- 以下資料類型應用於目錄 `/doc/project1` 中名為 `chmn.xxx` 其中 n 為 0 到 9，而 `xxx` 是任何三個字元的檔案名稱字尾。

```

DATA_ATTRIBUTES ChapterFiles
{
    DESCRIPTION      Chapter file for the project document.
    ICON              chapter
    ACTIONS           Edit,Print
}

DATA_CRITERIA Chapter_Criteria
{
    PATH_PATTERN          /doc/project1/ch[0-9][0-9].???
    DATA_ATTRIBUTES_NAME    ChapterFiles
}

```

使用檔案模式作為類型條件

使用 MODE 欄位來指定所需的許可權。

模式條件通常用於以名稱為基礎、以位置為基礎或以內容為基礎的資料類型的組合中。它們允許您將資料類型限制為檔案或目錄，或指定所需的讀取、寫入與執行許可權。

MODE（模式）欄位可包括邏輯運算子（表 13-4）與字元（表 13-5）。

表 13-4 模式欄位邏輯運算子與說明

運算子	說明
!	邏輯運算子 NOT
&	邏輯運算子 AND
	邏輯 OR

表 13-5 模式欄位字元與說明

字元	說明
f	資料類型只應用於檔案
d	資料類型只應用於目錄
r	檔案可由任何使用者讀取
w	檔案可由任何使用者寫入
x	檔案可由任何使用者執行
l	檔案為連結

特定模式的預設值為模式沒有關係。

範例

- 下列模式欄位限制如下所示的資料類型：

f&!w—唯讀檔案

!w—唯讀檔案及目錄

f&x—可執行檔

f&r&x—同時可寫入與可執行的檔案

x|!w—可執行或唯讀的檔案

- 以下資料類型定義建立了唯讀且不可執行的檔案的資料類型，其檔案名稱遵循命名慣例*.doc。這種定義假定資料類型的「查看」動作已定義了。

```
DATA_ATTRIBUTES ReadOnlyDocument
{
```

```

        ICON                read_only
        DESCRIPTION         This document is not writable.
Double- \                  clicking runs your editor with a \
                             read-only copy of the file.
        ACTIONS            View
    }

DATA_CRITERIA ReadOnlyDocument_Criteria
{
    NAME_PATTERN            *.doc
    MODE                    !d&!x&!w
    DATA_ATTRIBUTES_NAME  ReadOnlyDocument
}

```

內容型資料類型

使用 CONTENT 欄位以指定以檔案內容為基礎的資料類型。內容型資料類型可以在以名稱或位置為基礎的資料類型組合中使用。

類型可以以檔案的字串或數值內容為基礎。檔案中的第一個位元組為數字 0。

- 對於檔案的字串內容，請使用此語法：

```
CONTENT starting_byte 字串 字串
```

- 對於檔案的數字內容，請使用此語法：

```
CONTENT starting_byte 位元組 數目
```

```
CONTENT starting_byte 短 數目
```

```
CONTENT starting_byte 長 數目
```

- 對於目錄的內容，使用此語法：

```
CONTENT 0 檔案名稱 "file_name"
```

使用八進位（前導 0）與十六進位（前導 0X）數字的標準 C 記號。

註解 – 使用以內容為基礎的資料類型將會導致較慢的系統效能。在可能之處使用以名稱與位置為基礎的類型來代替。

例如，以下的資料類型 Writable_Wingz 應用至具有包含字串 WNGZ 於檔案開始之處的寫入許可權的所有檔案。

```

DATA_ATTRIBUTES Writable_Wingz
{
    ...
}

DATA_CRITERIA Writable_Wingz_Criteria

```

```

{
    CONTENT                0 string WNGZ
    MODE                    w&!d
    DATA_ATTRIBUTES_NAME  Writable_Wingz
}

```

▼ 建立具有數項獨立條件的資料類型

您可以建立具有數項獨立條件的資料類型—也就是說，如果檔案符合條件之一（或兩者），就會被指派至資料類型。

1. 建立新的「前方面板」的 `DATA_ATTRIBUTES` 定義。
2. 建立每項條件的 `DATA_CRITERIA` 定義。

使用 `DATA_ATTRIBUTES_NAME` 欄位將每項條件連接至相同的 `DATA_ATTRIBUTES` 定義。

例如，下列定義會建立 Mif 資料類型。類型是以名稱或內容為基礎。

```

DATA_ATTRIBUTES Mif
{
    ICON                    Frame
    ACTION_LIST             Open,Print
}

DATA_CRITERIA Mif_Name_Criteria
{
    DATA_ATTRIBUTES_NAME  Mif
    NAME_PATTERN           *.mif
}

DATA_CRITERIA Mif_Content_Criteria
{
    DATA_ATTRIBUTES_NAME  Mif
    CONTENT                 1 string MIFFile
}

```

建立本土化資料類型

資料類型的搜尋路徑包括語言相依的位置。桌上管理系統使用 `LANG` 的值來決定搜尋資料類型的位置。

本土化資料類型的位置

本土化的資料類型定義必須放在動作搜尋路徑沿途的正確語言相依目錄中。

預設搜尋路徑為：

- 個人動作：*HomeDirectory* / *.dt/types*
- 整個系統的動作：*/etc/dt/appconfig/types/language*
- 內建動作：*/usr/dt/appconfig/types/language*

▼ 如果要本土化資料類型

1. 在合適的語言相依目錄中建立檔案（例如，在 */etc/dt/appconfig/types/japanese* 中）。
2. 將資料類型定義複製到語言相依的配置檔案。
3. 本土化資料類型定義中的一個或多個欄位。

第 14 章

建立桌上管理系統的圖示

桌上管理系統圖示與以下有關聯：

- 「檔案管理者」與「應用程式管理者」中的動作檔案與資料類型
- 「前方面板」控制
- 最小化的應用程式視窗
- 由應用程式所使用的圖形，例如調色板與工具列
- 工作區背景定位
- 第 209 頁“圖示影像檔案”
- 第 211 頁“圖示關聯”
- 第 214 頁“圖示設計建議”

註解 – 開發環境文件包含有關桌上管理系統圖示的其它資訊。請參閱第 4 章，*Common Desktop Environment: Style Guide and Certification Checklist* 中的“視覺設計”。

圖示影像檔案

如果要讓桌上管理系統使用圖示影像，圖示影像檔案必須：

- 是正確的格式。
- 使用正確的檔案命名慣例。
- 使用桌上管理系統大小慣例。
- 位於圖示搜尋路徑沿途的目錄中。
- 由桌上管理系統結構使用正確的語法呼叫。例如，如果您建立「前方面板」的新控制，請使用「前方面板」定義中的 ICON 欄位來指定用於控制的圖示影像。

圖示檔案格式

對於彩色顯示器，請使用一般具有 .pm 字尾的 X 像素對映 (XPM) 格式圖示檔案。否則，請使用一般具有 .bm 字尾的 X 點陣圖 (XBM) 格式檔案。如果像素對映檔案使用透明圖案，建立 .bm 檔案時會產生遮罩檔案 (_m.bm)。請參閱 第 142 頁“圖示搜尋路徑”，取得桌上管理系統如何尋找這些檔案的詳細資訊。

圖示檔案名稱

每個圖示與背景定位影像都以個別的檔案儲存。一般而言，圖示是以其檔案名稱的基本部分來指定的。例如，當檔案真正儲存為下列時，可用名稱郵件參考圖示：

```
/usr/dt/appconfig/icons/language /mail.l.pm
```

增加字尾的檔案命名慣例藉由大小與類型來輔助說明群組圖示。桌上管理系統元件的圖示名稱是用這些一般格式：

基本名稱.大小.格式

或

基本名稱.格式

其中：

基本名稱—用於參考影像的影像基本名稱

大小—指出大小的字母：l (大) m (中) s (小) t (微)

格式—檔案格式：pm (圖素對映) bm (點陣圖)

圖示大小慣例

表 14-1 顯示建議的桌上管理系統圖示的像素尺寸。

表 14-1 圖示大小與檔案名稱

圖示大小	點陣圖名稱	像素對映名稱
16 乘 16 (微)	<i>name</i> .t.bm	<i>name</i> .t.pm
24 乘 24 (小)	<i>name</i> .s.bm	<i>name</i> .s.pm
32 乘 32 (中)	<i>name</i> .m.bm	<i>name</i> .m.pm
48 乘 48 (大)	<i>name</i> .l.bm	<i>name</i> .l.pm

表 14-2 顯示由桌上管理系統元件所使用的圖示大小。在某些情況中，使用的圖示大小要視顯示器的解析度而定。

表 14-2 桌上管理系統元件與其圖示大小

桌上管理系統元件	高解析度	中解析度	低解析度
檔案管理者與應用程式管理者（由名稱與圖示查看）	中型	中型	中型
檔案管理者與應用程式管理者（由名稱和小圖示查看）	小型	小型	小型
主要前方面板控制	大型	大型	中型
前方面板子面板	中型	中型	小型
前方面板開關控制	小型	小型	小型
最小化的視窗	大型	大型	中型

例如，如果您指定名為郵件的圖示給資料類型，有彩色顯示器，並已設定小圖示的「檔案管理者」個人喜好，使用的圖示影像為 `mail.t.pm`。

圖示搜尋路徑

桌上管理系統藉由搜尋目錄清單中的檔案來尋找圖示檔案或影像。稱為**圖示搜尋路徑**的目錄清單是由數個環境變數的值所決定的。將使用哪些變數及如何將它們放在一起來建立圖示搜尋路徑，將在第 142 頁“圖示搜尋路徑”中加以討論。

預設搜尋路徑為：

- 內建圖示：`/usr/dt/appconfig/icons/language`
- 整個系統的圖示：`/etc/dt/appconfig/icons/language`
- 個人圖示：`HomeDirectory /.dt/icons`

經由網路存取圖示

桌上管理系統可以在遠端系統上存取圖示。有關建立圖示伺服器的資訊，請參閱第 124 頁“配置資料庫、圖示及輔助說明服務”。

圖示關聯

如果要啟動較快的物件認可，您可以使圖示與以下的项目產生關聯：

- 動作與資料類型
- 前方面板與輔助面板中的控制

- 最小化的應用程式視窗

指定圖示檔案

對於用於動作、資料類型與前方面板或子面板中的圖示，只要指定圖示的基本名稱（無字尾）。正確的字尾會依據您的顯示器解析度、色彩支援以及「檔案管理者」查看選項（例如小圖示）自動增加上去。

如果要置換搜尋路徑，請提供圖示的完整路徑與名稱。

▼ 如果要使圖示與動作或資料類型關聯

1. 使用 ICON 欄位來指定圖示。

如果您遵循圖示檔案的合適命名慣例，只要指定圖示的基本名稱。正確的圖示將會根據您的顯示器的解析度與色彩支援來顯示。

2. 建立下列的圖示大小：

- 動作：大型、中型與微型
- 資料類型：中型與微型

動作定義的範例

以下範例是啟動 Island Paint 繪圖工具的動作定義。圖示 `Ipaint.l` 和 `Ipaint.s` 與動作有關聯。

```
ACTION      IslandPaintOpenDoc
{
    WINDOW_TYPE      NO-STDIO
    ICON              Ipaint
    EXEC_STRING      /usr/bin/IslandPaint%Arg _1"開啓的檔案："%
}
```

如果您正在使用色彩圖示，尋找實際的圖示檔案時，桌上管理系統會先附加 `.pm`。否則（或是如果找不到符合 `.pm` 者），桌上管理系統會附加 `.bm`。

資料類型定義的範例

以下的資料類型定義使得具有壓縮檔的圖示相關聯 `comprsd.l` 與 `comprsd.s`：

```
DATA_ATTRIBUTES  COMPRESSED
{
    ICON          comprsd
    ACTION        Uncompress
}
```

```

        DESCRIPTION      壓縮的 檔案已由 \
        '壓縮' 指令來減少空間。
    }

```

▼ 如果要顯示前方面板控制中的圖示

1. 使用 ICON 欄位來指定影像名稱。

如果控制在監視檔案 (MONITOR_TYPE 被設定為郵件或檔案)，請使用 ALTERNATE_ICON 欄位指定當偵測到變更時所使用的圖示。

您也可以提供按鈕與定位區域控制的動畫。

2. 建立下列的圖示大小：

- 前方面板與輔助面板：大型、中型與微型
- 工作區開關：小型

範例

當名為 report (報告) 的檔案放在 /doc/ftp/pub/ 目錄中時，以下控制會變更外觀。當檔案不在該處時，NoReport.pm 圖示會顯示出來；當檔案在該處時，Report.pm 會顯示出來。

```

CONTROL MonitorReport
{
    CONTAINER_NAME container_name
    TYPE             ICON
    MONITOR_TYPE     file
    FILE_NAME        /doc/ftp/pub/report
    ICON             NoReport
    ALTERNATE_ICON   Report
}

```

▼ 如果要關聯圖示與應用程式視窗

1. 設定「工作區管理者」的 iconImage 資源如下：

Dtwm*clientname*iconImage: icon_file_name

如果要決定 clientname 的正確值，請開啓「應用程式管理者」並連按兩下桌面工具應用程式群組中的「視窗屬性」。當您選擇視窗時，會列出其屬性。WM_CLASS 屬性以引號顯示視窗的類別名稱。

有關設定資源的更多資訊，請參閱 第 257 頁“設定應用程式資源”。

2. 從「工作區」功能表選擇「重新啓動工作區管理者」。

如果要驗證「工作區管理者」已認可圖示，將最小化您試圖修改的視窗的圖示。

註解 – 有些應用程式不允許其預設視窗圖示被置換。

▼ 使用檔案管理者作為圖示瀏覽器

1. 將檔案 `/usr/dt/examples/language/IconBrowse.dt` 複製到 `HomeDirectory/.dt/types/Iconbrowse.dt` 目錄。
2. 開啓「應用程式管理者」並連按兩下桌面工具應用程式群組中的「重新載入動作」。
當您變更包含圖示（.bm 與 .pm 檔案）的目錄時，每個圖示都會顯示在其名稱旁。例如，如果您變更爲 `/usr/dt/appconfig/icons/language` 目錄，您將會看到許多桌上管理系統圖示。

註解 – 在低記憶體系統上啓用圖示瀏覽可能會使「檔案管理者」顯示目錄的速度更緩慢。大於 256 x 256 的影像不會顯示在預設配置中。

如果要取消圖示瀏覽：

1. 刪除您個人的 `IconBrowse.dt` 檔案副本。
2. 開啓「應用程式管理者」並連按兩下桌面工具應用程式群組中的「重新載入動作」。

圖示設計建議

使用相關圖示之間共同主題。例如，如果您正在設計應用程式的圖示，就可以有應用程式的圖示與資料檔案圖示之間有意義的相似之處。

確定您所設計的任何色彩圖示的雙色版本是可接受的。如果圖示顯示於單色或灰階顯示器上（或是如果沒有足夠的色彩可供使用），圖示會自動以其雙色形式來顯示。

如果要保存系統色彩用法，請試著限制由桌上管理系統所提供的色彩的圖示色彩使用。（使用「圖示編輯程式」所建立的圖示將只使用桌上管理系統色彩。）

有關桌上管理系統元件所使用的大小，請參閱表 14-1。

色彩用法

桌上管理系統圖示使用 22 色的調色板，包括：

- 八種靜態灰色

- 八種靜態色彩：紅色、藍色、綠色、靛色、深紅色、黃色、黑色與白色
- 六種動態色彩：前景、背景、頂部陰影、底部陰影、選取與透明

此調色板建立了有吸引力且讀取簡易的圖示，而且不超過其它應用程式所需的色彩資源。桌上管理系統提供的大部分圖示都使用以色彩強調的灰色。

透明色彩對於建立具有非長方形的幻覺圖示是很有用的，因為圖示後面的色彩會透通顯示出來。

第 15 章

進階前方面板自訂

使用者可以使用其即現式功能表與子面板中的「安裝圖示」控制來自訂「前方面板」。

本章涵蓋藉由建立與編輯配置檔案來自訂「前方面板」。

- 第 217 頁“前方面板配置檔案”
- 第 219 頁“管理使用者介面自訂”
- 第 220 頁“前方面板定義的組織”
- 第 223 頁“修改主面板”
- 第 228 頁“建立及修改輔助面板”
- 第 231 頁“定義前方面板控制”
- 第 237 頁“自訂工作區切換”
- 第 239 頁“一般前方面板配置”
- 有關「前方面板」控制與配置的參考資訊，請參閱 `dtfpfile(4X)` 線上援助頁。
- 有關「工作區管理者」的參考資訊，請參閱 `dtwm(1)` 與 `dtwmrc(4)` 線上援助頁。

前方面板配置檔案

「前方面板」定義於配置檔案的資料庫中。

配置檔案提供自訂「前方面板」的方法。特定的修改只可藉由編輯配置檔案來完成，包括：

- 增加新的控制位置至「主面板」。
- 增加特殊類型的控制，例如用戶端視窗。
- 變更特定的預設行為—例如，「前方面板」控制是否要回應按一下或連按兩下。

如果要提供配置面板的最大彈性，這些檔案可以是個人的、整個系統的或是位於其它系統上的。

「前方面板」是由「工作區管理者」所建立與管理的。

預設前方面板配置檔案

預設「前方面板」定義於「前方面板」配置檔案
`/usr/dt/appconfig/types/language/dtwm.fp` 中。

不應改變此檔案。

前方面板配置檔案的搜尋路徑

「前方面板」定義可以分配於任何數目的位於本機或遠端系統上的檔案之間。

用於定義「前方面板」的檔案必須符合這些需求：

- 檔案名稱必須以 `.fp` 結尾；例如，`mail.fp`。
- 檔案必須位於動作資料庫搜尋路徑沿途。

預設動作資料庫搜尋路徑包括這些以如下順序搜尋的目錄：

個人自訂—`HomeDirectory/.dt/types`

整個系統的自訂—`/etc/dt/appconfig/types/language`

內建面板與控制—`/usr/dt/appconfig/types/language`

另外的目錄 `HomeDirectory/.dt/types/fp_dynamic` 是用於以使用者介面製作的個人自訂。請勿使用此目錄於人工自訂。

動作資料庫搜尋路徑可包括增加以配置網路系統的其他目錄。特別是當配置系統以存取應用程式伺服器時，會增加其他的遠端位置。有關更多的資訊，請參閱第 140 頁“資料庫 (動作/資料類型) 搜尋路徑”。

如何組合前方面板：優先順序規則

「前方面板」是由所有位於動作資料庫搜尋路徑上的配置檔案所組合的。

在定義的元件之間有衝突之處，優先順序規則會決定使用何種定義。當兩種元件有以下情況時會彼此衝突：

- 有相同的控制名稱，`CONTAINER_NAME` 與 `CONTAINER_TYPE`。
- 爭相同的位置（藉由有不同的名稱但相同的 `CONTAINER_NAME`，`CONTAINER_TYPE` 與 `POSITION_HINTS`）。

「前方面板」使用下列的優先順序規則：

- 如果元件有相同的控制名稱以及容器名稱與類型，則使用先讀取的元件。
例如，如果整個系統與內建的控制都包含這些欄位，但其它部分不同時：

```
CONTROL TextEditor
{
    CONTAINER_TYPE          BOX
    CONTAINER_NAME          Top
    ...
}
```

那麼整個系統的控制就會有優先順序。

- 如果兩個元件爭相同的位置，則以讀取的順序來放置。
例如，如果使用者建立「主面板」（CONTAINER_TYPE BOX 與 CONTAINER_NAME 頂端）的新個人控制並指派它 POSITION_HINTS 5，個人控制將會把內建控制與所有其它具有較高位置編號的控制往右推一個位置。

註解 – 當您正藉由建立新的整個系統或其個人版本來修改控制時，新的控制定義必須指定相同的控制名稱，CONTAINER_NAME 與 CONTAINER_TYPE。否則，除了現存控制外將會出現新的控制。

動態建立的前方面板檔案

當使用者使用「安裝圖示」控制與即現式功能表來自訂「前方面板」時，檔案會寫入至目錄 *HomeDirectory/.dt/types/fp_dynamic*。

「前方面板」建立另外的檔案 *HomeDirectory/.dt/sessions/dtwmfp.session*，它是用以儲存與回復每個階段作業的自訂「前方面板」的狀態。

管理使用者介面自訂

使用者可以使用即現式功能表與「前方面板」控制的「安裝圖示」控制來擴展自訂「前方面板」。

本節描述如何：

- 防止特定的個人自訂。例如，您可能想要讓使用者不能刪除控制。
- 還原個人自訂。例如，使用者可能會要求您回復意外刪除的單一控制。

▼ 防止個人自訂

1. 如果控制是內建的控制，請從 `/usr/dt/appconfig/types/language/dtwm.fp` 複製其定義至 `/etc/dt/appconfig/types/language/name.fp`。
2. 將下行增加至控制定義：

```
LOCKED    True
```

▼ 回復已刪除的控制或輔助面板

桌面工具應用程式群組中的「回復前方面板」動作刪除了所有以使用者介面作的「前方面板」自訂。使用者可以使用此動作來刪除他們所有以「前方面板」的即現式功能表所作的個人自訂。

使用以下程序來回復個別控制。

- 在 *HomeDirectory* `/.dt/types/fp_dynamic` 目錄中，刪除在使用者刪除控制時所建立的檔案。控制將會有與被刪除的原始控制相同的名稱。

例如，如果使用者刪除「圖示編輯程式」控制，`fp_dynamic` 目錄中的檔案將會包含：

```
CONTROL IconEditor
{
    ...
    DELETE    True
}
```

當使用者刪除輔助面板時，會為輔助面板及輔助面板中的每個控制建立個別的動態檔案。

前方面板定義的組織

「前方面板」是藉由組合其元件的定義所建立的。這些元件每個都需要有定義元件放在「前方面板」中何處、元件看起來像什麼以及其如何行為的語法。

前方面板元件

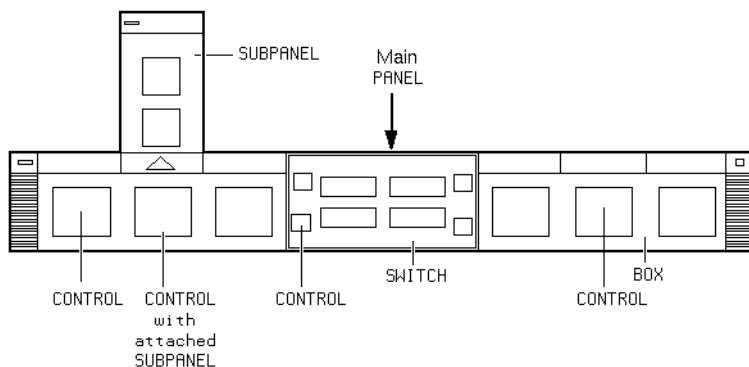


圖 15-1 前方面板元件

「前方面板」是由外向內組合的：

- 此 PANEL 是整個「前方面板」的頂層容器或父項。
- 此面板 是一個或多個 BOX 的容器。
- 所謂 BOX 是一個或多個 CONTROL（控制）。

有兩種特殊類型的容器：

- 輔助面板 與特定控制有關聯（控制是輔助面板的容器）。輔助面板可以從與其有關聯的控制“向上調整”得到。
- 此 SWITCH（開關） 包含變更工作區以及其它控制的按鈕。

前方面板定義的一般語法

「前方面板」中每個元件都使用此語法分別定義：

```
COMPONENT name
{
    KEYWORD      value
    KEYWORD      value
    ...
}
```

有些關鍵字是需要的，其它則是選擇性的。有關更多的資訊，請參閱 dtfprofile(4X) 線上援助頁。

PANEL 定義

此 PANEL 是頂層的元件。其定義包括：

- 「前方面板」名稱

- 描述整個「前方面板」的一般外觀與行為的欄位

```

面板 front_panel_name
{
    KEYWORD value
    KEYWORD value
    ...
}

```

此 *front_panel_name* 是「前方面板」的唯一名稱。預設名稱爲“FrontPanel”。

BOX 定義

BOX（方塊）定義描述：

- BOX 名稱
- 方塊所在的 PANEL (CONTAINER_NAME)
- PANEL 中方塊的位置 (POSITION_HINTS)
- 應用至整個方塊的描述外觀與行為的欄位

```

BOX box_name
{
    CONTAINER_NAME front_panel_name
    POSITION_HINTS position
    KEYWORD value
    KEYWORD value
    ...
}

```

CONTROL 定義

CONTROL（控制）定義描述：

- 此 CONTROL（控制）名稱
- 控制是在方塊、輔助面板或開關中 (CONTAINER_TYPE)
- 控制在哪個方塊、輔助面板或開關 (CONTAINER_NAME) 中
- BOX 中 CONTROL 的位置 (POSITION_HINTS)。
- 描述控制的外觀與行為的欄位

```

CONTROL control_name
{
    CONTAINER_TYPE BOX 或 輔助面板或 SWITCH
    CONTAINER_NAME box_name 或 subpanel_name 或 switch_name
    TYPE control_type
    POSITION_HINTS position
    KEYWORD value
    KEYWORD value
    ...
}

```

```
}
```

SUBPANEL 定義

SUBPANEL (輔助面板) 定義描述：

- 此 SUBPANEL 名稱
- 附加輔助面板的控制名稱 (CONTAINER_NAME)
- 描述輔助面板特定的外觀與行為的欄位

```
SUBPANEL subpanel_name
{
    CONTAINER_NAME control_name
    KEYWORD value
    KEYWORD value
    ...
}
```

SWITCH 定義

SWITCH (開關) 定義描述：

- 此 SWITCH 名稱
- SWITCH 所在的 BOX (CONTAINER_NAME)
- SWITCH 在 BOX 內的位置 (POSITION_HINTS)
- 描述 SWITCH 外觀和行為的欄位。

```
SWITCH switch_name
{
    CONTAINER_NAME box_name
    POSITION_HINTS position
    KEYWORD value
    KEYWORD value
    ...
}
```

修改主面板

「主面板」是除了輔助面板以外的「前方面板」的視窗。

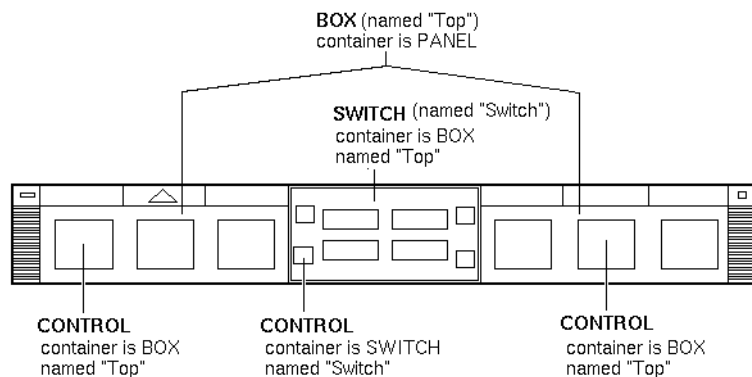


圖 15-2 主面板容器

您可以作的修改包括：

- 增加或刪除控制
- 交換控制的位置

▼ 增加控制至主面板

1. 建立「前方面板」配置檔案：

- 整個系統的：`/etc/dt/appconfig/types/language/*.fp`
- 個人的：`HomeDirectory/.dt/types/*.fp`

2. 在檔案中定義控制。

使用 `CONTAINER_NAME` 與 `CONTAINER_TYPE` 欄位以指定控制的容器：

```
CONTAINER_NAME 頂端          CONTAINER_TYPE  BOX
```

使用 `POSITION_HINTS` 來指定由左到右的控制位置。既然自訂對於內建控制有優先順序，新的控制會把具有該位置的現存控制往右“推”一個位置。

3. 儲存配置檔案。

4. 建立「前方面板」控制的圖示。

請參閱 第 227 頁“指定控制所使用的圖示”。

5. 從「工作區」功能表選擇「重新啟動工作區管理者」。

例如，放在檔案 `/etc/dt/appconfig/types/language/audio.fp` 中的以下控制定義會插入「時鐘」與「日曆」控制之間的聲訊應用程式控制。

```
CONTROL AudioApplication
{
    TYPE          icon
```



```

CONTAINER_NAME    Top
CONTAINER_TYPE    BOX
ICON              AudioApp
POSITION_HINTS    2
PUSH_ACTION       StartAudioApplication
PUSH_RECALL       true
}

```

▼ 移除控制

1. 建立「前方面板」配置檔案：

- 整個系統的：`/etc/dt/appconfig/types/language/name.fp`
- 個人的：`HomeDirectory/.dt/types/name.fp`

2. 將您想要刪除的控制定義複製到新檔案。

如果控制是內建的，其定義是在 `/usr/dt/appconfig/types/language/dtwm.fp` 中。

您不需要複製整個定義。但是，您所複製的部分必須包括欄位 `CONTAINER_NAME` 與 `CONTAINER_TYPE`。

3. 增加 DELETE（刪除）欄位至定義：

```
DELETE            True
```

4. 儲存配置檔案

5. 從「工作區」功能表選擇「重新啟動工作區管理者」。

例如，放在檔案 `/etc/dt/appconfig/types/language/TrashCan.fp` 中的以下控制定義會從「前方面板」中移除「垃圾桶」控制。

```

CONTROL Trash
{
  CONTAINER_NAME    Top
  CONTAINER_TYPE    BOX
  DELETE            True
}

```

▼ 修改控制

當您需要修改控制定義時，請使用此程序—例如，如果要變更其圖示影像。

1. 從 `/usr/dt/appconfig/types/language/dtwm.fp` 複製整個控制定義至：

- 整個系統的：`/etc/dt/appconfig/types/language/name.fp`
- 個人的：`HomeDirectory/.dt/types/name.fp`。

2. 編輯您想要變更的欄位。您也可以增加其它的欄位。

3. 儲存檔案
4. 從「工作區」功能表選擇「重新啟動工作區管理者」。

▼ 交換控制的位置

1. 複製您想要從 `/usr/dt/appconfig/types/language/dtwm.fp` 複製整個控制定義至：
 - 整個系統的：`/etc/dt/appconfig/types/language/name.fp`
 - 個人的：`HomeDirectory/.dt/types/name.fp`。
您必須複製整個控制定義，這樣才能移動每個控制。
2. 交換控制定義的 `POSITION_HINTS` 欄位的值。

3. 儲存檔案

4. 從「工作區」功能表選擇「重新啟動工作區管理者」。

例如，放在檔案 `/etc/dt/appconfig/types/C/MailHelp.fp` 中的下列定義會交換「郵件」與「輔助說明管理者」的位置，並鎖定這些對於個人變更的控制。

```
CONTROL Mail
{
    POSITION_HINTS                12
    LOCKED                       True
    ...the rest of the control definition
}

CONTROL Help
{
    POSITION_HINTS                5
    LOCKED                       True
    ...the rest of the control definition
}
```

▼ 取代前方面板控制

- 建立另一個控制定義，其具有相同的：
 - `control_name`
 - `CONTAINER_NAME` value例如，下列兩個控制定義於兩個不同的配置檔案中。控制有相同的控制名稱與容器名稱，因此可視為相同的控制。
- `/etc/dt/appconfig/types/C/SysControls.fp` 中的定義：

```
CONTROL ImportantApplication
{
```

```

CONTAINER_NAME          TOP
CONTAINER_TYPE          BOX
POSITION_HINTS          2
...
}

```

- *HomeDirectory* / .dt/types/MyControls.fp 中的定義：

```

CONTROL ImportantApplication
{
CONTAINER_NAME          TOP
CONTAINER_TYPE          BOX
POSITION_HINTS          6
...
}

```

個人控制有優先順序，因此控制會位於位置 6。

指定控制所使用的圖示

控制定義的 ICON（圖示）欄位定義用於控制的圖示影像。

ICON（圖示）欄位的值可以是：

- 基本檔案名稱。

基本檔案名稱是包含圖示影像減去檔案名稱的大小（m 與 t）與影像類型（bm and pm）字尾的檔案名稱。例如，如果檔案名為 MyGame.l.pm 與 MyGame.m.pm，使用 MyGame。

如果您使用基本檔案名稱，圖示檔案必須放在圖示搜尋路徑上的目錄中：

- 個人圖示：*HomeDirectory* / .dt/icons
 - 整個系統的圖示：*/etc/dt/appconfig/icons/language*
- 到圖示檔案的絕對路徑，包括完整的檔案名稱。

唯有當圖示檔案不在圖示搜尋路徑上時才應使用絕對路徑。

您需要的大小圖示取決於控制的位置：

位置大小

主面板 – 48 乘 48 像素（*name.l.pm* 或 *name.l.bm*）

輔助面板 – 24 乘 24 像素（*name.s.pm* 或 *name.s.bm*）

將圖示檔案置於這些位置之一：

- 個人圖示：*HomeDirectory* / .dt/icons
- 整個系統的圖示：*/etc/dt/appconfig/icons/language*

建立及修改輔助面板

使用者可以使用「前方面板」即現式功能表來建立與修改輔助面板。

本節討論如何提供整個系統的自訂，其需要您修改「前方面板」配置檔案。

輔助面板是“附加”至「主面板」中的控制。

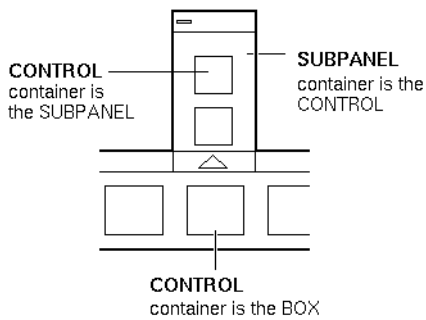


圖 15-3 輔助面板的容器是附加於容器的那個控制

附加完成於輔助面板定義中。CONTAINER_NAME 欄位指定輔助面板附加於其中的控制：

```
CONTROL control_name
{
    ...
}

SUBPANEL subpanel_name
{
    CONTAINER_NAME control_name
    ...
}
```

▼ 建立新的整個系統的輔助面板

1. 找出位於「主面板」中，您要附加輔助面板於其中的控制的 *control_name* 。
如果控制是內建控制之一，其定義是在 `/usr/dt/appconfig/types/language/dtwm.fp` 中。
2. 建立新檔案 `/etc/dt/appconfig/types/language/* .fp` 。
3. 定義輔助面板：

```

SUBPANEL subpanel_name
{
    CONTAINER_NAME          control_name
    TITLE                   value
    KEYWORD                 value
    ...
}

```

4. 儲存新的配置檔案。
5. 從「工作區」功能表選擇「重新啟動工作區管理者」。

自訂內建的輔助面板

您可以修改內建輔助面板的一般屬性（例如標題）與內容。

▼ 如果要修改內建輔助面板的一般屬性

1. 建立新的「前方面板」配置檔案：
 - 整個系統的： `/etc/dt/appconfig/types/language/name.fp`
 - 個人的： `HomeDirectory/.dt/types/name.fp`
2. 複製整個預設的 SUBPANEL 定義（其定義在 `/usr/dt/appconfig/types/language/dtwm.fp` 目錄中），到一個新的檔案：

```

SUBPANEL subpanel_name
{
    ...
}

```

3. 修改輔助面板定義。
4. 儲存新的配置檔案。
5. 從「工作區」功能表選擇「重新啟動工作區管理者」。

例如，以下放在檔案 `/users/janice/.dt/types/PerApps.fp` 中的定義變更了「個人應用程式」輔助面板的名稱：

```

SUBPANEL PersAppsSubpanel$
{
    CONTAINER_NAME          TextEditor$
    TITLE                   Janice's Applications$
}

```

▼ 如果要增加整個系統的控制至內建的輔助面板

1. 建立「前方面板」配置檔案 `/etc/dt/appconfig/types/language/name.fp`。
2. 定義檔案中整個系統的控制。

使用CONTAINER_NAME 與 CONTAINER_TYPE 欄位以指定控制的容器：

```
CONTROL control_name
{
    CONTAINER_NAME          subpanel_name
    CONTAINER_TYPE          SUBPANEL
    ...
}
```

請參閱 第 231 頁“定義前方面板控制”。

3. 儲存配置檔案。

4. 從「工作區」功能表選擇「重新啟動工作區管理者」。

例如，新檔案 `/etc/dt/appconfig/types/language/DigitalClock.fp` 中所定義的以下控制會將 DigitalClock（在桌面工具應用程式群組中）增加至所有使用者的「個人應用程式」輔助面板。

```
CONTROL DigitalClockControl
{
    TYPE          icon
    CONTAINER_NAME      PerAppsSubpanel
    CONTAINER_TYPE      SUBPANEL
    ICON            Dtdgclk
    PUSH_ACTION        DigitalClock
    PUSH_RECALL        True
}
```

▼ 如果要從內建的輔助面板刪除控制

- 使用與刪除「主面板」控制相同的程序。請參閱 第 225 頁“移除控制”。

▼ 移除安裝圖示控制

- 將以下欄位增加至輔助面板定義：

```
CONTROL_INSTALL    False
```

▼ 變更輔助面板的自動關閉行爲

當使用者選擇控制時，輔助面板的預設行爲是關閉，除非使用者已從其原始位置移去輔助面板。

可以配置「前方面板」以保持輔助面板開啓至使用者確實地關閉爲止。

1. 建立新的「前方面板」配置檔案於：

- 整個系統的：`/etc/dt/appconfig/types/language/*.fp`

- 個人的：*HomeDirectory* /.dt/types/*.fp
2. 複製預設的 PANEL 定義（由 /usr/dt/appconfig/types/language/dtwm.fp 目錄中）到一個新的檔案：


```
PANEL FrontPanel
{
    ...
}
```
 3. 將以下欄位增加至 PANEL 定義：


```
SUBPANEL_UNPOST    Fasle
```
 4. 儲存新的配置檔案。
 5. 從「工作區」功能表選擇「重新啟動工作區管理者」。

定義前方面板控制

使用者可以藉由將圖示定位於「安裝圖示控制」上來建立個人控制。

當此動作個人控制提供簡易自訂性之時，其所提供的功能是「前方面板」控制能力的子集。例如，使用「安裝圖示控制」所建立的控制無法：

- 提供動畫
 - 顯示用戶端視窗
 - 在事件發生時（例如，在收到新郵件時）變更外觀
- 本節描述如何人工建立「前方面板」控制。
- 有關「前方面板」控制的語法的參考資訊，請參閱 [dtfpfile\(4X\)](#) 線上援助頁。

前方面板控制定義

「前方面板」控制定義的結構為：

```
CONTROL (控制) control_name
{
    TYPE                control_type
    CONTAINER_NAME     value
    CONTAINER_TYPE     value
    other fields defining appearance and behavior
}
```

控制類型

控制定義中的 TYPE 欄位指定控制的基本行爲。

控制類型	控制行爲
icon (預設值)。	當使用者按一下控制或於其上定位檔案時，控制將會執行指定的動作。
blank	用以調整控制的空間設定之位置保留符號。
busy	進行燈。當呼叫動作時，控制會閃爍（切換影像）
client	「前方面板」中的用戶端視窗。
clock	時鐘。
date	顯示目前日期。
file	代表檔案。選擇執行檔案的預設動作之控制。

▼ 建立新的控制

本節描述定義控制的一般步驟，以及描述如何建立各種類型的控制。

1. 如果控制將會有 PUSH ACTION 及/或 DROP ACTION，請建立動作定義。這些是當使用者按一下控制或於其上定位檔案時所執行的動作。
2. 建立控制的圖示影像檔案。
有關圖示大小、名稱與位置的資訊，請參閱 第 209 頁“圖示影像檔案”。
3. 建立新的「前方面板」配置檔案於：
 - 整個系統的：`/etc/dt/appconfig/types/language/*.fp`
 - 個人的：`HomeDirectory/.dt/types/*.fp`
4. 將控制定義增加至檔案。
5. 儲存檔案。
6. 從「工作區」功能表選擇「重新啓動工作區管理者」。

建立在按一下時執行動作的控制

使用這些欄位來定義控制的行爲：

- TYPE：設定爲圖示
- PUSH_ACTION：指定要執行的動作名稱
例如，將會放在「個人應用程式」輔助面板的以下控制會執行使用者已獲得的遊戲：


```

CONTROL Ball
{
    TYPE                icon
    CONTAINER_NAME      PersAppsSubpanel
    CONTAINER_TYPE      SUBPANEL
    ICON                ball
    PUSH_ACTION          RunBallGame
    HELP_STRING          "Choose this control to play Ball."
}
}

```

以下控制將會位於開關的左上角。它會啟動名為 `CutDisp` 的動作。

```

CONTROL StartCutDisp
{
    TYPE                icon
    CONTAINER_NAME      Switch
    CONTAINER_TYPE      SWITCH
    POSITION_HINTS       first
    ICON                cutdisp
    HELP_STRING          "Choose this control to run cutdisp."
    PUSH_ACTION          CutDisp
}
}

```

建立開啓檔案的控制

使用這些欄位來定義控制的行為：

- `TYPE`：設定為檔案
- `FILE_NAME`：指定要開啓的檔案路徑
- `PUSH_ACTION`：設定為開啓

必須要有一個為檔案的資料類型定義的「開啓」動作。

例如，以下控制將會位於「主面板」的最右邊。它會啟動具有資料檔案 `/users/ellen/PhoneList.txt` 的「文字編輯程式」。*.txt 檔案的「開啓」動作是預設動作資料庫的一部分。

```

CONTROL EditPhoneList
{
    TYPE                file
    FILE_NAME           /users/ellen/PhoneList.txt
    CONTAINER_NAME      Top
    CONTAINER_TYPE      BOX
    POSITION_HINTS       last
    ICON                PhoneBook
    HELP_STRING          "This control displays Ellen's phone
list."
    PUSH_ACTION          Open
}
}

```

建立作用如定位區域的控制

當使用者在控制上定位檔案時，使用 `DROP_ACTION` 欄位來指定執行的動作。動作必須要能夠接受檔案引數。

控制定義經常同時包括 `PUSH_ACTION` 與 `DROP_ACTION` 欄位。您可以使用相同的動作於放入與定位動作。

例如，位於「個人應用程式」輔助面板中的以下控制執行取得檔案引數的 X 用戶端 `xwud`。

```
CONTROL Run_xwud
{
    CONTAINER_NAME           PerAppsSubpanel
    CONTAINER_TYPE           SUBPANEL
    POSITION_HINTS            2
    ICON                     XwudImage
    PUSH_ACTION              RunXwud
    DROP_ACTION              RunXwud
}
```

建立監視檔案的控制

使用這些欄位來定義控制的行為：

- **TYPE**：指定下列其中一個值：
 - 圖示—如果您想要為控制指定 `PUSH_ACTION` 及/或 `DROP_ACTION`，請使用此類型。
 - 檔案—在「檔案管理者」中連按兩下檔案的圖示時，如果您想要此控制在選擇時作用有如檔案，請使用此類型。
- **ICON** 與 **ALTERNATE_ICON**：描述用於指示受監視檔案的未變更與變更的狀態的影像。
- **MONITOR_TYPE**：描述引起影像變更的條件。使用下列其中一個值：
 - 郵件—當資訊增加至檔案時，控制將會變更外觀。
 - 檔案—當指定的檔案變成非空白時，控制將會變更。
- **FILE_NAME**：指定要監視的檔案。

例如，以下的控制尋找您期望使用匿名 `ftp` 將其傳輸至您的系統的名為 `meetings`（會議）的檔案。控制是放在「個人應用程式」輔助面板的頂端。

```
CONTROL MonitorCalendar
{
    TYPE                     file
    CONTAINER_NAME           PersonalApps
    CONTAINER_TYPE           SUBPANEL
    POSITION_HINTS            first
    FILE_NAME                 /users/ftp/meetings
}
```

```

        MONITOR_TYPE          file
        ICON                  meetingsno
        ALTERNATE_ICON        meetingsyes
    }

```

建立一個實例的（切換）控制

一個實例的控制會檢查由 `PUSH_ACTION` 啟動的程序是否已經執行。如果程序未執行，則執行 `PUSH_ACTION`。如果程序已經執行，視窗會移動至目前工作區中視窗堆疊的頂端。

使用這些欄位來定義控制的行為：

- `PUSH_RECALL`：設定為 `True`。
- `CLIENT_NAME`：向控制指定用戶端的名稱。
`CLIENT_NAME` 的值必須符合應用程式頂層視窗上的 `WM_CLASS` 屬性中的第一個字串 (`res_name`)。有關更多的資訊，請參閱 `xprop(1)` 線上援助頁。
- `PUSH_ACTION`：描述當使用者按一下控制時所執行的動作。
 例如，以下控制執行其動作名為 `MyEditor` 的應用程式的一個實例。

```

CONTROL MyEditor
{
    TYPE          icon
    CONTAINER_NAME Top
    CONTAINER_TYPE BOX
    POSITION_HINTS 15
    PUSH_RECALL  True
    CLIENT_NAME   BestEditor
    PUSH_ACTION   StartMyEditor
    ICON          MyEd
}

```

▼ 建立用戶端視窗控制

用戶端視窗控制是內含於「前方面板」中的應用程式視窗。例如，您可以藉由建立 `xload` 用戶端視窗控制將系統載入表放在「前方面板」中。

1. 定義控制。

使用這些欄位來定義控制的行為：

- `TYPE`：設定為 `client`。
- `CLIENT_NAME`：指定要啟動的用戶端。
`CLIENT_NAME` 的值必須符合應用程式頂層視窗上 `WM_CLASS` 屬性中的第一個字串 (`res_name`)。有關更多的資訊，請參閱 `xprop(1)` 線上援助頁。
- `CLIENT_GEOMETRY`：指定「前方面板」視窗所需要的大小，以像素計算大小。

xwininfo(1) 線上援助頁描述如何以像素找出視窗的大小。

2. 從「工作區」功能表選擇「重新啟動工作區管理者」。
3. 從終端機模擬器指令行啟動用戶端。

例如，以下控制顯示 30x20 的像素載入表。

```
CONTROL LoadMeter
{
    TYPE                client
    CONTAINER_NAME      Top
    CONTAINER_TYPE      BOX
    CLIENT_NAME         xload
    CLIENT_GEOMETRY     30x20
}
```

如果在階段作業之間未儲存與回復用戶端，您可能會想要配置控制以在使用者按一下時啟動用戶端。例如，您可以藉由將下行增加至定義來配置 LoadMeter 控制以啟動

xload：

```
PUSH_ACTION          StartXload
```

以及建立動作：

```
ACTION StartXload
{
    WINDOW_TYPE        NO_STDIO
    EXEC_STRING        /usr/contrib/bin/X11/xload
}
```

▼ 動畫控制

當使用者選擇控制或於其上定位物件時，您可以附加要使用的動畫順序。

爲了要有動畫順序，控制必須：

- 爲類型圖示
- 具有 PUSH_ACTION 或 DROP_ACTION

1. 使用 ANIMATION (動畫) 元件：

```
ANIMATION animation_name
{
    ANIMATION    icon1                [milisecond_delay]
    ANIMATION    icon2                [milisecond_delay]
    ...
}
```

其 *icon1*、*icon2* 等等是圖示的名稱，而 *milisecond_delay* 是以毫秒計算的動畫圖示之間的時間延遲。預設時間延遲爲 200 毫秒。

2. 增加 PUSH_ANIMATION 及/或 DROP_ANIMATION 欄位至控制定義。此值為 ANIMATION (動畫) 順序的名稱。

例如，下列各行使啓動 BestEditor 應用程式的控制成爲動畫。圖示之間的時間延遲是 300 毫秒。範例假定您已建立圖示檔案 frame1、frame2 等等。

```
CONTROL BestEditor
{
    ...
    PUSH_ANIMATION BestEdAnimation
    ...
}

ANIMATION BestEdAnimation
{
    frame1          300
    frame2
    ...
}
```

提供前方面板控制的關於項目輔助說明

提供控制的輔助說明有兩種方法：

- 在控制定義中提供輔助說明字串。

當使用者呼叫控制的關於項目輔助說明時，輔助說明字串會顯示於輔助說明查看器中。輔助說明字串不得包括製作格式（例如標題）或連結。

如果要提供輔助說明字串，請在控制定義中指定它：

```
HELP_STRING          help_string
```

- 在註冊的輔助說明內容中指定輔助說明主題。

輔助說明主題是使用輔助說明系統的完整能力所著作的資訊。著作輔助說明主題需要您使用桌上管理系統「輔助說明設計者工具箱」。

如果要提供輔助說明主題，請指定控制定義中的輔助說明內容與主題 ID：

```
HELP_VOLUME          help_volume_name
HELP_TOPIC           topic_id
```

自訂工作區切換

自訂工作區切換的方法有數種：

- 變更工作區的數目
- 變更切換的佈局
- 變更切換中的控制

▼ 如果要變更預設的工作區數目

- 修改以下的「工作區管理者」資源：

```
Dtwm*workspaceCount: n
```

有關更多的資訊，請參閱 第 246 頁“以整個系統的基礎來變更工作區的數目”。

▼ 變更切換列的數目

- 修改 SWITCH 定義中的 NUMBER_OF_ROWS 欄位。

例如，以下的定義會定義一個三列的切換。

```
SWITCH Switch
{
    CONTAINER_NAME          box_name
    NUMBER_OF_ROWS          3
    ...
}
```

▼ 在工作區切換中變更或增加控制

1. 建立具有控制定義的「前方面板」配置檔案。

- 指定控制要在開關內：

```
CONTAINER_NAME Switch CONTAINER_TYPE SWITCH
```

- 指定切換中的位置：

```
POSITION_HINTS n
```

其中 n 為整數。位置是以由左至右、由上至下的順序來編號。（對於預設的兩列式開關，位置是 1 到 4。）

2. 建立控制的圖示。建議的大小為 16 乘 16 像素。

例如，以下控制會將「終端機」控制放在開關中。

```
CONTROL SwitchTerminal
{
    TYPE          icon
    CONTAINER_NAME Switch
    CONTAINER_TYPE SWITCH
    POSITION_HINTS 3
    ICON          Fpterm
    LABEL         Terminal
    PUSH_ACTION   Dtterm
    HELP_TOPIC    FPOnItemTerm
}
```

```
HELP_VOLUME          FPanel
}
```

控制使用內建的圖示以及由「個人應用程式」輔助面板中的「終端機」控制所使用的相同輔助說明主題。

一般前方面板配置

「前方面板」的 `PANEL` 語法允許您：

- 變更「前方面板」的位置
- 變更視窗裝飾
- 設定控制的一般外觀與行為

預設 `PANEL` 說明是在 `/usr/dt/appconfig/types/language/dtwm.fp` 中。

有關其它的資訊，請參閱 `dtfpfile(4X)` 線上援助頁。

一般步驟

1. 在 `/etc/dt/appconfig/types/language` 或 `HomeDirectory/.dt/types` 中建立新的「前方面板」配置檔案。
2. 複製預設的 `PANEL` 描述（由 `/usr/dt/appconfig/types/language/dtwm.fp` 目錄）到一個新的檔案。
3. 編輯 `PANEL` 說明。
新的 `PANEL` 說明對於預設的面板有優先順序。

▼ 變更預設前方面板位置

- 使用 `PANEL` 定義中的 `PANEL_GEOMETRY` 欄位來指定位置。

例如，以下面板在右上角。

```
PANEL SpecialFrontPanel
{
    PANEL_GEOMETRY    -1+1
    ...
}
```

▼ 標示主要面板中的控制

1. 增加下行至 PANEL 定義：

```
DISPLAY_CONTROL_LABELS True
```

2. 增加 LABEL (標籤) 欄位至每個控制。
如果未指定 LABEL，將使用 *control_name*。

▼ 變更控制的按一下行爲

- 使用 PANEL 定義中的 CONTROL_BEHAVIOR 欄位，以指定使用者如何執行控制的 PUSH_ACTION。欄位值如下：
single_click—使用者按一下控制以執行 PUSH_ACTION
double_click—使用者連按兩下控制以執行 PUSH_ACTION

▼ 建立全新的前方面板

當您想要作大規模的變更時，建立新的「前方面板」可能會比較好。

如果要避免與內建的「前方面板」元件發生衝突，全新的「前方面板」應該使用新的名稱於 PANEL 與其它容器。

1. 建立新的「前方面板」的 PANEL 元件。給它一個唯一的名稱：

```
PANEL front_panel_name
{
    ...
}
```

2. 使用新的容器名稱來建立新的方塊與控制。
如果您想要使用現存元件，您必須複製其定義並變更 CONTAINER_NAME 值。
3. 從「工作區」功能表選擇「重新啓動工作區管理者」。

建立具有三列的個人前方面板的範例

下例的範例變更了預設「前方面板」，使得其控制組織成爲三列。

1. 將 `/usr/dt/appconfig/types/language/dtwm.fp` 複製到 `HomeDirectory/.dt/types/MyFrontPanel.fp`。給予檔案寫入許可權。
此爲您將要編輯以提供新的「前方面板」的檔案。

2. 變更「前方面板」的名稱：

```
PANEL NewFrontPanel
```

3. 變更為 **Top** (頂端) 的方塊名稱，並編輯其容器名稱：

```
BOX NewFrontPanelTop
{
    CONTAINER_NAME          NewFrontPanel
    POSITION_HINTS           first
    ...
}
```

4. 增加中間與底部各列的方塊定義：

```
BOX NewFrontPanelMiddle
{
    CONTAINER_NAME          NewFrontPanel
    POSITION_HINTS           second
}

BOX NewFrontPanelBottom
{
    CONTAINER_NAME          NewFrontPanel
    POSITION_HINTS           second
}
```

5. 將下列控制的 **CONTAINER_NAME** 變更為 **NewFrontPanelTop**：

- 時鐘
- 日期
- 主目錄
- 文字編輯程式
- 郵件

6. 將下列控制的 **CONTAINER_NAME** 變更為 **NewFrontPanelBottom**：

- 印表機
- 樣式
- 應用程式
- 輔助說明
- 垃圾

7. 將開關的 **CONTAINER_NAME** 變更為 **NewFrontPanelMiddle**。

8. 從「工作區」功能表選擇「重新啟動工作區管理者」。

第 16 章

自訂工作區管理者

本章描述如何自訂桌上管理系統的「工作區管理者」。

- 第 244 頁“工作區管理者配置檔案”
- 第 246 頁“自訂工作區”
- 第 248 頁“工作區管理者功能表”
- 第 252 頁“自訂按鈕組合”
- 第 254 頁“自訂按鍵組合”
- 第 255 頁“在預設與自訂行為之間切換”

「工作區管理者」是由桌上管理系統所提供的視窗管理者。就像其它的視窗管理者一樣，其控制：

- 視窗框架元件的外觀
- 視窗的行為，包括其堆疊順序與焦點行為
- 按鍵組合與按鈕組合
- 最小化視窗的外觀
- 「工作區」與「視窗」功能表

此外，「工作區管理者」還控制這些桌上管理系統元件：

- **工作區**。「工作區管理者」控制工作區的數目，並注意每個工作區中開啓了哪些視窗。
- **工作區背景定位**。使用者使用「樣式管理者」來變更背景定位。然而，背景定位管理是「工作區管理者」的功能。
- **前方面板**。雖然「前方面板」使用它自己的配置檔案，但它是由「工作區管理者」所建立與管理的。

許多的這些元件都可以用「樣式管理者」來變更。只要您用一點點努力，「樣式管理者」就能夠使常用的變更更加快速。其它的資源必須人工設定。

「工作區管理者」是 dtwm。它是以「Motif 視窗管理者」為基礎的。

- 有關「工作區管理者」的參考資訊，請參閱 dtwm(1) 與 dtwmrc(4) 線上援助頁。
- 有關設定「工作區管理者」資源的資訊，請參閱 第 257 頁“設定應用程式資源”。

- 有關「前方面板」配置檔案的資訊，請參閱 第 15 章。

有關設定資源的其它資訊，請參閱 第 257 頁“設定應用程式資源”。

工作區管理者配置檔案

「工作區管理者」取得有關來自配置檔案的視窗功能表、工作區功能表、按鈕組合與按鍵組合的資訊。

其使用下列檔案之一：

- 個人檔案：*HomeDirectory/.dt/dtwmrc*
- 系統自訂檔案：*/etc/dt/config/language/sys.dtwmrc*
- 內建檔案：*/usr/dt/config/language/sys.dtwmrc*

「工作區管理者」以如上所示的順序搜尋配置檔案，並使用它所找到的第一個檔案。

對於使用一種以上的階段作業語言使用者，一個個人且具語言相依性的配置檔案 *HomeDirectory/.dt/language/dtwmrc* 可被建立為在 *HomeDirectory/.dt/dtwmrc* 上有優先順序。

▼ 建立或修改個人的配置檔案

個人的「工作區管理者」配置檔案為 *HomeDirectory/.dt/dtwmrc*。如果這個檔案存在，就使用此檔案。

1. 連按兩下桌面工具應用程式群組中的「編輯 *Dtwmrc*」。

如果您已經有個人的 *dtwmrc* 檔案，其會載入編輯程式中。如果沒有的話，*sys.dtwmrc* 會複製至 *HomeDirectory/.dt/dtwmrc*，然後其會載入編輯程式中。

2. 編輯檔案。

3. 結束編輯程式。

檔案會儲存為您個人的 *dtwmrc*，不論其原始資源為何。

▼ 建立整個系統的配置檔案

整個系統的「工作區管理者」配置檔案為 */etc/dt/config/language/sys.dtwmrc*。

- 將 */usr/dt/config/language/sys.dtwmrc* 複製到 */etc/dt/config/language/sys.dtwmrc*。

註解 – 如果 *HomeDirectory* / *.dt/dtwmrc* 存在，則不使用此檔案。

▼ 包含 (併入資源) 其他檔案

- 使用語法：

```
include
{
  path
  path
  ...
}
```

例如，下列各行納入檔案 */users/ellen/mymenu*：

```
包括
{
  /users/ellen/mymenu
}
```

包含陳述式對於提供其它的功能而不需複製整個配置檔案是很有用的。例如，使用者可能會想要建立新的按鍵組合，而不需管理整個配置檔案。使用者可以用此內容建立檔案 *HomeDirectory/.dt/dtwmrc*：

```
include
{
  /etc/dt/config/C/sys.dtwmrc
}
Keys DtKeyBindings
{
  Alt<Key>F5 root f.menu Applications
}
Menu Applications
{
  "GraphicsApp" f.exec "/usr/bin/GraphicsApp/GApp"
  ...
}
```

▼ 重新啓動工作區管理者

「工作區管理者」必須重新啓動才能使對配置檔案所作的變更生效。

- 從「工作區」功能表選擇「重新啓動工作區管理者」（當指標在背景定位上時，請按滑鼠鈕 3）。

自訂工作區

大部分的工作區自訂，例如變更工作區名稱與工作區數目，都可以由使用者使用桌上管理系統的介面來完成。不過，「工作區管理者」會提供設定整個系統的預設資源。

▼ 以整個系統的基礎來變更工作區的數目

預設桌上管理系統配置提供四個工作區。使用者可以使用與「工作區」開關關聯的即現式功能表來增加與刪除工作區。

此 `/usr/dt/app-defaults/C/Dtwm` 檔案使 `workspaceCount` 資源設定為下列預設工作區數目：

```
Dtwm*0*workspaceCount: 4
Dtwm*workspaceCount: 1
```

螢幕 0 上指定多重工作區。單一工作區是指定於任何其它的螢幕上。

您可以建立（或如果其存在則修改）`/etc/dt/config/C/sys.resources` 檔案以為工作站上所有的新使用者來變更預設的工作區數目。

- 使用 `0*workspaceCount` 資源來設定主螢幕上整個系統的預設值：

```
Dtwm*0*workspaceCount: 數目
```

例如，以下資源將主螢幕上整個系統的工作區數目設定為六：

```
Dtwm*0*workspaceCount: 6
```

有關設定「工作區管理者」資源的資訊，請參閱第 257 頁“設定應用程式資源”。

例如，以下資源將工作區的數目設定為六：

```
Dtwm*workspaceCount: 6
```

▼ 提供整個系統的工作區名稱

剛開始時，工作區是以編號慣例 `wsn` 來編號的，其中 n 為 0、1、2 等等。例如，預設的四個工作區內部的編號為 `ws0` 到 `ws3`。

- 使用 `title` 資源來變更指定的工作區名稱：

```
Dtwm*wsn: name
```

有關設定「工作區管理者」資源的資訊，請參閱第 257 頁“設定應用程式資源”。

例如，下列資源將預設的四個工作區設定為指定的名稱：

```
Dtwm*ws0*title:  
Anna Dtwm*ws1*title:  
Don Dtwm*ws2*title:  
Julia Dtwm*ws3*title:  
Patti
```

▼ 建立其他背景定位

1. 建立背景定位影像。它們可以是點陣圖或像素對映檔案。
2. 將背景定位放在下列目錄之一中。（您可能會需要建立目錄。）
 - 整個系統的背景定位：`/etc/dt/backdrops`
 - 個人的背景定位：`HomeDirectory /.dt/backdrops`
3. 從「工作區」功能表選擇「重新啓動工作區管理者」。
整個系統與個人的背景定位是增加至 `/usr/dt/backdrops` 中的內建背景定位。
您可以用建立具有相同名稱的個人或整個系統的背景定位來取代現存內建背景定位。

▼ 用圖形影像取代背景定位

背景定位是位於顯示器的 `root` 視窗上之階層。「樣式管理者背景定位」對話框提供 `NoBackdrop` 設定，其中背景定位是透明的。

在所有的工作區背景定位後面只有一個 `root` 視窗。因此，放在 `root` 視窗上的圖形影像持續經過所有工作區。您可以指定哪些工作區以背景定位遮蓋 `root` 視窗。然而，當 `NoBackdrop` 實際上對於每個工作區都是相同的時，影像是可見的。

1. 建立圖形影像。
影像必須是其中有工具存在以在 `root` 視窗上顯示它的格式。例如，如果您意欲使用 `xsetroot`，您便必須建立點陣圖檔案。
2. 如果它並非已經存在，請建立可執行的檔案 `HomeDirectory /.dt/sessions/sessionetc`。
每次使用者登入時都會執行 `sessionetc` 檔案。
3. 放置指令以在 `sessionetc` 檔案中顯示影像。
例如，以下指令非重疊顯示具有指定點陣圖的 `root` 視窗：

```
xsetroot -bitmap /users/ellen/.dt/icons/root.bm
```

工作區管理者功能表

「工作區管理者」有三種預設功能表：

工作區功能表—也稱為 root 功能表。當指標在背景定位上時，會在使用者按滑鼠按鈕 3 時顯示出來。功能表經由按鈕組合與滑鼠按鈕關聯。

視窗功能表—當指標在「視窗」功能表按鈕（視窗框架的左上角）上時，功能表顯示於使用者按滑鼠按鈕 1 或滑鼠按鈕 3 時。功能表藉由 windowMenu 資源而與按鈕關聯。

前方面板功能表—當指標在「前方面板的視窗」功能表按鈕上方時，功能表顯示於使用者按滑鼠按鈕 1 或滑鼠按鈕 3 時。

工作區管理者功能表語法

「工作區管理者」功能表有語法：

```
功能表 MenuName
{
    selection1 [ mnemonic ] [加速鍵] 功能 [argument]
    selection2 [ mnemonic ] [加速鍵] 功能 [argument]
    ...
}
```

其中：

選擇—出現在功能表上的文字或點陣圖。如果文字包括空間，請將文字附在引號中。對於點陣圖，請使用語法 @/路徑。

mnemonic—當功能表顯示時，作為鍵盤捷徑的單一字元。它係以格式 **字元** 來指定。

加速鍵—不論功能表顯示與否，均為作用中的鍵盤捷徑。加速鍵有語法 **modifier<Key> 按鍵名稱** 其中修飾符號為 Ctrl、Shift、Alt (擴充字元) 或 Lock。有關所有可能的按鍵名稱清單，請參照您的 "X11 包括" 目錄中的 keysymdef.h 檔案。

功能—當作此選擇時，則執行功能。有關功能清單，請參照 dtwmrc(4) 線上援助頁。

argument—功能引數。有關更多的細節資訊，請參照 dtwmrc(4) 線上援助頁。

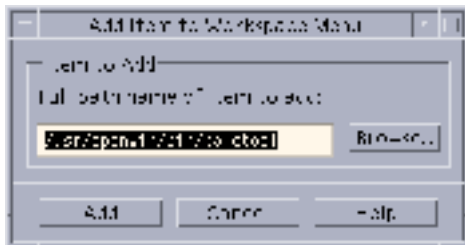
例如，以下加上「回復」標籤的功能表項目會使視窗一般化。當功能表顯示時，鍵入 "R" 也會回復視窗。按擴充字元 F5 也會回復視窗。

```
Restore    _R    Alt<Key> F5    f.normalize
```

註解—有關「工作區管理者」功能表語法的完整資訊，請參閱 dtwmrc(4) 線上援助頁。

▼ 新增功能表項目到工作區功能表

1. 按一下「前方面板」的「工具」輔助面板上的「增加項目至功能表」控制。
「增加項目至工作區功能表」對話框會顯示出來。

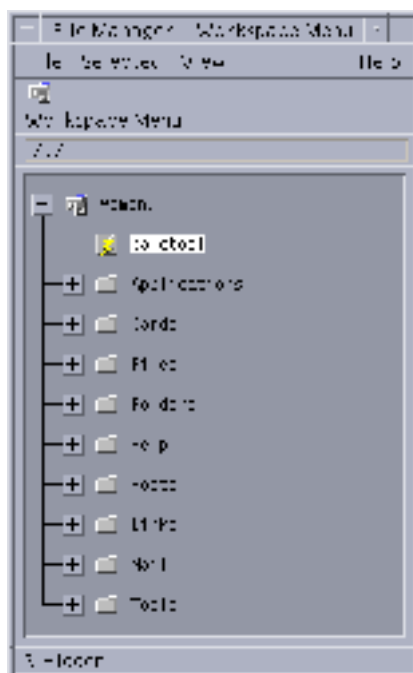


2. 鍵入完整路徑或按一下「瀏覽」，並選擇您想要新的功能表項目執行的檔案路徑。
您所指的檔案一般應是可供此主機使用的。此檔案應為註冊至主機的可執行檔案或資料類型的檔案（例如聲訊或線上援助頁）。
文字方塊中顯示包括檔案名稱的完整路徑。
3. 按一下「增加」以將檔案增加至功能表。
檔案增加至「工作區功能表」中的第一個位置。其預設圖示（如果有的話）與檔案名稱將會用作功能表項目的圖示與文字使用。

註解 – 如果要變更「工作區功能表」上的功能表項目的位置，請參閱 第 249 頁“修改工作區功能表”。

▼ 修改工作區功能表

1. 按一下「前方面板」的「工具」輔助面板上的「自訂工作區功能表」控制。
「檔案管理者」的顯示會顯示出「工作區功能表」的配置檔案夾。注意，此檔案夾中的檔案代表每個功能表項目，而子檔案夾代表每個子功能表。藉由重新安排「工作區功能表」檔案夾的內容，您可重新安排「工作區功能表」。



2. 將任何您希望移動的功能表項目拖曳及定位至此子檔案夾下的新位置。
有關使用「樹狀查看」模式中的「檔案管理者」的輔助說明，請參閱 *Solaris 共用桌上管理系統環境：使用者指南*。
3. 藉由將其檔案拖曳至「前面板」上的檔案垃圾桶，以刪除任何不想要的功能表項目。
4. 藉由編輯其檔案或檔案夾名稱來重新命名任何您想要重新命名的功能表項目。
5. 當您對您所作的變更感到滿意時，請從「檔案管理者」中的「檔案」功能表選擇「更新工作區功能表」，並結束「檔案管理者」。
現在「工作區功能表」會反映出您在「工作區功能」表檔案夾中所作的變更。

▼ 建立新的工作區 (根) 功能表

1. 開啓合適的檔案以編輯：
 - 個人的：`HomeDirectory /.dt/dtwmrc`
 - 整個系統的：`/etc/dt/config/language /sys.dtwmrc`
有關建立這些檔案的資訊，請參閱 第 244 頁“工作區管理者配置檔案”。
2. 建立新的功能表：

```
Menu menu_name
{
    ...
}
```

請參閱 第 248 頁“工作區管理者功能表語法”。

3. 建立或編輯按鈕組合以顯示新的功能表。

如果功能表取代現存功能表，請編輯顯示「工作區功能表」的按鈕組合。

```
<Btn3Down> root f.menu menu_name
```

如果功能表是附加的功能表，請建立新的滑鼠按鈕組合。例如，當在背景定位之上按滑鼠鈕 3 時，以下的按鈕組合會顯示功能表：

```
Shift<Btn3Down> root f.menu menu_name
```

4. 從「工作區」功能表選擇「重新啟動工作區管理者」。

▼ 建立新的視窗功能表

註解 – 「視窗」功能表建於「工作區管理者」中，而且一般都不是自訂。如果要保持視窗行為在應用程式之間的一致，您應該避免「視窗」功能表的大規模修改。

1. 開啓合適的檔案以編輯：

- 個人的：`HomeDirectory / .dt/dtwmrc`
 - 整個系統的：`/etc/dt/config/language / sys.dtwmrc`
- 有關建立這些檔案的資訊，請參閱 第 244 頁“工作區管理者配置檔案”。

2. 建立新的功能表：

```
Menu menu_name
{
    ...
}
```

3. 使用windowMenu 資源來指定新的功能表：

```
Dtwm*windowMenu: menu_name
```

4. 從「工作區」功能表選擇「重新啟動工作區管理者」。

自訂按鈕組合

按鈕組合關聯滑鼠按鈕作業與具有視窗管理者功能的可能鍵盤修飾碼。按鈕組合應用於所有的工作區。

在工作區管理者配置檔案中以名為 `DtButtonBindings` 的按鈕組合集來定義。

```
Buttons DtButtonBindings
{
    ...
}
```

按鈕組合語法

按鈕組合的語法為：

```
Buttons ButtonBindingSetName
{
    [modifier] < button_nameMouse_action> context function [argument]
    [modifier] < button_nameMouse_action> context function [argument]
```

其中：

button_name—`Btn1`—左邊滑鼠按鈕 `Btn2`—中間按鈕 (3 個按鈕的滑鼠) 或兩個按鈕 (2 個按鈕的滑鼠) `Btn3`—右邊按鈕 `Btn4`—在 3 個按鈕上按鈕 1 及按鈕 2 合在一起 `Btn5`—
在 3 個按鈕上按鈕 2 及按鈕 3 合在一起

modifier—`Ctrl`, `Shift`, `Alt`, `Lock`

mouse_action—`Down`—按住滑鼠；按鈕 `Up`—釋放滑鼠按鈕；`Click`—按住並釋放滑鼠按鈕；`Click2`—連按兩下滑鼠按鈕；`Drag`—當按住滑鼠按鈕時拖曳滑鼠

context—指示要使組合有效，指標必須放在何處。如果有必要，以 "|" 字元分開多重內容。

root—工作區視窗；*window*—用戶端視窗或視窗框架；*frame*—視窗框架, 不包括內容；*icon*—圖示；*title*—標題欄；*app*—用戶端視窗 (不包括框架)

function—視窗管理者功能之一。有關有效功能的清單，請參照 `dtwmrc(4)` 線上援助頁。

argument—所需的任何視窗管理者功能引數。有關細節資訊，請參照 `dtwmrc(4)` 線上援助頁。

例如，當指標在工作區視窗（但不在用戶端視窗中）之時按滑鼠按鈕 3，下行會使 `DtRootMenu` 中所描述的功能表顯示出來的。

```
<Btn3Down>      root      f.menu      DtRootMenu
```

註解 – 有關按鈕組合語法的完整資訊，請參閱 dtwmrc(4) 線上援助頁。

▼ 增加按鈕組合

1. 開啓合適的檔案以編輯：
 - 個人的：`HomeDirectory /.dt/dtwmrc`
 - 整個系統的：`/etc/dt/config/language /sys.dtwmrc`
有關建立這些檔案的資訊，請參閱 第 244 頁“工作區管理者配置檔案”。
2. 增加按鈕組合至 `DtButtonBindings` 定義。
請勿將相同的按鈕連結到不同的按一下與按的作業功能，並勿將一個以上的功能連結到相同的按鈕與上下文。
3. 從「工作區」功能表選擇「重新啓動工作區管理者」。

▼ 建立新的按鈕組合集

1. 開啓合適的檔案以編輯：
 - 個人的：`HomeDirectory /.dt/dtwmrc`
 - 整個系統的：`/etc/dt/config/language /sys.dtwmrc`
有關建立這些檔案的資訊，請參閱 第 244 頁“工作區管理者配置檔案”。
2. 建立新的按鈕組合集。請參閱 第 252 頁“按鈕組合語法”。
3. 將 `buttonBindings` 資源設定為新的名稱：

```
Dtwm*buttonBindings: ButtonBindingsSetName
```
4. 從「工作區」功能表選擇「重新啓動工作區管理者」。

註解 – 新的按鈕組合會取代您的現存按鈕組合。從 `DtButtonBindings` 複製您想保存的按鍵組合。

自訂按鍵組合

鍵盤組合也稱為**按鍵組合**，它是將「工作區管理者」功能和按鍵組合相關聯。按鍵組合應用於所有的工作區。

註解 – 關於使用共用按鍵組合作為鍵盤組合方面，請小心。例如，Shift-A 通常會在您目前的視窗中加入字母「A」。如果您將 Shift-A 連結至一項功能，您會失去其通常用法。

預設桌上管理系統按鍵組合

桌上管理系統預設按鍵組合在工作區管理者配置檔案中，以名為 `DtKeyBindings` 的按鈕組合集來定義。

```
Keys DtKeyBindings
{
    ...
}
```

按鍵組合語法

按鍵組合的語法為：

```
Keys KeyBindingSetName
{
    [Modifiers] <Key>key_name context function [argument]
    [Modifiers] <Key>key_name context function [argument]
    ...
}
```

其中：

Modifiers—Ctrl, Shift, Alt 及 Lock。允許多重修飾碼；以空格分開它們。

key_name—功能與其對映的按鍵。對於具有字母或數字的按鍵，*key_name* 名稱通常列印在按鍵上。例如「a」鍵的名稱為「a」，而「2」鍵名為「2」。「Tab」鍵名為「Tab」。「F3」鍵名為「F3」。

對於其它的按鍵，則是將名稱拼出來—例如，「+」鍵的 `plus`。位於系統相依目錄中的檔案 `keysymdef.h` 包含有關按鍵名稱的其它資訊。

context—必須有鍵盤焦點以使此動作有效的元素。如果組合應用於一項以上的上下文，這些可以串接在一起。多重上下文由 "|" 字元來分開。

root—工作區背景定位；*window*—用戶端視窗；*icon*—圖示

function—視窗管理者功能。有關有效功能的清單，請參照 dtwmrc(4) 線上援助頁。

argument—所需的任何視窗管理者功能引數。有關細節資訊，請參照 dtwmrc(4) 線上援助頁。

例如，以下的按鍵組合讓使用者藉由按 Alt+F6 將鍵盤焦點轉換至應用程式中的下一暫態視窗。

```
Alt<Key>F6      window      f.next_key      transient
```

註解 – 有關按鍵組合語法的完整資訊，請參閱 dtwmrc(4) 線上援助頁。

▼ 建立自訂的按鍵組合集

1. 開啓合適的檔案以編輯：
 - 個人的：`HomeDirectory/.dt/dtwmrc`
 - 整個系統的：`/etc/dt/config/language/sys.dtwmrc`
有關建立這些檔案的資訊，請參閱 第 244 頁“工作區管理者配置檔案”。
2. 用唯一的 *KeyBindingSetName*。建立新的按鍵組合。使用桌上管理系統預設按鍵組合組 `DtKeyBindings` 作為指南。
3. 將 `keyBindings` 資源設定新集名稱：

```
Dtwn*keyBindings: KeyBindingSetName
```
4. 從「工作區」功能表選擇「重新啓動工作區管理者」。

註解 – 新的按鍵組合取代了您的現存按鍵組合。從 `DtKeyBindings` 將任何您想要保存的按鍵組合複製到您的新集中。

在預設與自訂行為之間切換

如果要在 Motif 預設與 CDE 桌上管理系統視窗行為之間切換：

1. 按 **Alt+Shift+Ctrl+!**
2. 按一下對話框中的「確定」。
轉換至預設行為會移除「前方面板」以及任何自訂按鍵與按鈕組合。

第 17 章

管理應用程式資源、字型與色彩

您可以藉由使用「樣式管理者」或是藉由自訂其它字型與色彩資源來為您的顯示器選擇範圍很廣的色彩與字型。本章討論如何自訂字型與色彩資源。

本章也描述如何指定 DtEditor widget 應用程式的樣式轉換，例如桌上管理系統文字編輯程式 (dtpad) 與郵件程式 (dtmail) 以及與這些轉換衝突的 DtEditor widget 應用程式功能表加速鍵的替代鍵。

- 第 257 頁“設定應用程式資源”
- 第 258 頁“定義 UNIX 連結”
- 第 262 頁“管理字型”
- 第 267 頁“管理色彩”
- 第 274 頁“設定應用程式視窗的陰影濃度”

設定應用程式資源

應用程式使用資源來設定外觀與行為的特定方面。例如，「樣式管理者」(dtstyle) 提供使您能夠指定系統於何處尋找包含有關色彩調色板資訊的檔案的資源：

```
dtstyle*paletteDirectories: /usr/dt/palettes/C HomeDirectory /.dt/palettes
```

桌上管理系統應用程式的 App-default 檔案位於 /usr/dt/app-defaults/language 目錄中。

▼ 設定整個系統的資源

- 增加資源至檔案 /etc/dt/config/language/sys.resources。(您可能必須建立此檔案。)

例如，如果您在 /etc/dt/config/C/sys.resources 中，可以指定：

`AnApplication*resource: value`

則下次登入時資源 `AnApplication*resource` 將會設定在每一個使用者的 `RESOURCE_MANAGER` 屬性中。

註解 – 有關「樣式管理者」資源的更多資訊，請參閱 `dtstyle` 線上援助頁。有關「郵件程式」資源的更多資訊，請參閱 `dtmail` 線上援助頁。

▼ 設定個人資源

1. 增加資源至檔案 `HomeDirectory/.Xdefaults`。
2. 連按兩下桌面工具應用程式群組中的「重新載入資源」。

桌上管理系統如何載入資源

資源於階段作業啟動時由「階段作業管理者」所載入。有關「階段作業管理者」如何將資源載入 `RESOURCE_MANAGER` 的資訊，請參閱 第 42 頁“載入階段作業資源”。

處理管理者資源

下列為可供使用的「處理管理者資源」：

- `sampleNowTR`
- `postPopupMenuTR`
- `selectNextProcessTR`
- `selectPrevProcessTR`
- `selectFirstProcessTR`
- `selectLastProcessTR`
- `killSelectedProcessTR`

定義 UNIX 連結

按預設值，不啓用 UNIX 組合。

▼ 指定 EMACS 樣式轉換

以下程序指定：

- dtmail widget 應用程式的 EMACS 樣式轉換，如桌上管理系統文字編輯程式 (dtpad) 及郵件程式 (dtmail)
- 替代與這些轉換衝突的 DtEditor widget 應用程式功能表加速鍵。

1. 增加下行至 *HomeDirectory* / *.Xdefaults* 檔案：

```
#include "/usr/dt/app-defaults/language/UNIXbindings"
```

其中 *language* (語言) 是 LANG 環境變數。

2. 重新啟動您的階段作業。

▼ 如果要修改 EMACS 樣式轉換

1. 將檔案 */usr/dt/app-defaults/language/UNIXbindings* 的內容插入 *HomeDirectory* / *.Xdefaults*。
2. 編輯 *.Xdefaults* 檔案中的組合。
3. 當您完成時，重新啟動您的階段作業。

由 UNIXbindings 檔案所提供的 UNIX 連結

/usr/dt/app-defaults/language/UNIXbindings 檔案提供如下表所述的組合。

註解 – 啓用 UNIX 組合時，「刪除」鍵會刪除前一字元，而 Shift-Delete 會刪除下一字元。

表 17-1 列出與 UNIX 組合衝突的功能表加速鍵與加速鍵文字的 dtpad 置換。

表 17-1 dtpad 置換

功能表加速鍵與加速鍵文字	置換
Dtpad*fileMenu.print.acceleratorText:	
Dtpad*fileMenu.print.accelerator:	
Dtpad*editMenu.undo.acceleratorText:	Ctrl+_
Dtpad*editMenu.undo.accelerator:	Ctrl<Key>_
Dtpad*editMenu.paste.acceleratorText:	Shift+Insert
Dtpad*editMenu.paste.accelerator:	Shift<Key>osfInsert

表 17-1 dtpad 置換 (繼續)

功能表加速鍵與加速鍵文字	置換
Dtpad*editMenu.findChange.acceleratorText:	Ctrl+S
Dtpad*editMenu.findChange.accelerator:	Ctrl<Key>s

表 17-2 列出與 UNIX 組合衝突的功能表加速鍵與加速鍵文字的 dtmail 「撰寫」視窗置換。

表 17-2 dtmail 撰寫視窗置換

功能表加速鍵與加速鍵文字	置換
Dtmail*ComposeDialog*menubar*Edit.Undo.acceleratorText:	Ctrl+_
Dtmail*ComposeDialog*menubar*Edit.Undo.accelerator:	Ctrl<Key>_
Dtmail*ComposeDialog*menubar*Edit.Paste.acceleratorText:	Shift+Insert
Dtmail*ComposeDialog*menubar*Edit.Paste.accelerator:	Shift<Key>osfInsert
Dtmail*ComposeDialog*menubar*Edit.Find/Change.acceleratorText:	Ctrl+S
Dtmail*ComposeDialog*menubar*Edit.Find/Change.accelerator:	Ctrl<Key>s

下列的轉換提供 (GNU 樣式) EMACS 控制與複合按鍵組合加上一些其它的組合。合適時, 它們也允許使用 Shift 鍵與一般組合做組合, 以回轉作業的方向。例如, 因為 Ctrl+F 通常會往前移動一個字元, Ctrl+Shift+F 就將會使游標往後移動一個字元。

其它的組合為:

Ctrl+逗點—向後移動字

Ctrl+Shift+逗點—向前移動一字

Ctrl+句點—向前移動一字

Ctrl+Shift+句點—向後移動一字

Ctrl+Return—檔案結尾

Ctrl+Shift+Return—檔案開頭

GNU EMACS 將刪除前一字元()而非刪除下一字元()連結至「刪除」鍵。Meta+F 通常是「檔案」功能表的助憶, 因此將會忽略向前移動字()。使用其它向前移動字的組合之一(例如, Ctrl+句點)。

表 17-3 列出 DtEditor.text 轉換

表 17-3 DtEditor.text 轉換

修飾按鍵	按鍵	動作常式
c ~s	<Key>a:	行的開頭 ()\n\
c s	<Key>a:	行的結束 ()\n\
c ~s	<Key>b:	向後移動一字 ()\n\
c s	<Key>b:	向前移動一字 ()\n\
c ~s	<Key>b:	向後移動一字 ()\n\
c s	<Key>b:	向後移動一字 ()\n\
m ~s	<Key>b:	向後移動一字 ()\n\
m s	<Key>b:	向前移動一字 ()\n\
c ~s	<Key>d:	刪除下一字元 ()\n\
c s	<Key>d:	刪除前一字元 ()\n\
m ~s	<Key>d:	清除下一字 ()\n\
m s	<Key>d:	清除前一字 ()\n\
c ~s	<Key>e:	行的結束 ()\n\
c s	<Key>e:	行的開頭 ()\n\
c ~s	<Key>f:	向前移動一字 ()\n\
c s	<Key>f:	向後移動一字 ()\n\
m ~s	<Key>f:	向前移動一字 ()\n\
m s	<Key>f:	向後移動一字 ()\n\
c	<Key>j:	換行與縮排 ()\n\
c ~s	<Key>k:	清除至行的結束 ()\n\
c s	<Key>k:	清除至行的開始 ()\n\
c	<Key>l:	重畫顯示 ()\n\
c	<Key>m:	換行 ()\n\
c s	<Key>n:	往上處理 ()\n\
c ~s	<Key>n:	往下處理 ()\n\
c	<Key>o:	換行與備份 ()\n\
c ~s	<Key>p:	往上處理 ()\n\
c s	<Key>p:	往下處理 ()\n\

表 17-3 DtEditor.text 轉換 (繼續)

修飾按鍵	按鍵	動作常式
c ~s	<Key>u:	清除至行的開始 ()\n\
c s	<Key>u:	清除至行的結束 ()\n\
c ~s	<Key>v:	下一頁 ()\n\
c s	<Key>v:	前一頁 ()\n\
m ~s	<Key>v:	前一頁 ()\n\
m s	<Key>v:	下一頁 ()\n\
c	<Key>w:	清除選擇 ()\n\
c ~s	<Key>y:	取消清除 ()\n\
m	<Key>]:	向前移動一段 ()\n\
m	<Key>[:	向後移動一段 ()\n\
c ~s	<Key>comma:	向後移動一字 ()\n\
c s	<Key>comma:	向前移動一字 ()\n\
m	<Key>\\<:	檔案開始 ()\n\
c ~s	<Key>句點 :	向前移動一字 ()\n\
c s	<Key>句點 :	向後移動一字 ()\n\
m	<Key>\\>:	檔案結束 ()\n\
c ~s	<Key>Return:	檔案結束 ()\n\
c s	<Key>Return:	檔案開始 ()\n\
~c ~s ~m ~a	<Key>osfDelete:	刪除前一字元 ()\n\
~c s ~m ~a	<Key>osfDelete:	刪除下一字元 ()

管理字型

使用「樣式管理者字型」對話方塊，您可以為所有的應用程式選擇您想要的字型群組與大小。您也可以指令行上指定字型或使用資源以：

- 設定個別應用程式的字型資源
- 由「字型」對話方塊來指派要使用的不同字型

字型 是列印或顯示文字字元的類型樣式。桌上管理系統包括各種不同的樣式與大小的字型。

點陣圖字型 是由點矩陣所製造的。（按預設值，「樣式管理者」只配置點陣圖字型。）字型完全包含在一份檔案中。需要有許多檔案才能有完整範圍的大小、傾斜與重量。

字型被指定為資源的值以及指令的參數。「X 邏輯字型說明」（XLFD）名稱是要求想要的字型的方法。

設定桌上管理系統字型資源

「樣式管理者字型」對話方塊使您能夠選擇例如文字登錄與標籤的字型（最多七種大小）。它也使您增加或刪除字型群組。

由字型對話方塊所設定的資源

當選擇字型時，下列資源會寫入至 RESOURCE_MANAGER 屬性：

- SystemFont 是用於系統區域，例如功能表列、功能表窗格、按下按鈕、切換按鈕與標籤。以下資源是由 SystemFont 所設定的：
 - *FontList—顯示於使用 OSF/Motif 工具箱所建立的桌上管理系統用戶端與其它用戶端的系統區域中。
- UserFont 是用於輸入視窗中的文字。下列資源是由 UserFont 所設定的：
 - *字型—支援 X 應用程式的較早版本
 - *FontSet—主要的設定
 - *XmText*FontList—顯示於文字登錄方塊中
 - *XmTextField*FontList—顯示於文字登錄方塊中

由字型對話方塊所使用的資源

在 /usr/dt/app-defaults/Dtstyle 資源檔案中指定了「字型」對話方塊中用於每項選擇的字型。最多可以指定七種大小。

NumFonts—「字型」對話方塊中字型大小的數目

SystemFont [1-7]—指派特定字型至 SystemFont 的「字型」對話方塊選擇最多可到七種資源

UserFont [1-7]—最多有 7 個資源指定特定字型給字型對話方塊供 UserFont 選擇

註解 – 的「字型」對話方塊選擇最多可到七種資源 這些資源的預設字型已為不同顯示器的可讀性所選出。如果您想要應用程式的特定字型，請設定具有應用程式字型資源的字型而非變更這些桌上管理系統的字型。

有關應用程式字型的更多資訊，請參閱 DtStdAppFontNames(5) 及 DtStdInterfaceFontNames(5) 線上援助頁。

▼ 列出可用字型

1. 鍵入如下：

```
xlsfonts [-選項] [-fn 型樣]
```

會顯示您系統上可供使用的 XLFD 名稱與字型別名名稱的清單。點陣圖字型顯示在所有十四個 XLFD 欄位中的值。可測量的字體在 像素大小, 點大小, 解析度X與 解析度Y 位置中顯示為零。

2. 如果要檢查特定的字型，請使用 `xlsfonts` 的型樣符合的能力。使用通配符號來取代您並不試圖符合的部分型樣。
3. 如果 `xlsfonts` 未顯示任何以 `dt` 開始的字型名稱，您的字型路徑就不包括桌上管理系統字型。鍵如以下指令來將桌上管理系統字型包含在您可供使用的字型中：

```
xset +fp directory name
```

其 `directory name` 是包含桌上管理系統字型的目錄。由階段作業啟動所設定的預設位置為 `/usr/dt/config/xfonts/language`。

有關其它的資訊：

- `xset` 與 `xlsfonts` 線上援助頁列出可供使用的選項。
- 使用 X 視窗系統 來解釋字型別名名稱與 `xset` 用戶端。

▼ 如果要指定指令行上的字型

- 使用 `-xrm` 指令行選項來指定特定用戶端的字型資源。例如：

```
應用程式名稱 -xrm "*bitstream-charter-medium-r-normal-8-88-75-75-p-45-iso8859-1"
```

X 邏輯字型說明 (XLFD)

藉由列出十四種不同字元所指定的字型是由破折號 (-) 所分開的。此稱為「X 邏輯字型說明」(XLFD)。在某些情況中，清單中的屬性可以由 * 通配符號所取代，而屬性中的字元可以由 ? 通配符號來取代。表 17-4 列出字型屬性字串規格。

屬性字串規格的格式為：

```
"-Foundry- FamilyName- WeightName- Slant- SetwidthName- AddStyleName- 像素大小- 點大小- 解析度 X- 解析度 Y- Spacing- AverageWidth- CharSetRegistry- CharSetCoding"
```

表 17-4 字型屬性字串規格

屬性字串	定義
<i>Foundry</i>	識別字型設計者的字串

表 17-4 字型屬性字串規格 (繼續)

屬性字串	定義
<i>FamilyName</i>	識別字型商標名稱的字串
<i>WeightName</i>	給予字型的相對重量的字串，例如粗體
<i>Slant</i>	描述傾斜方向的代碼： R (羅馬-無傾斜) I (斜體-向右傾斜) O (傾斜-向右傾斜) RI (反斜體-向左傾斜) RO (反傾斜-向左傾斜)
<i>SetwidthName</i>	描述寬度的字串，例如壓縮的或擴充的
<i>AddStyleName</i>	提供唯一地識別字型所需要的任何其它資訊
像素大小	以像素計的 <i>em-square</i> 大小的整數
點大小	以小數點計的 <i>em-square</i> 大小的整數
解析度 X	以像素計的水平解析度的整數
解析度 Y	以像素計的垂直解析度的整數
<i>Spacing</i>	指定單元之間的空間設定代碼： M (Monospace-固定間距) P (比例空格-變數間距) C (字元儲存格)
<i>AverageWidth</i>	以 1/10 像素計的平均寬度的整數
<i>CharSetRegistry</i>	識別已註冊字型編碼的註冊授權字串
字元設定編碼	識別指定註冊中所設定的字元集的字串

範例

以下 XLF 名稱描述由支援 ISO8859-1 標準編碼的「位元串流」所製作的字型，其名稱為 *charter*：

```
-bitstream-charter-medium-r-normal--8-80-75-75-p-45-iso8859-1
```

其重量為中型，沒有特殊傾斜，且寬度一般。字型是以 8 像素或 8.0 點的 *em-square* 方塊而成比例的。水平與垂直解析度都是 75 像素。字元的平均寬度為 45 1/10 像素或 4.5 像素。

此字串的部分可由通配符號來取代。系統會使用其找到符合您所指定的部分的第一種字型。

如果您只是想要八像素的 charter 字體，您可以使用：

```
*-charter-*-*-*-8-*
```

查看選取字型的群組屬性

藉由從「樣式管理者字型」對話方塊選擇「屬性」按鈕，您可以查看下列的字型群組屬性：

- 字型群組
- 大小
- 別名
- 別名 XLFD
- 別名位置
- 字型
- 字型 XLFD

使用者的字型群組檔案系統儲存

當使用者增加字型群組時，新增的群組會儲存於以下位置中：

```
HomeDirectory / .dt/sdtfonts/host/locale/ typeface-nnnnnn
```

host 是本機工作站的主機名稱。

locale 是使用者目前的語言環境，例如 “C” 或 “ja”

typeface-nnnnnn 是由選取字型與所產生的唯一數目所得來的名稱。

此字型群組目錄包含下列三個檔案

- fonts.alias
- fonts.dir
- sdtfonts.group

fonts.alias 與 fonts.dir 檔案是一般的 X11 字型檔案，其適於增加至 Xserver 的字型路徑。sdtfonts.group 檔案包含使用者指定的字型群組名稱。

系統管理者字型群組建立

因此其它使用者可以存取工作站上的字型群組，系統管理者可以將字型群組複製到 /etc/dt/sdtfonts/locale 目錄或到 /usr/openwin/lib/X11/stdfonts/locale 目錄中。「階段作業管理者」先查看 HomeDirectory / .dt/stdfonts/host/locale，然後是 /etc/dt/sdtfonts/locale，最後才是 /usr/openwin/lib/X11/stdfonts/locale。

管理色彩

本節描述：

- 「樣式管理者」如何設定顯示器色彩。
- 由「樣式管理者」所使用來控制桌上管理系統色彩用法的資源。

色彩調色板

調色板是由一群色彩集所組成的。目前調色板的色彩集顯示於「樣式管理者色彩」對話方塊中

每個調色板都存在一個檔案。此 `paletteDirectories` 資源指定包含調色板檔案的目錄。按預設值，此資源包含：

- 內建調色板：`/usr/dt/palettes`
- 整個系統的調色板：`/etc/dt/palettes`
- 個人調色板：`HomeDirectory /.dt/palettes`

色彩集

目前的調色板中的每個色彩集都是由「樣式管理者色彩」對話方塊中的色彩按鈕所代表。每個色彩都由色彩集 ID 來識別—從 1 到 8 的數字。



圖 17-1 HIGH_COLOR 的色彩集 ID 值

每個色彩集最多是由五種色彩所構成的。每個色彩按鈕顯示色彩集的背景色彩。每個色彩集中的五種色彩代表下列的顯示器元件資源：

前景—應用程式視窗或視窗框架的前景。其永遠是黑色或白色的。此通常是用於視窗與標題內的文字。

背景—應用程式或視窗框架的背景。

上方陰影色彩—應用程式控制（例如按下按鈕）與視窗框架的頂端與左邊斜角的色彩。

下方陰影色彩—應用程式控制與視窗框架的底部與右邊斜角的色彩。

選擇按鈕—指示特定控制作用中狀態的色彩，例如作用中的切換與按鈕。

由每個調色板所使用的色彩集數目是由 colorUse 資源所決定的，使用者可以使用「樣式管理者使用的色彩數目」對話方塊來設定它。

指定色彩值

撰寫其調色板檔案的色彩資訊時，「樣式管理者」使用 RGB 值。RGB 數字的語法為：

```
# RedGreenBlue
```

紅色、**綠色**與**藍色**是十六進位數字，每個長度為 1 到 4 個數字，其指出該色彩的使用總數。每個色彩都必須要有相同數目的數字。因此，有效的色彩值是由 3、6、9 或 12 個十六進位數字所組成的。

例如，白色可以用下列任何一種方法來指定：

```
#fff
#ffffff
#ffffffff
#ffffffffffff
```

如果您直接設定色彩資源，您可以使用色彩名稱或是 RGB 值任一種。檔案 `/usr/lib/X11/rgb.txt` 列出所有命名的色彩。

色彩集如何對映至資源

桌上管理系統經由資源將色彩集對映至各種不同的顯示元素，並作出如表 17-5 所示的指定。

表 17-5 將色彩集對映至資源

資源	顯示元素
<code>activeColorSetId</code>	作用中的視窗框架色彩
<code>inactiveColorSetId</code>	非作用中的視窗框架色彩
<code>textColorSetId</code>	文字登錄區域
<code>primaryColorSetId</code>	應用程式的主要背景區域
<code>secondaryColorSetId</code>	應用程式的功能表列、功能表與對話方塊

這些資源取色彩集 ID 為其值。以「樣式管理者」選取新的調色板時，具有色彩集 ID 的顯示元素會允許元素動態變更成新的色彩計畫。

您可以使用這些個別應用程式的資源。例如，下行顯示您如何藉由使用其主要色彩的色彩集 8 來以視覺分組所有的 `dtterm` 視窗。

```
dtterm*primaryColorSetId: 8
```

預設色彩集指定

用於顯示元素色彩集 ID 取決於「樣式管理者」中的色彩數目設定：

表 17-6 顯示高色彩（8 色彩集）的色彩集 ID — 「樣式管理者」設定「桌上管理系統」的「更多色彩」。

表 17-6 高色彩

色彩集 ID	顯示元素
1	作用中的視窗框架色彩

表 17-6 高色彩 (繼續)

色彩集 ID	顯示元素
2	非作用中的視窗框架色彩
3	未使用 (按預設值)
4	文字登錄區域
5	應用程式的主要背景區域
6	應用程式的功能表列、功能表與對話方塊
7	按預設值未使用
8	前方面板背景

表 17-7 顯示中色彩 (4 色彩集) 的色彩集 ID—「樣式管理者」設定「應用程式」的「更多色彩」。

表 17-7 中色彩

色彩集 ID	顯示元素
1	作用中的視窗框架色彩
2	非作用中的視窗框架色彩
3	應用程式與前方面板背景色彩
4	文字登錄區域

表 17-8 顯示低色彩 (2 色彩集) 的色彩 ID—「樣式管理者」設定「應用程式」的「大部分色彩」。

表 17-8 低色彩

色彩集 ID	顯示元素
1	作用中的視窗框架，工作區選擇按鈕
2	所有其它的顯示元素

以樣式管理者控制色彩

您可以經由「樣式管理者」動態變更桌上管理系統應用程式與其它合作的應用程式的色彩。由「樣式管理者」所設定的前景與背景可供非合作的應用程式使用。

如果要用戶端回應「樣式管理者」的色彩變更，用戶端必須使用桌上管理系統 Motif 程式庫。以其它工具箱撰寫的用戶端無法動態變更色彩以回應「樣式管理者」的變更。這些用戶端的色彩變更要到用戶端重新啟動時才會生效。

必定沒有其它用戶端應用的特定色彩資源。此包括使用者指定的資源、appdefaults 與建於應用程式內的資源。

用戶端可以指定 primaryColorSetId 與 secondaryColorSetId 資源來使用桌上管理系統調色板內的特定色彩。

由樣式管理者所使用的色彩數目

由「樣式管理者」所使用的色彩數目取決於下列資源的值：

colorUse—配置桌上管理系統使用的色彩數目

shadowPixmaps—指示桌上管理系統以像素對映來取代兩種陰影色彩

foregroundColor—指定前景色彩是否要動態變更

dynamicColor—當您轉換調色板時，控制應用程式是否要變更色彩

表 17-9 列出由桌上管理系統所分配的色彩的最大數目。

表 17-9 桌上管理系統色彩的數目

顯示器	色彩的最大數目	數目得自
B_W	2	黑色與白色
LOW_COLOR	12	兩個色彩集乘以五種色彩加上黑色與白色
MEDIUM_COLOR	22	四個色彩集乘以五種色彩加上黑色與白色
HIGH_COLOR	42	八個色彩集乘以五種色彩加上黑色與白色

如果要決定色彩的最大數目：

1. 調色板中色彩集的數目乘以每個色彩集內的色彩數目。
2. 加 2（黑色與白色）。

但是，以下的配置，您的調色板中會只有十種色彩：四個色彩集乘以每集中的兩種色彩（背景 與 selectColor）加上黑色與白色：

```
*colorUse: MEDIUM_COLOR *shadowPixmaps: 正確 *foregroundColor: 白色
```

註解 – 多重色彩圖示使用十四種其它的色彩。

colorUse 資源

colorUse 資源的預設值為 MEDIUM_COLOR。此資源的值影響到調色板中所使用的色彩集數目。其它的資源影響到用於製造陰影的色彩數目。colorUse 資源的值也影響到多重色彩圖示的使用。

值	說明
B_W	“黑色與白色”「樣式管理者」設定 以 1 到 3 個色彩平面的顯示器色彩集的數目：2 色彩的最大數目：2 色彩的預設數目：2 無多重色彩圖示
LOW_COLOR	“應用程式的大部分色彩”「樣式管理者」設定 以 4 到 5 個色彩平面的顯示器色彩集的數目：2 色彩的最大數目：12 色彩的預設數目：12 無多重色彩圖示
MEDIUM_COLOR	“應用程式的較多色彩”「樣式管理者」設定 以 6 個色彩平面的顯示器色彩集的數目：4 色彩的最大數目：22 色彩的預設數目：22 多重色彩圖示
HIGH_COLOR	“桌上管理系統的較多色彩”「樣式管理者」設定 以 7 個或以上的色彩平面的顯示器色彩集的數目：8 色彩的最大數目：42 色彩的預設數目：42 多重色彩圖示
預設	桌上管理系統為該顯示器選擇正確的值。（如果要減少由高色彩顯示器的桌上管理系統所使用的色彩數目，請將預設 colorUse 資源設定為 MEDIUM_COLOR。）

shadowPixmap 資源

shadowPixmap 資源指示桌上管理系統以像素對映來取代兩個陰影色彩。這些像素對映將背景色彩與黑色或白色混合，以模擬頂端或底部的陰影值。既然陰影色彩不需要配置色彩儲存格，此舉會將所需色彩的數目減少兩種。

值	說明
正確	桌上管理系統建立 topShadowPixmap 與 bottomShadowPixmap 來代替陰影色彩使用。
錯誤	使用來自調色板的 topShadowColor 與 bottomShadowColor。

shadowPixmap 的預設值取決於您所擁有的 colorUse 資源以及顯示器的硬體支援。

foregroundColor 資源

此foregroundColor 資源指定如何在調色板中配置前景。

設定	結果
白色	前景設定為白色。
黑色	前景設定為黑色。
動態（預設）	前景動態設定為黑色或白色，視 background 的值而定。例如，白色的字在黃色的背景上是很難閱讀的，因此系統會選擇黑色。

如果 foregroundColor 設定為黑色或是白色，色彩集中的色彩數目會減少一種，而前景將不會變更，以回應背景色彩中的變更。

foregroundColor 的預設值為「動態」，除了其中 colorUse 的值為 B_W。

dynamicColor 資源

此 dynamicColor 資源控制應用程式是否會動態變更色彩；也就是當您轉換調色板時，用戶端是否會變更色彩。

值	說明
正確	當選擇新的調色板時，用戶端會動態變更色彩。此為預設值。
錯誤	用戶端不會動態變更色彩。選擇新的調色板時，用戶端將會在重新啟動階段作業時使用新的色彩。

當 dynamicColor 資源的值為「正確」時，無法動態變更色彩的用戶端（非 Motif 應用程式）會在色彩對映中配置與可以動態變更色彩的用戶端不同的儲存格，即使您所看到的是相同的色彩。

註解 – 既然所有的用戶端都可以共用相同的色彩儲存格，將 dynamicColor 設定為「錯誤」可減低您的桌上管理系統所消耗的色彩數目。

設定應用程式視窗的陰影濃度

桌上管理系統為應用程式視窗中的元件定義一個像素的陰影濃度，例如按鈕陰影與焦點選取部分。Motif 1.2 應用程式使用此資源值；其它的應用程式可能不會獲得此資源值，因此在顯示器上的出現將會不同。

如果要設定非 Motif 1.2 應用程式的一個像素的陰影濃度：

1. 登入為 `root`。
2. 建立 `/etc/dt/config/language/sys.resources` 檔案。
3. 在 `/etc/dt/config/language/sys.resources` 中指定特定的應用程式資源如下：

```
application_class_name *XmCascadeButton*shadowThickness: 1
```

有關置換系統預設資源以及為所有的桌上管理系統使用者指定其它資源的更多資訊，請參閱 第 42 頁“載入階段作業資源”

第 18 章

配置本土化桌面階段作業

如果要配置本土化的桌上管理系統階段作業，您將會需要：

- 將 LANG 環境變數與其它的「國家語言支援」（NLS）環境變數
- 存取語言相依的訊息目錄與資源檔案
- 經由國際化的系統遠端執行應用程式

- 第 275 頁“管理 LANG 環境變數”
- 第 278 頁“尋找字型”
- 第 278 頁“將 app- 預設資源檔案本土化”
- 第 278 頁“本土化動作與資料類型”
- 第 279 頁“將圖示及點陣圖本土化”
- 第 280 頁“將輔助說明卷次本土化”
- 第 281 頁“將訊息目錄本土化”
- 第 281 頁“在遠端執行本土化桌上管理系統應用程式”
- 第 281 頁“重設您的鍵盤對映”

管理 LANG 環境變數

此 LANG 環境變數必須為桌上管理系統設定以使用作業系統的語言感應式常式。桌上管理系統支援：

- 西歐、以拉丁語為基礎的語言
- 日文
- 繁體中文
- 簡體中文
- 韓文

註解 – 其它語言的支援可能已由您的桌上管理系統廠商所增加。

您可以將 LANG 設定為由作業系統所支援的任何值。登入螢幕中的「選項」功能表會顯示支援的語言與領域的清單。

設定桌上管理系統的 LANG 有四種方法：

- 藉由編輯 Xconfig 檔案中的資源
- 使用登入螢幕中的「選項」功能表
- 經由建立可執行的 sh 或 ksh Xsession.d 程序檔。（請參閱第 40 頁“找出 Xsession.d 程序檔的來源”，取得如何使用 Xsession.d 程序檔的資訊。）
- 藉由編輯使用者的 .dtprofile 檔案

設定 LANG 時，桌上管理系統會使用下列語言相依的檔案來決定本土化的介面。

色彩—/usr/dt/palettes/desc.language

背景定位—/usr/dt/backdrops/desc.language

設定多個使用者的語言

如果您藉 Xconfig 檔案設定語言，登入螢幕會本土化且會為所有使用者設定 LANG。這是變更多重顯示系統中所有顯示器的 LANG 的唯一方法。（如果要修改 Xconfig，請將 /usr/dt/config/Xconfig 複製到 /etc/dt/config/Xconfig。）

語言是藉由將下行放在 /etc/dt/config/Xconfig 中所設定的：

```
dtlogin.host_display.language: language
```

例如，下行會將 LANG 設定為顯示器 my_host:0 上的 Swedish_locale。

```
dtlogin.my_host_0.language: Swedish_locale
```

dtlogin 用戶端讀取該語言合適的訊息目錄，並帶出本土化的登入螢幕。然後 dtlogin 用戶端使用 /etc/dt/config/Xresources 資源檔案中的下列資源來決定語言環境的清單：

- dtlogin*language
- dtlogin*languageList
- dtlogin*languageName

Xconfig 檔案可能需要適當地為選擇的語言設定 NLSPATH 環境變數。如果情況並非如此，或是如果您想要自己設定 NLSPATH，請參閱。

設定一個階段作業的語言

如果要設定一個階段作業的語言，請使用登入螢幕「選項」功能表。登入螢幕會本土化且會為使用者設定 LANG。LANG 會於階段作業結束時返回其預設值（設定於 dtlogin 中）。

設定一個使用者的語言

使用者可以置換登入的 LANG 設定在 *HomeDirectory/.dtprofile* 檔案中。登入螢幕未本土化，但是 LANG 已為使用者設定好了。

- 如果您使用 sh 或 ksh:

```
LANG=language      export LANG
```

- 如果您使用 csh:

```
setenv LANG language
```

LANG 環境變數與階段作業配置

此 LANG 環境變數變更搜尋您的階段作業配置檔案的目錄名稱。

本土化的階段作業配置檔案為：

- */usr/dt/config/language/Xresources* (登入管理者資源檔案)
- */usr/dt/config/language/sys.font* (階段作業管理者資源檔案)
- */usr/dt/config/language/sys.resources* (階段作業管理者資源檔案)
- */usr/dt/config/language/sys.session* (階段作業管理者可執行的 shell)
- */usr/dt/config/language/sys.dtwmrc* (視窗管理者資源檔案)
- */usr/dt/appconfig/types/language/dtwm.fp* (視窗管理者前方面板)

設定其他 NLS 環境變數

除了 LANG，還有其它的 NLS 環境變數，例如 LC_CTYPE 與 LC_ALL。這些變數既不是由 dtlogin 語言資源也不是由登入螢幕「選項」功能表所影響。它們必須設定在下列檔案中：

- 整個系統的變數：*/etc/dt/config/Xsession.d*
- 個人變數：*HomeDirectory/.dtprofile*

NLSPATH 環境變數

此 NLSPATH 環境變數決定應用程式搜尋訊息目錄的目錄路徑。必須設定 LANG 與 NLSPATH 兩者以使用這些訊息目錄。有關本土化訊息的位置，請參照第 281 頁“將訊息目錄本土化”。大部分的桌上管理系統用戶端都會在啟動時將 NLSPATH 字首加至路徑。

尋找字型

桌上管理系統所包含的字型是在 `/usr/lib/X11/fonts` 目錄中。每個目錄都包含一個目錄檔案 `fonts.dir` 以及一個別名檔案 `fonts.alias`。有關建立 `fonts.dir` 與 `fonts.alias` 檔案的資訊，請參閱 `mkfontdir` 線上援助頁。

如果要列出伺服器所有可供使用的字型，請使用 `xlsfonts` 指令。如果要增加或刪除伺服器的字型，請使用 `xset` 指令。

將 app- 預設資源檔案本土化

桌上管理系統用戶端的 `app-defaults` 檔案的預設位置為 `/usr/dt/app-defaults/language`。例如，如果 `LANG` 設定為 `Swedish_locale`，則應用程式將會在 `/usr/dt/app-defaults/Swedish_locale` 中的尋找 `app-defaults` 檔案。如果未設定 `LANG`，會忽略 `language`，而應用程式會在 `/usr/app-defaults/C` 中尋找 `app-defaults` 檔案。

如果要變更 `app-defaults` 的位置，請使用 `XFILESEARCHPATH` 環境變數。例如，如果要將 `app-defaults` 移至 `/users`，請將 `XFILESEARCHPATH` 設定為 `/usr/app-defaults/language/classname`。

如果您設定 `XFILESEARCHPATH` 於 `HomeDirectory/.dtprofile`，則其值會套用到所有的桌上管理系統。非客戶將不會找到其資源檔案，除非您將其連結或複製到由 `XFILESEARCHPATH` 指定的目錄。

本土化動作與資料類型

註解 – 如果要在 `/usr/dt/appconfig` 目錄中自訂檔案，請在自訂之前將檔案複製到 `/etc/dt/appconfig` 目錄。

動作的搜尋路徑與資料類型定義檔案包括語言相依的目錄於：

- 個人的：`HomeDirectory/dt/types`
- 整個系統的：`/etc/dt/appconfig/types/language`
- 內建的：`/usr/dt/appconfig/types/language`

「應用程式管理者」的配置檔案搜尋路徑為：

- 個人的：*HomeDirectory* /dt/appmanager
- 整個系統的：*/etc/dt/appconfig/appmanager/language*
- 內建的：*/usr/dt/appconfig/appmanager/language*

此目錄中的檔案與目錄名稱都已本土化。

將圖示及點陣圖本土化

如果要本土化圖示，請以「圖示編輯程式」來編輯圖示並儲存於：

```
/etc/dt/appconfig/icons/language
```

如果您將其儲存於不同的目錄中，請設定 *XMICONSEARCHPATH* 環境變數以包含您儲存圖示的目錄。此 *XMICONBMSEARCHPATH* 環境變數控制用於搜尋圖示的路徑。

將背景定位名稱本土化

背景定位的本土化是經由使用說明檔案 (*desc.language* 與 *desc.backdrops*) 所完成的。背景定位檔案沒有特定的本土化目錄存在 (例如 */usr/dt/backdrops/language*)。所有的語言環境都使用相同的背景定位檔案集，但其自己的 *desc.language* 檔案都包含轉換的背景定位名稱。

說明檔案包含轉換的背景定位名稱的資源規格。例如：

```
Backdrops*Corduroy.desc:  
Velours Backdrops*DarkPaper.desc:  
PapierKraft  
Backdrops*Foreground.desc:  
AvantPlan
```

desc.language 檔案是用於尋回語言環境 *language* 的背景定位說明，以在「樣式管理者」中顯示背景定位。如果有說明規格，其將會顯示於「樣式管理者」背景定位清單中。否則，將會使用背景定位檔案名稱。

使用者可以將其自己的背景定位說明增加至 *HomeDirectory* */.dt/backdrops/desc.backdrops* 檔案中。此檔案是用於尋回由使用者所增加全部背景定位的背景定位說明，不管語言環境為何。

說明 檔案的搜尋路徑為：

- 個人的：*HomeDirectory* */.dt/backdrops/desc.backdrops*

- 整個系統的：`/etc/dt/backdrops/desc.language`
- 內建的：`/usr/dt/backdrops/desc.language`

將調色板名稱本土化

調色板的本土化是經由使用說明檔案 (`desc.language` 與 `desc.palettes`) 所完成的。沒有特定的本土化目錄存在 (例如 `/usr/dt/palettes/language`)。所有的語言環境都使用相同的調色板檔案集，但其自己的 `desc.palettes` 檔案都包含轉換的調色板名稱。

說明檔案包含轉換的調色板名稱的資源規格。例如：

```
Palettes*Cardamon.desc: Cardamone
Palettes*Cinnamon.desc: Cannelle
Palettes*Clove.desc: Brun
```

此 `desc.language` 檔案是用於尋回語言環境 `language` 的調色板說明，以在「樣式管理者」清單中顯示調色板。如果有說明規格，其將會顯示於「樣式管理者」調色板清單中。否則，將會使用調色板檔案名稱。

使用者可以在 `HomeDirectory /.dt/palettes/desc.palettes` 檔案中新增他們自己的調色板說明。此檔案是用於尋回由使用者所增加全部調色板的調色板說明，不管語言環境為何。

說明檔案的搜尋路徑為：

- 個人的：`HomeDirectory /.dt/palettes/desc.palettes`
- 整個系統的：`/etc/dt/palettes/desc.language`
- 內建的：`/usr/dt/palettes/desc.language`

將輔助說明卷次本土化

如果您已本土化輔助說明卷次，您必須將其儲存於下列目錄之一中。最先找到哪一個輔助說明卷次就使用哪一個。以下面的順序來搜尋目錄：

- 個人的：`HomeDirectory /.dt/help`
- 整個系統的：`/etc/dt/appconfig/help/language`
- 內建的：`/usr/dt/appconfig/help/language`

將訊息目錄本土化

如果您已本土化訊息目錄，將其儲存於以下目錄中：

```
/usr/dt/lib/nls/msg/language.
```

這些目錄包含 *.cat 檔案。

在遠端執行本土化桌上管理系統應用程式

您可以呼叫有類似的本土化桌上管理系統安裝的任何遠端執行主機的本土化桌上管理系統應用程式。啟動應用程式時，呼叫應用程式主機上與 NLS 有關的環境變數的值會傳送至遠端主機。但是，環境變數不包含任何主機資訊。

重設您的鍵盤對映

如果您看到非預料中的字元與行為，或是無法顯示或鍵入的字元，您可能會需要重設或安裝您的鍵盤對映或是變更您的輸入方法。

輸入方法是由 LC_CTYPE，LANG 或 LC_ALL 環境變數或是由 -lang 選項所指定的語言所決定的。

例如，如果使用者要在 IX shell 中開啓具有 C 語言環境的終端機：

```
LANG=C dtterm
```

此新的終端機使用包括 C 輸入方法與字型的 C 語言環境。如果您正在使用特定的語言鍵盤，輸入方法可能不會接受任何輸入的擴充字元。使用具有語言特定鍵盤的 C 語言環境時，使用者需要在呼叫終端機之前將 LC_CTYPE（或 LANG 或 LC_ALL）環境變數設定為合適的值。

例如，如果要使用具有德文鍵盤的 C 語言環境，請鍵入：

```
LANG=C LC_CTYPE=DeDE dtterm
```

如果 X 伺服器已重設且鍵對映已初始化，則您可以使用 `txmodmap` 指令，在伺服器中重設適當的鍵盤對映。

dtconfig(1) Man Page

NAME

dtconfig - desktop configuration utility

SYNOPSIS

dtconfig [-d |-e |-kill |-reset |-p |-inetd |-inetd.ow]

DESCRIPTION

Desktop configuration utility. Integrates CDE with the operating system of the underlying platform. System root login privilege is required to use dtconfig.

OPTIONS

-d Disables desktop auto-start feature. At end of boot cycle, platform's native text based login mechanism will be used.

-e Enable's desktop auto-start feature. Desktop login

-kill

Kill desktop (window based) login process and any user sessions associated with it. Return control to system's native text based console.

-reset

Tell desktop (window based) login process to reread

its

configuration file to incorporate any changes.

-p Printer actions for any printer known to platform will

be created if such print actions do not already exist in the platform's actions database. This option is executed automatically at boot time if desktop auto-start has been enabled.

-inetd

Adds /usr/dt/bin daemons to the /etc/inetd.conf file. Specific CDE background daemon setup includes rpc.ttdbserverd (ToolTalk), rpc.cmsd (Calendar Manager), and dtspcd (subprocess control). This

-inetd

option is called automatically by Solaris CDE package installs. This -inetd option is also useful for CDE daemon setup outside of normal Solaris CDE install, including system setup where /usr/dt has simply been mounted from some remote fileserver exporting the /usr/dt directory.

-inetd.ow

Switches the ToolTalk and Calendar Manager daemons (rpc.ttdbserverd & rpc.cmsd) start lines in /etc/inetd.conf back to the older /usr/openwin/bin area. This option is called automatically by Solaris CDE package remove scripts when needed. It is also useful outside of normal Solaris package remove operations when /usr/dt is about to be manually removed or unmounted.

RETURN VALUES

0 Successful completion
>0 Error condition

FILES

/usr/dt/bin/dtconfig location of dtconfig utility

SEE ALSO

dtlogin (1), dtprintinfo (1)

索引

編號和符號

* 通配符號, 202
? 通配符號, 202
%B, 137
%DatabaseHost%, 188
%DisplayHost%, 188
%H, 137
%L, 137
%LocalHost%, 188
%M, 137
%SessionHost%, 188
?Xconfig 檔案, 設定語言, 276

A

ACTIONS 欄位, 200
activeColorSetId 資源, 269
ALTERNATE_ICON 欄位, 234
ANIMATION 定義, 236
AnswerBook 文件, 從網路增加, 106
app_root
 參閱 應用程式根目錄
app-預設值
 桌上管理系統應用程式, 257
 語言相依, 278
/appconfig, 94
/app-defaults, 94
Apple Macintosh 應用程式環境, 107
Arg_1 syntax, 151
ARG_CLASS 欄位, 177
ARG_COUNT 欄位, 177, 186
ARG_MODE 欄位, 177

Arg_n 語法, 179
ARG_TYPE 欄位, 177, 200
 列印, 133
ARG_TYPE欄位, 186

B

/bin, 93
.bm 檔案延伸名稱, 210
bottomShadowColor 資源, 268
BOX 定義, 221
 語法, 222
BROADCAST, 在 XDMCP-間接中使用, 25
-broadcast 旗標, 104
buttonBindings 資源, 253

C

C 語言環境, 103
CDE-MIN 檔案, 119
CDE-TT 檔案, 119
CHOOSER 字串, 25
CLIENT_GEOMETRY 欄位, 235
CLIENT_NAME 欄位, 235
colorUse 資源, 43, 271, 272
COMMAND 動作, 170
 必需的欄位, 178
 執行字串, 178
 範例, 172
/config, 94
CONTAINER_NAME 欄位, 222, 223, 226

CONTAINER_TYPE 欄位, 222
CONTENT 欄位, 201, 205
CONTROL_BEHAVIOR 欄位, 240
CONTROL 定義, 語法, 222
COPY_TO_ACTION 欄位, 201
cpp 敘述, 46
current.old 目錄, 48

D

DATA_ATTRIBUTE, 與 DATA_ATTRIBUTES 配對, 196
DATA_ATTRIBUTES
 定義, 196, 199
 語法, 197
DATA_CRITERIA
 多重, 206
 定義, 201
 語法, 197
DataBaseHost 關鍵字, 188
DELETE 欄位, 220
DESCRIPTION 欄位, 174, 199
 /dev/console, 95
 directory, 資料類型條件, 204
DISPLAY_displayname 巨集, 46
DISPLAY 變數, 登入管理者設定, 33
DisplayHost 關鍵字, 188
DROP_ACTION 欄位, 234
DROP_ANIMATION 欄位, 236
dt 檔案, 171
dtaction
 用來變更使用者, 190
 語法, 190
dtappgather, 41, 57
dtappintegrate, 84
 功能, 85
 移除應用程式, 63
 語法, 85
DTAPPSEARCHPATH 變數
 定義, 136
 組合, 138
DtButtonBindings, 252
dtchooser 檔案, 35
dtconfig(1) man page, 283
dtconfig 指令, 26

dtCreate
 參閱 動作
DTDATABASESEARCHPATH 變數
 使用, 176
 定義, 136
 組合, 141
DtEditor, 樣式轉換, 258
Derrors file, 26
dtgreet 檔案, 35
DTHELPSEARCHPATH 變數
 定義, 136
 組合, 145
dtlogin
 參閱 登入管理者
Dtlogin*language 資源, 31
dtlp, 105
dtmailpr, 105
DTMOUNTPOINT 變數
 必需的程序, 120
 由使用者繼承, 120
 使用的程序, 120
 設定, 120
Dtpid 檔案, 21
.dtprofile, 51, 53
.dtprofile 檔案
 建立, 39
 設定 LANG, 277
 設定環境變數, 45
 尋源, 39
 資源, 39
 語法, 45
DtRootMenu, 250
dtsearchpath, 41, 136, 176
dtsmcmd 指令, 47
DTSOURCEPROFILE 變數, 42
dtspcd, 119, 120, 121
 配置, 121
 認證目錄, 116, 121
DTSPSYSAPPHOSTS 變數
 定義, 136
 修改, 62
 語法, 138
DTSPSYSDATABASEHOSTS 變數
 EXEC_HOST 的影響, 126
 定義, 136, 141
 語法, 141
DTSPSYSHELP 變數, 144
 定義, 136

DTSPYSHELP 變數 (繼續)
語法, 144
DTSPYSICON 變數
定義, 136
語法, 143
DTSPUSERAPPHOSTS 變數
定義, 136
修改, 63
語法, 138
DTSPUSERDATABASEHOSTS 變數
定義, 136, 141
語法, 141
DTSPUSERHELP 變數
定義, 136
語法, 144
DTSPUSERICON 變數
定義, 136
語法, 143
dtstart_appgather 變數, 41
dtstart_searchpath 變數, 41
dtstart_ttsession 變數, 42
dtterm, 54
dtwm
參閱 工作區管理者
dtwm.fp 檔案, 218
dtwmfp.session 檔案, 219
dtwmrc 檔案, 244
編輯, 244
dynamicColor 資源, 43, 271, 273

E

EMACS 樣式轉換, 258
EMACS 轉換, 258
/etc/dt, 94
/etc/rmmount.conf, 108
/examples, 94
EXEC_HOST
參閱 執行主機
EXEC_HOST 欄位, 188
多重值, 126
受到資料庫搜尋路徑影響, 126
預設值, 126, 188
EXEC_STRING, 105

EXEC_STRING 欄位
參閱 執行字串

F

FILE_NAME 欄位, 233, 234
FontSet 資源, 263
foregroundColor 資源, 43, 273
fp_dynamic 目錄, 218

G

getty, 23, 38
GID, 116

H

HELP_STRING 欄位, 237
HELP_TOPIC 欄位, 237
HELP_VOLUME 欄位, 237
HIGH_COLOR, 272
home.old 目錄, 48
HOME 變數, 33
HomeDirectory, 94

I

ICON 欄位
可容許的值, 175
有效值, 199
前方面板中, 227
資料類型的, 199
inactiveFrameColorId 資源, 269
/include, 94
-indirect 選項, 24
inetd.conf, 120
IS_TEXT 欄位, 201

K

keyBindings 資源, 255

L

LABEL 動作欄位, 175
LANG 變數, 275
 在 .dtprofile, 277
 登入管理者設定, 33
 資料類型上的效果, 206
/lib, 94
LINK_TO_ACTION 欄位, 201
localTerminal 資源, 185
LOCKED 欄位, 220
.login, 52
.login 檔案, 39
 登入管理者未讀取, 45
 資源, 42
LOGNAME 變數, 33
LOW_COLOR, 272
lp
 列印排存器, 117
 指令, 117
LPDEST 變數, 132

M

mailx, 117
/man, 94
MAP 動作, 154
 定義中的名稱, 170
 範例, 173
MEDIA 欄位, 201
MEDIUM_COLOR, 272
MIME_TYPE_MEDIA 欄位, 201
mkfontdir 指令, 編譯檔案, 278
MODE 欄位, 201
 語法, 204
MODE 欄位中的 AND 運算子, 204
MODE 欄位中的 NOT 運算子, 204
MODE 欄位中的 OR 運算子, 204
MONITOR_TYPE 欄位, 234
Motif 視窗管理員
 參閱 工作區管理者
MOVE_TO_ACTION 欄位, 201

N

NAME_PATTERN 欄位, 201
NFS, 116

NLS 遠端執行, 281
NLS 環境變數, 277
NO_STDIO 視窗支援, 184
NoBackdrop 設定, 247
NoPrint 動作, 156
NUMBER_OF_ROWS 欄位, 238

O

-once 選項, 104
OpenWindows, 環境, 95

P

/palettes, 94
PANEL_GEOMETRY 欄位, 239
PANEL 定義, 221
 語法, 221
PATH_PATTERN 欄位, 201
 語法, 202
PATH 變數, 179
 登入管理者設定, 33
PERM_TERMINAL 視窗支援, 184
.pm 檔案延伸名稱, 210
POSITION_HINTS 欄位, 226
primaryColorSetId 資源, 269
primaryColorSetId 資源, 271
.profile, 52
.profile 檔案, 39
 登入管理者未讀取, 45
 資源, 42
PUSH_ACTION 欄位, 232
PUSH_ANIMATION 欄位, 236

Q

-query 選項, 24

R

README 檔案, 83
RESOURCE_MANAGER 屬性, 43, 46
rgb.txt 檔案, 269
RGB 色彩值, 269

RGB 值, 268
rpc.cmsd, 122
rpc.ttdbserver, 119, 120

S

.sdl 檔案, 77
sdtcm_convert 程序檔, 106
secondaryColorSetId 資源, 269, 271
selectColor 資源, 268
sessionetc 檔案, 47
sessionexit 檔案, 48
SessionHost 關鍵字, 188
shadowPixmap 資源, 43, 272
/share, 94
shell
 亦請參閱環境變數, .profile、.login、.dtprofile, 45
 在動作中, 181
 在動作中使用, 181
 找出 .profile 或 .login 的資源, 42
 個人自訂, 39
 執行字串中的語法, 179
 整個系統的自訂, 39
SHELL 變數, 登入管理者設定, 33
Solaris CDE, 取消桌上管理系統, 96
SPC, 120
 安全性, 121
startlog 檔案, 48
stty, 52
SUBPANEL 定義, 221
SWITCH 定義, 223
sys.dtprofile 檔案, 39
sys.dtwmrc 檔案, 244
sys.resources 檔案, 43, 45, 257
sys.session 檔案, 44, 46
systemPath 資源, 34

T

TERMINAL 視窗支援, 184
textColorSetId 資源, 269
timeZone 資源, 34

ToolTalk
 動作
 參閱 TT_MSG 動作
 傳訊常駐程式, 39, 42
 資料庫伺服器
 參閱 rpc.ttdbserver
 應用程式, 動作, 192
ToolTalk 訊息伺服器
 參閱 ttsession
topShadowColor 資源, 268
tset, 52
TT_MSG 動作
 建立, 192
 關鍵字, 193
ttsession, 122
 啟動, 42
tty 設定, 53, 54
ttyModes, 54
TYPE 欄位, 232
TZ 變數, 33, 34

U

UID, 116
UNIX 按鍵組合, 257, 258
UNIXbindings 檔案, 259
user-prefs.dt 檔案, 65
USER 變數, 33
userPath 資源, 34
/usr/dt, 93

V

/var/dt, 94

W

WINDOW_TYPE 欄位, 184
windowMenu 資源, 248
WM_CLASS 屬性, 213
wmStartupCommand 資源, 44
workspaceCount 資源, 246
writeXrdbColors 資源, 44
wscon 主控台日誌檔案, 52

X

- X400_TYPE 欄位, 201
- X11 伺服器, 95
- X 伺服器
 - 變更存取, 31
 - 變更環境, 31
- X 授權, 117
- X 終端機, 101, 103, 118
 - CHOOSE 字串, 25
 - Xaccess 清單, 25
 - XDMCP-直接, 24, 25
 - XDMCP-間接, 24, 25
 - 可能的登入伺服器配置, 21
 - 作為工作站, 103
 - 使用, 102
 - 取得階段作業服務, 112
 - 非 XDMCP 顯示器, 24
- X 資源
 - 參閱 資源
- X 邏輯字型說明, 263
- Xaccess 檔案, 24
- XAUTHORITY 變數, 登入管理者設定, 33
- Xconfig, 95, 99
- Xconfig 檔案
 - 修改, 21
 - 設定資源, 30
- .Xdefaults, 53, 54
- .Xdefaults 檔案, 43, 258
- XDM, 95, 102, 104
 - 協定, 102
- XDMCP, 19
 - 定義, 23
 - 直接存取, 25
 - 直接要求, 24
 - 查詢模式, 24
 - 間接存取, 25
 - 間接要求, 24, 35
- Xerrors 檔案, 26
- Xfailsafe 檔案, 32, 33, 35
- xlsfonts 指令
 - 列出伺服器中的字型, 278
 - 安裝, 278
- XMICONBMSEARCHPATH 變數
 - 用法, 143
 - 定義, 136
 - 組合, 143
- XMICONSEARCHPATH 變數
 - 用法, 143

- XMICONSEARCHPATH 變數 (繼續)
 - 組合, 143
- XMICONSEARCHPATH 變數, 定義中的名稱, 136
- XmText*FontList 資源, 263
- Xpid 檔案, 21
- Xreset 檔案, 33
- Xresources 檔案, 28, 29
- Xserver, 104
 - 清除, 96
- Xservers 檔案, 95
 - 啟動伺服器, 22
 - 預設, 22
 - 管理本機顯示器, 34
 - 語法, 22
- Xsession
 - 程序檔, 51
- Xsession.d 目錄, 39, 45
 - 自訂, 40
 - 程序檔, 40
- Xsession 檔案, 39
 - 啟動階段作業管理者, 38
 - 設定 PATH, 33
 - 透過登入伺服器執行, 32
 - 整個系統的自訂, 40
- Xsetup 檔案, 32
- Xstartup 檔案, 32
- xterm, 54
 - 啟動, 97
- XUSERFILESEARCHPATH 變數, 278

子

- 子程序控制服務
 - 參閱 SPC
- 子程序控制常駐程式
 - 參閱 dtspcd

工

- 工作站, 作為 X 終端機, 103
- 工作區
 - 名稱, 246
 - 自訂, 246

工作區 (繼續)
背景定位
參閱 背景定位
數目, 246
變更預設數目, 238
工作區切換
工作區數目, 238
列數, 238
自訂, 237
定義, 221
定義的語法, 223
增加控制, 238
工作區功能表
定義, 248
建立, 250
修改, 249
語法, 248
增加功能表項目, 249
工作區管理者
功能, 248
功能表, 248
包含其他檔案, 245
定義, 244
按鈕組合, 252
重新啓動, 245
個人自訂, 244
配置檔案, 244
啓動, 39, 44
管理前方面板, 244
整個系統的自訂, 244
變更為 Motif, 255

內
內容型資料類型, 205

切
切換
參閱 工作區切換

引
引數
不可交換的, 動作, 182

引數 (繼續)
非檔案, 181
限制動作, 185
動作的, 151, 178
動作的可交換, 183
動作的多重, 182
動作數目, 186
提示, 180

文
文字編輯程式, 變更, 64

日
日期類型控制, 232
日曆常駐程式, 122

主
主目錄
共用, 116
網路的, 116
主要配置檔案, 95
主要階段作業, 37, 38
主從配置
參閱 網路

功
功能表
工作區管理者, 248
使用動作, 149

包
包含陳述式, 在工作區管理者檔案中, 245

可
可執行檔, 資料類型條件, 204

外

外來顯示器類型, 23

本

本土化, 280
 訊息目錄, 281
 動作, 191
 動作標籤, 191
 登入螢幕, 29
 資料類型, 206
 圖示, 279, 280
 調色板名稱, 280
本機顯示器類型, 23

用

用戶端
 伺服器的, 配置, 118
 定義, 110
 前方面板中的視窗, 235
用戶端類型控制, 232

目

目前階段作業, 37, 38

列

列印
 不同資料類型, 133
 測試, 117
 資料類型的配置, 155
 預設目標, 132
 管理, 129
 觀念, 133
列印工作更新間隔, 130
列印排存器, lp, 117
列印管理者, 130
 工作更新間隔, 130

印

印表機
 工作更新間隔, 130
 刪除, 130
 裝置名稱, 117
 預設, 132
 圖示影像, 130
 遠端存取, 117
 增加, 129
 標籤, 131
 關於項目輔助說明, 131

名

名稱型資料類型, 202

回

回復前方面板動作, 220

多

多重螢幕, 99
多重顯示器, 登入管理者, 30

字

字元顯示主控台, 23
字串動作引數, 181
字型
 X 邏輯字型說明, 263, 264
 xlsfonts 指令, 278
 主要目錄, 278
 列出可用的, 264
 系統, 樣式管理者中, 263
 指定屬性字串, 264
 透過 mkfontdir 指令尋找, 278
 透過目錄檔案尋找, 278
 透過別名檔案尋找, 278
 資源, 為註冊修改, 71
 資源, 設定, 263
 管理, 262
 樣式管理者的數目, 263
 點陣圖, 263

字型資源, 263
字型路徑, 103
字型對話方塊, 263

安

安裝, 目錄位置, 93
安裝圖示控制, 移除, 230

自

自訂郵件列印, 105
自動裝載程式, 120

色

色彩

工作視窗框架, 269
文字登錄區域, 269
以 shadowPmaps 資源建立陰影, 272
色彩集, 267, 268
使用的數目, 271
非作用中視窗框架, 269
指定前景, 273
值, 268
控制, 以 dynamicColor 資源, 273
控制, 以樣式管理者, 270
最大的分配數目, 271
資源, 268
預設, 269
圖示中的用法, 214
管理, 267
調色板, 267
應用程式視窗, 269
色彩伺服器, 39
 啓動, 43
 資源, 43
色彩集, 267
 預設, 269
 對映到顯示元素, 269
色彩資源, 為註冊修改, 72
色彩調色板, 267

位

位置型資料類型, 202

伺

伺服器

定義, 110
配置, 118
動作, 124
登入, 118
階段作業, 118
 伺服器, 110
資料庫, 118
資料類型, 124
圖示, 114, 118
輔助說明, 114, 118
應用程式, 110, 118
檔案, 110
類型, 114

依

依資料類型隱藏檔案, 200

使

使用 X 終端機, 102
使用者, 變更動作, 190
使用者 ID, 116
使用檔案管理者瀏覽圖示, 214

協

協定, XDM, 102

卸

卸載已裝載的 CDE 目錄, 98

定

定位區域

- 前方面板控制, 234
- 動作圖示, 179
- 定位檔案, 接受... 的動作, 179

放

- 放置, 圖示, 54

空

- 空白類型控制, 232

前

- 前方面板, 240
 - 工作區管理者管理, 244
 - 元件, 220
 - 功能表, 248
 - 用戶端, 235
 - 自訂, 217
 - 使用動作, 148
 - 定位區域控制, 234
 - 定義的組織, 220
 - 修改, 223
 - 配置檔案, 217
 - 動畫, 236
 - 動態自訂, 218, 219
 - 控制
 - 參閱 控制
 - 控制個人自訂, 220
 - 組合的優先順序, 218
 - 註冊套裝軟體中的控制, 83
 - 搜尋路徑, 218
 - 新的, 240
 - 語法, 221
 - 輔助說明, 237
 - 增加列, 240
 - 螢幕上的位置, 239
 - 檔案的命名慣例, 218
 - 顯示圖示, 213
- 前方面板的動畫, 236
- 前景資源, 268

建

- 建立動作, 157
 - 介紹, 157
 - 主視窗, 159
 - 功能, 157
 - 使用, 159
 - 指定圖示, 166
 - 指定檔案引數, 160
 - 限制, 158
 - 配置檔案建立的, 157
 - 動作名稱, 160
 - 動作指令語法, 160
 - 啓動, 159
 - 資料類型名稱, 162
 - 資料類型建立, 152, 161
 - 檔案提示, 161
- 建立動作中的動作名稱欄位, 160
- 建立動作中的資料類型清單, 162

按

- 按鈕組合, 252
 - 建立新集, 253
 - 語法, 252
 - 增加, 253
- 按鍵組合
 - 建立新集, 255
 - 預設, 254
 - 語法, 254

指

- 指令行登入, 23

背

- 背景定位, 244
 - 使用圖形影像, 247
 - 增加, 247
 - 檔案位置, 247
- 背景資源, 268

重

- 重新載入動作動作, 173
- 重新載入資源動作, 43
- 重新載入應用程式動作, 63

個

- 個人資料類型及動作, 建立, 198
- 個人應用程式群組, 61

時

- 時區, 變更, 34
- 時鐘類型控制, 232

根

- 根功能表
 - 參閱 工作區功能表
- 根視窗, 247

桌

- 桌上管理系統, 網路的, 100
- 桌上管理系統啓動
 - 多重螢幕, 99
 - 問題, 51
- 桌上管理系統搜尋路徑, 39
- 桌面工具應用程式群組, 修改, 61
- 桌面環境檔案, 107

訊

- 訊息目錄, 277

配

- 配置檔案, 95
 - location, 94
 - 工作區管理者, 244
 - 在註冊套裝軟體中, 70

配置檔案 (繼續)

- 前方面板, 217
- 動作, 171
- 登入管理者, 35
- 視窗管理員, 244
- 階段作業管理者, 49
- 資料類型, 196

動

動作

- ... 的圖示, 166, 175
- COMMAND, 170
- MAP, 170
- TT_MSG, 170
- 人工建立, 169, 171
- 人工建立的理由, 169
- 不可交換的引數, 182
- 不同連接兩下及定位功能, 186
- 介紹, 147
- 引數, 178
- 引數, 非檔案, 181
- 引數個數限制, 186
- 以不同使用者執行, 190
- 代表... 的圖示, 174
- 代表... 的檔案
 - 參閱 動作檔案
- 功能表中使用, 149
- 可交換的引數, 183
- 本土化, 191
- 用途, 152
- 名稱, 157, 174
- 字串變數, 189
- 伺服器, 124
- 沒有引數, 179
- 沒有資料, 179
- 沒有顯示輸出, 161
- 使圖示與... 產生關聯, 212
- 定義中的變數, 188
- 非檔案引數, 158, 181
- 前方面板使用, 148
- 建立動作的限制, 158
- 建立應用程式的圖示, 149
- 重新載入, 173
- 限制引數, 185
- 修改, 176
- 配置檔案, 171

動作 (繼續)

參數

參閱 引數

執行字串, 178

執行其他動作, 190

執行遠端應用程式, 125, 187

接受多重定位檔案, 183

接受定位檔案, 152, 179

接受定位檔案或提示符號, 180

終端機支援, 185

終端機選項, 184

提示檔案, 180

提供 shell, 181

視窗支援, 161, 184

註冊要求的, 75

搜尋路徑

參閱 資料庫搜尋路徑

資料類型的關係, 153

資料類型限定, 154, 186

預設終端機, 185

預設圖示, 175

對映, 154

與 dtappintegrate 整合, 85

與資料類型產生關聯, 199

標籤, 175, 191

範例, 172, 173

編輯, 176

優先順序規則, 177

檔案引數, 151

環境變數, 189

類型, 169, 171

動作伺服器

參閱 資料庫伺服器

動作定義中的字串變數, 189

動作定義檔案, 建立動作建立的, 157

動作的指令行, 178

動作的視窗支援, 161

動作搜尋路徑

參閱 資料庫搜尋路徑

動作圖示, 150, 174

建立, 174

桌上管理系統必需的, 78

動作圖示控制, 建立動作中的, 160

動作圖示資源, 175

動作檔案, 157, 174

內容, 150

定義, 150

建立, 82, 174

問

問題, 桌上管理系統啟動, 51

唯

唯讀資料類型條件, 204

國

國家語言支援, 國際化, 275

國際化

app-預設值, 278

LANG 變數, 275

NLS 環境變數, 277

字型, 278

設定語言, 276

疑難排解, 281

基

基本檔案名稱, 175, 199

執

執行主機

EXEC_HOST 欄位指定, 188

建立動作, 187

指定, 125

配置, 126

執行字串, 178

shell 語法, 179

一般功能, 178

多重檔案引數, 182

沒有引數, 179

定位檔案, 179

指定可執行檔, 179

提示字串, 181

提示檔案, 180

絕對路徑, 179

控

控制

- 一個實例, 235
- 工作區切換中, 238
- 切換, 235
- 外觀, 227
- 用戶端, 235
- 回復, 220
- 取代, 226
- 定位區域, 234
- 定義, 231
- 建立, 232
- 按一下對連按兩下, 240
- 修改, 225
- 動畫, 236
- 從前方面板移除, 225
- 開啓檔案的, 233
- 圖示, 227
- 監視, 234
- 與另一個控制交換, 226
- 增加到主面板, 224
- 標示, 240
- 鎖定, 220
- 關於項目輔助說明, 237
- 類型, 232
- 控制的監視類型, 234

授

- 授權, X, 117
- 授權資源, 31

啓

- 啓動錯誤登錄, 52

符

符號連結

- 已在註冊期間建立, 85
- 資料類型條件, 201
- 檔案名稱一致性, 117

終

- 終端機模擬, 54
- 終端機模擬器
 - 動作自動關閉選項, 161
 - 動作的, 184
 - 動作的指令行選項, 184
 - 預設動作, 185
 - 變更, 64

許

- 許可權型樣, 以建立動作指定, 163

連

- 連結, 資料類型條件, 204

陰

- 陰影濃度, 視窗, 274

單

- 單視窗階段作業, 32
- 單視窗階段作業選項, 52
 - 登入螢幕, 97

尋

- 尋找集對話方塊, 166

提

- 提示動作, 180

智

- 智慧型桌上管理系統應用程式, 60

無

無點陣圖顯示器, 執行登入伺服器, 22

登

登入伺服器

在沒有本機顯示器下執行, 22

字元顯示主控台, 23

沒有點陣圖顯示器, 22

系統 shell, 34

系統啟動時停用, 96

系統啟動時啟動, 96

使用者路徑, 33

取消, 96

取消程序 ID, 26

指令行登入, 22

配置, 21, 118

停止, 26

停用, 26

從指令行啟動, 20, 96

控制存取, 24

啟動, 20

啟動階段作業, 19

連接顯示器, 19

程序 ID, 21

概述, 19

疑難排解, 26

認證使用者, 19

環境, 33

變更時區, 34

顯示登入螢幕, 19

登入帳戶, 115

登入啟動檔案, 51

登入管理者, 19

自訂, 19

定義, 19

配置檔案, 35

發出指令, 32

資源, 28, 30

管理, 34

錯誤, 26

登入語言環境, 103

登入認證, 51

登入螢幕

X 伺服器存取, 31

X 伺服器環境, 31

本土化, 29

登入螢幕 (繼續)

字型, 29

自訂, 27

單視窗階段作業選項, 97

結束, 96

資源, 28

歡迎, 28

變更外觀, 28

變更行爲, 29

變更預設語言, 31

變更語言功能表的內容, 31

變更歡迎訊息, 28

顯示在網路顯示器上, 23

顯示器相依的行爲, 30

登入錯誤日誌, 52

發

發送郵件, 117

視

視窗功能表

定義, 248

新的, 251

語法, 248

視窗管理者, 變更, 44

視窗管理員, 244

註

註冊, 82

dtappintegrate, 84

一般步驟, 70

目的, 70

字型修改, 71

色彩修改, 72

定義, 59, 70

修改資源, 71

提供的功能, 68

概述, 68

資源修改, 71

圖示要求, 78

輔助說明檔案, 77

需要的動作, 75

註冊 (繼續)
需要的資料類型, 75
範例, 86, 92
應用程式根目錄, 72
應用程式群組, 79
註冊套裝軟體, 59, 82
README 檔案, 83
目的, 68
目錄, 73
定義, 70
前方面板控制, 83
建立的範例, 86
應用程式群組內容, 82
應用程式圖示, 82

郵

郵件列印, 自訂, 105
郵件類型控制, 234

開

開啓動作, 154

階

階段作業
主要, 37, 38
目前, 37, 38
回復, 48
在啓動時執行指令, 47
在登出時執行指令, 48
定義, 37
初始, 38
記錄錯誤, 48
啓動, 38
第一個, 46
備份, 48
單視窗, 32
結束時執行的程序檔, 33
資源, 39
預設, 37, 38
顯示器的特定, 38
顯示器特定, 47
階段作業目錄, 47

階段作業伺服器

參閱 登入伺服器
階段作業管理者, 38
介紹, 37
用戶端, 42
目錄, 49
在登出時執行指令, 48
收集應用程式, 41
自訂應用程式啓動, 44
執行其它的指令, 47
啓動, 38
啓動工作區管理者, 44
啓動應用程式, 44
設定搜尋路徑, 41
備份階段作業, 48
載入資源, 42
疑難排解, 48
整個系統的自訂, 40
錯誤登錄, 48, 52
檔案, 49
歡迎訊息, 39

搜

搜尋路徑
本土化, 145
目前的值, 136
前方面板定義, 218
桌上管理系統定義的, 135
動作, 171
設定, 136
階段作業管理者設定, 41
圖示, 211
輔助說明, 144
輸入變數, 136
輸出變數, 136
應用程式, 56, 137
環境變數, 136

新

新增, 工作區功能表項目, 249

滑

滑鼠組合

參閱 按鈕組合

群

群組 ID, 116

裝

裝載, 已安裝的 CDE, 98

裝載, 執行應用程式, 127

資

資料庫

重新載入, 173

重新載入動作, 173

資料庫主機, 125, 187

資料庫伺服器, 114, 125

用戶端的, 125

建立, 124

配置, 118, 124

資料庫搜尋路徑, 140, 171

組合, 141

預設, 140

與應用程式搜尋路徑相關, 139, 141

語法, 141

影響 EXEC_HOST, 126

環境變數, 141

資料類型

... 的圖示, 78, 166, 199

人工建立, 195

人工建立的要求, 158, 195

介紹, 147, 152

內容型, 205

分類條件, 201

以建立動作建立, 152, 161

可執行, 204

本土化, 206

列印, 155

名稱型, 163, 202

多重條件, 206

位置型, 202

伺服器, 124

資料類型 (繼續)

使圖示與... 產生關聯, 212

定位行為, 155

定義, 197

定義中的變數, 188

建立的目的, 69

建立動作的限制, 158

重新載入, 173

重新載入動作, 173

限制動作依據, 186

配置檔案, 196

動作的關係, 153

區分, 201

唯讀, 204

條件, 201

許可權型樣, 164

連按兩下行爲, 155

註冊要求的, 75

搜尋路徑

參閱 資料庫搜尋路徑

路徑型, 202

與 dtappintegrate 整合, 85

與動作產生關聯, 199

輔助說明, 199

模式條件, 204

範例, 198

隱藏, 200

屬性, 199

資料類型中的通配符號, 202

資料類型文字欄位的名稱, 162

資源

app-預設值, 257

colorUse, 272

foregroundColor, 273

shadowPixmap, 272

字型, 263

個人, 258

設定, 45, 257

視窗的陰影濃度, 274

階段作業, 39

載入, 42

預設桌上管理系統, 43

語言相依, 278

整個系統, 257

顯示器特定, 46

路

路徑

系統, 34

使用者, 34

路徑型資料類型, 202

電

電子郵件, 配置, 117

圖

圖示

大小慣例, 210

代表應用程式, 82

本土化, 279, 280

印表機影像, 130

如何尋找檔案, 210

色彩用法, 214

伺服器

參閱 圖示伺服器

使用檔案管理者作為圖示瀏覽器, 214

命名慣例, 210

非英文, 279, 280

前方面板, 213, 227

動作, 166, 175

動作圖示, 174

基本檔案名稱, 199

啟動應用程式, 78

產生關聯, 211

設計建議, 214

透過檔案管理者瀏覽, 214

尋找集對話方塊, 166

註冊要求的, 78

搜尋路徑

參閱 圖示搜尋路徑

資料類型, 78, 166, 199

與 dtappintegrate 整合, 86

與動作或資料類型產生關聯, 212

與應用程式視窗產生關聯, 213

應用程式的, 149

應用程式群組, 78, 80

檔案名稱, 210

檔案格式, 210

圖示大小, 210

圖示伺服器, 114

圖示 (繼續)

用戶端的, 125

建立, 124

配置, 118, 124

圖示放置, 54

圖示搜尋路徑, 142

組合, 143

預設, 142

與應用程式搜尋路徑相關, 139, 142

語法, 143

環境變數, 142

圖示類型控制, 234

圖素對映

如何尋找檔案, 210, 211

命名慣例, 210

搜尋路徑, 211

旗

旗標, -broadcast, 104

疑

疑難排解桌上管理系統啟動的問題, 51

網

網路, 109

X 授權, 117

一般配置步驟, 115

必需的檔案, 118

服務類型, 110

配置用戶端及伺服器, 118

基本配置, 115

透過裝載執行應用程式, 127

概述, 109

電子郵件, 117

網路的桌上管理系統, 100

語

語言, 使用 Xconfig 檔案設定, 276

語言功能表, 自訂, 31

語言功能表項目, 103

語言環境，C， 103

認

認證, 登入, 51
認證目錄, 116, 121

輔

輔助面板

回復已刪除的, 220
自訂內建面板, 229
定義, 223
建立, 228
修改, 228
容器, 223
新的, 228
與主面板產生關聯, 228
語法, 223
整個系統的自訂, 228
變更自動關閉行為, 230

輔助說明, 77

印表機圖示, 131
完全整合, 77
使用建立動作指定, 161
前方面板, 237
動作檔案上, 174
部分整合, 77
資料類型, 199

輔助說明伺服器, 114

用戶端的, 125
建立, 124
配置, 118, 124

輔助說明卷次

主輔助說明檔案, 77
本土化, 280
註冊套裝軟體中的位置, 77
整合層次, 77

輔助說明開發者工具箱, 77

輔助說明搜尋路徑, 135, 144

組合, 145
預設, 144
與應用程式搜尋路徑相關, 139, 144
語法, 144
環境變數, 144

輔助說明檔案

在註冊套裝軟體中, 77
與 dtappintegrate 整合, 85

遠

遠端執行

本國語言支援, 281
來自應用程式的遠端動作, 125
配置應用程式伺服器, 122
經由動作, 187
遠端檔案的裝載點, 119

影

影像

參閱 圖示

樣

樣式管理者

用來指定色彩, 269
字型整合, 71, 72

標

標示控制, 240

標題資源, 246

標籤

「前方面板」控制, 240
動作, 175, 191

編

編輯 Dtwmrc 動作, 244

編輯動作, 176

調

調色板, 267

本土化名稱, 280

輸

- 輸入方法,國際化, 281
- 輸入變數, 136
- 輸出變數, 136

選

- 選定的功能表, 149, 154
- 選項功能表, language, 103
- 選擇程式, 100, 101, 102

錯

- 錯誤登錄
 - Xsession, 52
 - 位置, 52
 - 啓動, 52
 - 登入, 52
 - 階段作業管理者, 52

優

- 優先順序
 - 前方面板組合, 218
 - 動作資料庫組合, 177

應

- 應用程式
 - app_root 目錄, 72
 - root 目錄, 72
 - 已註冊的, 定義, 59
 - 不需註冊即可增加, 59
 - 收集, 57
 - 收集到應用程式管理者, 56
 - 取消註冊, 63
 - 建立圖示, 149
 - 要求的動作, 75
 - 要求的資料類型, 75
 - 重新載入, 63
 - 移除, 63
 - 透過裝載在本端執行, 127
 - 智慧型桌上管理系統, 60
 - 登入時啓動, 39, 44

應用程式 (繼續)

- 註冊
 - 參閱 註冊
 - 註冊的功能, 68
 - 階段作業管理者收集, 41
 - 搜尋路徑, 56
 - 資料類型的目的, 69
 - 增加到現存的應用程式群組, 60
 - 增加到應用程式管理者, 58
 - 增加的方法, 59
- 應用程式伺服器
 - 用戶端的, 123
 - 定義, 110
 - 配置, 118, 123
 - 配置用戶端, 123
 - 管理, 122
 - 增加, 62
 - 標準配置, 123
 - 應用程式的可用性, 70
- 應用程式套裝軟體
 - 參閱 註冊套裝軟體
- 應用程式根目錄, 72
- 應用程式視窗, 使圖示與... 產生關聯, 213
- 應用程式註冊
 - 參閱 註冊
- 應用程式搜尋路徑
 - 本土化, 145
 - 用來收集應用程式, 56
 - 定義, 137
 - 個人, 63
 - 修改, 62
 - 修改的理由, 62
 - 修改優先順序, 139
 - 組合, 138
 - 預設, 62, 138
 - 語法, 138
 - 整個系統, 62
 - 環境變數, 138
- 應用程式群組
 - ... 的圖示, 78, 80
 - README 檔案, 83
 - 內容, 82
 - 目錄, 79
 - 名稱, 80
 - 在註冊套裝軟體中建立, 79
 - 收集, 56
 - 自訂, 61
 - 命名, 61

應用程式群組 (繼續)

- 定義, 55
- 建立的範例, 90
- 個人, 61
- 動作, 81
- 資料類型, 81
- 預設, 57
- 管理, 61
- 與 dtappintegrate 整合, 86
- 整個系統, 61
- 優先順序, 57

應用程式資源

參閱 資源

應用程式圖示, 172

- 使用建立動作, 157
- 定位檔案, 152
- 建立, 150, 174
- 桌上管理系統必需的, 78
- 連按兩下, 152

應用程式管理者

- 一般管理, 63
- 收集應用程式, 56
- 更新, 63
- 符號連結, 57
- 說明, 55
- 增加應用程式, 58
- 優先順序規則, 57
- 檔案系統位置, 56

應用程式管理員

- 收集應用程式, 41
- 整合應用程式, 68

檔

檔案

- 分散式存取, 116
- 名稱一致性, 117
- 依據資料類型隱藏, 200
- 裝載, 116
- 裝載點, 119
- 網路必需的, 118
- 遠端存取, 116
- 遠端資料, 119
- 檔案, 登入啓動, 51
- 檔案, 資料類型條件, 204
- 檔案引數
 - 以建立動作指定, 160

檔案引數 (繼續)

- 在動作中使用, 151
- 檔案共用, 116
- 檔案名稱, 圖示, 210
- 檔案名稱一致性, 117
- 檔案名稱資料庫伺服器, 119
- 檔案名稱對映, 119
- 檔案伺服器, 110, 111
- 檔案提示, 以建立動作指定, 161
- 檔案管理者, 作為圖示瀏覽器, 214
- 檔案類型
 - 參閱 資料類型
- 檔案類型控制, 232, 234

環

- 環境檔案, 桌面, 107
- 環境變數
 - 在 .dtprofile 中, 39
 - 找出 .login 或 .profile 的資源, 42
 - 個人, 45
 - 動作定義, 189
 - 設定, 45
 - 登入管理者, 34
 - 匯出, 45
 - 搜尋路徑, 136
 - 資料庫搜尋路徑, 141
 - 預設, 39
 - 像素對映搜尋路徑, 142
 - 圖示搜尋路徑, 142
 - 輔助說明搜尋路徑, 144
 - 遠端執行, 121
 - 整個系統, 45
 - 應用程式搜尋路徑, 138
 - 點陣圖搜尋路徑, 142

鍵

- 鍵盤對映, 重設, 281

點

- 點陣圖, 210
 - 如何尋找檔案, 210, 211
 - 命名慣例, 210

點陣圖 (繼續)
搜尋路徑, 211

識

識別特性
對話方塊, 163
欄位, 165

歡

歡迎訊息
自訂, 40
預設, 28
變更, 28
顯示, 39, 40

變

變數
動作定義中, 188
環境, 請參閱環境變數, 45

顯

顯示色彩, 最大的分配數目, 271
顯示器的特定階段作業, 47
顯示器特定的資源, 46

