

IBM

@server

iSeries

效能





@server

iSeries

效能

目錄

效能	1
V5R2 新增功能	1
新增功能：監督程式	2
新增功能：資料收集服務	2
新增功能：效能工具授權程式	3
新增功能：Performance Management/400	5
新增功能：效能探究程式	5
列印此主題	7
效能規劃	8
設定系統基準性能測試	9
決定擴充系統的時機及方法	9
Capacity Upgrade on Demand	10
Capacity Upgrade on Demand 概念	10
Capacity Upgrade on Demand 試用期	11
Capacity Upgrade on Demand 的軟體授權注意事項	11
Processors On Demand 啟動碼	12
Electronic Service Agent 基本要求	12
準備 Capacity Upgrade on Demand	12
Capacity Upgrade on Demand 的容量規劃	12
決定啟動處理器的時間	13
設定 Capacity Upgrade on Demand 的環境	13
訂購 Capacity Upgrade on Demand	14
傳送重要產品資料到 IBM	14
啟動待用處理器	15
試用期間啟動待用處理器	15
永久啟動待用處理器	16
實務範例：Capacity Upgrade on Demand	17
選取效能管理策略	17
設定環境以管理效能	19
管理 iSeries 效能	19
追蹤效能	20
調查效能問題	20
識別效能問題	21
識別及解決常見的效能問題	21
收集系統效能資料	22
收集系統資源使用率相關資訊	23
收集應用程式效能相關資訊	23
傾出追蹤資料	24
實務範例：在升級或移轉後增進效能	24
顯示效能資料	25
調整效能	25
基本效能調整	26
自動調整效能	27
管理電子商業效能	28
從屬站效能	28
網路效能	28
OS/400 中的 Java 效能	29
HTTP Server 效能	30
WebSphere 效能	30

用於效能管理的應用程式	31
資料收集服務	33
從資料收集服務資料建立資料庫檔案	36
從現有的資料收集物件建立資料庫檔案	37
自訂資料資料收集	37
資料收集服務的時區注意事項	38
資料收集服務中使用者定義的種類	38
資料收集程式建議及需求	39
範例：實施使用者定義的種類	40
管理資料收集物件	47
使用者定義的異動	47
C++ 範例：將使用者定義的異動整合到資料收集服務中	48
Java 範例：將使用者定義的異動整合到資料收集服務中	51
效能資料檔	55
含有時間間隔資料的效能資料檔	56
效能資料檔：檔案縮寫	57
效能資料檔：資料收集服務系統種類與檔案關係	57
效能資料檔案：配置資料庫檔案的欄位資料	59
效能資料庫檔案：追蹤資料庫檔案的欄位資料	59
iSeries 領航員監督程式	60
監督程式概念	61
配置監督程式	61
監督計量單位	61
實務範例：iSeries 領航員監督程式	63
實務範例：系統監督程式	63
實務範例：工作監測程式	64
實務範例：訊息監督程式	65
圖形歷程	66
圖形歷程概念	66
使用圖形歷程	67
Performance Management/400	67
PM/400 概念	68
PM/400 的優點	68
PM/400e 提供的作業支援服務	68
PM/400 的資料收集注意事項	69
配置 PM/400	69
啓動 Performance Management/400	70
自訂 PM/400	76
管理 PM/400	78
停用 PM/400	78
變更 PM/400 聯絡資訊	79
使用 PM/400 排程工作	79
從 PM/400 分析省略項目	80
暫時關閉 PM/400	80
顯示 PM/400 狀態	80
檢視 PM/400 報告	81
效能工具	81
效能工具概念	81
效能工具提供的功能	82
管理程式和代理程式特性比較	82
顯示效能資訊	83
安裝和配置效能工具	84

效能工具報告	84
效能探究程式	86
效能探究程式概念	87
效能探究程式定義	88
效能探究程式資料庫檔案	90
效能探究程式報告	91
效能探究程式優點	91
配置效能探究程式	92
結束效能探究程式	92
iDoctor for iSeries	93
Performance Trace Data Visualizer (PTDV)	93
效能管理 API	94
“使用” OS/400 效能的指令	94
延伸的適應性快速記憶體	94
延伸的適應性快速記憶體概念	95
延伸的適應性快速記憶體 限制和注意事項	95
啓動延伸的適應性快速記憶體	96
延伸的適應性快速記憶體模擬器	96
取得延伸的適應性快速記憶體	97
Workload Estimator for iSeries	97
iSeries Navigator for Wireless	97
PATROL for iSeries (AS/400) - Predict	98
實務範例：效能	98
相關資訊	98

效能

在系統效能的管理方面您做了多少投資？您企業需求的改變有時比預期的更快。為有效反應企業變更，您的系統也必須跟著變更。乍看之下，管理系統又像一個費時的工作。但由於系統執行效率提高所以投資很快成功，這會反映在您的企業上。有效率的原因是因為所做的變更都有規劃及管理。

管理 iSeries 伺服器效能是一項複雜的作業，需要對 iSeries 工作管理充份瞭解。要瞭解影響系統效能的所有不同程序，對經驗不足的使用者而言是一種挑戰。解決效能問題需要有效使用到大量工具，每一個工具都有自己的唯一需求及支援的功能。即使在完成效能資料的收集及分析後，您可能還是無法知道如何運用該資訊。

本主題將協助您逐步瞭解與效能管理相關的作業及工具。

V5R2 新增功能

本主題說明此版次的新增資訊或重要的變更。

列印此主題

如果您想要列印此資訊，請在此[列印 PDF](#)。

效能規劃

設定 iSeries 伺服器的效能目標可讓您使用可衡量的效能基準性能測試來比較效能資料。本主題說明如何設定基準性能測試及如何在稍後使用它們。

設定您的環境以管理效能

iSeries 伺服器加入了功能強大的應用程式來管理系統效能。不過，為符合您唯一企業環境的特定需求，這些應用程式必須經過適當配置。瞭解如何配置應用程式來定期收集、監督及分析效能資料。

管理效能

效能管理是最佳化電腦系統使用率所必需的，其方法是測量現行功能、認清趨勢，以及經由適當調整來滿足一般使用者及管理需求，例如回應時間或工作產能。這對於維護企業效率及避免長期暫停正常企業活動是必要的。因此，管理效能是您每日作業的一部份。

用於效能管理的應用程式

管理 iSeries 系統效能需要使用到各種特殊的應用程式。每一個應用程式皆提供系統效能的深入瞭解。本主題說明數個應用程式及各應用程式的預定用途。

實務範例：效能

瞭解效能管理的最佳方式之一，就是參考說明如何在您企業環境中使用這些應用程式或工具的範例。請參考這些範例。

相關資訊

IBM^(R) 相關資訊包含技術、竅門及「作法」資訊。

註：本主題包含程式碼範例。有關重要的法律資訊，請參閱程式碼不保事項聲明。

V5R2 新增功能

下面說明本版次的新增功能：

- **iSeries 領航員監督程式**

找出新監督程式支援及現有監督程式的變更。

- **資料收集服務**
讀取效能資料庫檔案變更的相關資訊，以及新功能的相關資訊。
- **效能工具**
讀取報告變更的相關資訊。
- **Performance Management/400**
讀取 PM/400 如何自動收集資料的相關資訊。
- **效能探究程式**
讀取效能探究程式資料庫檔案變更的相關資訊，以及新功能的相關資訊。
- **Capacity Upgrade on Demand**
R閱讀有關 Capacity Upgrade on Demand 的 iSeries 支援。
- **管理電子商業效能**
閱讀有關在電子商業環境中，可用來管理 iSeries 效能的作業、注意事項及資源。
- **主記憶體關係**
閱讀有關處理和緒如何達到記憶體與處理器資源的增進關係。

新增功能：監督程式

下面說明本版次的新增功能：

- **檔案監督程式**：可讓您管理多重平台環境的功能。您將可以執行指令、監督檔案並在「iSeries 領航員」中管理多重平台端點系統。
- **B2B 活動監督程式**：一種活動監督程式，可讓您在指定的合併系統上檢視具有詳細資訊的異動資料，並自動執行 OS/400 指令以控制整體異動計數或異動持續時間。
- **系統監督程式計量單位**現在包含與「點對點通信協定」相關的資訊。數種計量單位已新增至 CPU 使用率內容。這些新計量單位關聯檔案系統、串流檔、異動日誌作業及計數器組。

新增功能：資料收集服務

下面說明本版次的新增功能：

安裝新版次

在升級至新版次時，您有數種選擇可以決定如何處理效能資料。您可以使用「轉換效能資料 (CVTPFRDTA)」指令，繼續在集合檔案庫中轉換資料。您可以使用新的自動資料收集支援。如果您沒有轉換效能資料庫檔案，且如果「資料收集服務」已啟動且您已指定自動建立資料庫檔案的選項，則會建立前一版次檔案的效能檔案庫。效能資料庫檔案會移至該檔案庫。此動作會建立新檔案，並從前一版次中保留現有的資料檔。最後，如果您沒有轉換效能資料庫檔案且沒有執行「資料收集服務」，則可以使用「刪除檔案 (DLTF)」指令 (DLTF 檔案庫/QAPM*) 來刪除 QAPMxxxx 檔案。

使用者定義的種類

V5R2 之前，所有效能集合種類都是預先定義的，且系統會隨附所有資料收集程式。從 V5R2 開始，您可以定義您自己的效能收集種類。使用者定義的效能收集種類可讓您：

- 新增效能集合種類至收集器。登記收集器資料種類 (QypsRegCollectorDataCategory) 會在「管理中心」中「資料收集服務」功能的一或多個收集器定義中，新增使用者定義的資料種類。
- 提供新種類資料收集程式以收集資料。
- 將收集的種類資料儲存在管理資料收集物件中。
- 依照系統定義種類的相同方式，排程及執行種類的資料收集
- 使用管理資料收集物件 API，在管理資料收集物件中存取新種類的收集資料。

Domino 及 HTTP 伺服器 (Apache 驅動) 使用此新特性，將效能資料整合到「資料收集服務」。

使用者定義的異動

從 V5R2 開始，「資料收集服務」作業可利用效能探究程式異動界限 API，讓應用程式使用者可以定義自己的異動。「資料收集服務」資料種類是 USRTNS，且與使用者定義異動相關的效能資料庫檔案是 QAPMUSRTNS。這些 API 包括下列：

- 啟動異動
- 結束異動

效能資料庫檔案

所有資料庫檔案 (QAPMAPPN 除外) 現在都是交談式的，表示您可以在直欄中按照字母順序來排序文字，且可以在表格中搜尋特定的字組。

下表顯示新的及變更的資料庫檔案。

資料庫檔案	說明
QAPMJOBWT	含有工作、作業及緒等候條件之資料的新檔案。
QAPMJOBWTD	含有檔案 QAPMJOBWT 中計數器組說明的新檔案。
QAPMHTTPB	含有 IBM HTTP Server (Apache 驅動) 之基本資料的新檔案。
QAPMHTTPD	含有 IBM HTTP Server (Apache 驅動) 之詳細資料的新檔案。
QAPMDOMINO	含有 Domino for iSeries 資料的新檔案。
QAPMPPP	含有「點對點通信協定」資料的新檔案。
QAPMUSRTNS	含有使用者定義異動資料的新檔案。
QAPMSYSTEM	支援檔案系統計數器及異動日誌計數器的新欄位。
QAPMSYS	新欄位，提供在單一系統的不同分割區上，判斷系統時鐘間差別的方法。
QAPMECL	新欄位，報告未支援的通信協定中捨棄的訊框數。
QAPMETH	新欄位，報告未支援的通信協定中捨棄的訊框數。
QAPMCONF	新記錄鍵。
QAPMIOPD	支援 I/O 配接卡資料的新欄位。
QAPMMIOP	支援處理程序時間的新欄位。
QAPMJOBMI	新欄位，支援檔案系統計數器及異動日誌計數器。
QAPMDISK	新增及變更的欄位，支援獨立的 ASP 相關變更。
QAPMJOBOS	支援檔案系統計數器的新欄位。

新增功能：效能工具授權程式

下面說明本版次的新增功能：

- BEST/1 將從「效能工具」中撤消，並自本版次開始生效。您可以考慮使用調整容量需求的選擇方案，包括 (但不限於) IBM Workload Estimator (整合 PM/400) 及適用於分散式系統的 BMC PATROL for iSeries (AS/400) - Predict 產品。將從「效能工具」授權程式中撤消下列指令或功能表選項的支援，並自 V5R2 開始生效：
 - 啟動 BEST/1 (Start BEST/1)
 - 分析 BEST/1 模型 (ANZBESTMDL)
 - 建立 BEST/1 模型 (CRTBESTMDL)
 - 轉換 MDLSYS 檔案 (CVTMDLSYSF)
 - 刪除 BEST/1 模型 (DLTBESTMDL)


- 列印 BEST/1 檔案 (PRTBESTF)
- 檢查指定效能資料成員的版本 (QCYCHKV)
- 將 System/36 檔案從 MDLSYS 轉換至 BEST/1 (QCYCVTBD)
- 尋找指定效能資料成員的第一個及最後一個日期 (QCYFLDT)
- 接收前一層的訊息並重送至呼叫層 (QCYRSNDM)
- 啟動 MDLSYS (MDLSYS)
- 產能規劃/模型化已從 PERFORM 功能表中移除。
- 現在可以從「iSeries 領航員」中，以「效能工具」外掛程式的方式使用與顯示效能資料 (DSPPFRDTA) 指令相關的功能。介面提供的功能幾乎與 DSPPFRDTA 指令相等。它可讓您產生及檢視「效能工具」授權程式報告的子集。
- 更新「列印元件報告 (PRTCPTRPT)」指令以反映數種加強功能，其中包括下列：
 - 新選項種類，可讓您選取要列印的工作類型。此選項對於在伺服器模型分析互動式工作非常有用。
 - 新選項種類，可讓您選取要併入報告中的工作優先順序。當系統具有高額外執行時間時，對於取得高優先順序工作總計及尋找所有優先順序 00 工作，此選項非常有用。
 - 新直欄，報告寫入快取超限運轉的百分比 (% 寫入快取超限運轉)。
 - 「資料庫日誌登載摘要」區段含有新的次區段，顯示與異動日誌計數器與作業相關的資訊。新計數器的資料儲存在 QAPMJOBMI 檔案中。
- 「列印系統報告 (PRTSYSRPT)」指令已更新，以反映數種加強功能，包括下列各項：
 - 「通信摘要」現在包含點對點通信協定 (PPP) 資訊。PPP 通信協定現在也會顯示在「資源報告」的「通信明細」區段。「顯示效能資料 (DSPPFRDTA)」指令也會顯示「顯示通信線路明細」顯示畫面中的「點對點通信協定」資訊。
 - 新增區塊，顯示「HTTP 伺服器」工作處理的異動相關資訊。此資訊會在報告底端顯示每一個間隔的摘要與平均行。
 - 在「工作量」、「資源使用率」及「資源使用率擴充」區段中併入新種類，以顯示有關非交談式伺服器工作的其餘明細。
- 「使用系統活動」顯示畫面會顯示現行處理產能。在共用處理器環境中，此資訊是非常有用的，特別是在分割區容量變更，但虛擬處理器數目未變更的情況下。「使用系統活動」顯示畫面也會顯示工作或作業處於等待狀態時所花費的時間總計百分比。您可以使用選項 16 (等待明細) 來檢視等待種類的清單。在 QAITMON 檔案中新增數個與等待時間統計資料相關的欄位，該檔案是由 WRKSYSACT 指令建立。
- 「異動報告」的「物件的擷取/鎖定衝突摘要」區段最多可顯示 500 個物件。追蹤資料可能會有 500 個以上的擷取/鎖定衝突。因此，將會在此區段中顯示 500 個最重要的物件。
- 「啟動效能追蹤 (STRPFRTRC)」指令現在支援較大的追蹤資料表。
- 報告中部份欄位的大小已增加，以顯示較大的值。變更會影響下列報告。

報告	區段	欄位
系統	資源使用率	每秒磁碟 I/O
系統	資源使用率擴充	實體磁碟 I/O、邏輯資料庫 I/O、磁碟 I/O (同步與非同步)、DIO/秒 (同步與非同步)
元件	元件間隔活動	每秒磁碟 I/O
工作間隔	非交談式作業摘要	每秒 I/O 數
工作間隔	非交談式作業明細	Nbr I/O /秒

新增功能：Performance Management/400

下面說明本版次的新增功能：

PM/400 可自動收集效能資料，除非您是從前一版次中升級，且在前一版次中關閉 PM/400 效能資料收集。在您准許將它傳送至 IBM 之前，資料不會傳輸至 IBM。自動收集效能資料的好處是當您需要資料時，就可以取得它，且只要您一啟動 PM/400，就可以接收 PM/400 報告。

如需 PM/400e 報告的最新資訊，請跳至 PM/400 網站 。

轉換效能資料

因為在您安裝新版次後，效能資料庫檔案是屬於舊版，您必須執行容許收集效能資料的動作才能繼續。如果您使用 Performance Management/400、「資料收集服務」或「效能工具」授權程式，且您已在安裝新版次之前收集資料，則您可以選取下列選項，以決定如何處理前一版次的效能資料：

- 轉換資料
使用「轉換效能資料 (CVTPFRDTA)」指令，轉換集合檔案庫中的資料。
- • 使用自動資料收集支援
如果您沒有轉換效能資料庫檔案，且如果「資料收集服務」已啟動且您已指定自動建立資料庫檔案的選項，則會建立前一版次檔案的效能檔案庫。效能資料庫檔案會移至該檔案庫。此動作會建立新檔案，並從前一版次中保留現有的資料檔。您應瞭解下列項目：
 - 建立的檔案庫名稱是 QPFRDvrmmn，其中 vrm 是現行版本、版次與修改，nn 是從 01 開始的唯一序號，例如，QPFRD52001。
 -
 - 建立的檔案庫具有 *EXCLUDE 公用權限。檔案庫是有 QSYS 使用者設定檔所擁有，且原始檔案庫的擁有者具有 *ALL 權限。
 -
 - 所有 QAPMxxxx 檔案已移動。
 -
 - 如果您不想保留前一版次中的資料，則可以使用「刪除檔案庫 (DLTLIB)」指令 (DLTLIB qprdrvrmnn) 刪除 QPFRDvrmmn 檔案庫
- • 刪除 QAPMxxxx 檔案
如果您沒有轉換效能資料庫檔案且沒有執行「資料收集服務」，則可以使用「刪除檔案 (DLTF)」指令 (DLTF 檔案庫/QAPM*) 來刪除 QAPMxxxx 檔案。

新增功能：效能探究程式

下面說明本版次的新增功能：

- 效能探究程式指定現在需要使用者具有 *SERVICE 權限。
- 新的 CL 指令可讓您建立過濾器，以減少您擷取的資料數量。
 - 新增 PEX 過濾器 (ADDPEXFTR)
 - 使用 PEX 過濾器 (WRKPEXFTR)
 - 移除 PEX 過濾器 (RMVPEXFTR)
 - 已強化 STRPEX 指令以支援過濾器。

- 效能探究程式現在使用管理資料收集物件 (*MGTCOL)，將收集的資料轉換到資料庫檔案。「建立 PEX 資料 (CRTPEXDTA)」指令可執行轉換。
- 新增下列事件至「新增 PEX 定義 (ADDPEXDFN)」指令：
 - 應用程式事件 (APPEVT)
 - 「可攜式應用程式解決方案環境 (PASE)」事件 (PASEEVT)
 - 異動日誌事件 (JRNEVT)
 - iSeries NetServer、「檔案伺服器」及「網路檔案系統伺服器與從屬站」事件 (FILSVREVT)
 - 同步化事件 (SYNCEVT)
 - 專家快取事件 (EXPCCHEVT)
- 在 ADDPEXDFN 指令的「作業系統」事件 (OSEVT) 參數 *HOSTSVRCNN 中新增值。此值與 *DBSVRCNN 值相同。*HOSTSVRCNN 是建議值。
- 提供可讓您使用效能探究程式定義的新指令：「使用 PEX 定義 (WRKPEXDFN)」指令。
- 在結束效能探究程式階段作業時，可以顯示未完成的新集合狀態。此狀態表示集合異常結束。
- 此版次的新資料庫檔案包括下列：

新檔案	說明
QAYPEPROCI	此檔案含有與集合資料中每一個指示位址相關的程序資訊。其中含有之前是在 QAYPEMII、QAYPELICI、QAYPENMI 及 QAYPENLIC 檔案中的資料。
QAYPECFGI	此檔案含有集合定義資訊。其中含有之前是在 QAYPECOFG、QAYPECICFG、QAYPESTCFG、QAYPETRCFG 及 QAYPEHWCFG 檔案中的資料。
QAYPEFTRI	此檔案含有集合過濾器定義資訊。
QAYPELTASK	此檔案含有定義資訊的作業名稱與號碼。
QAYPERINF	此檔案含有雜項解決方案資訊，例如，每一個 IP 位址的字串版本。
QAYPETSWSW	此檔案含有作業切換事件專用的資訊。其中含有之前是在 QAYPEBASE 檔案中的資料。

- 此版次中刪除的資料庫檔案包括下列：

刪除的檔案	資料移動位置
QAYPECICFG	QAYPECFGI
QAYPECOCFG	QAYPECFGI
QAYPEHWCFG	QAYPEPROCI
QAYPELICI	QAYPECFGI
QAYPELNAMT	QAYPELTASK
QAYPELNUMT	QAYPELTASK
QAYPEMII	QAYPEPROCI
QAYPENLIC	QAYPEPROCI
QAYPENMIC	QAYPEPROCI
QAYPEPERD	因為已不再支援週期性模式，所以已不再需要表格。
QAYPEPSUM	現在可以從 QAYPEPPANE 及 QAYPETASKI 資料計算出設定檔摘要資料。
QAYPEPWDW	此檔案含有未使用的設定檔摘要資料。

刪除的檔案	資料移動位置
QAYPESTCFG	QAYPECFGI
QAYPES36	不再支援 System/36 事件。
QAYPETRCFG	QAYPECFGII
QAYPETRCPT	QAYPETIDX
QAYPEUNKWN	QAYPEUSRDF

- 此版次中資料庫檔案的變更包括下列：
 - 字元資料欄位現在使用預設的 CCSID 65535。
 - 8 位元組位址、指示與區段現在均儲存在 8 位元組十六進位欄位中，而不是將每一個位元組轉換成 2 十六進位字元並儲存在 16 位元組字元欄位或 16 位元組十六進位欄位。
 - 儲存在壓縮十進位數欄位中的 TRUE 數值資料已變更為整數欄位：
 - 5 位數壓縮十進位數欄位變更為 9B 整數欄位。
 - 10 位數壓縮十進位數欄位變更為 18B 整數欄位。
 - 20 位數壓縮十進位數欄位變更為 18B 整數欄位，除非值大於 8 位元組帶正負號的整數。

列印此主題

如果要檢視或下載效能主題的 PDF 版本，請選取效能 (大約 350 KB 或 95 頁)。這個 PDF 不包含效能資料庫表格資訊。

若要檢視或下載效能資料庫表格資訊的 PDF 版本，請選取效能資料庫表格 (大約 350 KB 或 125 頁)。

您也可以檢視或下載下列相關主題：

- 管理中心 (大約 250 KB 或 55 頁) 包含關於如何設定「管理中心」端點系統和系統群組的資訊，也包含關於「管理中心」可協助您簡化伺服器管理作業的所有方法，例如：
 - 管理使用者與群組
 - 包裝和傳送資料
 - 執行指令
 - 使用「管理中心」排程器或「進階工作排程器」排程作業或工作。
- 工作管理 (大約 173 KB 或 40 頁) 說明這些工作管理概念：
 - 工作的期限
 - 每日工作管理
 - 系統結構
 - 如何完成工作








您也可以檢視或列印下列 PDF：

- 手冊：
 - Performance Tools for iSeries  (大約 400 頁)
 - iSeries Performance Capabilities Reference 

鏈結到含有本書的最新版本的網站。這個參照提供關於對效能評比、產能規劃和規劃伺服器效能來說很有用的伺服器效能的高度技術性資訊。


– 這個參照提供關於對效能評比、產能規劃和規劃伺服器效能來說很有用的伺服器效能的高度技術性資訊。

- 紅皮書：

- AS/400 Performance Management  (大約 504 頁)
- AS/400 HTTP Server Performance and Capacity Planning  (大約 205 頁)
- Java and WebSphere Performance on IBM eserver iSeries Servers  (大約 235 頁)
- Managing OS/400 V5R1 with Operations Navigator  (大約 550 頁)
- DB2 UDB/WebSphere Performance Tuning Guide  (大約 360 頁)
- Lotus Domino for AS/400: Performance, Tuning, and Capacity Planning  (大約 550 頁)
- AS/400 Performance Explorer Tips and Techniques  (大約 550 頁)

如果要在工作站儲存要檢視或列印的 PDF：

1. 使用瀏覽器開啓 PDF (按一下上面的鏈結)。
2. 在瀏覽器的功能表，按一下**檔案**。
3. 按一下**另存新檔 ...**
4. 瀏覽到您要儲存 PDF 的目錄。
5. 按一下**儲存**。

若您需要 Adobe Acrobat Reader 以檢視或列印這些 PDF，您可以從 the Adobe 網站 (www.adobe.com/prodindex/acrobat/readstep.html)  下載複本。

效能規劃

規劃系統的效能需要設定效能目標，根據那些目標建立基準性能測試，以及規劃系統的成長。本節引導您執行必要步驟以規劃系統的效能。

在規劃系統的效能時，您必須完全瞭解您的系統處理的業務需求，而且能夠將那些業務需求轉換成效能目標。請記住當業務需求擴大，效能目標也必須擴大。

也許最好的開始方式是預估在尖峰業務期間電腦系統需要的最大每小時和每日交談式異動產能。在預估該產能之後，您可以決定本端和遠端工作站可接受的平均回應時間。您應該思考一般批次處理要多久，如何排程它們以便及時完成它們來達成業務需求。

然後您可以建立一組基本統計值，並將它們記錄在效能目標計劃中，包括：

- 每小時尖峰異動數目
- 每天尖峰異動數目
- 本端工作站的可接受的平均回應時間
- 尖峰交談式異動數目
- 主要排定的批次工作清單 (含有執行它們的時間) 和它們的預期持續時間

- 可能需要的其他未排定的批次工作清單

如果要規劃效能，請完成下列作業：

設定系統基準性能測試

設定良好基準性能測試會提供適當調整系統的效能資料。來自系統變更前後的這些效能基準性能測試提供疑難排解和規劃的重要資訊。

決定擴充系統的時機及方法

當業務需求改變時，系統也必須跟著改變。如果要準備進行變更，您必須建立現行系統的模型，然後看看如果變更系統、配置或工作量會發生什麼狀況。

Capacity Upgrade on Demand

Capacity Upgrade on demand 可在變更企業需要更多資源時，讓您新增處理器至 iSeries 伺服器。您可以訂購及啟動處理器，以試用、暫時或永久為基礎。

選取效能管理策略

不同業務需求需要不同效能管理策略。下面是三個基本企業運作模式以及它們的建議效能管理策略。

設定系統基準性能測試

您應該建立系統基準性能測試之後才對系統配置執行主要變更，例如新增新交談式應用程式或執行系統升級。維護正確基準性能測試資訊可以提供基本的疑難排解資訊。至少基準性能測試應包含來自資料收集服務的現行資料收集物件。根據您的環境您可能需要使用效能探究程式來維持更詳細的資訊。

設定基準性能測試需要下列各項：

- 有正確 iSeries 配置
- 應用程式和資料具代表性且有效
- 有要使用的程式和軟體的正確版本
- 有必要數目的使用者和工作站可執行測試
- 為每一位使用者完整定義異動

若沒有可讓您模擬工作站使用者的特殊設備，則要執行交談式工作量的有意義基準性能測試幾乎是不可能的。當然，要執行批次基準性能測試不會像測試交談式應用程式效能那麼複雜，因此對於這類測試來說，上述最前面三點仍然有效。不過，對於並行批次和交談式工作（這通常是實際客戶環境）設定系統基準性能測試來說，也需要適當的使用者和工作站數目。

決定擴充系統的時機及方法

隨著業務需求發展，您的系統需求也會增加。如果要規劃將來系統需求及成長，您必須決定萬一變更系統、配置或工作量時會發生的狀況。這個處理稱為趨勢分析而且應每月執行它。當系統接近資源產能引導線時，您可能需要更經常地收集這個資料。

應分別為交談式環境和批次環境執行趨勢分析。如果公司過度使用某個應用程式，您可能需要為該應用程式執行趨勢分析。另一個需要追蹤的重要環境是月底處理。持續收集趨勢分析資料很重要。如果系統的尖峰工作時間介於 10:00 AM 與 2:00 PM 而您剛好收集這段期間的趨勢分析資料，那麼請勿比較這個資料與收集自其他時段的資料。

如果要執行適當產能規劃和效能分析工作，您必須收集、分析、維護和保存效能資料。IBM 提供一些工具，這些工具協助您進行產能規劃、資源預估和調整大小：

Performance Management/400 (PM/400)

PM/400 完全自動收集資料、分析資料和保存資料，而且提供容易瞭解的彙總效能和產能資訊。PM/400 透過持續分析重要效能指示符協助您規劃和管理系統資源。這個功能檢附於 OS/400 授權程式。除了啟動此功能和定期地檢查是否有收集資料並傳給 IBM 之外，您什麼都不必做。所有的資料收集站都受到網路保護，而且 PM/400e 服務只將非專屬效能資料傳給 IBM。轉送的時間完全由您來控制。

Workload Estimator

Workload Estimator 是一項工具，它協助您根據特定工作量類型的預估工作量調整系統需求大小。透過 web 型應用程式，您可以調整必要的 iSeries 系統的升級大小，以便按照 PM/400 的報告來容納現存系統的使用率、效能和成長。當作另一種選擇，調整大小也可以包含可新增像 Domino、Java 和 WebSphere 等特定應用程式的能力，或包含能夠在一個系統上合併多重 AS/400 或 iSeries 傳統 OS/400 工作量的能力。這種功能可讓您根據您本身系統的現存使用率資料來規劃將來的系統需求。這個應用程式不支援「處理器即時要求 (Processor On-Demand)」環境或含有邏輯分割區的环境。

PATROL for iSeries (AS/400) - Predict

本產品透過自動化高可用性和最佳效能需要的許多日常管理作業來協助管理 iSeries 效能。此外，它提供詳細產能規劃資訊協助您規劃 iSeries 環境的成長。

關於建立和實施效能策略的詳細資訊，請參閱選取效能管理策略。

Capacity Upgrade on Demand

Capacity Upgrade on Demand (CUoD) for iSeries 是一種特性，提供您可動態啟動一或多個選擇伺服器模型之中央處理器的能力，而您不需要重新啟動伺服器或岔斷業務。您可以訂購及啟動已安裝在伺服器上的試用或永久的待用處理器。

使用 CUoD，您可以啟動額外的處理器，僅需要付費增加需求所造成的新處理動力。您可以增加處理器容量，而不中斷現行作業。

使用此主題以瞭解 CUoD 的運作方式，以及活用它的方法。

Capacity Upgrade on Demand 概念

從此處開始，概念上瞭解 CUoD。

準備 Capacity Upgrade on Demand

考慮重要規劃和準備基本要求。

訂購 Capacity Upgrade on Demand

您必須訂購啟動特性以取得啟動碼，才能啟動待用處理器。

啟動待用處理器

使用此資訊以啟動 iSeries 伺服器上一或多個待用處理器。

實務範例：Capacity Upgrade on Demand

使用此資訊的範例步驟，以讓管理者可逐步規劃、訂購及啟動額外的容量。

Capacity Upgrade on Demand 概念

Capacity Upgrade on Demand (CUoD) 提供您在免費 14 天的試用期時，在選取的 iSeries 機型上啟動額外的處理器的功能，或透過購買永久處理器啟動。此功能藉由啓用快速及經濟的方式新增新工作量的容量，新增 iSeries 伺服器的有效值，可讓您的伺服器適應非預期的效能需求。

選取 iSeries 機型會與**啟動處理器**數量一起出貨。啟動處理器是當出貨時，已在 iSeries 伺服器中啟動的處理器。**待用處理器**是與伺服器一起出貨的處理器，但要啟動後才能使用。待用處理器可以在 14 天的試用期暫時啟動，或藉由購買啟動特性及輸入提供的啟動碼可永久啟動。下表列出每個機型可用的啟動及待用處理器數量。

機型	處理器特性	啟動處理器	待用處理器	安裝的處理器
825	2473	3	3	6
830	2349	4	4	8
830	2351*	1	7	8
840	2352	8	4	12
840	2353	12	6	18
840	2354	18	6	24
840	2416	8	4	12
840	2417	12	6	18
840	2419	18	6	24
870	2486	8	8	16
890	2487	16	8	24
890	2488	24	8	32
890	2497	16	8	24
890	2498	24	8	32

* 限制可用性

下列概念提供您需要利用 CUoD 的資訊。

Capacity Upgrade on Demand 試用期

利用免費的 14 天試用期，量規額外處理器容量的益處。

Capacity Upgrade on Demand 的軟體授權注意事項

閱讀啟動 CUoD 如何影響軟體層級。

Processors on demand 啟動碼

一旦決定要永久啟動部份或全部待用處理器，您必須訂購及購買一或多個 Processors On Demand 啟動特性。這將提供您可讓您啟動待用處理器的啟動碼。

Electronic Service Agent 基本要求

使用 Electronic Service Agent 快速及準確傳輸啟動 CUoD 待用處理器必需的重要產品資料。

Capacity Upgrade on Demand 試用期: 客戶想要在不付費的情況下評估使用待用處理器時，Capacity Upgrade on Demand 提供 14 天的試用期。一旦啟動，起始啟動伺服器之後試用期使用一次，且每次輸入啟動碼時也能使用一次。

一旦啟動，試用期的期限為 14 個開啓電源的天數。試用期時鐘只有在伺服器電源開啓時才起作用。所有處理器 (包括啟動和待用) 都會在試用期間啟動。也有可能停止及重新啟動試用期。

請參閱試用期間啟動待用處理器，以試驗此特性。

Capacity Upgrade on Demand 的軟體授權注意事項: 許多商業夥伴使用伺服器上可用的處理器特性碼系統值，以設定軟體的軟體授權費。此系統值繼續可在具有待用處理器的伺服器上使用，但應注意處理器特性碼系統值仍保持一樣，不管啟動多少待用處理器都一樣。

在伺服器上以處理器數量為基礎收軟體授權費的軟體提供者，根據經驗使用「軟性」承諾方法。每個處理器啟動時，會使客戶的責任為通知和付款與伺服器上之任何軟體相關的必要軟體授權費，此伺服器具有由處理器數量所授權的軟體。


啟動處理器不會變更軟體等級。

註：透過其中一項 IBM 配置器訂購 Processors on Demand 啟動特性時，IBM 軟體授權的額外收費是為目前安裝在客戶伺服器 (由處理器授權) 之軟體產品所建立。

Processors On Demand 啟動碼： 當您決定要永久啟動部份或全部待用處理器時，您必須訂購及購買一或多個 Processors On Demand (POD) 啟動特性。

下訂單後，訂單記錄會結合從伺服器收集的重要產品資料 (VPD)。此資訊可用於產生伺服器特定的 POD 啟動碼。您可以想成是用來啟動待用處理器的容量授權碼。


此啟動碼將登記在 IBM 網站以快速存取。這通常會在一個工作天 (24 小時) 內發生，假設訂單與從伺服器收集的必要 VPD 皆有。一旦產生啟動碼，您可在 iSeries Capacity Upgrade on Demand 網站使用系統類型和序號存取：

<http://www.ibm.com/servers/eserver/iseries/hardware/ondemand> 

有關訂購 POD 啟動特性及接收 POD 啟動碼的指示，請參閱訂購 Capacity Upgrade on Demand。

Electronic Service Agent 基本要求： 當您下訂單購買 Capacity Upgrade on Demand 時，IBM 會結合您的訂單資訊及伺服器的重要產品資料 (VPD)，以建立解除鎖定及啟動伺服器之待用處理器所需的啟動碼。

可使用 Electronic Service Agent 以電子方式將 VPD 資訊傳送到 IBM，這是「最大支援」的一部份，是 IBM 對 IBM iSeries 的綜合技術服務和獨家支援服務。Electronic Service Agent 是伺服器上不收費的授權程式 (5798-RZG)，其設計目的在於以週期性、客戶定義的時間表監視事件，並將伺服器庫存資訊傳輸到 IBM。

有關 Electronic Service Agent 的完整文件，包括安裝資訊，請參閱 Electronic Service Agent for iSeries 使用手冊 

準備 Capacity Upgrade on Demand

Capacity upgrade on Demand 可讓您新增處理器容量而不用岔斷伺服器作業。然而，若要完整地整合新容量，您應該在下訂單之前準備伺服器。

Capacity Upgrade on Demand 的容量規劃

此主題包括在 CUoD 型伺服器機型執行容量規劃的注意事項與資源。

決定啟動處理器的時間

此主題討論可監督資源使用率趨勢的工具及建議需要額外容量的時間，以及工具如何在具有待用處理器的伺服器上報告 CPU 使用率。

設定 Capacity Upgrade on Demand 的環境

準備伺服器與新處理器容量整合，並在必要時準備訂購額外的容量。

Capacity Upgrade on Demand 的容量規劃： 具有待用處理器之伺服器的容量規劃基本上使用同樣的程序和資源，用於估算其他 iSeries 機型。綜合的工具、資源和供應組可用於協助決定已更新必要的伺服器，以支援具有待用處理器的伺服器。

有關容量規劃的說明，請參照下列資源：

決定擴充系統的時機及方法

此主題概觀討論數個工具，可用於協助您執行容量規劃，以及識別資源使用率中的趨勢。

iSeries Benchmark Center

使用此 IBM 網站取得評比應用程式環境的說明。

iSeries Solutions Center — Capacity Planning Services

此 IBM 諮詢服務可協助您規劃符合您業務成長需求的伺服器解決方案。

註： Workload Estimator 支援 CUoD，但僅傳回建議之作用中處理器數量的預估。待用處理器提供的額外容量不在建議中。

決定啓動處理器的時間： Capacity upgrade on Demand 在您的工作量需要額外資源時，提供可將處理器加入系統的能力。您應該監督系統 CPU 使用率及 CPU 使用率中的趨勢，以決定啓動額外處理器的時機，和需要幾個處理器。

有許多效能工具可用來報告 CPU 使用率資訊。此外，PM/400 可以識別資源使用率中的趨勢，而「iSeries 領航員」監視器提供您較多使用資源的詳細資訊，並在使用率達到預先定義的嚴重層次時警示您。

測量具有待用處理器之伺服器的 CPU 使用率

計算所有可用處理器的平均用量時，系統功能報告 CPU 使用率不包括 CPU 容量總數中的待用處理器。待用處理器在各種報告 CPU 使用率百分比的系統功能中不視為作用中。已使用 CPU 容量（「iSeries 領航員」中 CPU 使用率 CPU 百分比）的百分比，是基於處理器在經歷時間中為作用中的時間量所計算的度量。通常報告為百分比，其中 100% 表示處理器在整個經歷時間中都在工作中。出現多個處理器時，必須調整 CPU 時間以表示所有處理器的平均用量，因此使用率總是以總可用容量的百分比報告。

交談式功能由購買的交談式效能決定。此功能不會因待用處理器的數量而受到影響，且在待用處理器成為作用中時亦不會變更。報告為交談式功能百分比的交談式使用率不會受到 Capacity Upgrade on Demand 技術的影響。在「iSeries 領航員」中，此度量稱之為「CPU 使用率」（交談式效能）。

交談式 CPU 使用率亦報告為總系統 CPU 的百分比。在「管理中心」中，此度量稱之為「CPU 使用率」（交談式作業）。此度量如上述之系統 CPU 使用率，以相同的方式影響具有 CUoD 的伺服器。

設定 Capacity Upgrade on Demand 的環境： 訂購 Processor On Demand (POD) 啓動碼之前，您應該準備環境以訂購及整合額外的容量。

準備訂購

購買 POD 啓動碼時，您也必須提供伺服器的重要產品資料 (VPD)。您可以手動、傳真、電子方式、或使用 Electronic Service Agent 這樣做。使用 Electronic Service Agent 傳送 VPD 資料可避免與手動方法相關的延遲，並可讓 POD 啓動碼在接收訂單的 24 小時內登記。使用試用的其他處理器時，您應該預留傳送及處理訂單得必要的時間，以避免效能岔斷。

若要使用電子方法傳送 VPD 資料，請設定 Electronic Service Agent。

準備額外的容量

若要確定伺服器可完全使用啓動的處理器，您可能須採取下列準備動作：

- 執行 I/O 設定條件
- 執行記憶體升級
- 準備邏輯分割區 (LPAR)

針對使用 LPAR 的伺服器，待用處理器一律與主分割區結合。您必須指派所有處理器給分割區。

設定 Electronic Service Agent: 您可以使用 Electronic Service Agent 傳送 VPD 資料。若要設定 Electronic Service Agent，請遵循下列步驟：

1. 安裝 Electronic Service Agent。

請參照 Electronic Service Agent for iSeries 使用手冊 ，以取得設定及安裝指示。

2. 確定已設定且已啟動 TCP/IP。

有關 Electronic Service Agent 的詳細資訊，請參照 Electronic Service Agent 基本要求。

訂購 Capacity Upgrade on Demand

您可以為新伺服器、機型升級或安裝的伺服器訂購 Processors on Demand (POD) 啟動特性。針對新伺服器或機型升級，您的訂單會包含一或多個 POD 啟動特性。在此範例中，會在伺服器送給您之前輸入 POD 啟動碼。


針對已安裝的伺服器，一旦您決定要永久啟動部份或全部待用處理器，您需要訂購一或多個 POD 啟動特性，然後使用產生的 POD 啟動碼啟動待用處理器。

註：

1. 可能需要數天處理訂單，並登記產生的 POD 啟動碼。您可以在試用期啟動待用處理器 14 天，以滿足工作量基本要求，同時已完成額外容量永久啟動的訂單。
2. 若雜項特性包含在訂單中，將可以快一點處理 POD 啟動特性的訂單。

若要訂購一或多個 POD 啟動特性，請遵循下列步驟：

1. 決定您要啟動的待用處理器數量。
相關資訊，請參閱決定啟動處理器的時間主題。
2. 聯絡 IBM 企業夥伴或 IBM 業務代表，以訂購一或多個 POD 啟動特性。美國的客戶可選擇在 IBM 網站下訂單：

<http://www.ibm.com> 

3. 從伺服器傳送重要產品資料到 IBM。
處理訂單之前，您的訂單與從伺服器收集的重要產品資料 (VPD) 結合一起。接著使用該資訊產生伺服器特定的 POD 啟動碼。POD 啟動碼將郵寄給您，並登記在 iSeries Capacity Upgrade on Demand 網站以快速存取。
4. 在伺服器上輸入 POD 啟動碼，以永久啟動待用處理器。

傳送重要產品資料到 IBM: 當您訂購一或多個「依需求處理器 (POD)」啟動特性時，必須為您的 iSeries 伺服器提供重要產品資料 (VPD) 給 IBM。您可以使用 Electronic Service Agent^(TM) 或列印及傳真資料，收集 VPD 並將之送到 IBM。

若要使用 Electronic Service Agent 收集及傳送 VPD，請遵循下列步驟：

1. 確定已設定 Electronic Service Agent。
2. 啟動 Electronic Service Agent 精靈。

3. 選取**硬體**，以收集及傳送必要的 VPD 資訊。

若要列印及傳真 VPD，請遵循下列步驟：

1. 從系統主控台，在指令行上輸入 STRSST，以啟動系統服務工具 (SST)。登入 SST。

註：若要使用系統服務工具，您需要具有「系統」容量的有效服務工具使用者 ID - 管理者專用權。

2. 選取選項 6 (使用系統容量)，然後按 Enter 鍵。

3. 選取選項 1 (顯示系統容量資訊)，然後按 Enter 鍵。出現「顯示永久系統容量」畫面。

4. 按 F6 以列印資訊。

5. 準備下列資訊的傳真文件：

• 傳真接收者資訊：

– 傳送至：Capacity on Demand Administrator (507-253-7019)

– 傳真號碼：507-253-4553

– 位置：Rochester, Minnesota

• 傳真發送者資訊：

– 客戶名稱：

– 客戶聯繫名稱：

– 客戶地址：

– 客戶電話號碼：

– 客戶傳真號碼：

6. 將 VPD 傳真至下列傳真號碼：

507-253-4553

啓動待用處理器

您可以啓動一或多個已安裝在伺服器上的試用或永久的待用處理器。

試用期間啓動待用處理器

您可以在試用期 14 天內，啓動所有安裝在伺服器上的所有待用處理器。當您的額外容量永久啓動訂單完成時，這可讓您在決定是否購買額外容量或滿足尖峰工作量之前，先行評估額外容量能對您的環境帶來多少益處。

永久啓動待用處理器

您可以永久啓動指定數量的待用處理器。

試用期間啓動待用處理器： 若要在試用期間，啓動所有安裝在伺服器上的所有待用處理器，請遵循下列步驟：

1. 從系統主控台，在指令行上輸入 STRSST，以啓動系統服務工具 (SST)。登入 SST。

註：若要使用系統服務工具，您需要具有「系統」容量的有效服務工具使用者 ID - 管理者專用權。

2. 選取選項 6 (使用系統容量)，然後按 Enter 鍵。出現「使用系統容量」畫面。

3. 選取選項 3 (使用暫時系統容量)，然後按 Enter 鍵。出現「確認啓動暫時系統容量啓動」畫面。

註：若暫時系統容量啓動在使用中或已使用，則不會出現這個選項。

4. 按 Enter 鍵確認系統容量啓動。

5. 跳出「使用系統容量」畫面。

6. 若伺服器是分割的，在您使用新容量之前，您必須先將最近可用的處理器指派到邏輯分割區。

有關將啟動的處理器指派到邏輯分割區的資訊，請參閱處理動力的動態移動。

7. 若伺服器未分割，請遵循下列步驟：
 - a. 選取選項 5 (使用系統分割區)，然後按 Enter 鍵。
 - b. 選取選項 3 (使用分割配置)，然後按 Enter 鍵。
 - c. 在 PRIMARY 旁鍵入 2 (變更分割區處理資源)，然後按 Enter 鍵。
 - d. 輸入處理器新數量的值，可代表作用中處理器的總數。

註：畫面底端的配置錯誤指出已改變預設「主分割區」(邏輯分割區目前使用中)，您必須參照邏輯分割區計畫，以適當地將啟動的處理器指派給伺服器。

- e. 按 Enter 鍵確認變更。
8. 跳出 SST。


您現在可以開始使用新容量。

註：

1. 試用會在 14 天過後自動結束，或者輸入新 Processors on Demand (POD) 啟動碼時結束。
2. 若伺服器已分割，您必須在試用期結束時將待用處理器傳回主分割區。

永久啟動待用處理器：當您購買一或多個「依需求處理器 (POD)」啟動特性時，將收到 POD 啟動碼以啟動待用處理器。

若要永久啟動部份或全部待用處理器，請遵循下列步驟：

1. 擷取 POD 啟動碼，如下所示：
 - a. 存取 iSeries Capacity Upgrade on Demand 網站：
<http://www.ibm.com/servers/eserver/series/hardware/ondemand> 
 - b. 輸入伺服器的系統類型和序號。
 - c. 記錄在網站上顯示的 POD 啟動碼。
2. 在指令行上輸入 STRSST，以啟動系統服務工具 (SST)。登入 SST。

註：若要使用系統服務工具，您需要具有「系統」容量的有效服務工具使用者 ID - 管理者專用權。

3. 選取選項 6 (使用系統容量)，然後按 Enter 鍵。
4. 選取選項 2 (啟動永久系統容量)，然後按 Enter 鍵。
5. 在「依需求處理器」啟動碼欄位上輸入 POD 啟動碼，然後按 Enter 鍵。出現「確認啟動系統容量」畫面。
6. 按 Enter 鍵確認系統容量啟動。
7. 跳出「使用系統容量」畫面。
8. 若伺服器是分割的，在您使用新容量之前，您必須先將最近可用的處理器指派到邏輯分割區。

有關將啟動的處理器指派到邏輯分割區的資訊，請參閱處理動力的動態移動。

9. 若伺服器未分割，請遵循下列步驟：
 - a. 選取選項 5 (使用系統分割區)，然後按 Enter 鍵。
 - b. 選取選項 3 (使用分割配置)，然後按 Enter 鍵。
 - c. 在 PRIMARY 旁鍵入 2 (變更分割區處理資源)，然後按 Enter 鍵。

- d. 輸入處理器新數量的值，可代表作用中處理器的總數。

註：畫面底端的配置錯誤指出已改變預設「主分割區」(邏輯分割區目前使用中)，您必須參照邏輯分割區計畫，以適當地將啟動的處理器指派給伺服器。

- e. 按 Enter 鍵確認變更。

10. 跳出 SST。

註：輸入 POD 啟動碼時，待用處理器立即變為作用中。然而，必須執行伺服器 15 分鐘，以儲存 POD 啟動碼。若在執行 15 分鐘之前即輸入 POD 啟動碼並關機伺服器，當您啟動伺服器時，可能需要重新輸入 POD 啟動碼。

您現在可以開始使用新容量。

實務範例：Capacity Upgrade on Demand

Capacity Upgrade on Demand 可讓客戶在工作量有需要時啟動待用處理器。下列實務範例引導您完成規劃、訂購及合併此特性的步驟。

1. 特性碼 2416 的機型 840 伺服器與八個作用中處理器與四個待用處理器一起操作。當伺服器工作量增加時，可用的 CPU 資源使用率持續地接近或超過 70% 的可用容量。預期需要其他資源，管理者決定考慮啟動一些待用處理器。
2. 啟動處理器之前，管理者為 Capacity Upgrade on Demand 準備伺服器。這樣會使執行趨勢分析學習將需要多少額外的處理器、讓伺服器準備合併其他處理器，並準備訂購新容量。
3. 若要探查啟動額外處理器的好處，管理者決定啟動試用期間的處理器。試用期間最多為 14 天。
4. 決定由啟動額外處理器所取得的效能增進，授權永久購買處理器之後，管理者聯絡 IBM 銷售業務或企業夥伴，或探訪 www.ibm.com 下四個 Processor on Demand (POD) 啟動碼的訂單。
5. IBM 業務代表在 IBM 配置器中下訂單，並從該訂單的伺服器接收傳送重要產品資料 (VPD) 的提示。可將 VPD 資料傳真到 IBM，或以 Electronic Service Agent 的電子方式傳送。
6. 管理者從 Web 擷取 POD 啟動碼，並啟動永久容量。這樣會在目標伺服器上輸入 POD 啟動碼，並指派處理器給邏輯分割區。

機型 840 現有有八個處理器可使用。

選取效能管理策略

開發良好效能管理策略有助於您管理系統的效能。效能管理策略大部份是根據您能夠花在管理效能上的時間量而定。如果您是經營一家小公司，您可能管理公司的許多不同業務，但無法花太多時間管理效能。許多大公司雇用效能專家來有效調整和執行系統。

為了決定基本效能管理策略和確認要使用哪些效能應用程式，請將公司分類成下列三個種類的其中之一：小公司、中型企業和大型企業。公司資源依每一個大小而不同，管理策略也會跟著不同。

小公司

小公司與較大公司相比之下，大都會將較少數資源用於管理效能。因此，儘可能使用自動化。您使用 PM/400 讓效能資料直接傳給 IBM，IBM 會編譯該資料並產生報告。這不僅節省您的時間，IBM 還能夠在 iSeries 伺服器需要升級時對您提出建議。

下列是小公司的建議效能應用程式清單：

資料收集服務

按使用者定義的間隔收集範例資料供以後分析。

圖形歷程

顯示使用「資料收集服務」收集的效能資料。

PM/400

自動化系統效能資料的收集、歸檔和分析。

效能工具

收集、分析和維護系統效能資訊。

監督程式

觀察 iSeries 系統效能的圖形式呈現，自動回應預先定義的事件或狀況。

中型公司

中型公司大概會比小公司投注更多資源來管理效能。您仍可能需要儘量自動化而且也可以從使用 PM/400 獲益。

下列是中型公司的建議效能應用程式清單：

資料收集服務

按使用者定義的間隔收集取樣資料供以後分析。

圖形歷程

顯示使用「資料收集服務」收集的效能資料。

PM/400

自動化系統效能資料的收集、歸檔和分析。

效能工具

收集、分析和維護系統效能資訊。

監督程式

觀察 iSeries 系統效能的圖形式呈現，自動回應預先定義的事件或狀況。

效能探索

收集關於特定應用程式或系統資源的詳細資訊。

大型公司

大型公司有專用於管理效能的資源。

下列是大型公司的建議效能應用程式清單：

資料收集服務

按使用者定義的間隔收集取樣資料供以後分析。

圖形歷程

顯示使用「資料收集服務」收集的效能資料。

PM/400

自動化系統效能資料的收集、歸檔和分析。

效能工具

收集、分析和維護系統效能資訊。

監督程式

觀察 iSeries 系統效能的圖形式呈現，自動回應預先定義的事件或狀況。

效能探索

收集關於特定應用程式或系統資源的詳細資訊。

iDoctor for iSeries

分析追蹤資料以增進系統和應用程式效能。

Performance Trace Data Visualizer (PTDV)

從 Java 應用程式檢視追蹤資料。

設定環境以管理效能

iSeries 伺服器包含數種工具，可以定期收集系統效能資料並監督您的系統，以取得效能趨勢與潛伏的問題。您的唯一需求與環境將決定選擇的調查工具與配置選項。有效地設定系統可讓您在系統成長時執行精確的產能規劃，且可以發生效能問題時加以解決。

請使用下列主題來學習及配置收集、監督與分析系統效能所使用的工作。

資料收集服務

「資料收集服務」管理系統效能資料的日常收集。此工具可定期收集資料並建立稱為資料收集物件的保存檔。這些資料收集物件可以使用部份工具直接存取或轉換成資料庫檔案集，以便使用自訂查詢或由其他工具及報告加以分析。因為「資料收集服務」主要提供其他應用程式的資料，您使用的其他工具對配置選項有重大影響，包括收集資料的頻率、收集的資料類型及在系統上保存資料的時間長度。

Performance Management/400

PM/400 使用「資料收集服務」來收集非專屬效能資料，並將它傳送至 IBM 以進行儲存及專業分析。如此，您就不需要自行儲存及維護它。然後，您可以使用 Web 瀏覽器存取有關系統效能的明細報告與建議。

「iSeries 領航員」監督程式

「iSeries 領航員」中所含的監督程式使用「資料收集服務」資料來追蹤您特別感興趣的系統效能元素。尤其，在發生某些事件時，如 CPU 使用率百分比或工作狀態，它們可以採取指定的動作。請使用此主題來瞭解如何使用這些監督程式，以及如何在系統上設定它們。

管理 iSeries 效能

成功管理效能可確保您的系統有效地使用資源，以及確保您的 iSeries 伺服器提供最適合您使用者及公司需求的服務。此外，有效的效能管理可協助您快速回應系統變更，並可藉由延緩昂貴升級及服務費來節省成本。

瞭解影響系統效能的因素可協助您回應問題並做好長期規劃。有效的規劃可防止潛伏的效能問題發生，以及確保您的系統容量足以處理目前及成長中的工作量。

使用下列主題來瞭解如何維護系統效能及解決效能問題。

追蹤效能

追蹤系統效能一段時間可讓您規劃系統成長量，以及確保您擁有資料來協助隔離及識別效能問題的原因。瞭解要使用的應用程式及如何定期收集效能資料。

重新搜尋效能問題

有許多選項可協助您識別及解決效能問題。瞭解如何使用可用的工具及報告來協助您尋找效能問題來源。

顯示效能資料

在您收集效能資料之後，瞭解如何使用最適合您的工具來顯示資料。

調整效能

當識別出某個效能問題時，您要調整系統來解決它。

管理電子商業效能

在電子商業環境中管理效能介紹數個新問題給 iSeries 管理者。請參照此主題取得資訊和資源，以協助您規劃、追蹤及增進電子商業應用程式的效能。

追蹤效能

追蹤 iSeries 伺服器的系統效能有助於您識別趨勢，這可以協助您調整系統配置並選擇最佳的系統升級時間與方法。尤其，在發生問題時，收集意外事件前後的效能是非常重要的，可以縮小效能問題的原因範圍並找出適當的解決方案。

iSeries 伺服器含有數種應用程式，可以追蹤效能趨勢及維護 iSeries 效能資料的歷程記錄。大部份應用程式均使用「資料收集服務」所收集到的資料。您可以使用「資料收集服務」來查看下列區域中的趨勢：

- 系統資源使用率的趨勢。您可以使用此資訊來規劃及特別修整系統配置變更與升級。
- 在配置的實體元件上識別緊張度。
- 在尖峰及一般使用期間，平衡交談式作業與批次作業之間的系統資源使用。
- 配置變更。您可以使用「資料收集服務」資料來精確地預測變更的影響，如新增使用者群組、增加交談式作業及其他變更。
- 識別可能造成系統中其他活動問題的工作。

下列工具可協助您隨著時間監督系統效能：

資料收集服務

「資料收集服務」會收集使用者定義時間間隔的效能資料，然後將此資訊儲存在系統的資料收集物件中。許多其他工具 (包括監督程式、圖形歷程、PM/400) 及「效能工具」授權程式中的許多功能都依來這些資料收集物件來取得其資料。

圖形歷程

圖形歷程顯示使用「資料收集服務」以透過圖形使用者介面 (GUI)，在一段指定時期內收集到的效能資料。可用於顯示的時間長度取決於您要保留資料收集物件的時間長度，以及是否使用 PM/400。

PM/400

PM/400 會自動化系統效能資料的集合、保存與分析並傳回清除報告，以協助您管理系統資源與產能。

調查效能問題

大部份收集或分析效能的工具都使用追蹤或取樣資料。「資料收集服務」會在各種系統資源上定期收集取樣資料。數種工具可以分析或報告此取樣資料，且您可以使用此項來更廣泛地檢視系統資源使用率，並回答許多常見的效能問題。如需更詳細的效能資訊，數種工具會產生追蹤層次資料。通常，追蹤層次資料可以提供系統上工作與應用程式的行為及資源使用相關詳細資訊。「效能探究程式」與 STRPFRTRC 指令是產生追蹤資料的兩種常用工具。

例如，如果系統執行速度緩慢，則可以使用系統監督程式來尋找問題。如果您看到 CPU 使用率偏高，則可以識別出通常可能使用大量資源的任何工作。然後，您可以變更配置以更正問題。然後，有些問題需要其餘資訊。若要取得工作效能的詳細資訊，您可以啟動效能探究程式階段作業、在 iSeries 系統上收集該工作的行為相關詳細資訊，且可能變更原來的程式。

若要瞭解如何收集效能資料的其餘相關資訊，請使用下列主題來瞭解使用部份效能管理應用程式的方法與時間。

識別效能問題

瞭解識別效能問題所含的一般步驟。

識別及解決常見的效能問題

許多不同的效能問題通常會影響 iSeries 系統的通用區域。請學習如何調查及解決通用區域中的問題。

收集系統效能資訊

「資料收集服務」會定期收集系統效能的相關資訊。通常，分析效能資料是從此資訊開始。

收集系統資訊使用率的相關資訊

數種工具可以監督如何使用資源，如 CPU、磁碟空間、交談式容量及許多其他元素。您可以使用這些工具以開始識別問題區域。

收集應用程式效能的相關資訊

應用程式執行速度緩慢可能是因為各種原因。您可以使用 OS/400 中所含的數種工具及其他授權程式，協助您取得其餘資訊。

實務範例：升級或移轉之後，增進系統效能

在此實務範例中，您剛剛升級或移轉系統，且它現在的執行速度似乎比之前更慢。此實務範例將協助您識別及修正效能問題。

識別效能問題

嘗試識別效能問題時，評定硬體配置是否足以支援工作量是很重要的。是否有足夠的 CPU 容量？主要儲存體是否足以儲存各種類型的應用程式？先回答這些問題，也許透過容量模型技術，可以避免日後不必要的麻煩。

瞭解問題的徵狀及要達成的目標，分析師就可以製作出一個也許可以解釋問題原因的假設公式。分析師可以使用在 OS/400 作業系統上可用的指令與工具和「效能工具」授權程式，以測量系統效能。

複查測量的資料有助於進一步定義問題，並可協助驗證或拒絕假設。一旦隔離出明顯的原因，就可以提供解決方案。當您一次處理一個解決方案時，可以重新設計並測試程式。然後，在許多情況下，分析師的工具可以測量出解決方案的有效性，並尋找可能的副作用。

爲了達到最佳效能，您必須辨識出重要系統資源間的交互關係，並嘗試平衡這些資源，那就是 CPU、磁碟、主要儲存體，以及若用於通信，則爲遠端線路。每一個資源都可造成效能退化。

系統效能的增進（不論是交談式產能、交談式回應時間、批次產能或結合其中的一部份）可以有許多形式，從只調整活動層次或儲存區大小，到變更應用程式碼本身。在此案例中，活動層次是子系統的一種特性，指定可以在同一處理單位時間競爭的最大工作數。

識別及解決常見的效能問題

在 iSeries 伺服器上發生效能問題時，常常會先影響到系統的某些區域。請參照下表以取得部份可用的方法，以調查這些系統區上的效能。許多區域可以當成系統監督程式計量單位使用。然而，有數種方法可以存取它們的相關資訊。

區域	說明	可用的工具
處理器載入	判定系統上是否有太多工作，或是否有部份工作使用大量的處理器時間百分比。	<ul style="list-style-type: none"> 「處理作用中的工作 (WRKACTJOB)」指令。 「使用系統活動 (WRKSYSACT)」指令，這是「效能工具」授權程式的一部份。 「iSeries 領航員」中的工作管理功能。 「iSeries 領航員」系統監督程式中的 CPU 使用率計量單位。
主要儲存體	調查錯誤與等待合格的轉移。	<ul style="list-style-type: none"> 「使用磁碟狀態 (WRKDSKSTS)」指令 「iSeries 領航員」系統監督程式中的磁碟儲存體計量單位 「查看系統狀態 (WRKSYSSTS)」指令 「iSeries 領航員」中「工作管理」下的「記憶體儲存區」功能。
磁碟	判定是否讀寫臂太少，或是否讀寫臂太慢。	<ul style="list-style-type: none"> 「使用磁碟狀態 (WRKDSKSTS)」指令。 「iSeries 領航員」系統監督程式中的磁碟讀寫臂使用率計量單位。 「效能工具系統與元件」報告。
通信	尋找慢速線路、線路上的錯誤，或線路的使用者是否太多。	<ul style="list-style-type: none"> 效能工具元件報告。 「iSeries 領航員」系統監督程式中的 LAN 使用率計量單位。
IOP	判定是否有任何 IOP 不平衡，或是否沒有足夠的 IOP。	<ul style="list-style-type: none"> 效能工具元件報告。 「iSeries 領航員」系統監督程式中的 IOP 使用率計量單位。
軟體	調查鎖定與互斥 (mutexes)。	<ul style="list-style-type: none"> 「效能工具鎖定」報告 「效能工具追蹤」報告 「使用物件鎖定 (WRKOBJLCK)」指令。 在「iSeries 領航員」中，以滑鼠右鍵按一下「工作管理」下的不可使用工作，然後選取明細 → 鎖定的物件。

收集系統效能資料

收集資料是增進效能的一個重要步驟。當收集效能資料時，您所收集到的伺服器相關資訊可用來瞭解回應時間及產能。這可用來攫取有關完成工作的一或多個伺服器效能狀態。資料收集提供可稍後執行的任何比較及分析之環境或起點。當使用您的第一個資料收集時，提供您一個用於進一步改進的基準性能測試，同時您可以立即改善效能。您可以使用所收集的效能資料來進行調整、增加回應時間，以及協助您的系統達到最高效能。效能問題分析通常從簡單的問題開始：“What changed?” 效能資料可協助您回答該問題。

您可以使用「資料收集服務」來收集效能資料，使用「建立效能資料 (CRTPFDRDTA)」指令建立效能檔案，使用「轉換效能資料 (CVTPFDRDTA)」指令或透過 iSeries 中的「效能工具」外掛程式將它們轉換成現行版次，然後使用效能資料庫檔案中的資訊來建立報告或建立自己的查詢。

有關效能資料的詳細資訊，請參閱下列所示：

資料收集服務

瞭解如何收集用於分析的效能資料及如何自訂資料收集。

效能資料庫檔案

尋找可用的效能資料庫概觀及查看每個效能資料庫檔案的詳細欄位資料。

此外，您可以使用效能管理 API 來啟動、結束及循環資料收集，以及變更及擷取所收集資料的系統參數。有關其它 iSeries 效能工具及技術的詳細資訊，請參閱效能頁。

收集系統資源使用率相關資訊

許多工具可協助您監督及追蹤 iSeries 伺服器及您的應用程式使用可用資源的方式。您可使用此資訊作為問題分析的起點，以及找出協助您進行產能規劃及管理系統成長的趨勢。

請參閱下列主題以瞭解使用這些工具的方法及時機：

iSeries 領航員監督程式

併入「iSeries 領航員」中的監督程式提供各種計量單位的最新資料。另外，當發生某些事件時，您可以配置它們來採取指定的動作。

OS/400 效能指令

OS/400 併入了一些重要功能來協助您管理及維護系統效能。

PM/400

PM/400 使用「資料收集服務」來收集非專用效能資料並傳送給 IBM 進行儲存及專業分析。這可省略您自行儲存及維護它的需求。然後，您可以使用 Web 瀏覽器來存取有關系統效能及趨勢分析的明細報告及建議事項。

收集應用程式效能相關資訊

收集應用程式效能相關資訊與收集效能系統相關資訊有很大的不同。收集應用程式資訊只能透過某些效能應用程式來完成，例如「效能探究程式」、Performance Trace Data Visualizer (PTDV) 及 iDoctor。另外，您可以使用工作監測程式來追蹤個別的伺服器效能，以及使用效能工具來追蹤及分析伺服器工作，藉以取得應用程式效能概觀。

註：收集應用程式效能資料對系統效能有顯著的影響。開始收集之前，請確定您試過其它所有的資料收集選項。

效能探究程式

此工具可協助尋找無法使用執行一般效能監督功能的工具來識別的效能問題之原因。當您的電腦環境逐漸變大而且變得複雜時，您的效能分析也會變得更複雜。效能探究程式藉由收集複雜效能問題的相關資料來應付複雜性的提高。

效能探究程式係專為應用程式開發者而設計，這些人想要瞭解或增進其程式效能。它對於熟悉使用效能管理來幫助識別及解決複雜效能問題的使用者也很有幫助。

Performance Trace Data Visualizer for iSeries (PTDV)

此工具是一種 Java 應用程式，可用於 iSeries 上所執行應用程式的效能分析。PTDV 使用 OS/400 作業系統的效能探究程式功能，讓分析師檢視程式流程及取得追蹤、工作、緒及程序彙總的明細 (例如 CPU 時間、現行系統時間、循環次數及指令數量)。當顯現 Java 應用程式追蹤時，也會顯示所建立物件類型及數

量的其他明細，以及有關 Java 鎖定行為的資訊。也支援 WebSphere 應用程式伺服器所產生的效能探究程式事件。PTDV 容許直欄排序、資料的匯出及許多層次的資料彙總。

詳細資訊，請探訪 Performance Trace Data Visualizer  網站。

iDoctor for iSeries

iDoctor 中的「效能探究程式分析程式」功能併入一個特別準備來協助分析追蹤資料以增進系統和應用程式效能的工具。這個詳細的分析提供了磁碟作業、CPU 使用率、檔案開啓作業、機器介面 (MI) 程式、等待狀態、磁碟空間用量及其它等等的基本彙總。從屬站元件是「iSeries 領航員」外掛程式，可讓使用者以圖形方式壓縮及顯示 iSeries 追蹤資料。

「啓動效能追蹤 (STRPFRTRC)」指令

OS/400 併入一個用來收集多重程式設計及異動資料的指令。此指令可收集 STRPFRMON 在前版次所收集的資料。執行這個指令後，您可以使用「傾出追蹤 (DMPTRC)」指令，將資料匯出至資料庫檔案。

傾出追蹤資料： 決定何時傾出追蹤資料是一項重要的決策，因為傾出會影響系統效能。傾出追蹤 (DMPTRC) 指令可將內部追蹤表格中的資訊置入資料庫檔案。在已載入的系統活動尖峰期間或高優先順序 (交談式) 工作內，不建議傾出追蹤資料。您可以延遲追蹤傾出，但不要忘了傾出資料。如果追蹤表格因為任何原因被清除了，您將會遺失追蹤資料。不過，稍微延遲傾出，然後使用 DMPTRC 指令以批次工作傾出追蹤，可為使用者保持效能。

若要傾出追蹤資料，請使用下列指令：

```
DMPTRC MBR(member-name) LIB(library-name)
```

您必須指定儲存資料的檔案庫名稱及成員名稱。您可以在收集追蹤資訊的同時使用資料收集服務來收集範取樣式資料。當您像這樣同時收集取樣資料和追蹤資料時，您應該將其資料放置到名稱一致的成員中。換言之，您在 CRTPFRTA TOMBR 和 TOLIB 參數中提供的名稱應該與您在 DMPTRC MBR 和 LIB 參數中提供的名稱相同。

實務範例：在升級或移轉後增進效能

您最近將 iSeries 伺服器升級成最新的版次。在完成升級及回復正常作業後，您的系統效能明顯降低。您想要找出效能問題的原因並將系統還原至正常效能層次。

隔離變更

有許多問題會造成作業系統升級後效能降低。您可以使用 OS/400 和 Performance Tools 授權程式 (5722-PT1) 內含的效能管理工具，取得有關效能問題的詳細資訊及將一些可疑的問題範圍縮小成一個可能的原因。

1. 檢查 CPU 使用率。有時，在升級後工作無法存取某些必要的資源。這會造成單一工作耗用太多 CPU 資源數量。
 - 使用 WRKSYSACT、WRKSYSSTS、WRKACTJOB 或「iSeries 領航員」系統監督程式來尋找 CPU 總使用率。
 - 如果 CPU 使用率過高，例如超過 90%，請檢查作用中工作使用的 CPU 量。如果單一工作耗用的 CPU 資源超過 30%，則可能遺失檔案呼叫或物件。您可以向供應商取得供應商提供的程式，或向工作的擁有者或程式設計師尋求其它支援。
2. 使用 STRPFRTRC 指令啓動效能追蹤，然後使用系統及元件報告來識別及更正下列可能的問題：
 - 如果機器儲存區的尋頁錯失率超過每秒 10 個錯誤，請提供更多記憶體給機器儲存區，直到錯誤率降至這個層次以下。
 - 如果磁碟使用率超過 40%，請查看等待時間及服務時間。如果這些值可接受，您可能需要縮減工作量來管理優先順序。

- 如果 IOP 使用率超過 60%，請新增一個額外的 IOP 並指定一些磁碟資源給它。
 - 如果使用者儲存區中的尋頁錯失率太高，請參閱自動調整效能主題。
3. 執行工作摘要報告，然後參考**抓取鎖定衝突報告**。如果抓取或鎖定衝突數目太高，請確定存取路徑大小設為 1TB。如果抓取或鎖定衝突與使用者設定檔有關，而且被參考到的使用者設定檔擁有許多物件，請減少該設定檔所擁有的物件數量。
 4. 使用**作業切換**選項執行 iDoctor 五分鐘。然後，使用作業切換監督程式分析追蹤資料結果。識別並解決下列任一項：
 - 工作正等待 CPU
 - 工作錯誤
 - 掌握衝突

顯示效能資料

顯示效能資料可協助您更精確地分析系統效能。有許多不同的方式可顯示效能資料；不過，在某些狀況下您會發現某個效能應用程式更加適合。大部份應用程式顯示資料收集服務或效能追蹤所收集的資料。存取該資料的最佳方式根據您要嘗試解決效能問題、您要監督系統效能來規劃未來成長或您要識別趨勢而定。

顯示接近即時的效能資料

使用下列工具來顯示現行或最新的效能資訊：

OS/400 指令

基本作業系統中有許多指令可讓您檢視有關特定區域系統效能的現行資訊。

「效能工具」顯示畫面

「效能工具」授權程式併入了「iSeries 領航員」的外掛程式，它顯示「資料收集服務」資料收集物件中的效能資料。您也可以檢視有關系統工作的詳細資訊及列印「效能工具」報告。

系統與工作監測程式

這些監督程式顯示許多系統元素的效能資料。監督程式資料視資料收集物件而定，並且根據「資料收集服務」中的資料收集間隔，在收集資料時顯示資料。

顯示歷程效能資料

使用下列工具來檢視系統儲存的資料：

PM/400

PM/400 自動化系統效能資料的集合、保存及分析，並傳回明確報告來協助您管理系統資源及產能。

圖形歷程

圖形歷程根據「資料收集服務」中的保留期，提供最多一週的效能資料圖形顯示畫面。搭配 PM/400，「圖形歷程」可顯示更長期的資料收集。

調整效能

效能調整的主要目的是要讓伺服器可以最有效地利用系統資源，並讓工作量能夠盡可能地有效執行。效能調整是以手動或自動調整系統效能的方式。其中存有許多選項可以用來調整系統。每一個系統環境都是唯一的，您必須觀察其中的效能並進行最適合環境的調整；換句話說，您必須執行日常效能監督。如需效能調整前必須執行的效能監督步驟相關資訊，請參閱管理效能。

IBM 亦提供工具，可以減少從磁碟讀取的實體 I/O 要求數，以增進 I/O 子系統與系統回應時間。請瞭解您要如何使用「延伸的適應性快速記憶體」來增進系統效能。

如需效能調整的詳細資訊，請從下列主題中選取：

基本效能調整

若要調整系統效能，您必須設定起始調整值、觀察系統效能、複查值，然後決定要調整的項目。

自動調整效能

大部份的使用者應設定系統以自動執行效能調整。新系統在出貨時，即已配置自動調整。

基本效能調整

若要開始調整效能，您必須先決定起始機器與使用者儲存區大小以設定起始調整值。然後，您可以開始觀察系統效能。

設定起始調整值

設定起始調整值包括起始配置系統儲存區大小與活動層次所採取的步驟，以有效地調整系統。起始值是以估計值為依據；因此，當系統作用中時，可能需要進一步調整估計值。下列步驟可設定起始調整值：

- 決定起始機器儲存區大小
- 決定起始使用者儲存區大小

觀察系統效能

若要觀察系統效能，您可以使用「查看系統狀態 (WRKSYSSTS)」、「查看磁碟狀態 (WRKDSKSTS)」及「處理作用中的工作 (WRKACTJOB)」指令。在每一個觀察期間，您應針對效能目標來檢查及評估系統效能測量。

1. 移除任何無效的系統活動。可能會造成嚴重效能降低的無效活動為 (例如) 交談式程式編譯、通信錯誤復原程序 (ERP)、開啓查詢檔 (OPNQRYF)、應用程式錯誤及登出活動。
2. 請使用 WRKSYSSTS、WRKDSKSTS 及 WRKACTJOB 指令來顯示效能資料。您也可以使用「效能工具」指令「使用系統活動 (WRKSYSACT)」來顯示效能資料。
3. 容許系統收集最小值 5 分鐘的資料。
4. 針對效能目標，評估效能測量。一般測量包括：
 - 交談式產能與回應時間，可以從 WRKACTJOB 顯示畫面中取得。
 - 批次產能。觀察作用中批次作業的輔助輸入/輸出 (AuxIO) 及 CPU 百分比 (CPU%) 值。
 - 排存產能。觀察作用中寫出器的輔助輸入/輸出 (AuxIO) 及 CPU 百分比 (CPU%) 值。
5. 如果您觀察的效能資料不符合預期，請依照新資料來調整系統。請務必：
 - 測量及比較所有重要的效能測量。
 - 一次執行及評估一種調整。

複查效能

一旦您已設定良好的調整值，則應定期複查它們以確保系統能維持良好運作。進行中的調整包括觀察系統效能的各個方面，以及調整至建議的引導線。

若要收集重要的統計值，您應觀察一般活動層次期間的系統效能。例如，當沒有任何工作在系統上執行時所收集到的統計值，是屬於評定效能時的小值。如果您已盡最大努力，但仍然不滿意效能，則應評估配置的功能。若要符合您的目標，請考慮下列：

- 處理器升級
- 增加儲存裝置與控制器
- 增加主要儲存體

- 應用程式修改

套用一或多種方法，應可達到您的目標。如果，在合理的努力後，仍然無法達成目標，則應判定您的目標對執行的工作類型而言是否實際。

決定調整的項目

如果系統效能已下降且需要調整，請參照研究效能問題，以識別效能問題的來源並進行特定更正。

自動調整效能

系統可以自動設定效能值，以便有效地使用系統資源。您可以設定下列以自動調整系統效能：

- 調整儲存體儲存區大小與活動層次
- 調整儲存體儲存區分頁

調整儲存體儲存區大小與活動層次

使用 QPFRADJ 系統值，以控制儲存體儲存區與活動層次的自動調整。此值指示系統是否應在系統重新啟動 (IPL) 時或在重新啟動後定期調整值。

您可以設定系統，在 IPL 時、動態或兩者調整效能。

- 若要將系統設定為只在系統重新啟動 (IPL) 時調整，請選取「配置與服務 -> 系統值 -> iSeries 領航員的效能」。按一下「記憶體儲存區」標籤，然後選取**自動調整記憶體儲存區與活動層次下的系統重新啟動時**。這就等於將 QPFRADJ 系統值設定為 1。
- 若要將系統設定為在系統重新啟動 (IPL) 時執行儲存體儲存區調整，並在重新啟動後定期執行儲存體儲存區調整，請選取「配置與服務 -> 系統值 -> iSeries 領航員的效能」。按一下「記憶體儲存區」標籤，然後在**自動調整記憶體儲存區與活動層次下選取系統重新啟動時及重新啟動後定期**。這就等於將 QPFRADJ 系統值設定為 2。
- 若要將系統設定為在重新啟動後定期執行儲存體儲存區調整，而不在系統重新啟動 (IPL) 時執行，請選取「配置與服務 -> 系統值 -> iSeries 領航員的效能」。按一下「記憶體儲存區」標籤，然後在**自動調整記憶體儲存區與活動層次下選取重新啟動後定期**。這就等於將 QPFRADJ 系統值設定為 3。

儲存體儲存區值不會在系統重新啟動 (IPL) 時重設為起始值。

調整儲存體儲存區分頁

系統提供的動態調整支援會自動調整儲存區大小與共用儲存區的活動層次，以增進系統效能。此調整會將儲存體從最小使用的儲存體儲存區移至會因有較多儲存體而受益的儲存區。此調整也會設定活動層次，以平衡儲存區中的緒數與儲存區的配置儲存體。若要調整系統，調整者可使用以緒數為基礎所計算出來的引導線。

當動態調整生效時，下列效能值即會自動變更為適當的設定值：

- 機器 (*MACHINE) 記憶體儲存區大小 (QMCHPOOL 系統值)
- 基本 (*BASE) 記憶體儲存區活動層次 (QBASACTLVL 系統值)
- 儲存區大小與共用儲存區 *INTERACT 的活動層次
- 儲存區大小與共用儲存區 *SHRPOOL 的活動層次
- 儲存區大小與共用儲存區 *SHRPOOL1-*SHRPOOL60 的活動層次

當動態調整生效 (QPFRADJ 系統值設定為 2 或 3) 時，即會在系統中顯示設定檔 QSYS 下執行的工作 QPFRADJ 為作用中。

如需記憶體儲存區的詳細資訊，請參閱記憶體儲存區。

管理電子商業效能

電子商務環境中的效能呈現給 iSeries 系統管理者一些複雜的問題。除了在 iSeries 伺服器上定期調整，管理者也必須監督及最佳化硬體與服務，以支援其電子商業交易。

下列主題可協助您熟悉一些重要的注意事項，以最佳化伺服器的電子商業效能，也將提供一些其他資源的鏈結，讓您取得詳細建議和範例。

從屬站效能

當 iSeries 系統管理者通常可控制電子商業網路的從屬站端時，您可以使用這些建議確定已為電子商業環境最佳化該從屬站裝置。

網路效能

網路設計、硬體資源及傳輸壓力通常對電子商業應用程式的效能有顯著的影響。您可以使用本主題，以取得最佳化網路效能及調整 iSeries 通信資源的資訊。

OS/400 中的 Java 效能

OS/400 提供數個配置選項和資源，以最佳化 iSeries 伺服器上的 Java 應用程式效能或服務。使用此主題可學習有關 OS/400 中的 Java 環境，以及如何從 Java 型應用程式取得最佳的可能效能。


HTTP Server 效能

HTTP 伺服器通常是 iSeries 伺服器上電子商業效能的重要部份。IBM 提供數個選項和配置選項，可讓您取得最佳化此伺服器的選項。

WebSphere 效能

WebSphere Application Server 是 iSeries 伺服器的電子商業應用程式部署環境選擇。使用此主題可學習如何在 WebSphere 環境中規劃及最佳化效能。

除了這些特定的建議之外，管理者也應該熟悉下列主題：

- 工作管理
- Java for iSeries
- HTTP 伺服器
- Domino for iSeries sizing and performance tuning 

從屬站效能

包含 PC (具有 Web 瀏覽器) 的從屬站通常代表管理者有最小直接控制的 eBusiness 元件。然而，這些元件仍對 Web 應用程式的端對端回應時間有顯著的影響。

若要協助說明高層次的效能，從屬站 PC 應該：

- 有足夠的記憶體。資源密集 Applets 與使用複雜格式及圖形的介面，也可能要求從屬站處理器。
- 利用高速及最佳化網路連線。許多從屬站 PC 上的通信配接卡可能在沒有最佳化網路環境時仍作用中。詳細資訊，請參照通信硬體的文件。
- 使用完全支援必要技術的瀏覽器。此外，瀏覽器支援與效能必須是設計 Web 介面時的主要考量。

網路效能

網路通常扮演 Web 應用程式回應時間中主要的角色。此外，網路元件的效能影響通常很複雜也很難測量，因為網路壅塞和可用的頻寬可能會頻繁地變更，並且因系統管理者是否直接控制而產生影響。然而，有許多資源可協助您監督及調整 iSeries 伺服器上的通信資源。

詳細資訊，請參閱下列主題：

資料收集服務

「資料收集服務」以固定間隔收集通信資源的效能資料。由於特殊喜好，效能資料檔 QAPMTCP 和 QAPMTCPIFC 收集有關 TCP 伺服器的資訊。您可以藉由直接查詢檔案，或使用「效能工具授權程式」中的報告，以參考此資料。

系統監督程式

您可以使用系統監督程式提供有關在 iSeries 伺服器上使用系統資源 (包括通信硬體) 的資訊。其中，系統監督程式中的線路使用率和 IOP 度量可提供有關網路效能的有用資料。

追蹤效能

有些應用程式和工具可讓您在 iSeries 伺服器上定期收集通信資源的效能資料，並可監督超過時間的效能。

iSeries 效能功能參照

「效能功能參照」提供詳細資訊、報告和範例，可協助您配置或調整 iSeries 伺服器以達最佳效能。其中，請參閱第 5 章：通信效能，以協助您規劃及管理 iSeries 通信資源。

iSeries Network.com

本網站掌控許多資源，可最佳化您的網路規劃和資源。請參照文章 “Cultivate your AS/400 Network” 及 “8 工具，以取得較佳網路效能。”

OS/400 中的 Java 效能

Java 通常是 Web 型應用程式的語言選項。然而，Java 應用程式可能需要一些最佳化 (OS/400 執行環境與 Java 應用程式)，以取得最佳效能。

使用下列資源學習 OS/400 中的 Java 環境，以及分析與增進 Java 效能可用的秘訣。

Java 效能

有數個重要配置選項與工具可協助您從 Java 型應用程式取得最佳效能。

收集應用程式效能的相關資訊

有數個工具可協助您監督及調整 OS/400 中的應用程式效能。使用此主題學習如何使用效能追蹤、效能探查 (PEX) 與類似的工具，以協助您測量及增進應用程式效能。

iSeries 效能功能參照

「效能功能參照」提供詳細資訊、報告和範例，可協助您配置或調整 iSeries 伺服器以達最佳效能。請特別參閱第 7 章：Java 效能，以協助您最佳化 Java 應用程式的效能，及學習 Java 程式設計的效能秘訣。

OS/400 中的 Java 與 WebSphere 效能

使用此「紅皮書」以學習如何規劃及配置作業環境，以最佳化 Java 和 WebSphere 效能，並協助您收集及分析效能資料。

IBM eServer iSeries 伺服器的 WebSphere J2EE 應用程式開發

此「紅皮書」提供 J2EE 的簡介，並提供建議與範例以協助您順利在 iSeries 伺服器上實作 J2EE 應用程式。

除了效能資訊外，Java 主題提供在 iSeries 伺服器上開發及部署 Java 應用程式的資源。

HTTP Server 效能

HTTP 伺服器在 Web 型應用程式的端對端效能中扮演了很重要的角色，一些新的改進可讓您更有效地監督及改進 Web 伺服器效能。特別來說，新「快速回應快取加速器 (FRCA)」可讓您明顯地增進 HTTP 伺服器效能，尤其在靜態環境中更為明顯。

請參考下列資源，以取得最佳化 HTTP 伺服器效能的資訊。

資料收集服務

您可使用「資料收集服務」收集 HTTP 伺服器效能資料，並監督超過時間的結果。效能資料檔 QAPMHTTPB 和 QAPMHTTPD 在每個資料收集間隔儲存 HTTP 伺服器資料。QAPMHTTPB 提供基本資訊，而 QAPMHTTPD 則提供較詳細的統計值。您可以直接查詢這些資料檔，或參照「效能工具授權程式」中的「系統與元件」報告。

IBM HTTP Server for iSeries

請參照此主題，以取得在 iSeries 上設定、配置及管理 HTTP 伺服器的資訊。此主題也包括本產品最新加強功能的說明，如「快速回應快取加速器 (FRCA)」。

iSeries 效能功能參照

「效能功能參照」提供詳細資訊、報告和範例，可協助您配置或調整 iSeries 伺服器以達最佳效能。請特別參照第 6 章：Web Server 與 Web Commerce，以取得 HTTP 伺服器效能規格、規劃資訊與效能秘訣。

HTTP 伺服器 (Apache 驅動)

使用「紅皮書」取得 OS/400 上 HTTP 伺服器 (Apache 驅動) 的深入說明，包括一般用法實務的配置 HTTP Server 範例。

AS/400 HTTP Server Performance and Capacity Planning

使用此「紅皮書」學習 HTTP 伺服器的效能調整和規劃影響。本出版品也包括使用 iSeries 效能管理工具收集、解譯及回應 Web 伺服器效能資料的建議。

WebSphere 效能

在 WebSphere 環境中管理 iSeries 伺服器效能帶給 iSeries 管理者數個挑戰。Web 型交易可能使用多個資源，並且是以不同於傳統通信工作量的方式使用。

請參照下列主題及資源，以學習如何規劃最佳效能，以及在 WebSphere 環境中調整數個資源。

WebSphere Application Server 效能注意事項

此網站提供資源給 iSeries 伺服器上的每個 WebSphere Application Server 版本，包括許多有用的效能秘訣及建議。此資源對使用 servlets、Java Server Pages (JSP) 與 Enterprise Java Beans (EJB) 的環境特別有用。

DB2 UDB/WebSphere Performance Tuning Guide

此「紅皮書」提供 WebSphere 和 DB2 環境的簡介，並提供一般效能問題的建議、範例與解決方案，以協助您最佳化 WebSphere 與 DB2 效能。

OS/400 中的 Java 與 WebSphere 效能

使用此「紅皮書」以學習如何規劃及配置作業環境，以最佳化 Java 和 WebSphere 效能，並協助您收集及分析效能資料。

此「紅皮書」提供在 iSeries 伺服器上最佳化 WebSphere V3 效能的詳細建議與範例。

iSeries 效能功能參照

「效能功能參照」提供詳細資訊、報告和範例，可協助您配置或調整 iSeries 伺服器以達最佳效能。請特別參照第 6 章：Web Server 與 Web Commerce，以取得 WebSphere 特定效能秘訣。

有關其他 WebSphere 與電子商業資訊資源，請參照主題 WebSphere 電子商業管理。

用於效能管理的應用程式

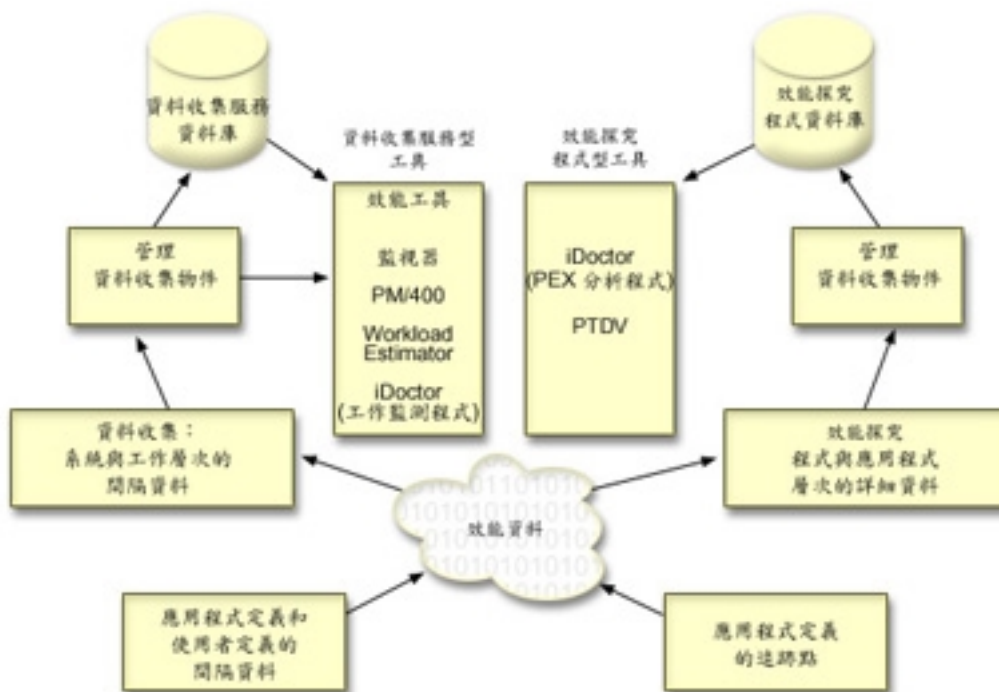
用於效能管理的許多應用程式都具有若干功能。要完全瞭解應用程式可用套件的哪一個元件最適合某給定的狀況，可能不是那麼簡單。下列主題提供有關每一個效能管理應用程式的詳細資訊，包括選項、用法及配置。

如下圖所示，基本上，iSeries 伺服器有兩個效能收集功能：

- 「資料收集服務」，它可收集系統與工作層次的間隔資料。您可以持續執行此功能來瞭解系統發生的狀況。所收集的間隔資料為應用程式或使用者的定義的間隔資料。
- 「效能探究程式」，它可收集程式與應用程式層次的詳細資料。它也追蹤應用程式的工作流程，並且可用來診斷困難的效能問題。收集的資料是根據應用程式定義的效能探究程式追蹤點，例如 Domino、NetServer 或 WebSphere。

這兩個集合功能將其資料存放在管理資料收集物件中。您可以從管理資料收集物件轉換資料，方法是對「資料收集服務」資料使用「建立效能資料 (CRTPFRDTA)」指令，或對效能探究程式資料使用「建立效能探究程式資料 (CRTPEXDTA)」指令。

本主題簡介可用於使用「資料收集服務」資料或效能探究程式資料的效能管理應用程式。



資料收集服務

「資料收集服務」會在使用者定義的時間間隔收集效能資料，然後將此資訊儲存於您系統的資料收集物件中。其它許多工具，包括監督程式、圖形歷程、PM/400，以及「效能工具」授權程式中的許多功能，皆根據這些資料收集物件來取得它們的資料。

效能資料庫檔案

您可以從「資料收集服務」所維護的資料收集物件產生資料庫檔案。此主題用來尋找這些資料庫檔案的名稱、說明及屬性。

監督程式

監督程式顯示有關您系統效能的現行資訊。另外，當發生特定事件時，您可以使用它們來執行預先定義的動作。您可以使用系統、訊息、工作、檔案及 B2B 異動監督程式來顯示及監督您系統的相關資訊。系統及工作監測程式使用「資料收集服務」所收集的效能資料。

圖形歷程

圖形歷程提供「資料收集服務」在指定時段所收集之效能資料的圖形顯示畫面。

PM/400

PM/400 自動化系統效能資料的收集、保存及分析，並傳回報告來協助您管理系統資源及產能。PM/400 使用「資料收集服務」所收集的效能資料。

效能工具

「效能工具」授權程式包含許多協助您收集、分析及維護系統效能資訊的功能。這包括協助您管理分散式網路效能、收集及報告摘要及追蹤資料，以及產能規劃。「效能工具」使用「資料收集服務」所收集的效能資料（取樣資料），以及取得自「啟動效能追蹤 (STRPFRTRC)」指令和「結束效能追蹤 (ENDPFRTRC)」指令的追蹤資料。

效能探究程式

「效能探究程式」可收集特定應用程式、程式或系統資源的詳細資訊，以及提供特定效能問題的深入探討。這包括執行若干追蹤類型及層次及執行明細報告的功能。

iDoctor for iSeries

iDoctor for iSeries 外掛程式由三個用於管理效能的軟體工具組成：「效能探究程式分析程式」可取得詳細的追蹤資料分析，「工作監測程式」可取得有關工作行為的追蹤層次資訊，以及「物件探究程式」可協助您查詢及管理系統上的物件。

Performance Trace Data Visualizer (PTDV)

Performance Trace Data Visualizer for iSeries (PTDV) 是一種 Java 應用程式，可用於 iSeries 上所執行應用程式的效能分析。

效能管理 API

「效能管理 API」提供了管理資料收集的服務。這些 API 可啟動、結束及循環資料收集，並且可變更及擷取所收集資料的系統參數。許多「效能管理 API」使用「資料收集服務」所收集的效能資料。

OS/400 效能指令

OS/400 併入了一些重要功能來協助您管理及維護系統效能。

延伸的適應性快速記憶體

「延伸的適應性快速記憶體」可增進系統效能，其方法是收集磁碟用量資料，然後使用那些統計值來建立大型快取記憶體，有效減少磁碟的實體 I/O 需求。

Workload Estimator for iSeries

Workload Estimator 可協助您規劃下一次升級的大小及時間需求。此工具通常與 PM/400 搭配使用來分析系統效能趨勢，及協助您有效管理 iSeries 伺服器的成長及擴充。

iSeries Navigator for Wireless

iSeries Navigator for Wireless 可讓您透過無線方式，利用個人數位助理 (PDA)、網際網路型電話或傳統 Web 瀏覽器來監督效能資料。iSeries Navigator for Wireless 使用「資料收集服務」所收集的效能資料。

PATROL for iSeries (AS/400) - Predict

PATROL for iSeries (AS/400) - Predict 可協助管理 iSeries 效能，方法是自動化執行許多必要的日常管理作業以達到高替代使用性及最佳化效能。另外，此產品提供詳細的產能規劃資訊來協助您規劃 iSeries 環境的成長。

資料收集服務

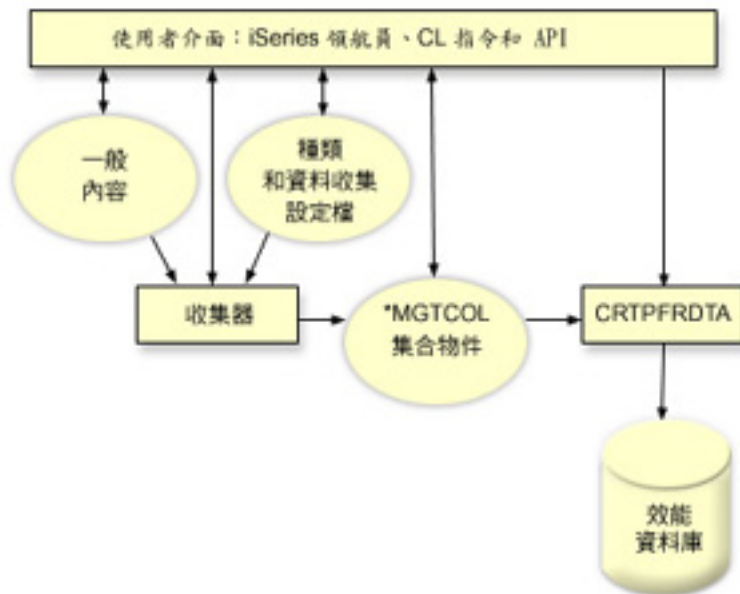
使用「資料收集服務」來收集效能資料，供 Performance Tools for iSeries 授權程式或其它效能報告應用程式、iSeries 領航員監督程式及圖形歷程功能日後分析之用。(如果您偏好檢視即時效能資料，系統監督程式提供一個易用的圖形介面來監督系統效能)。「資料收集服務」收集用於識別不同系統區域使用的相對系統資源量之資料。「資料收集服務」可用來：

- 輕易管理您的資料收集物件
- 使用最小系統額外執行時間來連續並自動地收集效能資料
- 控制收集的資料種類及資料的使用方式
- 在版次之間移動效能資料，而不需要轉換資料
- 建立「效能工具」使用的效能資料檔案
- 整合您自己的程式，將使用者定義的效能資料收集到「資料收集服務」中。

「資料收集服務」的使用方式

「資料收集服務」會置換 OS/400 效能監視器，您可以使用「啟動效能監視器 (STRPFRMON)」指令來呼叫它。「啟動效能監視器 (STRPFRMON 指令)」在 V4R5 之前不支援。當您使用 OS/400 效能監視器時，您所收集的資料可達 30 個資料庫檔案。

「資料收集服務」功能介紹一種新的收集效能資料處理程序。「資料收集服務」可將每個資料收集的資料儲存在單一資料收集物件中，您可以依需求來建立一些不同的資料庫檔案集。這表示系統收集效能資料時的額外執行時間較低。即使您選擇在資料收集期間建立資料庫檔案，您仍然會感受到 OS/400 效能監視器上的效能優點，因為「資料收集服務」使用低優先順序 (50) 批次工作來更新這些檔案。資料收集額外執行時間的縮減有助於更詳細且以更短的間隔持續收集效能資料。「資料收集服務」可讓您建立用於收集和保留效能資料的全網路系統原則，以及自動實施該原則。只要保留管理資料收集物件，一旦有需求，您就能夠回顧及分析直到所收集明細層次為止的效能相關事件。



「資料收集服務」可讓您收集對系統效能影響不大或沒有明顯影響的效能資料。您可以使用「iSeries 領航員」配置「資料收集服務」來收集您要經常收集的資料。資料收集物件 *MGTCOL 是作為保留大量效能資料的有效儲存媒體。一旦配置並啟動「資料收集服務」後，會持續收集效能資料。當需要使用效能資料時，您可以將需要的資料複製到一組效能資料庫檔案中。

上圖提供下列「資料收集服務」元素的概觀：

使用者介面

有一些方法可讓您存取不同的「資料收集服務」元素。例如，您可以使用 CL 指令、API 及「iSeries 領航員」介面。

一般內容

一般內容定義資料收集的完成方式且可控制自動資料收集屬性。

資料種類

資料種類可識別收集的資料類型。您可以配置種類來單獨控制收集的資料種類及收集資料的頻率。

資料收集設定檔

資料收集設定檔提供了儲存及啟動特定種類配置的方法。

效能收集器

效能收集器使用一般內容及種類資訊來控制效能資料的資料收集。您可以啟動及停止效能收集器，或配置它來自動執行。

資料收集物件

資料收集物件 *MGTCOL 作為保留大量效能資料的有效儲存媒體。

「建立效能資料 (CRTPFDRDTA)」指令


CRTPFDRDTA 指令處理儲存於管理資料收集物件中的資料並產生效能資料庫檔案。

效能資料庫

資料庫檔案儲存 CRTPFRTA 指令所處理的資料。資料庫檔案可分成這些種類：包含時間間隔資料的效能資料檔、配置資料檔及追蹤資料檔。

如何啟動「資料收集服務」

您可以使用下列任一方法來啟動「資料收集服務」。不過，「效能」主題中的資訊主要是關於「iSeries 領航員」方法。

啟動方法	說明
iSeries 領航員	以下區段說明如何利用 iSeries 領航員來執行各種「資料收集服務」作業。
效能管理 API	您可以使用效能管理 API 來啟動、自訂、結束及循環資料收集。此外，您可以使用 API 來處理管理資料收集物件或定義您自己的異動。
傳統功能表選項	在字元型介面中鍵入 GO PERFORM ，然後從「效能工具」主功能表中選取選項 2 (收集效能資料)。如需其它資訊，請參閱 Performance Tools  。
Performance Management/400	您可以啟動 PM/400 來自動啟動「資料收集服務」，然後在資料收集期間建立資料庫檔案。

「資料收集服務」作業

您可以使用「資料收集服務」和「iSeries 領航員」來執行下表所顯示的各種資料收集作業。


作業	說明
以各種方法啟動「資料收集服務」	在具有特定效能計量單位的個別系統或系統群組上建立自訂的效能資料收集。您也可以使用啟動程式中的「啟動收集器 API」來自動啟動效能資料收集。有關如何執行這些作業的詳細資訊，請參閱線上說明。您可以從「iSeries 領航員」視窗中取得詳細的作業說明。從功能表列按一下 說明 ，然後選取 說明主題 即可。選取 我可以做些什麼...? 來瞭解您可以做些什麼以及您必須在「iSeries 領航員」視窗的什麼位置才能達到目的。
建立資料庫檔案	使用「資料收集服務」來自動建立效能資料庫檔案。您也可以從資料收集物件建立資料庫檔案，資料收集之後會儲存在資料收集物件中。您可以搭配 PM/400 或「效能工具」授權程式來使用這些資料庫檔案，或是建立自己的查詢來對這些檔案執行。瞭解如何在您建立資料庫檔案時控制收集的資料種類。 有關可用的資料庫檔案及併入每個檔案中的欄位層次資料，請參閱效能資料庫檔案。
自訂資料收集	自訂您的資料收集。尋找有關控制您所收集的效能資料及收集資料的頻率的資訊。您也可以尋找有關重要時區注意事項的資訊。
新增使用者定義的種類至「資料收集服務」	您可以撰寫跳出程式並將其整合到「資料收集服務」中，從使用者應用程式中收集效能資料。然後，您可以在例行資料收集間隔期間收集此資料並將它儲存到資料收集物件中。有關如何實施的資訊，請參閱使用者定義的種類。
管理資料收集物件	尋找用於管理資料收集物件所需的資訊，包括資料收集物件內容、資料收集物件儲存多久，以及您可使用資料收集物件來做什麼。
收集追蹤資料	「資料收集服務」收集取樣資料。不過，它不收集追蹤資料。請瞭解如何收集追蹤資料。
收集使用者定義的異動之效能資料	「資料收集服務」提供了可讓您定義自己的異動之 API。請瞭解如何以使用者定義的異動來執行此作業。

從資料收集服務資料建立資料庫檔案

「資料收集服務」可將您收集的資料放置到管理資料收集物件中。若要使用此資料，您必須先將資料放置到一組特定的資料庫檔案中。若要在收集資料時自動建立資料庫檔案，只要選取**啟動資料收集服務**對話框中的**建立資料庫檔案**即可。當您想要從現有的管理資料收集物件中將資料匯出至資料庫檔案時，也可以稍後建立資料庫檔案。

有許多選項可讓您建立資料庫檔案。

- 當使用「資料收集服務」來收集效能資料時，您可以在收集資料時自動建立資料庫檔案。
- 您可以從資料收集後儲存在其中的管理資料收集物件建立資料庫檔案。您可以使用建立效能資料 (CRTPFRTA) 指令，根據儲存於管理資料收集 (*MGTCOL) 物件中的效能資訊，建立一組效能資料庫檔案。您可以使用「iSeries 領航員」或 CRTPFRTA 指令。
- 您可以啟動 Performance Management/400 來自動啟動「資料收集服務」，然後在收集期間建立資料庫檔案。

您可以使用 Performance Tools for iSeries 授權程式或其它應用程式所建立的資料庫檔案來產生效能報告。您可以收集一個系統上的效能資料，然後將管理資料收集物件 (*MGTCOL) 移到另一個系統上，以產生效能資料檔及執行「效能工具」報告。此動作可讓您分析另一個系統上的效能資料，而不影響來源系統效能。有關「效能工具」的詳細資訊，請參閱 Performance Tools 。

將資料儲存於管理資料收集物件中而非資料庫檔案中

為何您應該將資料儲存於管理資料收集物件中，而不是儲存於執行報告所需的資料庫檔案中？因為您可以分別從資料庫檔案管理管理資料收集物件，您可以使用小資料收集間隔 (例如 5 分鐘間隔) 收集您的效能資料，以及使用較長取樣間隔 (例如 15 分鐘間隔) 建立您的資料庫檔案。

從單一管理資料收集物件中，您可以藉由指定不同的資料種類、不同的時間範圍及不同的取樣間隔，來建立不同用途的許多不同資料庫檔案集。

例如，您可以使用 5 分鐘間隔來 24 小時收集所有種類的效能資料 (所有資料或 **Standard plus protocol** 設定檔)。從一個管理資料收集物件中，您可以建立不同用途的不同資料庫檔案集。您可以建立一組資料庫檔案來執行一般每日效能報告。這些檔案可能包含取樣間隔 15 分鐘之所有種類的資料。然後，若要分析特定效能問題，您可以建立另一組資料庫檔案。這些檔案可能僅包含分析所需之單一種類的資料、24 小時內的特定時段，以及 5 分鐘更小的取樣間隔。

此外，單一管理資料收集物件可讓您將資料當作單一物件來管理，而非當作許多檔案來管理。單一資料收集物件可讓您在不同版次間移動效能資料，而不需要轉換資料。只要保留資料收集物件，您就可以回顧及分析效能相關事件，直到所收集的明細層次為止。

匯出收集的資料

若要從管理資料收集物件中將效能資料匯出至資料庫檔案，請遵循下列步驟：

1. 在「iSeries 領航員」中，選取**管理中心**下的端點系統，或選取**我的連線** (或您的作用中環境) 下您有直接連線的系統。
2. 展開**配置與服務**。
3. 按一下**資料收集服務**。
4. 以滑鼠右鍵按一下您要匯出至資料庫檔案的管理資料收集物件，然後選取**建立資料庫檔案**。
5. 在**建立資料庫檔案**對話框中，選取資料收集物件中要併入資料庫檔案的種類。只要資料收集物件包含支援您選項的資料，您也可以選取不同的時段及取樣間隔。
6. 按一下**確定**。

從現有的資料收集物件建立資料庫檔案： 您可以從現有的管理資料收集物件中將效能資料匯出至資料庫檔案。請遵循下列步驟：

1. 展開已收集效能資料的系統之**配置與系統**。
2. 選取**資料收集服務**。
3. 以滑鼠右鍵按一下您要從其中將資料匯出至資料庫檔案的管理資料收集物件。
4. 您可以先選取**內容**來顯示資料收集物件中的資料性質。在「資料內容」頁中，您可以看到這個資料收集物件中所收集的資料種類，以及收集的時間間隔。您可以利用此資訊來選取想要匯出的資料。複查此資訊後，按一下**確定**。
5. 以滑鼠右鍵再按一下管理資料收集物件，然後選取**建立資料庫檔案**。使用線上說明來完成欄位。
6. 按一下**確定**。

在轉換資料庫檔案中的資料後，您可以使用 Performance Tools for iSeries授權程式或其它應用程式來產生效能報告。

自訂資料資料收集

當使用「資料收集服務」來收集效能資料時，您可以控制收集哪些資料及多久收集一次。您可以在提供的資料收集設定檔中選擇。**標準**設定檔對應到前版次「啓動效能監視器 (STRPFRMON)」指令所提供 OS/400 效能監視器功能中的系統資料設定值。**Standard plus protocol** 設定檔對應到所有資料的 STRPFRMON 指令設定值。或者，您可以選取**自訂**來建立自訂的設定檔。其它可用設定檔的詳細說明，請參閱線上說明。針對您的自訂設定檔，您可以從可用的資料種類清單中選取，例如「系統 CPU」、「本端回應時間」、「磁碟儲存體」及 IOP (輸入/輸出處理器)。

針對您收集的每一個資料種類，您可以指定多久收集資料一次。若有許多種類，而您想選取預設資料資料收集間隔，您可以從預先定義的設定值 15 秒和 60 分鐘之間設定。(建議的設定值為 15 分鐘。)

註：當預設值設為任何指定的時間，所有種類皆使用指定的時間，但具有明確時間間隔的種類 (例如磁碟儲存體及輸入/輸出處理器) 及通信相關的種類除外。

儲存於管理資料收集物件 (類型 *MGTCOL) 中的已收集資料稱為一個資料收集。為防止這些管理資料收集物件太大，資料收集必須在固定間隔循環。循環資料收集意指建立一個新資料收集物件並在原始資料收集物件中資料收集停止的同時開始於新資料收集物件儲存資料。您可以根據資料使用方式來指定 1 小時至 24 小時的任何間隔。

若要在系統上自訂「資料收集服務」，請遵循下列步驟：

1. 在「iSeries 領航員」中，選取**管理中心**下的端點系統，或選取**我的連線** (或您的作用中環境) 下您有直接連線的系統。
2. 展開**配置與服務**。
3. 以滑鼠右鍵按一下**資料收集服務**，然後選取**內容**。
4. 在**一般**頁上，您可能要指定一個比預設值 1 天長的保留期。「資料收集服務」會在保留期到期後的任何時間，刪除系統中的管理資料收集物件及其內含的資料。當建立管理資料收集物件後，會對其指定一個到期日。即使您將資料收集物件移到另一個檔案庫中，「資料收集服務」仍會刪除到期的物件。如果您不希望「資料收集服務」對新資料收集物件指定到期日，您可以指定**永久**。如此的話，您就必須手動刪除這些資料收集物件。

若要檢視「圖形歷程」視窗，您必須指定「圖形」或「摘要」的「資料收集」保留期。當指定這些選項時，您可利用歷程報告功能來對超過延長期的多重系統做計量單位比較。

您也可以指定儲存資料收集的位置路徑、循環資料收集的週期，以及預設資料資料收集間隔。您可以選擇在資料收集期間自動建立資料庫檔案。

5. 按一下**收集資料**標籤。
6. 對於**使用的資料收集設定檔**，選取**自訂**。您可以對自訂清單中選取的每一個種類指定資料資料收集間隔。
7. 按一下**確定**來儲存自訂值。

一旦將「資料收集服務」自訂為您偏好的設定值後，您可以再使用滑鼠右鍵按一下**資料收集服務**，然後選取**啟動資料收集服務**來開始收集效能資料。

資料收集服務的時區注意事項

當您複查及分析效能資料時，集合的實際本端時間很重要。例如，您可能需要確定一天當中最忙碌時所收集的是什麼資料，它代表複查下的系統所遇到的最大工作量。如果您從中收集效能資料的某些系統位於不同的時區，則您應該瞭解以下這些注意事項：

- 當您對某個系統群組啟動「資料收集服務」的同時，您也對該群組中所有系統啟動了「資料收集服務」。由於某些系統位於不同的時區，所以系統日期與時間設定值的任何差異不納入考量。
- 如果您使用「管理中心」排程器啟動「資料收集服務」，則排程器啟動作業的時間視「管理中心」中您的中央系統日期與時間而定。
- 每一個端點系統的管理資料收集物件根據該端點系統及您中央系統的 `QTIME` 和 `QUTCOffset` (世界標準時間 (UTC) 位移)系統值來反映開始和結束時間。如果端點系統與您的中央系統位於不同的時區，而且兩系統上已正確設定這些系統值，則對資料收集物件報告的開始和結束時間就是端點系統的實際時間。換言之，開始和結束時間反映端點系統上的 `QTIME` 值，就像發生那些事件時它位於實際時間點一樣。
- 效能集合的排程可跨越從標準時間到日光節約時間或從日光節約時間到標準時間的界限。如果是這種情況，當排定開始時間時此時差應納入考慮。否則，實際開始和結束時間可能比預期的早或慢。此外，除非每次日光節約時間的變更生效時有調整 `QUTCOffset` 系統值，否則對管理資料收集物件報告的開始和結束時間會受到此差異的影響。

有關使用「資料收集服務」來收集效能資料的詳細資訊，請參閱資料收集服務。

資料收集服務中使用者定義的種類

資料收集服務中使用者定義的種類功能，可讓應用程式將效能資料收集整合到「資料收集服務」中。這可讓您撰寫資料收集程式、登錄它並與「資料收集服務」整合，而從應用程式收集資料。「資料收集服務」會在每一個資料收集間隔呼叫資料收集程式，並將資料儲存於資料收集物件中。您應該使用下面列示的「資料收集物件 API」來存取儲存於資料收集物件中的資料。只要資料已收集或資料收集物件已保留，您就可以即時存取該資料。

若要實施此功能，您必須：

1. 開發程式來收集「資料收集服務」中新種類的效能資料。詳細資訊，請參閱資料收集程式建議事項及需求。
2. 為資料收集程式建立工作說明。QGPL 中的工作說明 `QPMUSRCAT` 提供了範例，但不代表預設值或建議值。
3. 登錄新種類並指定資料收集程式。詳細資訊，請參閱 API 說明：
 - 登錄：`QypsRegCollectorDataCategory`
 - 取消登錄：`QypsDeregCollectorDataCategory`登錄該種類之後，「資料收集服務」將它併入可用的資料收集種類清單中。
4. 新增種類至您的「資料收集服務」設定檔中，然後循環「資料收集服務」
5. 開發程式來查詢資料收集物件。詳細資訊，請參閱 API 說明：
 - 擷取作用中管理資料收集物件名稱：`QpmRtvActiveMgtcolName` (僅用於即時查詢資料收集物件。)
 - 擷取管理資料收集物件屬性：`QpmRtvMgtcolAttrs`

- 開啟管理資料收集物件：QpmOpenMgtcol
- 關閉管理資料收集物件：QpmCloseMgtcol
- 開啟管理資料收集物件儲存庫：QpmOpenMgtcolRepo
- 關閉管理資料收集物件儲存庫：QpmCloseMgtcolRepo
- 讀取管理資料收集物件資料：QpmReadMgtcolData

您自訂的資料收集程式目前依每個資料收集間隔執行，而收集的資料保存在資料收集物件中。

您也可實施這些 API 的 Java 版本。必要的 Java 類別已併入 IFS 目錄 QIBM/ProdData/OS400/CollectionServices/lib 中的 ColSrv.jar。Java 應用程式應將這個檔案併入它們的類別路徑中。有關 Java 施行的詳細資訊，請參閱 javadocs。

如需施行範例，請參閱範例：實施使用者定義的種類

即時查詢資料收集物件

如果您的應用程式需要即時查詢資料收集物件，則必須將查詢與「資料收集服務」同步。若要這麼做，應用程式應建立一個資料佇列並對「資料收集服務」登錄它。登錄之後，收集器會在每一個資料收集間隔及資料收集循環結束時發出通知。應用程式應維護資料佇列，包括完成時移除資料佇列及處理異常終止。若要登錄及取消登錄資料佇列，請參閱下列 API 說明：

- 新增收集器通知：QypsAddCollectorNotification
- 移除收集器通知：QypsRmvCollectorNotification

資料收集程式建議及需求：「資料收集服務」會在啟動資料收集循環期間呼叫資料收集程式一次，在每個資料收集間隔呼叫一次，並且在結束資料收集循環時再呼叫一次。資料收集程式必須執行任何資料收集，並將該資料傳回「資料收集服務」所提供的資料緩衝區。除了提供資料緩衝區之外，「資料收集服務」還提供了工作區，可讓資料收集程式維護資料收集間隔之間的某些狀態資訊。

資料收集程式應儘快收集資料並執行最小格式化。此程式不應執行任何資料處理程序或排序。雖然使用者定義的種類之資料不會被轉換成資料庫檔案，不過「資料收集服務」會自動執行 CRTPFRTA 指令，並在每個資料收集間隔結束時將資料從資料收集物件新增至資料庫檔案中。如果資料收集程式無法在資料收集間隔內完成作業，則無法正確執行 CRTPFRTA 指令。

您可以在許多環境中建立資料收集程式：

- *PGM for OPM 語言。此環境不可用來查詢資料收集物件，因此可能導致效能降低。不過，較舊的程式設計語言有支援它。
- *SRVPGM，服務程式中的進入點。這代表 ILE 語言。
- *JVAPGM，併入 ColSrv.jar 中的必要 Java 類別。此檔案併入 QIBM/ProdData/OS400/CollectionServices/lib 的 IFS 中。下載 javadocs.zip 檔並開啟 index.html，取得 API 的 Java 施行說明。

「資料收集服務」將下列要求傳至資料收集程式：

要求	說明
啟動資料收集	資料收集程式應起始設定資料收集期間使用的任何介面或資源。選用性地，它也會起始設定「資料收集服務」提供的工作區，其中保留了資料收集間隔之間的狀態資訊。如果您要在收集的資料之前併入控制記錄，資料收集程式也會將少量資料寫入資料緩衝區。一般而言，這個控制記錄在資料處理期間用來協助解譯資料。

要求	說明
資料收集間隔	「資料收集服務」針對每個資料收集間隔傳送間隔要求。資料收集程式應收集資料並傳回資料緩衝區。然後，「資料收集服務」將該資料寫入資料收集物件中的間隔記錄。 如果資料量對資料緩衝區而言太大，資料收集程式應設定“More data”旗號。這個動作使「資料收集服務」傳送另一個間隔要求，它具有指示連續的修飾元。「資料收集服務」會在每個呼叫前重設其它資料旗號。這個程序會重複執行，直到所有資料移到資料收集物件中為止。
資料收集結束	當含有資料收集程式的種類之資料收集結束時，「資料收集服務」便會傳送此要求。資料收集程式應執行任何清除且可選擇性地傳回資料收集控制記錄。資料收集程式也應傳送一個指示資料收集結果的回覆碼。
清除並終止 (關閉)	如果異動終止是必要的，「資料收集服務」會傳送這個要求。當資料收集程式結束時，作業系統資源會自動釋出，但其它任何關閉作業應由資料收集程式執行。資料收集程式可在任何時候接收這個要求。

有關這些參數、工作區、資料緩衝區及回覆碼的詳細說明，請參閱位於 QSYSINC 中的標頭檔 QPMDCPRM。

資料收集物件中的資料儲存體

資料收集物件有一個用於每個資料收集種類的儲存庫。這個儲存庫是在啟動該種類的資料收集時由「資料收集服務」所建立的。每一個儲存庫由下列記錄組成：

記錄	說明
控制	這個選用性記錄可以是資料收集程式產生的第一個或最後一個記錄，也可能在這兩個位置同時出現。一般而言，它應該包含解譯記錄資料所需的任何資訊。
間隔	每一個資料收集間隔會建立間隔記錄，即使是空的。間隔記錄包含資料收集間隔期間寫入資料緩衝區的資料。其大小不可超過 4 GB。
停止	「資料收集服務」會自動建立這個記錄來指示資料收集階段作業結束。如果使用者定義的種類之資料收集重新啟動，但沒有結束或循環「資料收集服務」，您可以選擇性地在停止記錄後併入一個控制記錄，接著是其它間隔記錄。

範例：實施使用者定義的種類： 下列程式範例說明如何使用提供的 API 來將自訂的資料收集整合到「資料收集服務」中。

- 範例資料收集程式 (C++)
- 登錄資料收集程式 (C++) 的程式範例
- 查詢資料收集物件 (Java) 的程式範例

程式碼不保事項聲明

IBM 授與您非專用著作權來使用所有程式設計碼範例，您可以從中產生適合您特定需求的類似功能。

IBM 提供的所有程式碼範例僅供說明之用。這些範例並未在所有的條件下進行過完整的測試。因而，IBM 無法保證這些程式的可靠性、有用性或功能性。

此處包含的所有程式皆以「現狀」提供，不做任何保證。在此明確聲明放棄對於不侵權、適售性及特定用途的適用性之隱含保證。

範例：資料收集程式： 下列程式收集一些測試資料並儲存於資料緩衝區中，「資料收集服務」從中將資料複製至資料收集物件。有關資料收集程式的詳細資訊，請參閱資料收集程式建議事項及需求。

註：有關重要的法律資訊，請參閱程式碼不保事項聲明。

C++ 範例程式碼

```
#include "string.h"           // memcpy(), memset(), strlen()
#include "stdio.h"           // printf()
#include "qpmDCprm.h"        // data collection program interface
#include "time.h"

extern "C"
void DCPentry( Qpm_DC_Parm_t *request, char *dataBuffer,
              char *workArea, int *returnCode )
{
    static char testData[21] = "Just some test stuff";
    int i;

    /* Print contents of request structure */

    printf( "DCP called with parameters:\n" );
    printf( "  format name: \"%8.8s\"; category name: \"%10.10s\";\n",
            request->formatName, request->categoryName );
    printf( "  rsvd1: %4.4X; req type: %d; req mod: %d; buffer len: %d;\n",
            *(short *) (request->rsvd1), request->requestType,
            request->requestModifier, request->dataBufferLength );
    printf( "  prm offset: %d; prm len: %d; work len: %d; rsvd2: %8.8X;\n",
            request->parmOffset, request->parmLength, request->workAreaLength,
            *(int *) (request->rsvd2) );
    printf( "  rec key: \"%8.8s\"; timestamp: %8.8X %8.8X;\n",
            request->intervalKey,
            *(int *) (request->intervalTimestamp),
            *(int *) (request->intervalTimestamp + 4) );
    printf( "  return len: %d; more data: %d; rsvd3: %8.8X %8.8X;\n",
            request->bytesProvided, request->moreData,
            *(int *) (request->rsvd3),
            *(int *) (request->rsvd3 + 4) );

    switch ( request->requestType )
    {
        /* Write control record in the beginning of collection */
        case PM_DOBEGIN:
            printf( "doBegin(%d)\n", request->requestModifier );
            switch ( request->requestModifier )
            {
                case PM_CALL_NORMAL:
                    memcpy( dataBuffer, testData, 20 );
                    *(int *)workArea = 20;
                    request->moreData = PM_MORE_DATA;
                    request->bytesProvided = 20;
                    break;

                case PM_CALL_CONTINUE:
                    if ( *(int *)workArea < 200 )
                    {
                        memcpy( dataBuffer, testData, 20 );
                        *(int *)workArea += 20;
                        request->moreData = PM_MORE_DATA;
                        request->bytesProvided = 20;
                    }
                    else
                    {
                        *(int *)workArea = 0;
                        request->moreData = PM_NO_MORE_DATA;
                        request->bytesProvided = 0;
                    }
                    break;
            }
    }
}
```

```

        default:
            *returnCode = -1;
            return;
    }
    break;
/* Write control record in the end of collection */
case PM_DOEND:
    printf( "doEnd(%d)\n", request->requestModifier );
    switch ( request->requestModifier)
    {
        case PM_CALL_NORMAL:
            memcpy( dataBuffer, testData, 20 );
            *(int *)workArea = 20;
            request->moreData = PM_MORE_DATA;
            request->bytesProvided = 20;
            break;

        case PM_CALL_CONTINUE:
            if ( *(int *)workArea < 200 )
            {
                memcpy( dataBuffer, testData, 20 );
                *(int *)workArea += 20;
                request->moreData = PM_MORE_DATA;
                request->bytesProvided = 20;
            }
            else
            {
                *(int *)workArea = 0;
                request->moreData = PM_NO_MORE_DATA;
                request->bytesProvided = 0;
            }
            break;

        default:
            *returnCode = -1;
            return;
    }
    break;

/*Write interval record */
case PM_DOCOLLECT:
    printf( "doCollect(%d)\n", request->requestModifier );
    for ( i = 0; i < 10000; i++ )
        dataBuffer[i] = i % 256;
    request->bytesProvided = 10000;

    switch ( request->requestModifier)
    {
        case PM_CALL_NORMAL:
            *(time_t *)workArea = time( NULL );
            *(int *)workArea = 1;
            request->moreData = PM_MORE_DATA;
            break;

        case PM_CALL_CONTINUE:
            *(int *)workArea += 1;
            if ( *(int *)workArea < 20 )
                request->moreData = PM_MORE_DATA;
            else
            {
                *(time_t *)workArea = time( NULL );
                printf( "doCollect() complete in %d secs (%d bytes transferred)\n",
                    *(time_t *)workArea - *(time_t *)workArea, 10000 * 20 );
                request->moreData = PM_NO_MORE_DATA;
            }
            break;
    }

```

```

        default:
            *returnCode = -1;
            return;
    }
    break;
/* Clean-up and terminate */
case PM_DOSHUTDOWN:
    printf( "doShutdown\n" );
    *returnCode = 0;
    return;
    break;

default:
    *returnCode = -1;
    return;
    break;
}
}/* DCPentry() */

```

範例：登錄資料收集程式的程式： 下列程式使用「資料收集服務」登錄上一個範例中的資料收集程式。執行後，「資料收集服務」會在資料收集種類清單中顯示該資料收集程式。

C++ 範例程式碼

```

#include "stdlib.h"
#include "stdio.h"
#include "string.h"
#include "qypscoll.cleinc"

int main( int argc, char *argv[] )
{
    int    CCSID = 0;
    int    RC = 0;
    Qyps_USER_CAT_PROGRAM_ATTR    *pgmAttr;
    Qyps_USER_CAT_ATTR            catAttr;
    char   collectorName[11] = "*PFR ";
    char   categoryName[11] = "TESTCAT ";
    char   collectorDefn[11] = "*CUSTOM "; /* Register to *CUSTOM profile only */

    if ( argc > 2 )
    {
        int len = strlen( argv[2] );

        if ( len > 10 ) len = 10;
        memset( categoryName, ' ', 10 );
        memcpy( categoryName, argv[2], len );
    }

    if ( argc < 2 || *argv[1] == 'R' )
    {
        pgmAttr = (Qyps_USER_CAT_PROGRAM_ATTR *)malloc( 4096 );
        memset( pgmAttr, 0x00, sizeof(pgmAttr) );
        pgmAttr->fixedPortionSize = sizeof( Qyps_USER_CAT_PROGRAM_ATTR );
        memcpy( pgmAttr->programType,    "*SRVPGM ", 10 );
        memcpy( pgmAttr->parameterFormat, "PMDC0100", 8 );
        memcpy( pgmAttr->ownerUserId,    "USERID ", 10 );
        memcpy( pgmAttr->jobDescription, "QPMUSRCAT QGPL ", 20 );
        memcpy( pgmAttr->qualPgmSrvpgmName, "DCPTEST LIBRARY ", 20 );
        pgmAttr->workAreaSize = 123;
        pgmAttr->srvpgmEntrypointOffset = pgmAttr->fixedPortionSize;
        pgmAttr->srvpgmEntrypointLength = 8;
        pgmAttr->categoryParameterOffset = pgmAttr->srvpgmEntrypointOffset +
                                           pgmAttr->srvpgmEntrypointLength;
        pgmAttr->categoryParameterLength = 10;
        /* Set entry point name */
    }
}

```

```

memcpy( (char *)pgmAttr) + pgmAttr->srvpgmEntrypointOffset,
        "DCPentry", pgmAttr->srvpgmEntrypointLength ); /* Set parameter string */
memcpy( (char *)pgmAttr) + pgmAttr->categoryParameterOffset,
        "1234567890", pgmAttr->categoryParameterLength );

memset( &catAttr, 0x00, sizeof(catAttr) );
catAttr.structureSize = sizeof( Qyps_USER_CAT_ATTR );
catAttr.minCollectionInterval = 0;
catAttr.maxCollectionInterval = 0;
catAttr.defaultCollectionInterval = 30; /* Collect at 30 second interval */
memset( catAttr.qualifiedMsgId, ' ', sizeof(catAttr.qualifiedMsgId) );
memcpy( catAttr.categoryDesc,
        "123456789012345678901234567890123456789012345678901234567890", sizeof(catAttr.categoryDesc) );

QypsRegCollectorDataCategory( collectorName,
                             categoryName,
                             collectorDefn,
                             &CCSID,
                             (char*)pgmAttr,
                             (char*)&catAttr,
                             &RC
                             );
}
else
if( argc >= 2 && *argv[1] == 'D' )
    QypsDeregCollectorDataCategory( collectorName, categoryName, &RC );
else
    printf("Unrecognized option\n");
}/* main() */

```

範例：查詢資料收集物件的程式： 下列程式範例說明如何使用 QIBM/ProdData/OS400/CollectionServices/lib 中 ColSrv.jar 檔提供的 Java 類別來查詢儲存於資料收集物件中的資料。

Java 範例程式碼

```

import com.ibm.iseries.collectionservices.*;

class testmco2
{
    public static void main( String argv[] )
    {
        String    objectName = null;
        String    libraryName = null;
        String    repoName = null;
        MgtcolObj mco = null;
        int       repoHandle = 0;
        int       argc = argv.length;
        MgtcolObjAttributes
            attr = null;
        MgtcolObjRepositoryEntry
            repoE = null;
        MgtcolObjCollectionEntry
            collE = null;
        int       i,j;

        if ( argc < 3 )
        {
            System.out.println("testmco2 objectName libraryName repoName");
            System.exit(1);
        }

        objectName = argv[0];
        libraryName = argv[1];
        repoName = argv[2];

        if ( ! objectName.equals( "ACTIVE" ) )

```

```

    mco = new MgtcolObj( objectName, libraryName );
else
    try
    {
        mco = MgtcolObj.rtvActive();
    } catch ( Exception e)
    {
        System.out.println("rtvActive(): Exception " + e );
        System.exit(1);
    }
System.out.println("Object name = " + mco.getName() );
System.out.println("Library name = " + mco.getLibrary() );

try
{
    attr = mco.rtvAttributes( "MCOA0100" );
} catch ( Exception e)
{
    System.out.println("rtvAttributes(): MCOA0100: Exception " + e );
    System.exit(1);
}

System.out.println("MCOA0100: Object " + mco.getLibrary() + "/" + mco.getName() );
System.out.println("  size = " + attr.size + " retention = " + attr.retentionPeriod +
    " interval = " + attr.dftInterval + " time created = " + attr.timeCreated +
    " time updated = " + attr.timeUpdated );
System.out.println("  serial = " + attr.logicalPSN + " active = " + attr.isActive +
    " repaired = " + attr.isRepaired + " summary = " + attr.sumStatus +
    " repo count = " + attr.repositoryCount );
if ( attr.repositoryInfo != null )
    for(i = 0; i < attr.repositoryCount; i++ )
    {
repoE = attr.repositoryInfo[ i ];
System.out.println("      name = " + repoE.name + " category = " + repoE.categoryName +
    " size = " + repoE.size );
for( j = 0; j < repoE.collectionInfo.length; j++ )
    {
        collE = repoE.collectionInfo[ j ];
        System.out.println("          startTime = " + collE.startTime + " endTime = " + collE.endTime +
            " interval = " + collE.interval );
    }
    }

try
{
    attr = mco.rtvAttributes( "MCOA0200" );
} catch ( Exception e)
{
    System.out.println("rtvAttributes(): MCOA0200: Exception " + e );
    System.exit(1);
}

System.out.println("MCOA0200: Object " + mco.getLibrary() + "/" + mco.getName() );
System.out.println("  size = " + attr.size + " retention = " + attr.retentionPeriod +
    " interval = " + attr.dftInterval + " time created = " + attr.timeCreated +
    " time updated = " + attr.timeUpdated );
System.out.println("  serial = " + attr.logicalPSN + " active = " + attr.isActive +
    " repaired = " + attr.isRepaired + " summary = " + attr.sumStatus +
    " repo count = " + attr.repositoryCount );
if ( attr.repositoryInfo != null )
    for(i = 0; i < attr.repositoryCount; i++ )
    {
repoE = attr.repositoryInfo[ i ];
System.out.println("      name = " + repoE.name + " category = " + repoE.categoryName +
    " size = " + repoE.size );
for( j = 0; j < repoE.collectionInfo.length; j++ )
    {

```

```

    collE = repoE.collectionInfo[ j ];
    System.out.println("        startTime = " + collE.startTime + " endTime = " + collE.endTime +
        " interval = " + collE.interval );
}
}

if ( repoName.equals("NONE") )
return;

try
{
    mco.open();
} catch ( Exception e)
{
    System.out.println("open(): Exception " + e );
    System.exit(1);
}

try
{
    repoHandle = mco.openRepository( repoName, "MCO0100" );
} catch ( Exception e)
{
    System.out.println("openRepository(): Exception " + e );
    mco.close();
    System.exit(1);
}
System.out.println("repoHandle = " + repoHandle );

MgtcolObjReadOptions  readOptions = new MgtcolObjReadOptions();
MgtcolObjRecInfo  recInfo = new MgtcolObjRecInfo();

readOptions.option = MgtcolObjReadOptions.READ_NEXT;
readOptions.recKey = null;
readOptions.offset = 0;
readOptions.length = 0;

while ( recInfo.recStatus == MgtcolObjRecInfo.RECORD_OK )
{
    try
    {
        mco.readData( repoHandle, readOptions, recInfo, null );
    } catch ( Exception e)
    {
        System.out.println("readData(): Exception " + e );
        mco.close();
        System.exit(1);
    }

    if( recInfo.recStatus == MgtcolObjRecInfo.RECORD_OK )
    {
        System.out.print("Type = " + recInfo.recType );
        System.out.print(" Key = " + recInfo.recKey );
        System.out.println(" Length = " + recInfo.reclength );
    }

}/* while ... */

mco.closeRepository( repoHandle );
mco.close();

}/* main() */
}/* class testmco2 */

```

管理資料收集物件

當使用「資料收集服務」來收集效能資料時，每一個集合儲存於單一物件中。若要查看任何管理資料收集物件中資料的摘要，請遵循下列步驟：

1. 在「iSeries 領航員」中，選取**管理中心**下的端點系統，或選取**我的連線** (或您的作用中環境) 下您有直接連線的系統。
2. 展開**配置與服務**。
3. 選取**資料收集服務**。
4. 以滑鼠右鍵按一下清單中任何管理資料收集物件，然後選取**內容**來查看集合的一般資訊及其內含的資料摘要。

您可以使用滑鼠右鍵按一下任何資料收集物件，然後選取建立資料庫檔案來指定資料種類、資料收集期間內的時間範圍，以及您要併入資料庫檔案中的取樣間隔。

您可以使用滑鼠右鍵按一下任何資料收集物件，然後選取 **圖形歷程**，以圖形方式檢視管理資料收集物件中的資料。

刪除或保留舊的管理資料收集物件

您可以從系統中刪除資料收集物件，方法是以滑鼠右鍵按一下該物件，然後選取**刪除**。如果您沒有手動刪除物件，「資料收集服務」會在到期日期及時間之後自動刪除它們。

「資料收集服務」僅刪除**循環**的管理資料收集物件。已**循環**的狀態意指「資料收集服務」已停止收集資料及將資料儲存在物件中。當您展開**配置與服務**並選取**資料收集服務**時，資料收集物件清單中顯示每一個管理資料收集物件的狀態。

當「資料收集服務」下次啟動或循環集合時，它會刪除已屆到期日期和時間之已循環的資料收集物件。到期日與管理資料收集物件有關。即使您將資料收集物件移到另一個檔案庫中，「資料收集服務」仍會刪除到期的物件。

每一個管理資料收集物件的到期日顯示在該資料收集物件的「內容」中。若要使系統上的物件保留更長的時間，您只要變更「內容」頁上的日期即可。以滑鼠右鍵按一下清單中任何管理資料收集物件，然後選取**內容**來查看該集合的相關資訊。如果您不要「資料收集服務」為您刪除管理資料收集物件，您可以指定**永久**。

使用者定義的異動

「資料收集服務」及效能探究程式現在可收集您在應用程式中定義的效能資料。透過提供的 API，您可以利用「資料收集服務」將異動資料整合到定期排定的取樣資料集中，並執行效能探究程式來取得有關異動的追蹤層次資料。

有關詳細說明及用法注意事項，請參閱下列 API 說明：

- 啟動異動：QYPESTRT, qypeStartTransaction
- 結束異動：QYPEENDT, qypeEndTransaction
- 記載異動：QYPELOGT, qypeLogTransaction (僅適用效能探究程式)
- 新增追蹤點：QYPEADDT, qypeAddTracePoint (僅適用效能探究程式)

註： 您只需要指定一次應用程式。「資料收集服務」及「效能探究程式」使用相同的 API 呼叫來收集不同類型的效能資料。

將使用者定義的異動資料整合到「資料收集服務」中

您可以選取使用者定義的異動作為「資料收集服務」配置中資料收集的種類。「資料收集服務」會在每個資料資料收集間隔收集異動資料並將該資料儲存於資料收集物件中。CRTPFRDTA 指令可將此資料匯出至使用者定義的異動效能資料庫檔案 QAPMUSRTNS。「資料收集服務」會依照異動類型來組織資料。您可以指定所需要的異動類型；不過，「資料收集服務」會報告前 15 個異動類型。其它異動類型的資料會被合併及儲存為 *OTHER 異動類型。在每一個資料收集間隔，「資料收集服務」會對每個唯一工作的每一個異動類型建立一個記錄。如需詳細說明，請參閱「啟動異動 API」中的用法注意事項。

「資料收集服務」收集一般異動資料，例如異動回應時間。您也可以併入最多 16 個選用性應用程式定義的計數器，來追蹤應用程式特定資料，例如用於異動的 SQL 陳述式數目或其它增量測量。您的應用程式應使用「啟動異動 API」來指示新異動的開端，而且應併入對應的「結束異動 API」來將異動資料傳遞至「資料收集服務」。詳細資訊，請參閱 QAMUSRTNS 檔案說明及 API 說明。

有關施行範例，請參閱 C++ 或 Java 中的範例。

註：有關重要的法律資訊，請參閱程式碼不保事項聲明。

使用效能探究程式收集使用者定義的異動之追蹤資訊

您可以在效能探究程式階段作業期間使用啟動、結束及記載異動 API 來建立追蹤記錄。效能探究程式將現行緒的系統資源使用率 (例如 CPU 使用率、I/O 及掌握/鎖定活動) 儲存於這些追蹤記錄中。另外，您可以選擇併入應用程式特定的效能資料，然後將它傳至每一個 API 中的效能探究程式。您也可以使用新增追蹤點 API 來識別效能探究程式應收集追蹤資料的應用程式特定事件。

若要針對您的異動來啟動效能探究程式階段作業，請對您的效能探究程式定義的 (OSEVT) 參數指定 *USRTRNS。輸入 ENDPX 指令之後，效能探究程式會將應用程式提供的資料寫入 QAYPEMIUSR 效能探究程式資料庫檔案中的 QMUDTA 欄位。啟動、結束及任何日誌記錄之系統提供的效能資料儲存於 QAYPEMIUSR 和 QAYPETIDX 資料庫檔案中。

如需詳細說明，請參閱「啟動異動 API」說明中的用法注意事項及 API 說明。

C++ 範例：將使用者定義的異動整合到資料收集服務中： 下列 C++ 程式範例顯示如何使用啟動異動和結束異動 API 來將使用者定義的異動效能資料整合到「資料收集服務」中。

註：有關重要的法律資訊，請參閱程式碼不保事項聲明。

```
//*****  
// tnstst.C  
//  
// This example program illustrates the use  
// of the Start/End Transaction APIs (qypeStartTransaction,  
// qypeEndTransaction).  
//  
//  
// This program can be invoked as follows:  
// CALL lib/TNSTST PARM('threads' 'types' 'transactions' 'delay')  
// where  
// threads = number of threads to create (10000 max)  
// types = number of transaction types for each thread  
// transactions = number of transactions for each transaction  
// type  
// delay = delay time (millisecs) between starting and  
// ending the transaction  
//  
// This program will create "threads" number of threads. Each thread  
// will generate transactions in the same way. A thread will do  
// "transactions" number of transactions for each transaction type,  
// where a transaction is defined as a call to Start Transaction API,  
// then a delay of "delay" millisecs, then a call to End Transaction
```



```

// API. Thus, each thread will do a total of "transactions" * "types"
// number of transactions. Each transaction type will be named
// "TRANSACTION_TYPE_nnn" where nnn ranges from 001 to "types". For
// transaction type n, there will be n-1 (16 max) user-provided
// counters reported, with counter m reporting m counts for each
// transaction.
//
// This program must be run in a job that allows multiple threads
// (interactive jobs typically do not allow multiple threads). One
// way to do this is to invoke the program using the SBMJOB command
// specifying ALWMLTTHD(*YES).
//
//*****

#define _MULTI_THREADED

// Includes
#include "pthread.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "qusec.h"
#include "lbcpynv.h"
#include "qypesvpg.h"

// Constants
#define maxThreads 10000

// Transaction pgm parm structure
typedef struct
{
    int types;
    int trans;
    int delay;
} tnsPgmParm_t;

// Error code structure
typedef struct
{
    Qus_EC_t error;
    char Exception_Data[100];
} error_code_t;

//*****
//
// Transaction program to run in each secondary thread
//
//*****

void *tnsPgm(void *parm)
{
    tnsPgmParm_t *p = (tnsPgmParm_t *)parm;

    char tnsTyp[] = "TRANSACTION_TYPE_XXX";
    char pexData[] = "PEX";
    unsigned int pexDataL = sizeof(pexData) - 1;
    unsigned long long colSrvData[16] = {1,2,3,4,5,6,7,8,
                                          9,10,11,12,13,14,15,16};

    unsigned int colSrvDataL;
    char tnsStrTim[8];

    struct timespec ts = {0, 0};

    error_code_t errCode;

    _DPA_Template_T target, source; // 用於 LBCPYNV MI instr

```

```

unsigned int typCnt;
unsigned int tnsCnt;
int rc;

// Initialize error code
memset(&errCode, 0, sizeof(errCode));
errCode.error.Bytes_Provided = sizeof(errCode);

// Initialize delay time
ts.tv_sec = p->delay / 1000;
ts.tv_nsec = (p->delay % 1000) * 1000000;

// Loop doing transactions
for (tnsCnt = 1; tnsCnt <= p->trans; tnsCnt++)
{
    for (typCnt = 1; typCnt <= p->types; typCnt++)
    {
        // Set number field in transaction type
        source.Type = _T_UNSIGNED;
        source.Length = 4;
        source.reserved = 0;
        target.Type = _T_ZONED;
        target.Length = 3;
        target.reserved = 0;
        _LBCPYNV(tnsTyp + 17, &target, &typCnt, &source);

        // Set Coll Svcs data length in bytes
        colSrvDataL = (typCnt <= 16) ? (typCnt - 1) : 16;
        colSrvDataL = colSrvDataL * 8;

        // Call Start Transaction API
        qypeStartTransaction(tnsTyp,
            (unsigned int *)&tnsCnt,
            pexData,
            (unsigned int *)&pexDataL,
            tnsStrTim,
            &errCode);

        // Delay specified amount
        rc = pthread_delay_np(&ts);

        // Call End Transaction API
        qypeEndTransaction(tnsTyp,
            (unsigned int *)&tnsCnt,
            pexData,
            (unsigned int *)&pexDataL,
            tnsStrTim,
            (unsigned long long *)&colSrvData[0],
            (unsigned int *)&colSrvDataL,
            &errCode);
    }
}

return NULL;
}

//*****
//
// Main program to run in primary thread
//
//*****

void main(int argc, char *argv[])
{

```

```

// Integer version of parms
int threads; // # of threads
int types;   // # of types
int trans;   // # of transactions
int delay;   // Delay in millisecs

pthread_t threadHandle[maxThreads];
tnsPgmParm_t tnsPgmParm;
int rc;
int i;

// Verify 4 parms passed
if (argc != 5)
{
    printf("Did not pass 4 parms\n");
    return;
}

// Copy parms into integer variables
threads = atoi(argv[1]);
types   = atoi(argv[2]);
trans   = atoi(argv[3]);
delay   = atoi(argv[4]);

// Verify parms
if (threads > maxThreads)
{
    printf("Too many threads requested\n");
    return;
}

// Initialize transaction pgm parms (do not modify
// these while threads are running)
tnsPgmParm.types = types;
tnsPgmParm.trans = trans;
tnsPgmParm.delay = delay;

// Create threads that will run transaction pgm
for (i=0; i < threads; i++)
{
    // Clear thread handle
    memset(&threadHandle[i], 0, sizeof(pthread_t));
    // Create thread
    rc = pthread_create(&threadHandle[i], // Thread handle
                       NULL,             // Default attributes
                       tnsPgm,           // Start routine
                       (void *)&tnsPgmParm); // Start routine parms

    if (rc != 0)
        printf("pthread_create() failed, rc = %d\n", rc);
}

// Wait for each thread to terminate
for (i=0; i < threads; i++)
{
    rc=pthread_join(threadHandle[i], // Thread handle
                    NULL);          // No exit status
}

} /* end of Main */

```

Java 範例：將使用者定義的異動整合到資料收集服務中： 下列 Java 程式範例顯示如何使用啓動異動和結束異動 API 來將使用者定義的異動效能資料整合到「資料收集服務」中。

註：有關重要的法律資訊，請參閱程式碼不保事項聲明。

```

import com.ibm.iseries.collectionservices.PerformanceDataReporter;

// parameters:
// number of TXs per thread
// number of threads
// log|nolog
// enable|disable
// transaction seconds

public class TestTXApi
{
    static TestTXApiThread[]    thread;

    static private String[] TxTypeString;
    static private byte[][] TxTypeArray;

    static private String TxEventString;
    static private byte[] TxEventArray;

    static
    {
        int i;

        // initialize transaction type strings and byte arrays

        TxTypeString = new String[20];
        TxTypeString[ 0] = "Transaction type 00";
        TxTypeString[ 1] = "Transaction type 01";
        TxTypeString[ 2] = "Transaction type 02";
        TxTypeString[ 3] = "Transaction type 03";
        TxTypeString[ 4] = "Transaction type 04";
        TxTypeString[ 5] = "Transaction type 05";
        TxTypeString[ 6] = "Transaction type 06";
        TxTypeString[ 7] = "Transaction type 07";
        TxTypeString[ 8] = "Transaction type 08";
        TxTypeString[ 9] = "Transaction type 09";
        TxTypeString[10] = "Transaction type 10";
        TxTypeString[11] = "Transaction type 11";
        TxTypeString[12] = "Transaction type 12";
        TxTypeString[13] = "Transaction type 13";
        TxTypeString[14] = "Transaction type 14";
        TxTypeString[15] = "Transaction type 15";
        TxTypeString[16] = "Transaction type 16";
        TxTypeString[17] = "Transaction type 17";
        TxTypeString[18] = "Transaction type 18";
        TxTypeString[19] = "Transaction type 19";

        TxTypeArray = new byte[20][];
        for ( i = 0; i < 20; i++ )
            try
            {
                TxTypeArray[i] = TxTypeString[i].getBytes("Cp037");
            } catch(Exception e)
            {
                System.out.println("Exception \"" + e + "\" when converting");
            }
    }

    /* static */

    public static void main( String[] args )
    {
        int    numberOfTXPerThread;
        int    numberOfThreads;
        boolean log;
    }
}

```

```

boolean enable;
int     secsToDelay;

// process parameters
if ( args.length >= 5 )
try
    {
        numberOfTXPerThread = Integer.parseInt( args[0] );
        numberOfThreads     = Integer.parseInt( args[1] );

        if ( args[2].equalsIgnoreCase( "log" ) )
log = true;
        else
            if ( args[2].equalsIgnoreCase( "nolog" ) )
                log = false;
            else
                {
                    System.out.println( "Wrong value for 3rd parameter!" );
                    System.out.println( "\tshould be log|nolog" );
                    return;
                }

            if ( args[3].equalsIgnoreCase( "enable" ) )
enable = true;
            else
                if ( args[3].equalsIgnoreCase( "disable" ) )
                    enable = false;
                else
                    {
                        System.out.println( "Wrong value for 4th parameter!" );
                        System.out.println( "\tshould be enable|disable" );
                        return;
                    }

            secsToDelay = Integer.parseInt( args[4] );

        } catch (Exception e)
        {
            System.out.println( "Oops! Cannot process parameters!" );
            return;
        }
        else
        {
            System.out.println( "Incorrect Usage." );
            System.out.println( "The correct usage is:" );
            System.out.println( "java TestTXApi numberOfTXPerThread numberOfThreads log|nolog enable|disable secsToDelay");
            System.out.println("\tlog will make the program cut 1 log transaction per start / end pair");
            System.out.println("\tdisable will disable performance collection to minimize overhead");
            System.out.print("\nExample: \"java TestTXApi 10000 100 log enable 3\" will call ");
            System.out.println("cause 10000 transactions for each of 100 threads");
            System.out.println("with 3 seconds between start and end of transaction");
            System.out.println("Plus it will place additional log call and will enable reporting." );
            return;
        }

        System.out.println( "Parameters are processed:" );
        System.out.println( "\tnumberOfTxPerThread = " + numberOfTXPerThread );
        System.out.println( "\tnumberOfThreads = " + numberOfThreads );
        System.out.println( "\tlog = " + log );
        System.out.println( "\tenable = " + enable );
        System.out.println( "\tsecsToDelay = " + secsToDelay );

// cause initialization of a PerformanceDataReporter class
    {
        PerformanceDataReporter pReporter = new PerformanceDataReporter();
        pReporter.enableReporting();
    }

```

```

TestTXApi t = new TestTXApi( );

    System.out.println( "\nAbout to start ..." );
    t.prepareTests( numberOfTxPerThread, numberOfThreads, log, enable, secsToDelay );

long startTime = System.currentTimeMillis();

    t.runTests( numberOfThreads );

    // wait for threads to complete
for ( int i = 0; i < numberOfThreads; i++ )
    try
    {
        thread[i].join( );
    } catch(Exception e)
    {
        System.out.println( "***Exception \"" + e + "\" while joining thread " + i );
    }

long endTime = System.currentTimeMillis();

    System.out.println( "\nTest runtime for " + ( numberOfTxPerThread * numberOfThreads ) +
        " TXs was " + ( endTime - startTime ) + " msec" );

}/* main() */

private void prepareTests( int numberOfTxPerThread,
                          int numberOfThreads, boolean log, boolean enable, int secsToDelay )
{
    System.out.println( "Creating " + numberOfThreads + " threads");
    thread = new TestTXApiThread[numberOfThreads];
    for ( int i = 0; i < numberOfThreads; i++ )
        thread[i] = new TestTXApiThread( i, numberOfTxPerThread,
                                         log, enable, secsToDelay );
}/* prepareTests() */

private void runTests( int numberOfThreads )
{
    for ( int i = 0; i < numberOfThreads; i++ )
        thread[i].start( );
}/* runTests() */

private class TestTXApiThread extends Thread
{
    private int    ordinal;
    private int    numberOfTxPerThread;
    private boolean log;
    private boolean enable;
    private int    secsToDelay;

    private PerformanceDataReporter    pReporter;

    private long    timeStamp[];
    private long    userCounters[];

    public TestTXApiThread( int ordinal, int numberOfTxPerThread,
                          boolean log, boolean enable, int secsToDelay )
    {
        super();
        this.ordinal    = ordinal;
        this.numberOfTxPerThread = numberOfTxPerThread;
        this.log        = log;
        this.enable     = enable;
    }
}

```

```

        this.secsToDelay      = secsToDelay;

        pReporter = new PerformanceDataReporter( false );
        if ( enable )
            pReporter.enableReporting();
        timeStamp = new long[1];
        userCounters = new long[16];
        for ( int i = 0; i < 16; i++ )
            userCounters[i] = i;

    }/* constructor */

    public void run()
    {
        int i;

        for ( i = 0; i < numberOfTxPerThread; i++ )
        {
            pReporter.startTransaction( TxTypeArray[i%20], i, TxTypeArray[i%20], 20, timeStamp );
            // pReporter.startTransaction( TxTypeArray[i%20], i, TxTypeString[i%20], timeStamp );
            if ( log )
                pReporter.logTransaction( TxTypeArray[i%20], i, TxTypeArray[i%20], 20 );
            // pReporter.logTransaction( TxTypeArray[i%20], i, TxTypeString[i%20] );
            if (secsToDelay > 0)
                try
                {
                    Thread.sleep(secsToDelay * 1000);
                } catch(Exception e) { }
            pReporter.endTransaction( TxTypeArray[i%20], i, TxTypeArray[i%20], 20, timeStamp,
                userCounters );
            // pReporter.endTransaction( TxTypeArray[i%20], i, TxTypeString[i%20], timeStamp,
            // userCounters );
        }

    }/* run() */

}/* class TestTXApiThread */

}/* class TestTXApi */

```

效能資料檔

效能資料是關於系統 (或系統網路) 運作的資訊集，您可以利用該資訊瞭解回應時間和產能。您可以使用效能資料調整程式、系統屬性和作業。這些調整可改善回應時間和產能。調整也可以協助您預測系統、作業或程式某些變更的效果。

資料收集服務將效能資料收集到管理資料收集物件 (*MGTCOL)。「建立效能資料 (CRTPFDRDTA)」指令處理來自該資料收集物件的資料並將結果儲存到效能資料庫檔案。資料庫檔案分成下列種類：

含有時間間隔資料的效能資料檔

這些檔案包含按照每個間隔收集的效能資料。關於這些檔案清單、簡短說明以及關於每一個檔案的完整資訊的鏈結，請參閱含有時間間隔的效能資料檔資料。如果要瞭解這些檔案中的資料來自哪裡，請參閱系統種類與檔案關係。檢視這些檔案時，您也會發現檔案縮寫很有用。

配置資料檔

每個階段作業收集一次配置資料。如果要瞭解這些檔案中的資料來自哪裡，請參閱系統種類與檔案關係。您可以在配置資料檔找到 QAPMCONF、QAPMHDWR 和 QAPMSBSD 檔案。

追蹤資料檔

唯有選擇收集追蹤資料之後才會收集追蹤資料。您可以在追蹤資料檔找到 QAPMDMPT 檔。

使用系統提供的「顯示檔案欄位說明 (DSPFFD)」指令可以顯示其他欄位資訊，例如位元組數和緩衝區位置。例如，在任何指令行鍵入下列指令：

```
DSPFFD file(QSYS/QAPMCONF)
```

關於 iSeries 效能的詳細資訊，請參閱效能。

含有時間間隔資料的效能資料檔

如果要檢視關於某效能資料檔的完整資訊，請從下面清單選取要檢視的檔案 (按字母順序顯示)。

檔案	說明
QAPMAPPN	APPN 資料
QAPMASYN	非同步統計值 (一個鏈結一個)
QAPMBSC	二進位同步統計值 (一個鏈結一個)
QAPMBUS	匯流排計數器 (一個匯流排一個)
QAPMCIOP	通信 IOP 資料 (一個 IOP 一個)
QAPMDDI	分散式數位介面 (DDI) 資料 (一個鏈結一個)
QAPMDIOP	儲存裝置 IOP 資料 (一個 IOP 一個)
QAPMDISK	磁碟儲存資料 (一個讀寫頭一個)
QAPMDOMINO	Domino for iSeries 資料 (一個 domino 伺服器一筆記錄)
QAPMECL	記號環檔案登錄 (一個鏈結一個)
QAPMETH	乙太網路統計值 (一個鏈結一個)
QAPMFRLY	訊框傳送資料 (一個鏈結一個)
QAPMHDL	HDLC 統計值 (一個鏈結一個)
QAPMHTTPB	IBM HTTP 伺服器的基本資料 (Apache 驅動) (一台伺服器一個)
QAPMHTTPD	IBM HTTP 伺服器的詳細資料 (Apache 驅動) (一個伺服器元件一個)
QAPMIDLC	整合服務數位網路 (ISDN) 資料鏈結控制檔案登錄 (一個鏈結一個)
QAPMIOPD	延伸的 IOP 資料 (特殊 IOP 的其它資料) (一個 IOP 一個)
QAPMJOBMI	MI 工作資料 (一個工作、作業或緒一筆記錄)。在使用本文件時您會發現關於作業類型延伸器的資訊很有用
QAPMJOBOS	工作作業系統資料 (一項工作一筆記錄)
QAPMJOBS 和 QAPMJOBL	工作資料 (一項工作、作業或緒一筆記錄)
QAPMJOBWT	工作、作業和緒等候條件
QAPMJOBWTD	在 QAPMJOBWT 檔發現的計數器組的說明。
QAPMJSUM	按工作群組的工作摘要資料 (一個工作群組一筆記錄)
QAPMLAPD	整合服務數位網路 (ISDN) LAPD 檔案登錄 (一個鏈結一個)
QAPMLIOP	雙軸工作站控制器資料 (一個實體控制器一個)
QAPMMIOP	多功能 IOP (一個 IOP 一個)
QAPMPOOL 和 QAPMPOOLL	主要儲存體資料 (一個系統儲存體儲存區一個)
QAPMPOOLB	儲存體儲存區資料 (一個儲存區一個)
QAPMPOOLT	儲存體儲存區調整資料 (一個儲存區一個)
QAPMPPP	點對點通信協定 (PPP) 資料 (一個鏈結一個)
QAPMRESP	本端工作站回應時間 (一個工作站一個)
QAPMRWS	遠端工作站回應時間

檔案	說明
QAPMSAP	TRLAN、Ethernet、DDI 和 Frame Relay SAP 檔案登錄 (一個 SAP 登錄一個)
QAPMSNA	SNA 資料
QAPMSNADS	SNADS 資料 (一項 SNADS 工作一個)
QAPMSTND	DDI 站資料
QAPMSTNE	乙太網路站檔案登錄
QAPMSTNL	記號環站檔案登錄
QAPMSTNY	訊框傳送站檔案登錄
QAPMSYS 和 QAPMSYSL	系統效能資料
QAPMSYSCPU	系統 CPU 使用量資料
QAPMSYSTEM	系統層次效能資料
QAPMTCP	TCP/IP 資料
QAPMTCPIFC	個別 TCP/IP 介面的 TCP/IP 資料
QAPMUSRTNS	使用者定義的異動資料 (每一項工作有每一種異動的一筆記錄)
QAPMX25	X.25 統計值 (一個鏈結一個)

效能資料檔：檔案縮寫

在欄位和位元組資料表中效能資料檔使用縮寫。這些縮寫包含：

縮寫	說明
主檔案	這些檔案與種類相關並從種類產生。
C	「屬性」直欄中的字元。
PD	「屬性」直欄中的壓縮十進位數。
Z	「屬性」直欄中的區化十進位數。
IOP	輸入/輸出處理器 (IOP) 或 I/O 處理器。控制主電腦系統與其他裝置 (例如磁碟、顯示站和通信線路) 之間的活動的處理器。
DCE	資料電路終端設備。
MAC	媒體存取控制。通信 IOP 中的一個實體。
LLC	邏輯鏈結控制。通信 IOP 中的一個實體。
指標訊框	當環無法運作時傳送的訊框。
類型 II 訊框	Systems Network Architecture (SNA) 使用的連接導向訊框 (資訊訊框)。
資訊訊框 (I-frame)	一個資訊訊框。

效能資料檔：資料收集服務系統種類與檔案關係

當您使用資料收集服務收集效能資料時，該資料儲存於管理集合 (*MGTCOL) 物件中。CRTPFRTA 指令可從該管理資料收集物件匯出資料，然後將資料寫入效能資料檔中。「資料收集服務」可獨立控制及收集的每一種資料類型以一個資料種類代表。每一個資料種類包含或提供寫入一或多個效能資料檔的資料。若要建立資料庫檔案或成員，該檔案或成員相依的種類 (或種類群組) 必須存在且由 CRTPFRTA 處理。下表指示種類與檔案的關係。關係有三種類型：

關係	說明
主要檔案	這些檔案與產生自其中的種類相關。

關係	說明
相容性檔案	這些檔案為邏輯檔案，它結合主要檔案來提供與先前檔案結構的效能資料庫相容性。如果系統產生了所有參與的檔案 (主要種類)，也會產生相容性檔案。
次要檔案	這些檔案包含種類或主要檔案內含之資料所衍生的資料，而且與它們相關。不過，不是由該種類控制。

以下是使用者注意事項：

1. 唯有資料庫檔案是所選取種類的主要檔案時，CRTPFRTDA 指令才會產生該檔案。
2. 如果列示的是多個種類的主要檔案，您必須選取每一個種類來產生檔案。
3. 如果一個種類的主要檔案被列示為另一種類的次要檔案，您必須選取第二個種類以確保所產生資料庫檔案中的資訊完整。例如，如下表所示，若要產生完整資料庫檔案 QAPMECL，您必須選取 *CMNBASE 和 *CMNSTN。
4. 唯有當系統產生了所有相關的主要檔案，該系統才會產生相容性檔案。

下表說明系統種類與效能資料庫檔案之間的關係。

種類	主要檔案	相容性檔案	次要檔案
*SYSBUS	QAPMBUS		
*POOL	QAPMPOOLB	QAPMPOOLL	
*POOLTUNE	QAPMPOOLT	QAPMPOOLL	
*HDWCFG	QAPMHDWR		
*SUBSYSTEM	QAPMSBSD		
*SYSCPU	QAPMSYSCPU	QAPMSYSL	
*SYSLVL	QAPMSYSTEM	QAPMSYSL	
*JOBMI	QAPMJOBMI QAPMJOBWT QAPMJOBWTD QAPMJSUM	QAPMJOBL QAPMSYSL	QAPMSYSTEM
*JOBOS	QAPMJOBOS QAPMJSUM	QAPMJOBL QAPMSYSL	QAPMSYSTEM
*SNADS	QAPMSNADS		
*DISK	QAPMDISK		QAPMSYSTEM
*IOPBASE	QAPMLIOP QAPMDIOP QAPMCIOP QAPMMIOP		
*IPCS	QAPMIOPD QAPMTSK		

*CMNBASE	QAPMASYN QAPMBSC QAPMDDI QAPMECL QAPMETH QAPMFRLY QAPMHDLC QAPMIDLC QAPMLAPD QAPMPPP QAPMX25		
*CMNSTN	QAPMSTND QAPMSTNE QAPMSTNL QAPMSTNY 無		QAPMDDI QAPMETH QAPMECL QAPMFRLY QAPMX25
*CMNSAP	QAPMSAP		
*LCLRSP	QAPMRESP		
*APPN	QAPMAPPN		
*SNA	QAPMSNA		
*EACACHE	無		QAPMDISK (請參閱附註)
*TCPBASE	QAPMTCP		
*TCPIFC	QAPMTCPIFC		
*DOMINO	QAPMDOMINO		
*HTTP	QAPMHTTPB QAPMHTTPD		
*USRTNS	QAPMUSRTNS		
註：			
CRTPFRDTA 不能選取這個種類。不過，它可以讓 *DISK 種類報告其它資料。			

效能資料檔案： 配置資料庫檔案的欄位資料

每一個階段作業收集配置資料一次。下列效能資料檔案顯示系統配置資料、子系統資料及硬體配置資料的檔名、簡要說明和欄位資料明細 (如果有提供) 參照。有關資料收集服務如何產生此檔案及資料來源位置的資訊，請參閱系統種類和檔案關係。

欄位名稱	說明
QAPMCONF	系統配置資料。
QAPMHDWR	系統硬體配置。
QAPMSBSD	子系統資料。無欄位及位元組資料。

有關此主題的其它資訊，請參閱效能資料庫檔案概觀。

效能資料庫檔案： 追蹤資料庫檔案的欄位資料

追蹤資料包含內部系統追蹤資料。這是您爲了取得特定工作及異動的其餘相關資訊，所收集的詳細資料。除非您使用「效能工具」授權程式來分析資料，否則不應收集此類型的資料。下列是使用「啟動效能追蹤 (STRPFRTRC)」指令時，系統支援的效能資料檔。

檔名	說明
QAPMDMPT	系統追蹤資料 (無欄位或位元組明細)。

如需此主題的其餘相關資訊，請參照效能資料庫檔案概觀。

iSeries 領航員監督程式

「iSeries 領航員」包含的監督程式使用資料收集服務資料追蹤您感興趣的系統效能的元素。不過，當發生某些事件時 (例如 CPU 使用率的百分比或工作的狀態)，它們可以採取指定動作。您可以使用數個監督程式查看和管理發生在多個系統和系統群組的系統效能。

有了多個監督程式，您可以啟動一個監督程式，然後切換到伺服器 (在「iSeries 領航員」中) 或您的 PC 上的其他作業。事實上，您甚至可以關閉 PC。「iSeries 領航員」繼續監督和執行您指定的臨界指令或動作。監督程式會一直執行，直到您停止它為止。您也可以透過使用 iSeries Navigator for Wireless 存取監督程式，使用監督程式遠端管理效能。

「iSeries 領航員」提供下面幾類監督程式：

系統監督程式

當發生效能資料時收集和顯示效能資料或收集和顯示效能資料最多達 1 小時。詳細的圖形協助您瞭解當它發生時伺服器執行什麼工作。從不同計量單位選擇 (效能測量) 來精確找出系統效能的特定層面。例如，如果在伺服器監督平均 CPU 使用率，您可以按一下圖形上的任何集合點，查看一個顯示最高 CPU 使用率的 20 項工作的明細圖表。然後您可以在任何這些工作按一下滑鼠右鍵來直接處理工作。

工作監測程式

根據工作名稱、工作使用者、工作類型、子系統或伺服器類型監督一項工作或工作清單。從不同計量單位選擇一個計量單位去監督工作的效能、狀態或錯誤訊息。如果要直接處理某工作，請在「工作監測程式」視窗所顯示的清單中，在該工作上按一下滑鼠右鍵。

訊息監督程式

確認是否順利完成應用程式或監督對於您的業務需求來說是很重要的訊息。從「訊息監督程式」視窗，您可以查看訊息明細、回答訊息、傳送訊息以及刪除訊息。

B2B 活動監督程式

如果有配置像 Connect for iSeries 的應用程式，您可以使用 B2B 活動監督程式監督 B2B 異動。您可以持續檢視活動異動圖，而且您可以在觸發臨界值時自動執行指令。您可以搜尋和顯示特定異動以及檢視該特定異動詳細步驟的長條圖。

檔案監督程式

監督一個或數個選取的檔案是否含有特定字串、監督特定大小的一個或數個選取的檔案，或監督一個或數個選取的檔案是否有發生修改。

如果要進一步瞭解監督程式，請參閱下列主題：

監督程式概念

監督程式可以顯示即時效能資料。此外，它們可以繼續監督系統以便在到達指定臨界值時執行選取的指令。瞭解監督程式如何運作、它們可以監督什麼以及它們如何能夠回應特定的效能狀況。

配置監督程式

您可以在「iSeries 領航員」配置監督程式。使用本主題瞭解如何設定監督程式以及如何配置它充分運用可用的選項。

實務範例

本主題提供一些實務範例，它們說明如何使用部份不同類型監督程式來查看系統效能的特定部份。

監督程式概念

系統監督程式顯示資料收集服務所產生及維護之資料收集物件中儲存的資料。系統監督程式顯示所收集的資料最多 1 小時。若要檢視更長時間的資料，您應該使用圖形歷程。您可以變更監督程式內容中資料收集的頻率，這會置換「資料收集服務」中的設定值。

您可以使用監督程式來追蹤及重新搜尋許多不同的系統效能元素，並且可同時執行許多不同的監督程式。當同時使用時，監督程式提供一個尖端工具來觀察及管理系統效能。例如，當實施新的交談式應用程式時，您可以使用系統監督程式來排定以下的優先順序：工作的資源使用率、工作監測程式監視及處理任何有問題的工作，以及在您任何系統上發生指定訊息時訊息監督程式通知您。

設定臨界值及動作

當建立新監督程式時，您可以指定系統計量單位到達指定臨界值層次或發生事件時所要採取的動作。當臨界值層次或事件發生時，您可以選擇在端點系統上執行 OS/400 指令，例如傳送訊息或保留工作佇列。另外，您可以選擇讓監督程式執行一些預先定義的動作，例如更新事件日誌及在您的 PC 上發出警鈴聲來通知您，或啓動監督程式。最後，您可以藉由指定第二個臨界值層次來自動重設監督程式，第二個臨界值層次可使監督程式回復正常活動。

配置監督程式

系統監督程式為高度交談式工具，您可以用來收集及顯示來自端點系統的即時效能資料。從**新監督程式**視窗開始建立新監督程式是一個既快速又容易的程序：

1. 在「iSeries 領航員」中，展開「管理中心」，選取**監督程式**，以滑鼠右鍵按一下**系統**，然後選取**新監督程式**。
2. 指定一個監督程式名稱。從**新監督程式-一般**頁上，為您的監督程式指定名稱。提供一個簡要說明，以便可在監督程式清單中找到該監督程式。
3. 選取計量單位。使用**新監督程式-計量單位**頁來選取計量單位。您可以監督任何端點系統或系統群組上任何的計量單位。
4. 檢視及變更您的計量單位資訊。使用**新監督程式-計量單位**頁來編輯每個計量單位的內容。您可以編輯資料資料收集間隔、最大繪製值，以及所選取每個計量單位的顯示時間。
5. 設定臨界值指令。使用**計量單位**頁上的**臨界值**標籤，來啓用臨界值並指定觸發或重設臨界值時在端點系統上執行的指令。
6. 設定臨界值動作。使用**新監督程式-動作**頁來指定觸發或重設計量單位臨界值時所要採取的動作。
7. 選取您的系統與群組。使用**新監督程式-系統與群組**頁來選取您要啓動監督程式所在的端點系統或系統群組。

建立您的監督程式之後，請以滑鼠右鍵按一下監督程式名稱，然後選取**啓動**來執行監督程式並開始處理監督程式圖形。

監督計量單位

若要有效地監督系統效能，您必須決定監督哪一方面的系統效能。「管理中心」提供了各種效能測量 (稱為**計量單位**) 來協助您找出不同方面的系統效能。

新監督視窗中的**計量單位**頁，可讓您檢視及變更所要監督的計量單位。若要存取此頁，請選取**監督**，以滑鼠右鍵按一下**系統**，然後選取**新監督**。填寫必要的欄位，然後按一下**計量單位**標籤。

當配置監督程式時，您可以從要併入監督程式的清單中使用任何計量單位、一組計量單位或全部計量單位。在您監督程式中可用的計量單位類型如下：

計量單位群組： CPU 使用率	計量單位說明： 您系統上的工作所耗用之可用處理單元時間百分比。從您監督程式使用的下列 CPU 使用率計量單位類型中選擇： <ul style="list-style-type: none">• CPU 使用率 (平均值)• CPU 使用率 (交談式作業)• CPU 使用率 (交談式效能)• CPU 使用率 (資料庫功能)• CPU 使用率 (次要工作量)• 基本 CPU 使用率 (平均值) 若要詳細瞭解這些計量單位及其用法，請參閱「iSeries 領航員」中 監督程式內容 視窗或 新監督程式 視窗的 一般 標籤上可用的線上說明。
交談式回應時間 (平均值和最大值)	交談式作業在您系統經歷的回應時間。
異動率 (平均值)	您系統上所有工作完成的每秒異動數。
異動率 (交談式)	您系統上下列工作類型完成的每秒異動數： <ul style="list-style-type: none">• 交談式• 多重要求程式終端機 (MRT)• System/36 交談式環境• 透通
批次邏輯資料庫 I/O	系統上批次工作目前所執行邏輯資料庫輸入/輸出 (I/O) 作業平均值。
磁碟讀寫臂使用率 (平均值和最大值)	您收集資料期間系統目前使用的磁碟讀寫臂產能百分比。
磁碟儲存體 (平均值和最大值)	您收集資料期間系統最大的磁碟讀寫臂儲存體百分比。
磁碟 IOP 使用率 (平均值和最大值)	您收集資料期間系統上磁碟輸入/輸出處理器 (IOP) 的忙碌程度。
通信 IOP 使用率 (最大值和平均值)	您收集資料期間系統上通信輸入/輸出處理器 (IOP) 的忙碌程度。
通信線路使用率 (平均值和最大值)	您所有通信線路實際送出及接收的資料量。
LAN 使用率 (最大值和平均值)	您所有區域網路 (LAN) 通信線路實際送出及接收的資料量。
機器儲存區錯誤	系統上機器儲存區中每秒發生的錯誤數。
使用者儲存區錯誤 (最大值和平均值)	系統上所有使用者儲存區中每秒發生的錯誤數。

如果需要其它說明，請在**新監督程式-計量單位**視窗中按一下**說明**按鈕。一旦您熟悉「管理中心」計量單位，則要選取哪個計量單位視您運算環境所需的資訊而定。在選定以您想要查看的資訊為目標之計量單位後，您可以開始檢視及變更對監督程式所選取之每個計量單位的詳細計量單位資訊。

實務範例：iSeries 領航員監督程式

併入「iSeries 領航員」中的監督程式提供一組功能強大的工具來重新搜尋及管理系統效能。有關「iSeries 領航員」提供的監督程式類型概觀，請參閱 iSeries 領航員監督程式。

有關詳細使用範例及範例配置的資訊，請參閱下列實務範例：

實務範例：系統監督程式

如果 CPU 使用率太高，範例系統監督程式會對您提出警告，並暫時保留任何低優先順序工作，直到其它資源可用為止。

實務範例：訊息監督程式

範例訊息監督程式顯示您任何 iSeries 伺服器上的訊息佇列之任何查詢訊息。此監督程式會開啓並顯示所偵測到的訊息。

實務範例：工作監測程式

範例工作監測程式可追蹤指定工作的 CPU 使用率並在 CPU 使用率太高時警告工作擁有者。

實務範例：系統監督程式:

狀況

身為系統管理者，您必須確定 iSeries 系統有足夠資源以滿足使用者和業務需求的目前要求。對於您的系統來說，CPU 使用率特別重要。您希望在 CPU 使用率變得太高時系統能夠提出警告，而且暫時保留低優先順序工作直到有其他資源可用為止。

爲了完成這個目的，您可以設定在 CPU 使用率超過 80% 時傳給您訊息的系統監督程式。另外，它也可以將工作保留在 QBATCH 工作佇列直到 CPU 使用率降到 60%，當降到 60% 時會釋放工作，然後回復正常作業。

配置範例

如果要設定系統監督程式，您必須定義要追蹤的計量單位以及當計量單位到達指定層次時監督程式要執行的動作。如果要定義完成這項目標的系統監督程式，請執行下列步驟：

1. 在「iSeries 領航員」，展開**管理中心 > 監督程式**，在**系統監督程式**按一下滑鼠右鍵，然後選取**新監督程式 ...**
2. 在**一般**頁面，輸入這個監督程式的名稱和說明。
3. 按一下**計量單位**標籤，然後輸入下列值：
 - a. 從「可用的計量單位」清單選取 **CPU 使用率基本 (平均值)**，然後按一下**新增**。目前「CPU 使用率基本 (平均值)」列示在要監督的「計量單位」下面，視窗底端顯示這個計量單位的設定值。
 - b. 對於**資料收集間隔**來說，請指定要收集這個資料的頻率。這樣會置換「資料收集服務」設定。對於這個範例來說，請指定 **30 秒**。
 - c. 如果要變更這個計量單位的監督程式圖的垂直軸的比例，請變更**最大圖形值**。如果要變更這個計量單位的圖形的水平軸的比例，請變更**顯示時間**這個值。

d. 按一下計量單位設定值的**臨界值 1** 標籤，然後輸入下列值以便在「使用率」大於或等於 80% 時傳送查詢訊息：

- 1) 選取**啟用臨界值**。
- 2) 對於臨界觸發值來說，請指定 **>= 80** (大於或等於百分之 80 在工作中)。
- 3) 對於**持續時間**來說，請指定 **1** 個間隔。
- 4) 對於 **OS/400 指令**來說，請指定下列項目：
SNDMSG MSG('Warning,CPU...') TOUSR(*SYSOPR) MSGTYPE(*INQ)
- 5) 對於臨界值重設值來說，請指定 **< 60** (小於百分之 60 在工作中)。這樣會在 CPU 使用率低於 60% 時重設監督程式。

e. 按一下**臨界值 2** 標籤，然後輸入下列值以便在經過 5 個資料收集間隔期間 CPU 使用率一直高於 80% 時，將所有工作保留在 QBATCH 工作佇列：

- 1) 選取**啟用臨界值**。
- 2) 對於臨界觸發值來說，請指定 **>= 80** (大於或等於百分之 80 在工作中)。
- 3) 對於**持續時間**來說，請指定 **5** 個間隔。
- 4) 對於 **OS/400 指令**來說，請指定下列項目：
HLDJOBQ JOBQ(QBATCH)
- 5) 對於臨界值重設值來說，請指定 **< 60** (小於百分之 60 在工作中)。這樣會在 CPU 使用率低於 60% 時重設監督程式。
- 6) 對於**持續時間**來說，請指定 **5** 個間隔。
- 7) 對於 **OS/400 指令**來說，請指定下列項目：

RLSJOBQ JOBQ(QBATCH)

這個指令會在經過 5 個資料收集間隔期間 CPU 使用率仍保持低於 60% 時釋放 QBATCH 工作佇列。

4. 按一下**動作**標籤，然後在**觸發和重設**直欄選取**日誌事件**。這個動作會在觸發和重設臨界值時在事件日誌建立登錄。
5. 按一下**系統與群組**標籤指定要監督的系統與群組。
6. 按一下**確定**儲存監督程式。
7. 從系統監督程式清單，在新監督程式按一下滑鼠右鍵並選取**啟動**。

結果

新監督程式根據指定的資料收集間隔顯示 CPU 使用率，每 30 秒新增一次新資料點。每次 CPU 使用率到達 80% 時監督程式會自動執行指定的臨界動作 (即使 PC 關閉也一樣)。

註：這個監督程式只追蹤 CPU 使用率。不過，您可以將任何數目的可用計量單位加入同一個監督程式，而且每一個計量單位可以有自已的臨界值和動作。您也可以同時執行數個系統監督程式。

實務範例：工作監測程式：

狀況

您目前正在 iSeries 伺服器執行新應用程式，您擔心部份新交談式工作會佔用太多資源。您希望當佔用太多資源的擁有者的工作消耗太多 CPU 容量時通知他們這個情況。

您可以設定工作監測程式監督來自新應用程式的工作，而且在一項工作消耗 30% CPU 容量時傳送訊息。

配置範例

如果要設定工作監測程式，您必須定義要監督的工作、要監督的工作屬性，以及當偵測到指定的工作屬性時監督程式應執行的動作。如果要設定完成這個目標的工作監測程式，請執行下列步驟：

1. 在「iSeries 領航員」，展開**管理中心 > 監督程式**，在**工作監測程式**按一下滑鼠右鍵，然後選取**新監督程式...**
2. 在**一般**頁面，輸入下列值：
 - a. 指定這個監督程式的名稱和說明。
 - b. 在**要監督的工作**標籤，輸入下列值：
 - 1) 對於**工作名稱**來說，請指定要監督的工作名稱 (例如，MKWIDGET)。
 - 2) 對於**子系統**來說，請指定 QINTER。
 - 3) 按一下**新增**。
3. 按一下**計量單位**標籤，然後輸入下列資訊：
 - a. 在**可用的計量單位**清單，展開**彙總數值**，選取 **CPU 百分比使用率**，然後按一下**新增**。
 - b. 在計量單位設定值的**臨界值 1** 標籤，輸入下列值：
 - 1) 選取**啟用觸發程式**。
 - 2) 對於**臨界觸發值**來說，請指定 **>= 30** (大於或等於百分之 30 在工作中)。
 - 3) 對於**持續時間**來說，請指定 **1** 個間隔。
 - 4) 對於 **OS/400 觸發程式指令**來說，請指定下列項目：
SNMSG MSG ('您的工作超過 30% CPU 容量') TOUSR(&OWNER)
 - 5) 按一下**啟用重設**。
 - 6) 對於**臨界值重設值**，請指定 **< 20** (小於百分之 20 在工作中)。
4. 按一下**資料收集間隔**標籤，然後選取 **15 秒**。這樣會置換「資料收集服務」設定。
5. 按一下**動作**標籤，然後在**觸發和重設**直欄選取**日誌事件**。
6. 按一下**伺服器和群組**標籤，然後選取要監督這項工作的伺服器和群組。
7. 按一下**確定**儲存新監督程式。
8. 從工作監測程式清單，在新監督程式按一下滑鼠右鍵並選取**啟動**。

結果

新監督程式每 15 秒檢查一次 QINTER 子系統，如果 MKWIDGET 這個工作消耗超過百分之 30 的 CPU，監督程式會傳送訊息給工作擁有者。當工作使用小於 20% CPU 容量時監督程式會重設。

實務範例：訊息監督程式:

狀況

貴公司同時執行數台 iSeries 伺服器，不過檢查每一個系統的訊息佇列很費時。身為系統管理者，您必須在系統出現查詢訊息時知道有這些訊息。

您可以設定訊息監督程式顯示發生在 iSeries 系統的訊息佇列的查詢訊息。當此監督程式偵測到訊息時它會開啓和顯示訊息。

配置範例

如果要設定訊息監督程式，您必須定義想要監督的訊息類型以及當出現這些訊息時要監督程式執行的動作。如果要設定完成這目標的訊息監督程式，請執行下列步驟：

1. 在「iSeries 領航員」，展開**管理中心 > 監督程式**，在**訊息監督程式**按一下滑鼠右鍵，然後選取**新監督程式...**
2. 在**一般**頁面，輸入這個監督程式的名稱和說明。
3. 按一下**訊息標籤**，然後輸入下列值：
 - a. 對於**要監督的訊息佇列**來說，請指定 **QSYSOPR**。
 - b. 在**訊息集 1** 標籤，選取**查詢類型**，然後按一下**新增**。
 - c. 選取**到達下面訊息計數時觸發**，然後指定 **1** 則訊息。
4. 按一下**資料收集間隔**標籤，然後選取 **15 秒**。
5. 按一下**動作**標籤，然後選取**開啓監督程式**。
6. 按一下**系統與群組**標籤，然後選取要監督查詢訊息的系統與群組。
7. 按一下**確定**儲存新監督程式。
8. 從訊息監督程式清單，在新監督程式按一下滑鼠右鍵並選取**啓動**。

結果

新訊息監督程式在被監督的 iSeries 伺服器顯示傳給 QSYSOPR 的查詢訊息。

註：這個監督程式只回應傳給 QSYSOPR 的查詢訊息。不過，您可以將兩個不同訊息集加入一個監督程式，而且您可以同時執行數個訊息監督程式。收到指定的訊息時訊息監督程式也可以執行 OS/400 指令。

圖形歷程

圖形歷程提供幾天前、幾週前、幾個月前或幾年前使用資料收集服務所收集效能資料的圖形檢視畫面。您不需要執行系統監督程式來檢視效能資料。只要使用「資料收集服務」來收集資料，您就可以檢視「圖形歷程」視窗。

• 圖形歷程概念

圖形歷程可用的歷史資料量大多根據「資料收集服務」中的資料收集保留期值及 PM/400 是否啓用而定。有關管理及顯示效能資料記錄的可用選項說明，請參閱本主題。

• 使用圖形歷程

圖形歷程可透過「iSeries 領航員」來存取。有關逐步指示，請參閱本主題。

有關監督系統效能的詳細資訊，請參閱追蹤效能資料主題。

圖形歷程概念

圖形歷程顯示資料收集服務所建立之資料收集物件內含的資料。因此，可用的資料類型及數量視您的「資料收集服務」配置而定。

可以繪製的資料量取決於您在「資料收集服務」內容中所選取的設定值，特別是集合保留期。使用「iSeries 領航員」在多個系統上啓動 PM/400。當啓動 PM/400 時，您可以使用圖形歷程功能來查看幾天前、幾週前或幾個月前的資料。您可以超越即時監督功能，並存取摘要或詳細資料。如果沒有啓用 PM/400，圖形資料欄位支援 1 至 7 天。如果啓用 PM/400，您可以定義管理資料收集物件保留在系統上多長的時間：

- **詳細資料**

管理資料收集物件被刪除前保留在檔案系統中的時間長度。您可以選取特定時間週期 (以小時或日為單位)，或選取**永久**。如果您選取**永久**，則不會自動刪除管理資料收集物件。

- **圖形資料**

「圖形歷程」視窗中的明細及內容資料被刪除前保留在系統中的時間長度。如果您沒有啟動 **PM/400**，則可指定 1 至 7 天。如果您啟動了 **PM/400**，則可指定 1 至 30 天。預設值為 1 小時。

- **摘要資料**

圖形的資料收集點在被刪除前可於「圖形歷程」視窗中顯示或保留在系統中的時間長度。沒有可用的明細或內容資料。您必須啟動 **PM/400** 才能啟用摘要資料欄位。預設值為 1 個月。

使用圖形歷程

圖形歷程併入於「iSeries 領航員」中。若要檢視使用資料收集服務所監督資料的圖形歷程，請執行下列步驟：

1. 若要在單一系統或系統群組上啟動「資料收集服務」，請遵循「iSeries 領航員」線上說明指示。
2. 從**啟動資料收集服務 - 一般**頁上，選取**啟動 Performance Management/400** (如果需要的話)。
3. 變更資料收集保留期的其它值。
4. 按一下**確定**。
5. 您可以檢視圖形歷程，方法是使用滑鼠右鍵按一下系統監督程式或「資料收集服務」物件，然後選取**圖形歷程**。
6. 按一下**重新整理**來查看圖形檢視畫面。

一旦啟動圖形歷程，會出現一個顯示一系列圖形資料收集點的視窗。圖形線上的這些資料收集點由三個不同的圖形識別，這三個圖形對應到三個可用的資料層次：

- 正方形資料收集點，代表同時包含詳細資訊和內容資訊的資料。
- 三角形資料收集點，代表包含詳細資訊的彙總資料。
- 圓形資料收集點，代表不含詳細資訊或內容資訊的資料。

Performance Management/400

Performance Management/400 是容易使用的自動化和自我管理工具。**PM/400** 自動觸發「資料收集服務」從伺服器收集非專屬效能和產能資料，然後傳送資料給 **IBM**。所有資料收集站都受到網路保護而且轉送的時間完全由您來控制。傳送資料給 **IBM** 時，您不必自己儲存所有趨勢資料。**IBM** 會為您儲存資料並提供一系列顯示伺服器成長與效能的報告和圖形。您可以使用傳統瀏覽器以電子方式存取報告。這些報告可透過不斷分析重要效能指示符來協助您規劃和管理系統資源。

IBM Operational Support Services for PM/400e 提供選項包含一組報告、圖形和設定檔，它們協助您將現行應用程式和硬體效能最大化 (透過使用效能趨勢分析)。這項提供也協助您更瞭解 (透過使用產能規劃) 您的業務趨勢與必要的硬體升級 (例如 **CPU** 或磁碟) 時機的關聯。趨勢系統使用率資源和產量資料提供產能規劃資訊，您可以將它視為伺服器的早期警告系統。將 **PM/400e** 視為通知您關於系統的「健康」狀態的虛擬資源。

PM/400 使用小於中央處理單元 (**CPU**) 的百分之一。它使用大約 **58 MB** 磁碟空間，根據硬體機型和資料收集間隔大小決定磁碟空間。

PM/400 概念

瞭解 **PM/400** 可以提供的功能和優點以及重要的實施注意事項。

配置 PM/400

如果要開始使用 **PM/400**，您必須啟動它，請設定一個可讓您傳送資料和接收報告的傳輸方法，最後再自訂資料收集和儲存體。

管理 PM/400

現在您已設定網路，您可以使用 PM/400 執行各種不同作業。

PM/400 報告

您可以配置 iSeries 伺服器使用 PM/400 將「資料收集服務」的資料直接傳給 IBM。然後 IBM 會產生數份您可以在 Web 檢視的報告，或直接將報告傳給您。啟動 PM/400 自動產生報告不僅節省您的時間和資源，也能讓您透過預測未來成長需求來預先做規劃。

PM/400 概念

PM/400 使用「資料收集服務」從伺服器收集非專屬效能和產能資料，然後傳送資料給 IBM。這個資訊可以包含 CPU 使用率和磁碟容量、回應時間、產能、應用程式和使用者用量。傳送資料給 IBM 時，您不必親自儲存所有趨勢資料。IBM 為您儲存資料而且提供一系列顯示伺服器成長和效能的報告和圖形。您可以使用傳統瀏覽器以電子方式存取報告。

PM/400 的優點

PM/400 可以使管理系統資源和產能規劃變得很容易。了解更多特定方法來利用 PM/400。

PM/400 提供選項

PM/400 提供許多種選項。使用這個資訊決定最適合您需求的服務組合。

資料收集注意事項


PM/400 使用資料收集服務收集效能資料。瞭解 PM/400 和「資料收集服務」如何搭配來提供您需要的資料。

PM/400 的優點： 使用 PM/400 服務時，您獲得下列好處：

- **協助您避免不幸意外。**
您可避免令人失望的意外。您控制著系統的成長與效能的管理，這表示由您來管理系統。而不是系統管理您。
- **節省時間。**
您透過自動收集與報告效能資料而省略既費力、成本又高的作業。這項好處讓您能夠將資源集中在管理系統和應用程式。
- **可讓您事先規劃最大效率。**
您可以積極規劃財務需求來以最高效率執行系統。
- **提供容易瞭解的資訊。**
在您瞭解資訊之後，當上級主管問「為什麼我們需要升級？」這種問題時，您就能夠輕易地向他報告
- **可讓您預測未來。**
您可以根據實際趨勢資訊預測資料處理的成長。
- **可讓您識別系統問題。**
PM/400 資料可讓您識別效能瓶頸。
- **可讓您協助預估下一個升級的大小。**

您可以將 PM/400 資料上傳到 Workload Estimator for iSeries  來調整下一個升級的大小。

關於使用 PM/400 之前需要執行什麼作業的詳細資訊，請參閱 Performance Management/400。

PM/400e 提供的作業支援服務： 您可以採取電子方式或列印方式接收圖形和報告。您可以每月接收電子圖。您每月或每季接收列印圖形。PM/400e 服務費根據您選擇接收效能資訊的頻率以及選擇的格式（電子式或列印格式）而不同。這些報告選項中有些是免費的，有些則要付費。每一個國家中的行銷和服務組織可以提供您關於可用的支援的明細。請探訪 PM/400e 網站  取得關於免費和付費選項的資訊。

關於使用 PM/400 之前需要執行什麼作業的詳細資訊，請參閱 Performance Management/400。

PM/400 的資料收集注意事項： 建立系統使用率、工作量和效能測量的正確趨勢時，最重要的要求就是一致性。理想的情況應該是每天 24 小時收集效能資料。因為 PM/400 與「資料收集服務」之間的關係，您必須知道當使用 PM/400 時會出現的含意。

下列一些準則協助您在使用 PM/400 時定義資料收集：

- **選取 QMPGDATA 檔案庫儲存資料。**

當 PM/400 是作用中時，**要儲存資料收集的位置**欄位使用 /QSYS.LIB/QMPGDATA.LIB 預設值。如果以其他值置換 QMPGDATA，PM/400 會在整點時循環資料收集並將它變回到 QMPGDATA。如果要將資料收集到另一個檔案庫，請變更 PM/400 尋找資料的位置。從指令行鍵入 **GO PM400**，選取選項 3 (處理自訂)，然後變更檔案庫名稱。

- **使用「資料收集服務」連續收集資料。**

PM/400 使用「資料收集服務」一天 24 小時收集資料來滿足這項要求。PM/400 每 15 分鐘收集一次效能資料。PM/400 使用 15 分鐘間隔預設值，但不變更設定的間隔。15 分鐘間隔是建議的間隔。

- **選取 Standard plus 通信協定設定檔。**

Standard plus 通信協定是資料收集設定檔的預設值。資料收集設定檔指示收集什麼資料。Standard plus 通信協定設定檔中的資料種類對應「啟動效能監視器 (STRPFRMON)」指令上的 DATA 參數的 *ALL 值。如果將這個值變更成其他值，PM/400 會在整點時將它變更回原來的值。即使選取「自訂」並包含所有種類也一樣。變更立即生效。資料收集不會循環 (除非有其他理由而必須要循環)。執行這個動作收集 PM/400 報告的足夠資訊。

- **當 PM/400 是作用中時避免臨時變更資料收集。**

例如，在啟動 PM/400 時會勾選**在收集期間建立資料庫檔案**欄位作為預設值。如果變更預設值，PM/400 會在整點時將它變更回預設值。變更立即生效。資料收集不會循環 (除非有其他理由而必須要循環)。

- **結束資料收集服務**

您可以隨時從「iSeries 領航員」結束「資料收集服務」。如果結束「資料收集服務」，那麼當執行 PM/400 時請考慮下列注意事項：

- PM/400 排程器在下一個小時開頭啟動「資料收集服務」。
- 含有少數資料或沒有收集資料的日期不併入趨勢計算。因此您不可經常岔斷「資料收集服務」。

如果不要啟動「資料收集服務」，您可以暫時關閉 PM/400。

配置 PM/400

PM/400 透過「資料收集服務」自動收集效能資料。只要檔案庫常駐在基本輔助儲存體儲存區 (ASP)，您就可以指定要放置資料的檔案庫。檔案庫不可以移到獨立的輔助儲存體儲存區，因為獨立的輔助儲存體儲存區有可能被轉斷，轉斷就會停止 PM/400 收集處理。如果還沒有檔案庫，那麼在啟動期間 PM/400 會建立檔案庫。

如果要開始使用 PM/400，您必須執行下列作業：

啟動 PM/400

PM/400 隨附於 OS/400，但您必須啟動它才能使用它的收集功能。

決定要使用哪一種傳輸方法

決定要如何傳送資料。您可以使用「管理中心」收集資料，使用「電子服務代理程式 (最大支援)」傳送資料，或者可以讓 PM/400 透過 SNA 通信協定收集資料和傳送它。

自訂 PM/400

現在您已設定網路，您可能需要自訂 PM/400 以滿足您的需求。

啓動 Performance Management/400: 您必須啓動 PM/400 才能利用它的資料收集功能。您可以使用下列其中一種方法啓動 PM/400：

使用「iSeries 領航員」

使用「iSeries 領航員」透過多個系統啓動 PM/400。啓動 PM/400 時，您可以使用圖形歷程功能，查看幾天前、幾週前或幾個月前收集的資料。您可以超越即時監督程式功能。您有彙總資料或詳細資料的存取權。在沒有啓用 PM/400 情況下，圖形資料欄位支援 1 到 7 天。在有啓用 PM/400 情況下，您可選擇保留資料的時間長度。

如果要從「iSeries 領航員」啓動 PM/400，請執行下列步驟：

1. 在「iSeries 領航員」，展開要在那裡啓動 PM/400 的系統。
2. 展開**配置和服務**。
3. 在**資料收集服務**按一下滑鼠右鍵。
4. 選取 **Performance Management/400**。
5. 選取**啓動**。
6. 選取要在那裡啓動 PM/400 的系統。
7. 按一下**確定**。

回答 QSYSOPR 訊息佇列中的 CPAB02A 訊息

啓動 QSYSWRK 子系統時，這則訊息會詢問您是否要啓動 PM/400。

1. 從字元型介面，對 QSYSOPR 中的訊息“您要啓動 PM/400 嗎？(I G C)”回答 G。QSYSOPR 訊息佇列接收表示已啓動 PM/400 的訊息。
2. 更新聯絡資訊。發出 **GO PM400** 指令並指定選項 1。

發出「配置 PM/400 (CFGPM400)」指令

從字元型介面，您可以發出「配置 PM/400 (CFGPM400)」指令。

您可以繼續執行設定處理中的下一個步驟，此步驟將決定使用哪一種傳輸方法傳送資料給 IBM。

關於 iSeries 效能主題的概觀，請參閱效能。

決定要使用的 PM/400 傳輸方法: 從 V5R1 開始，PM/400 傳輸處理會利用您以管理中心執行的網路配置，來設定中央系統和端點系統。不過，您仍可以使用字元型介面配置 PM/400。選擇要使用的傳輸方法：

- 使用「透過最大支援的電子服務代理程式」傳送資料
如果您選擇這種傳輸方法，您必須配置 PM/400 讓「管理中心」收集資料。如果伺服器有安裝 V4R5 或以上版本的作業系統（您也必須套用 Universal Connection 修訂程式），請請為 PM/400 執行這項配置。如果使用「最大支援」，那麼您需要選擇這種方法。
- 使用 SNA 通信協定傳送資料
如果選擇這種傳輸方法，您必須使用字元型介面配置 PM/400。PM/400 使用 SNA 收集資料和傳輸資料。如果伺服器有安裝 OS/400 V4R5 或以下的版本，請為 PM/400 執行這項配置。

一旦實施要使用的傳輸方法之後，您可以執行其他作業來管理 PM/400。

使用透過最大支援的服務代理程式傳送 PM/400 資料 (廣用連線): PM/400 使用「資料收集服務」來收集您伺服器中的非生產性效能及產能資料。收集此資料之後，您可以使用透過最大支援的服務代理程式來傳送資料至 IBM。

若要利用這些功能，您的伺服器上必須安裝 V5R1 或 V5R2，或是套用了廣用連線修訂程式的 V4R5。以下是配置 PM/400 所需遵循的步驟：

1. 啓動 PM/400
您必須啓動 PM/400 來使用其資料收集功能。
2. 設定您的「管理中心」網路。
定義哪一個伺服器是您的中央系統，以及哪些伺服器是您的端點系統。在傳送資料至 IBM 之前，您可以使用此網路階層來傳送端點系統中的資料至中央位置。
3. 使用 廣用連線 連接至 IBM 來傳輸資料。
這是「管理中心」用來傳輸 PM/400 資料至 IBM 的連線。在前版次中，您是使用透過 SNA 執行的電子客戶支援中心 (ECS) 連線。當使用廣用連線時，您可以透過 TCP/IP 來傳輸資料。
4. 收集 PM/400 效能資料。
使用「管理中心」庫存功能來收集資料。
5. 傳送您的資料至 IBM。
使用「管理中心」階層中的「最大支援」之下可用的「電子服務代理程式」來傳送資料至 IBM。「電子服務代理程式」使用廣用連線。

您也可以使用 SNA 通訊協定傳送資料。

一旦完成 PM/400 配置後，您就可以開始執行其它作業來管理 PM/400。

收集 PM/400 效能資料： 如果有執行下列作業，那麼您可以使用「管理中心」收集 PM/400 效能資料：

1. 啓動 PM/400
2. 配置廣用連線
3. 設定「管理中心」網路
4. 確認系統有安裝「電子服務代理程式」。

如果要在端點系統或系統群組收集 PM/400 效能資料，請遵循下列步驟：

1. 在「iSeries 領航員」，展開**管理中心**。
2. 展開**端點系統**或**系統群組**。
3. 在端點系統或系統群組按一下滑鼠右鍵，然後選取**庫存**。
4. 選取**收集**。
5. 選取要收集的一個或多個庫存。在這個狀況下，您要選取 **PM/400 效能資料**。
6. 如果要在收集完成時在中央系統執行某動作，請從清單選取此動作。
7. 按一下**確定**立即開始收集資料，或按一下**排程**指定要收集資料的時間。

一旦配置伺服器之後，您可以執行其他作業來管理 PM/400。

使用 SNA 通訊協定傳送資料： 如果您選擇不要使用透過最大支援的電子服務代理程式傳送資料，仍然可使用字元型介面來傳輸資料。PM/400 會詢問您有關伺服器配置及使用的一連串問題。「配置 PM/400」顯示畫面會詢問您有關伺服器如何傳送及接收 PM/400 效能資料的問題。此處理程序的第一部份涉及網路設定。第二部份會詢問您如何傳輸資料。如果您是使用字元型介面，您可以使用直撥線路來傳輸資料。

若要使用 SNA 來傳送資料，請遵循下列步驟：

1. 啓動 PM/400
您必須啓動 PM/400 來使用其資料收集功能。
2. 選取您要使用的網路配置。
決定您要用來傳輸資料的網路配置。選擇您要使用直撥線路、現有的網際網路服務供應商 (ISP) 或虛擬專用網路 (VPN) 來連接 IBM。如果您要使用 ISP 或 VPN，則必須配置廣用連線。

如果您決定使用直撥線路來向 IBM 報告資料，您有若干選擇來決定如何配置網路。選取適合您網路的配置，然後從「配置 PM/400」顯示畫面執行該特定配置所指示的步驟：

- 當作一個單一伺服器，將資料直接傳給 IBM 。
 - 當作一個主電腦伺服器，這表示您要伺服器從其它伺服器 (遠端伺服器) 接收效能資料，然後將資料轉遞給 IBM。主電腦伺服器的版次不能比其它伺服器舊。換言之，主電腦伺服器的版次至少要與其它伺服器相同，或者更新。
 - 當作一個遠端伺服器，這表示您可以將效能資料傳送至主電腦伺服器。您在「配置 PM/400」顯示畫面上得知需要一個遠端伺服器，然後從 PM/400 功能表中使用選項 5 (使用遠端 iSeries 系統) 來定義遠端伺服器。
3. 使用遠端伺服器。
如果您選擇針對主電腦伺服器來設定網路，則必須識別那些要將資料傳送至您主電腦伺服器的伺服器。如果您是使用單一伺服器或遠端伺服器，則可忽略此步驟。
4. 自訂 PM/400。
在完成網路配置後，您需要對 PM/400 軟體作業建立廣域參數。如果您要使用直撥線路連接 IBM，則必須定義 PM/400 資料電話號碼。

一旦完成伺服器配置後，您就可以開始執行其它作業來管理 PM/400。

單一伺服器的 PM/400 網路：單一伺服器將它的資料直接傳給 IBM。唯有 PM/400 透過 SNA 收集資料和傳輸資料時，您才必須遵循下列步驟，為單一伺服器配置 PM/400。從伺服器上的「配置 PM/400 (CFGPM400)」畫面：

1. 從指令行鍵入 **CFGPM400**。
2. 對**傳送效能資料給 IBM** 欄位指定 *YES。
3. 對**接收效能資料**欄位指定 *NO。
4. 接受 QMPGDATA 的預設檔案庫。
5. 如果對「傳送效能資料給 IBM」指定 *YES，您會看到指出是否有適當通信物件的其他資訊。如果沒有這些通信物件，PM/400 會建立通信物件讓您進行傳輸。適當地回應其他畫面。
6. 在「使用聯絡資訊」畫面鍵入公司的聯絡資訊。

如果覺得單一伺服器設定情況不是您想要的設定，您可以選擇另一個 SNA 配置選項。

一旦配置伺服器之後，您可以執行其他作業來管理 PM/400。

主電腦伺服器的 PM/400 網路：主電腦伺服器接收來自其他伺服器的效能資料，然後轉遞資料給 IBM。唯有 PM/400 透過 SNA 收集資料和傳輸資料時，您才需要遵循下列步驟來為主電腦伺服器配置 PM/400：

1. 從主電腦伺服器上的「配置 PM/400」畫面
 - 從指令行鍵入 **CFGPM400**。
 - 對**傳送效能資料給 IBM** 欄位指定 *YES。
 - 對**接收效能資料**欄位指定 *YES。
 - 接受 QMPGDATA 的預設檔案庫。
2. 在主電腦伺服器從「使用遠端 iSeries 系統」畫面
 - 按 F6 (建立) 確認哪些伺服器會將它們的資料傳給主電腦伺服器。
 - 完成欄位並按 Enter 鍵。

註：唯有 PM/400 透過 SNA 收集資料和傳輸資料時才會發生下列狀況。如果有一個系統網路，建議您在「iSeries 領航員」中使用廣用連線和「管理中心」，收集和傳輸那些系統的資料。

PM/400 自動排定在從遠端伺服器收到資料之後隔天，從主伺服器傳輸資料給 IBM。如果自動排程不適合您的工作管理架構，您可以從主伺服器手動排程資料傳輸。

下面是您在排程資料傳輸時應記住的要訣。在整個星期內，對主伺服器平均排程資料傳輸。這個動作會將主伺服器的效能影響降到最低程度。例如，在含有 12 部伺服器的網路，您可以擁有三組四個系統。您可以排程每一個群組在星期一、星期三和星期五傳送它們的資料。這個動作平均分送要傳給主伺服器的資料數量。

如果您判斷主電腦伺服器設定情況不是您想要的設定，您可以選擇另一個 SNA 配置選項。

一旦配置伺服器之後，您可以執行其他作業來管理 PM/400。

遠端伺服器的 PM/400 網路：遠端伺服器將它的效能資料傳給主電腦伺服器。唯有 PM/400 透過 SNA 收集資料和傳輸資料時，您才必須遵循下列步驟為遠端伺服器配置 PM/400。從遠端伺服器上的「配置 PM/400」畫面 (CFGPM400)，執行下列步驟：

1. 從指令行鍵入 **CFGPM400**。
2. 對**傳送效能資料給 IBM** 欄位指定 *NO。
3. 對**接收效能資料欄位**指定 *NO。
4. 接受 QMPGDATA 的預設檔案庫。

註：如果有一個系統網路，建議您使用「iSeries 領航員」的庫存功能，收集資料，然後透過廣用連線傳輸那些系統的資料。

如果判斷遠端伺服器設定情況不是您想要的設定，您可以選擇另一個 SNA 配置選項。

一旦配置伺服器之後，您可以執行其他作業來管理 PM/400。

使用遠端伺服器：在部份站台，網路中的主電腦伺服器會將必要的效能資料傳給 IBM 來進行處理。使用主電腦伺服器網路時，您讓網路中的其他伺服器傳送它們的效能資料給這部主電腦伺服器以便傳輸到 IBM。如果要設定網路使用主電腦伺服器，您必須識別其他遠端伺服器並設定它們的資料傳輸的排程。「使用遠端 iSeries 系統」畫面可讓您定義這些其他伺服器。

註：

1. 如果您正在設定網路作為遠端伺服器或單一伺服器，那麼您不必使用這個畫面。唯有 PM/400 透過 SNA 收集資料和傳輸資料時才需要執行這項作業。
2. 如果有一個系統網路，建議您使用「iSeries 領航員」的庫存功能收集資料，然後透過 Universal Connection 傳輸那些系統的資料。

請遵循下列步驟定義遠端伺服器：

1. 從指令行鍵入 **GO PM400**。
2. 從 PM/400 功能表鍵入 5 (使用遠端 iSeries 系統) 並按 Enter 鍵。一開始沒有顯示遠端伺服器。您必須建立新的遠端位置。
3. 按 F6 (建立) 建立新遠端位置。
4. 記錄下列資訊的值。使用「顯示網路屬性 (DSPNETA)」指令從遠端系統顯示這些值。
 - 區域網路 ID
 - 預設本端位置

「使用遠端 iSeries 系統」畫面顯示遠端伺服器清單。本清單包含伺服器的狀態 (作用中或非作用中) 和每一部伺服器的說明。

5. 使用「PM/400 遠端管理」畫面或「變更遠端 iSeries」畫面建立或變更遠端伺服器的說明。在遠端伺服器之間遠端位置名稱必須是唯一的。

PM/400 自動排定從遠端伺服器收到資料之後的隔天從主伺服器將資料傳輸給 IBM。如果自動排程不適合您的工作管理架構，您可以從主伺服器手動排程資料傳輸。如果要手動排程資料傳輸，請參閱 PM/400 排程器。

PM/400 軟體假設您已在接收資料的伺服器 (主電腦伺服器) 與傳送資料的伺服器 (遠端伺服器) 之間定義「進階點對點網路 (APPN)」鏈結。如果系統將 QCRTAUT (建立預設公用權限) 這個系統值設成 *EXCLUDE 或 *USE，請參閱建立遠端伺服器的裝置說明，取得關於如何定義控制器說明的資訊。如果網路不符合這些假設，請參閱非 APPN 網路注意事項，取得關於建立裝置對組以支援每一部遠端伺服器連線的資訊。

一旦定義遠端伺服器之後，您可以自訂 PM/400 來使用特定線路連線。

在非 APPN 網路使用遠端伺服器： 主伺服器接收來自其他伺服器的 PM/400 資料，然後傳送資料給 IBM。遠端伺服器傳送 PM/400 資料給主伺服器。下列資訊假設先前已定義被參照的控制器。

唯有 PM/400 透過 SNA 收集資料和傳輸資料時，您才必須建立裝置對組來支援連線遠端伺服器。

1. 使用「建立裝置說明 (APPC) (CRTDEVAPPC)」指令。在遠端伺服器，鍵入 CRTDEVAPPC。按 F4 要求輸入參數，然後使用下列資訊定義一些值：

遠端系統

DEV(D(Q1PLOC)	指定裝置說明的名稱。
RMTLOCNAME(Q1PLOC)	指定遠端位置的名稱。
ONLINE(*YES)	指定當啟動或重新啟動系統時是否將裝置轉接為連線。
LCLLOCNAME(Q1PRMxxx)	指定本端位置名稱。 Q1PRMxxx 符合主伺服器的 RMTLOCNAME，其中 xxx 對於每一個遠端位置來說是唯一的。
CTL(yyyyyy)	指定連接的控制器名稱，其中 yyyyyy 是連接主伺服器的控制器。
MODE(Q1PMOD)	指定模式名稱。
APPN(*NO)	指定裝置是否具有 APPN 功能。

2. 在主伺服器指定下列資訊。在指令行，鍵入 CRTDEVAPPC。按 F4 要求輸入參數，然後使用下列資訊定義一些值：

主伺服器

DEV(D(Q1PRMxxx)	指定裝置說明的名稱。在這裡使用的名稱符合遠端系統的裝置說明名稱。
RMTLOCNAME(Q1PRMxxx)	指定遠端位置的名稱。在這裡使用的名稱符合遠端伺服器的 LCLLOCNAME 值，其中 xxx 對於每一個遠端位置來說是唯一的。
ONLINE(*YES)	指定當啟動或重新啟動系統時是否將裝置轉接為連線。

LCLLOCNAME(Q1PLOC)	指定本端位置名稱。這個值符合遠端伺服器的 RMTLOCNAME。
CTL(aaaaaa)	指定連接的控制器名稱，其中 aaaaaa 是連接遠端伺服器的控制器。
MODE(Q1PMOD)	指定模式名稱。
APPN(*NO)	指定裝置是否具有 APPN 功能。

3. 定義 APPC 裝置之後轉接裝置 (轉換配置 (VRYCFG) 指令)。在遠端伺服器，鍵入 VRYCFG。按 F4 要求輸入參數。

轉接遠端系統

CFGOBJ(Q1PLOC)	指定配置物件。
CFGTYPE(*DEV)	指定配置物件類型。
STATUS(*ON)	指定狀態

4. 在 PM/400 功能表鍵入選項 5 來新增 Q1PRMxxx 作為遠端伺服器。關於如何新增遠端伺服器的指示，請參閱使用遠端伺服器。

現在您已完成配置 PM/400，請參閱管理 PM/400 瞭解可以使用 PM/400 執行的其他作業。

建立 PM/400 的裝置說明: 將「建立預設公用權限 (QCRTAUT)」系統值設成 *EXCLUDE 或 *USE 的遠端伺服器需要下列步驟。如果 QUSER 沒有裝置說明 Q1PLOC 的 *CHANGE 權限，那麼遠端傳輸會失敗。這些步驟確保不會自動建立或刪除裝置。

註: 唯有 PM/400 透過 SNA 收集資料和傳輸資料時才需要這項作業。

如果您允許自動建立裝置，就會根據為 QCRTAUT 設定的值使用 PUBLIC *EXCLUDE 或 *USE 權限建立裝置說明。控制器控制是否可以自動建立或刪除裝置。請參閱下列指令決定在系統要如何定義這些參數：

- 建立控制器說明 (APPC) (CRTCTLAPPC) 指令
- 變更控制器說明 (APPC) (CHGCTLAPPC) 指令
- 顯示控制器說明 (DSPCTLD) 指令

關於沒有配置為使用 APPN 的系統，請參閱在非 APPN 環境下使用遠端伺服器取得關於如何建立裝置說明的資訊。

下列資訊假設先前在遠端伺服器已定義會用來與主電腦伺服器通信的控制器。

在遠端伺服器，重建裝置說明 Q1PLOC：

```
VRYCFG  CFGOBJ(Q1PLOC)
         CFGTYPE(*DEV)
         STATUS(*OFF)
```

```
DLTDEVD  DEVD(Q1PLOC)
```

```

CRTDEVAPPC DEVD(Q1PLOC)
           RMTLOCNAME(Q1PLOC)
           ONLINE(*NO)
           LCLLOCNAME(name of remote system)
           RMTNETID(remote netid of primary (or central) system)
           CTL(name of controller that the device will be attached to)
           AUT(*EXCLUDE)

CRTOBJAUT  OBJ(Q1PLOC)
           OBJTYPE(*DEV)
           USER(QUSER)
           AUT(*CHANGE)

VRYCFG     CFGOBJ(Q1PLOC)
           CFGTYPE(*DEV)
           STATUS(*ON)

```

現在您已完成配置 PM/400，請參閱管理 PM/400 瞭解可以使用 PM/400 執行的其他作業。

自訂 PM/400: 「使用 PM/400 自訂」畫面讓您能夠執行下列作業：

為 PM/400 軟體作業建立廣域參數

廣域參數可讓您自訂下列項目。請參閱線上說明取得這些欄位的說明：

- 優先順序限制
- 趨勢和移位排程
- 效能資料檔案庫
- 除去規格

定義 PM/400 資料電話號碼

在美國與加拿大境外，您必須將接收資料的 IBM 位置的電話號碼提供給 PM/400。對於大部份位置來說，當您起始配置 PM/400 處理時，PM/400 會嘗試選取您的位置的正確電話資料號碼。

使用 PM/400 轉斷及轉接線路

「PM/400 線路控制」畫面可讓 PM/400 轉斷線路、傳輸 PM/400 資料，然後使線路回到連接擱置狀態。

如果要自訂廣域參數，請執行下列步驟：

1. 從指令行鍵入 **GO PM400**。
2. 從 PM/400 功能表鍵入 3 來顯示「使用 PM/400 自訂」畫面並按 Enter 鍵。

如果使用「資料收集服務」收集 PM/400 資料，您應該考慮某些 PM/400 的資料收集注意事項。

關於您可以使用 PM/400 執行的其他作業，請參閱管理 PM/400。

驗證 PM/400 資料號碼: 如果伺服器使用直接撥號連線來連接 IBM，您必須確認 PM/400 電話號碼是正確的。電話號碼也必須含有您的線路的正確字首。

註：這僅針對 SNA 傳輸。

如果要檢查電子客戶支援中心線路的電話號碼格式，請執行下列步驟：

1. 鍵入


```
DSPDTAARA DTAARA(QUSRSYS/QESTELE)
```

並按 Enter 鍵。

2. 判斷在位移 0 找到的連線號碼字首。例如，如果位移 0 是 'T9:1800xxxxxxx'，那麼字首是 T9:。
3. 鍵入

```
DSPDTAARA DTAARA(QUSRSYS/Q1PGTELE)
```

並按 Enter 鍵。

4. 位移 0 (零) 是會使用的撥號字串。(不會使用其他號碼。)
5. 如果您使用 ECS 線路訂購 PTF，您可以比較位移 0 (零) 中的格式與用於 ECS 線路的格式 (CALL QESPHONE)，記下使用的字串，然後比較它與在步驟 2 發現的值。
電話號碼會不同但字首應該會相同 (亦即，SST9:1800...、SST:1800... 等等)。

如果需要變更電話號碼，請使用「變更資料區 (CHGDTAARA)」指令：

鍵入 **CHGDTAARA**，其中 DTAARA 是 Q1PGTELE，LIB 是 QUSRSYS，子字串開始位置是 *ALL，新值是 'SST:18005475497'

註：新值應該是撥號字首，後面跟著 18005475497 (適用於美國和加拿大)。

現在您已完成 PM/400 配置，請參閱管理 PM/400 瞭解您可以管理的作業。

設定 PM/400 的直接撥號線： 對於大部份位置來說，PM/400 會嘗試選取您的位置的正确資料電話號碼。您一定要確認 PM/400 資料電話號碼是否正確。如果沒有包含 PM/400 資料電話號碼和 PM/400 支援號碼的資訊，請聯絡當地 IBM 支援人員。他們可以提供您適當電話號碼。

註：如果透過 Universal Connection 傳輸資料，就不需要這個電話號碼。唯有使用直接撥號線路才需要這個號碼。

如果要定義 PM/400 資料電話號碼或變更此號碼，請執行下列步驟：

1. 從指令行鍵入 **GO PM400**。
2. 從 PM/400 功能表鍵入 3 來顯示「使用 PM/400 自訂」畫面並按 Enter 鍵。
3. 在這個畫面，向前捲動直到出現顯示電話號碼欄位的畫面。
4. 在 **IBM PM/400 電話號碼欄位** 鍵入正確撥號順序。對於許多 IBM 數據機來說，您需要使用冒號 (:) 字元代表撥號音。

使用 PM/400 轉斷和轉接線路： 有時候 PM/400 使用的線路處於連接擱置狀態。這種狀態不允許 PM/400 存取線路來傳輸資料。「PM/400 線路控制」畫面可讓 PM/400 轉斷線路、傳輸資料，然後使線路回到連接擱置狀態。使用本畫面時，您可以變更 PM/400 傳輸作業 (Q1PCM1) 來檢查線路狀態和轉斷適當線路。一旦完成傳輸之後，相同線路會處於連接擱置狀態。

註：唯有 PM/400 透過 SNA 收集資料和傳輸資料時才需要這項作業。

如果要轉斷和轉接線路，請執行下列步驟：

1. 從指令行鍵入 **PMLINMON** 來啟動 PM/400 線路監督功能。您應該看到「PM/400 線路控制」畫面。
2. 閱讀顯示在第一個畫面的警告並按 Enter 鍵。
3. 定義 PM/400 需要轉斷的線路、控制器和裝置組合。
4. 使用您要 **PM/400 自動線路控制作用中嗎**？這個提示作為此功能的主要控制開關。如果指定 **YES**，那麼 PM/400 功能是作用中。如果指定 **NO**，就會停用此功能。

如果指定 **NO**，那麼在您指定 **YES** 時您不必重新定義線路控制清單。您可以僅指定線路來轉斷和轉接線路。您可以指定所有三個說明來轉斷和轉接線路、控制器和裝置。

5. 驗證您定義的線路、控制器和裝置。按 **Enter** 鍵查看您選擇的摘要。
6. 按 **Enter** 鍵確認您的選擇或按 **F12** 返回前一個畫面以變更您的輸入。

您也可以使用「配置 PM/400 (CFGPM400)」指令設定 PM/400 線路控制。

關於您可以使用 PM/400 執行的其他作業，請參閱管理 PM/400。

管理 PM/400

設定網路使用 PM/400 之後，您可以執行下列作業：

停用 PM/400

瞭解如何停止 PM/400。

變更聯絡資訊

瞭解如何從原始設定值變更聯絡資訊。

使用 PM/400 排程工作

瞭解如何使用 PM/400 排程工作。

從 PM/400 分析省略項目

瞭解在使用 PM/400 執行分析時如何省略工作、使用者和通信線路。

暫時關閉 PM/400

瞭解如何暫時停止 PM/400。

顯示 PM/400 狀態

瞭解如何使用「iSeries 領航員」或 PM/400 功能表顯示 PM/400 狀態。

檢視 PM/400 報告

請參閱 PM/400 報告範例以及如何解譯那些報告的說明。

檢視資料的圖形歷程

圖形歷程提供經過一段指定期間收集的效能資料的圖形畫面。瞭解如何檢視這項資料。

停用 PM/400: 如果要停止執行 PM/400，您可以使用下列其中一種方法：

使用「iSeries 領航員」

執行下列步驟：

1. 在「iSeries 領航員」，展開正在執行 PM/400 的系統。
2. 展開**配置和服務**。
3. 在**資料收集服務**按一下滑鼠右鍵。
4. 選取 **Performance Management/400**。
5. 選取**停止**。
6. 選取要在那裡停止 PM/400 的系統。
7. 按一下**確定**。

使用 API

使用結束 PM/400 (Q1PENDPM) API 停用 PM/400。

請參閱管理 PM/400 來瞭解您可以執行的其他作業。

變更 PM/400 聯絡資訊: 在配置 PM/400 軟體期間，您識別聯絡人並提供貴公司的郵寄資訊。如果後來您需要更新資訊，請使用「使用聯絡資訊」選項變更該資訊。如果要變更聯絡資訊，請執行下列步驟：

1. 從指令行鍵入 **GO PM400**。
2. 從 PM/400 功能表鍵入 1 並按 Enter 鍵。出現「使用聯絡資訊」畫面。
3. 適當地變更聯絡資訊，然後按 Enter 鍵。

關於您可以執行的其他作業，請參閱管理 PM/400。

使用 PM/400 排程工作: 排程器是 PM/400 軟體不可缺少的，排程器會自動啟動那些支援 PM/400 效能資料收集和分析需要的工作。

部份 PM/400 軟體啟動處理包含啟動一個稱為 Q1PSCH 的工作。如下表所示，這項工作依次啟動其他工作：

如果要存取 PM/400 排定的工作，請執行下列作業：

1. 從指令行鍵入 **GO PM400**。
2. 從 PM/400 功能表鍵入 2 並按 Enter 鍵。出現「使用自動排定的工作」畫面。
3. 您可以將每一項工作的狀態從作用中變成非作用中。在要變更的工作旁邊鍵入 2 (變更) 並按 Enter 鍵。顯示「變更自動排定的工作」畫面。

下表顯示可能的 PM/400 工作清單。

PM/400 排定的工作

工作	排程	功能
Q1PTEST	啟動時	驗證有啟動 PM/400，然後進入非作用中狀態。
Q1PCM1	每週	將減少的效能資料傳給 IBM。唯有使用直接撥號線路時這項工作才會作用中。
Q1PCM2	每日	將通信轉為離線。
Q1PPMSUB	每小時	確定「資料收集服務」正在收集資料。
Q1PDR	每日	執行資料縮減並清除效能資料。
Q1PPG	每月	清除減少的效能資料。
Q1PCM3	依需要	在直接撥號傳輸無法轉斷線路之後將通信轉為離線。
Q1PCM4	依需要	從遠端伺服器存取 PM/400 資料。唯有從 PM/400 功能表使用選項 5 新增遠端系統之後才會啟動這項工作。
Q1PPMCHK	每 4 小時	確認資料收集是作用中。

Q1PMONTH

每月

如果該月還需要其他傳輸，則允許每月傳輸。預設值是設成非作用中。唯有使用直接撥號線路時才能使用這項工作。

請參閱管理 PM/400 來瞭解您可以執行的其他作業。

從 PM/400 分析省略項目： PM/400 軟體應用程式彙總包含了批次工作、使用者和通信線路的前十個項目的分析。不過，部份工作、使用者或通信線路不適合這類分析。例如，您可能想要從執行時間種類排除超過正常執行時間的工作 (如自動啟動工作)。

您可以使用同屬省略功能，從最前面十項分析省略批次工作群組和使用者群組。例如，如果要省略開頭是 MYAPP 的工作，請指定：MYAPP*

如果要處理省略，請執行下列步驟：

1. 從指令行鍵入 **GO PM400**。
2. 從 PM/400 功能表鍵入 4 並按 Enter 鍵。出現「處理前面十項省略」畫面。
3. 根據要省略哪些項目鍵入適當選項號碼
 - 鍵入 1 來處理工作。
 - 鍵入 2 來處理使用者。
 - 鍵入 3 來處理通信線路。
4. 在適當欄位鍵入 1，從某特定種類省略使用者或工作。在通信線路狀況下，鍵入線路名稱，然後在適當欄位鍵入 1。

請參閱管理 PM/400 來瞭解您可以執行的其他作業。

暫時關閉 PM/400： 如果需要停止 PM/400 驗證「資料收集服務」是否正在收集資料，您可以使用排程器工作將日期變更成 Q1PPMSUB 工作的將來日期。

1. 從指令行鍵入 **GO PM400**。
2. 鍵入 2 (處理自動排定的工作)。
3. 在 Q1PPMSUB 工作旁邊鍵入 2 (變更)。
4. 將日期與時間變更成將來日期與時間。
5. 按 Enter 鍵。這個變更會暫時停止 PM/400 驗證「資料收集服務」是否正在收集資料。您必須結束目前所收集的資料。

註：在到達您設定 Q1PPMSUB 工作的日期和時間之前，PM/400 不會啟動、循環或變更「資料收集服務」。

請參閱使用 PM/400 排程工作瞭解您可以使用排程器來執行的其他事情。

請參閱管理 PM/400 來瞭解您可以執行的其他作業。

顯示 PM/400 狀態： 您可以在伺服器使用「iSeries 領航員」或 PM/400 功能表顯示 PM/400 的狀態。在一部或多部伺服器或群組使用「Performance Management/400 狀態」對話框檢視 PM/400 的整體狀態。例如，您會看到像 PM/400 是否在作用中之類的明細。使用 PM/400 功能表檢視「資料收集服務」狀態、PM/400 排程器狀態、效能資料版次、上次傳輸嘗試、效能資料成員以及效能資料大小。

如果要從「iSeries 領航員」檢視 PM/400 的整體狀態，請執行下列步驟：


1. 在「iSeries 領航員」，展開端點系統或系統群組。
2. 展開**配置和服務**。
3. 在**資料收集服務**按一下滑鼠右鍵。
4. 選取 **Performance Management/400**。
5. 選取**狀態**。

如果要從 PM/400 功能表檢視 PM/400 的詳細狀態，請執行下列步驟：

1. 從指令行鍵入 **GO PM400**。
2. 從指令行鍵入 6 並按 Enter 鍵。查看線上說明取得每一個欄位的說明。

請參閱管理 PM/400 來瞭解您可以執行的其他作業。

檢視 PM/400 報告： PM/400e 提供的輸出是以每月或每季為基礎的一組管理報告和圖形。PM/400e 提供有報告的兩個選項。

報告和圖形的用途是讓管理者清楚瞭解他們的伺服器的現行效能和正確成長趨勢。如果要詳細檢視報告和圖形並瞭解它們的一些好處及用法，請參閱 PM/400 網站 。

請參閱管理 PM/400 來瞭解您可以執行的其他作業。

效能工具

iSeries 授權程式的「效能工具」可讓您使用不同方法分析效能資料。「效能工具」是檢視、報告和圖形化效能資料的工具與指令集。您可以使用 Performance Tools for iSeries 檢視使用資料收集服務收集的效能資料，或檢視使用「啟動效能追蹤 (STRPFTRC)」指令收集的追蹤資料。然後您可以將資料彙總到報告中來研究系統效能問題。您也可以建立效能資料圖來查看某段時間的資源使用率。

Performance Tools for iSeries 包含基本產品和兩個特性 (管理程式和代理程式)。需要此基本產品加上其中一個特性。關於「效能工具」的「管理程式」和「代理程式」特性的詳細資訊，請參閱管理程式和代理程式特性比較這個主題。

效能工具概念

說明各種可協助您收集和**分析效能資訊**的工具。尋找哪些工具執行哪些功能以及它們如何運作的詳細資訊。

安裝和配置效能工具

請參閱本主題取得安裝和設定指示。

效能工具報告

「效能工具」報告提供的資訊是關於經過一段時間所收集的資料。使用這些報告取得關於系統資源的效能和使用量的其餘資訊。

關於如何使用「效能工具」收集有關系統、工作或程式的效能之資料的詳細資訊，請參閱 Performance Tools



。它也說明如何分析和列印資料以協助確認及更正問題。關於「效能工具」如何顯示這項作業使用的 CPU 時間，請參閱 HVLPTASK 作業。

效能工具概念

iSeries 授權程式的「效能工具」分析兩種不同效能資料：簡單資料和追蹤資料。資料收集服務收集範例資料，它是按固定時間間隔擷取的彙總資料。您收集範例資料進行趨勢分析和效能分析。資料與像儲存體儲存區和回

應時間相關。不過，「資料收集服務」不支援收集追蹤資料。追蹤資料是您收集的詳細資料，可從中獲得關於特定工作和異動的其他資訊。如果要收集追蹤資料，您可以使用「啟動效能追蹤 (STRPFTRC)」指令或效能探究程式。

併入效能工具的功能

「效能工具」包含收集、分析和報告效能資料的不同應用程式。瞭解有哪些可以使用的功能，以及哪些功能最適合某個作業會是一項複雜的工作。請參閱本主題取得併入這個授權程式的功能說明。




管理程式和代理程式特性比較

您可以使用「管理程式」和「代理程式」特性在分散式環境中有效地分配「效能工具」的必要功能。本主題包含這兩個特性的說明、它們每一個包含的功能以及關於如何最有效使用它們的資訊。

顯示效能資訊

您可以在「iSeries 領航員」檢視系統資源使用率資料。您可以檢視資料、以圖形呈現資料以及將資料彙總到報告。在這裡尋找關於如何存取這項功能的資訊。

效能工具提供的功能： 「效能工具」包含報告、交談式指令和其他功能。例如，「效能工具」包含下列工具：

工具	說明
處理系統活動 (WRKSYSACT) 指令	「處理系統活動 (WRKSYSACT)」指令可讓您交談式處理目前在系統執行的工作、緒和作業。WRKSYSACT 指令報告系統資源使用率，包含那些使用共用處理儲存區的分割區並以每個作業為基礎來計算的 CPU 使用率。關於這個指令如何顯示這項作業使用的 CPU 時間，請參閱 HVLPTASK 作業。
顯示效能資料	「顯示效能資料」圖形介面可讓您從「iSeries 領航員」檢視效能資料，將資料彙總到報告，建立圖形以顯示趨勢，以及分析系統效能的明細。
報告	報告使用邏輯和有用的格式組織「資料收集服務」效能資料。Performance Tools  詳細討論本工具。
圖形功能	「效能工具」圖形功能可讓您使用圖形格式處理效能資料。您可以以交談方式顯示圖形，或者您可以將資料列印、繪製或儲存到圖形資料格式 (GDF) 檔供其他公用程式使用。Performance Tools  詳細討論本工具。
效能探究程式	效能探究程式是一個資料收集工具，它可以協助您確認「資料收集服務」收集的取樣資料無法確認的效能問題的原因，或協助您確認那些在執行一般趨勢分析之後無法確認的效能問題的原因。在程式、程序、模組或方法層使用效能探究程式進行詳細的應用程式分析。例如，您可以在個別程式或程序 CPU 以及 I/O 統計值或個別物件 I/O 性質收集追蹤資料。Performance Tools  詳細討論本工具。

管理程式和代理程式特性比較： 透過兩個不同的可安裝的特性可以使用「效能工具」。本主題說明這兩個特性的差異協助您決定哪一個特性較適合您的應用程式。

管理程式特性

「效能工具管理程式」特性是完整功能套裝軟體，適合用於分散式環境中的中央站系統或用於單一系統。如果需要分析追蹤資料、以圖形方式檢視資料、即時檢視系統活動或管理及追蹤系統成長，那麼「效能工具」授權程式的「管理程式」特性就很有用。

代理程式特性

「效能工具代理程式」特性 (含有「管理程式」功能子集) 是具有更多基本功能的較便宜的套裝軟體。在分散式環境中，「代理程式」特性很適合網路中的受管理的系統，因為如果需要詳細分析時資料可以傳給「管理程式」。對於那些需要合理自足層次但沒有專家級技術的站台，它也是一項有效工具。

「效能工具」的「代理程式」特性提供一些功能來簡化效能資料的收集、管理、線上顯示、資料縮減以及分析。效能探究程式報告那些併入 iSeries 授權程式的「效能工具」中的基本選項的功能和它的相關指令，因此可以與「管理程式」特性或「代理程式」特性搭配使用。沒有併入「代理程式」特性的主要「效能工具」功能是效能和追蹤報告、效能公用程式（工作追蹤和選取檔案以及存取群組）、系統活動監督和效能圖。

顯示效能資訊：「效能工具」可以從「iSeries 領航員」顯示效能資料。從這個圖形介面，您可以檢視效能資料，將資料彙總到報告，建立圖形以顯示趨勢，以及分析系統效能的明細。

計量單位

「iSeries 領航員」顯示經過一段選定時間間隔的效能計量單位。您可以在「顯示效能資料」視窗的「圖形」窗格檢視的效能計量單位包含：

- 異動計數
- 異動回應時間
- CPU 使用率總計
- 交談式 CPU 使用率
- 批次 CPU 使用率
- 高磁碟使用率
- 機器儲存區尋頁錯失/秒
- 使用者儲存區尋頁錯失/秒
- 異常

「明細」窗格可讓您使用不同方法檢視選取的時間間隔的詳細效能資料。如果要分析系統效能，您可以檢視工作資料、子系統資料、儲存區資料或硬碟機資料。

報告

除了檢視圖形和明細資料，您也可以從「顯示效能資料」視窗列印報告。效能報告可讓您研究那些引起效能問題的系統區域。您可以執行不同報告以查看使用系統資源的地方。唯有在中央系統安裝 Performance Tools for iSeries (5722-PT1) 的選項 1 (管理程式特性) 之後，才能在「效能工具」中列印報告。關於「管理程式」特性的詳細資訊，請參閱管理程式和代理程式特性比較主題。

您可以從「顯示效能資料」視窗列印的報告包含：

- 系統
- 元件
- 工作
- 儲存區
- 資源

透過「iSeries 領航員」存取

您可以在「iSeries 領航員」中執行下列步驟存取「顯示效能資料」視窗：

1. 在「iSeries 領航員」，展開**我的連線** (或作用中環境)。
2. 展開含有要檢視的效能資料的伺服器。
3. 展開**配置和服務**。
4. 在**資料收集服務**按一下滑鼠右鍵，選取**效能工具**，然後選取**效能資料**。
5. 選取要顯示的效能資料檔。
6. 按一下**顯示**。

關於如何在「iSeries 領航員」使用「顯示效能資料」視窗的詳細資訊，請參閱「iSeries 領航員」線上說明。

安裝和配置效能工具

如果要安裝「效能工具」，您需要具有儲存系統 (*SAVSYS) 權限的使用者設定檔。您可以使用系統操作員設定檔取得這個權限。

必須在 QPFR 這個檔案庫中執行「效能工具」。如果系統有以這個名稱命名的檔案庫，請使用「更名物件 (RNMOBJ)」指令更名它之後才安裝「效能工具」。這個步驟會確保「效能工具」能正常運作。

使用下列指令將「效能工具」放到 QPFR 這個程式庫：

```
RSTLICPGM LICPGM(5722PT1) DEV(NAME) OPTION(*BASE)
```

然後您必須執行下列其中一項：

- 如果有購買「管理程式」特性，請使用下列指令：
RSTLICPGM LICPGM(5722PT1) DEV(*tape-device-name*) OPTION(1)
- 如果有購買「代理程式」特性，請使用下列指令：
RSTLICPGM LICPGM(5722PT1) DEV(NAME) OPTION(2)

如果要安裝數片 CD-ROM，那麼可能會發生下列狀況。在安裝第一片 CD-ROM 之後，您可能會收到一則表示已復置授權程式但沒有復置語言物件的訊息。如果發生這種狀況，請插入下一片 CD-ROM 並輸入下列指令：

```
RSTLICPGM LICPGM(5722PT1) DEV(NAME) RSTOBJ(*LNG) OPTION(*BASE)
```

安裝「效能工具」程式的另一種方法是鍵入 GO LICPGM 並使用功能表選項。

「效能工具」是處理器型程式。用法類型是並行，而且使用 *NOMAX 使用期限安裝程式。

Performance Tools  詳細討論本程式。

效能工具報告

「效能工具」提供一種簡易的方法，可讓您查看收集的資料。資料收集服務可使用「異動」、「鎖定」及「追蹤」報告的異常，提供大部份「效能工具」報告中的資料。您必須使用 STRPFRTRC 及 ENDPFRTRC 指令，以收集這三種報告的追蹤資訊。使用「效能工具」報告，您可以隔離出效能問題。在收集到一段時間內的效能資料後，可以列印報告以察看使用系統資源的方法與位置。報告可以引導您瞭解特定應用程式、使用者或造成整體回應時間較慢的無效率工作量。

下面清單說明每一個報告，並提供為何要使用特定報告的簡短概觀。如需每一個報告的詳細說明，請參閱

Performance Tools 。

註：您可以列印在之前的版次中，以「啟動效能監視器 (STRPFRMON)」指令收集到的資料報告。

「效能工具」報告概觀

報告	說明	顯示項目	如何使用資訊
----	----	------	--------

系統	使用「資料收集服務」資料，可以提供系統如何作業的概觀。報告含有工作量、資源使用、儲存體儲存區使用率、磁碟使用率及通信的彙總資訊。執行及列印此報告，通常可讓您瞭解系統的使用。	系統工作量。報告包含資料庫能力資料。	工作量投影
元件	使用「資料收集服務」資料以將相同的系統效能元件相關資訊提供為「系統報告」，但明細層次更高。此報告可協助您瞭解哪些工作正在使用大量的系統資源，如 CPU、磁碟等等。	資源使用、通信、系統及使用者工作。此報告包含資料庫能力資料與「交談式功能」使用率。	硬體成長與配置處理趨勢
異動	使用追蹤資料以提供有關效能資料收集期間所發生的異動詳細資訊。	CPU 的工作量及使用率、磁碟、主要儲存體、異動工作兩、物件競爭	工作量投影、儲存區配置、應用程式設計、檔案競爭及程式使用
鎖定	使用追蹤資料以提供系統作業期間的鎖定與攫取衝突相關資訊。使用此資訊，您可以判定在處理程序期間，工作是否因不符合鎖定要求或內部機器攫取衝突而延遲。這些狀況也稱為呼叫等待。如果發生這些狀況，您可以判定工作正在等待哪些物件，以及等待的長度。	按時間的檔案、記錄或物件競爭；保留工作或物件名稱；要求的工作或物件名稱	問題分析。減少或排除物件競爭。
追蹤	使用追蹤資料以顯示隨著時間所追蹤的各種工作類型 (例如，批次作業) 的進度。報告使用的資源、異常與狀態轉移。	工作類別時間片段與追蹤資料	問題分析與批次作業進度
工作	使用「資料收集服務」資料以顯示所有或選取的時間與工作相關資訊，包括交談式作業與非交談式作業的明細與彙總資訊。因為報告可能很長，您可能想要選取要併入的間隔與工作以限制輸出。	按間隔的工作	工作資料

儲存區	使用「資料收集服務」資料以提供子系統活動相關區段，以及儲存區活動相關區段。會顯示每一個取樣間隔的資料。因為報告可能很長，您可能想要選取要併入的間隔與工作以限制輸出。	按間隔的儲存區	儲存區資料
資源間隔	使用「資料收集服務」資料以提供所有或選取的間隔之相關資源資訊。因為報告可能很長，您可能想要選取要併入的間隔以限制輸出。	按間隔的資源	系統資源使用

效能探究程式及「資料收集服務」是個別的收集代理程式。它們會各自產生自己資料庫檔案集，其中含有收集的資料集群組。您可以同時執行這兩種資料收集功能。

如需其他工作的報告清單，請參閱下列：

- 效能探究程式報告
- Performance Management/400 報告

效能探究程式

效能探究程式是一個資料收集工具，它協助使用者確認那些無法透過使用「資料收集服務」收集資料來確認的效能問題的原因，或協助您確認那些在執行一般趨勢分析之後無法確認的效能問題的原因。使用效能探究程式的兩個理由如下：

- 針對系統資源、應用程式、程式、程序或引起此問題的方法來找出效能問題。
- 分析應用程式效能

效能探究程式的收集函數和相關指令是 OS/400 授權程式的一部份。報告功能和它的相關指令是 Performance Tools for iSeries 授權程式的基本選項的一部份，因此可以與「管理程式」特性或「代理程式」特性搭配使用。

AS/400 Performance Explorer Tips and Techniques  提供效能探究程式功能的其他範例和增強的效能探究程式追蹤支援範例。

效能探究程式是一項工具，它協助您找出那些透過使用執行一般效能監督的工具無法識別的效能問題的原因。當電腦環境變得愈大愈複雜時，您的效能分析也應該會變得複雜。效能探究程式收集關於複雜效能問題的資料來解決日益複雜的情況。

註：效能探究程式是您在嘗試過其他工具之後需要使用的工具。它收集一些特定形式的資料，這些資料可以更輕易找出與效能問題相關的因素；不過當您收集這個資料時，會明顯影響系統效能。

本工具適合有興趣瞭解或增進程式效能的應用程式開發者。對於瞭解效能管理的使用者來說，本工具在協助識別和區分複雜效能問題方面也很有用。

如果要進一步瞭解效能探究程式，請參閱下列效能探究程式主題。

效能探究程式概念

效能探究程式透過收集關於指定的系統處理或資源的詳細資訊來運作。本主題說明效能探究程式如何運作以及如何充分使用它。

配置效能探究程式

如果要收集詳細追蹤資訊，您必須修改效能探究程式，以便最佳化處理從那裡採取追蹤的應用程式處理。


效能探究程式報告

透過效能探究程式階段作業收集效能資料之後，您可以執行內含的報告或直接查詢資料庫檔案來檢視該資料。

關於其他詳細資訊，請參閱 Performance Tools 。

效能探究程式概念

就像資料收集服務，效能探究程式會收集資料以供日後分析。不過，它們收集非常不同類型的資料。「資料收集服務」按定期排程間隔收集各種系統資料 (消耗最少系統資源)。相對地，效能探究程式啟動一個收集追蹤層次資料的階段作業。這個追蹤產生大量關於應用程式、工作或緒使用的資源的詳細資訊。尤其是您可以使用「效能探究程式」回答關於像系統產生的磁碟 I/O、程序呼叫、Java 方法呼叫、尋頁錯失，以及其他追蹤事件

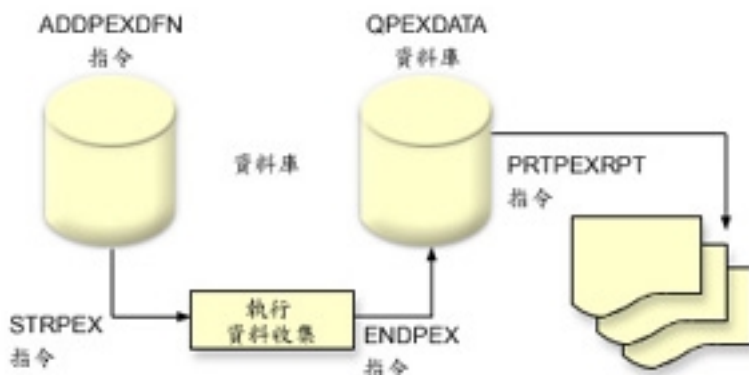
 等方面的特定問題。能夠收集極特定和非常詳細資訊的能力，使效能探究程式有效地協助找出效能問題。例如，「資料收集服務」可以告訴您磁碟儲存體空間正迅速耗用。您可以使用效能探究程式確認什麼程式和物件使用太多磁碟空間以及原因。

註：您可以同時收集效能探究程式資料及「資料收集服務」資料。

效能探究程式如何運作

下圖應該可協助您熟悉通往效能探究程式的一般路徑。關於這些步驟的明細，請參閱配置效能探究程式。此圖顯示由下列步驟組成的基本工作週期：

1. 定義效能探究程式的資料收集。您也可以新增過濾器透過指定特定事件的比較值來限制收集的資料量。
2. 啟動效能探究程式來根據定義收集資料。
3. 執行程式、指令或工作量。
4. 結束資料收集，這個動作會將收集的資料儲存到一組資料庫檔案。
5. 從資料庫檔案建立和列印報告。



如果要進一步瞭解效能探究程式，請參閱下列效能探究程式主題。

效能探究程式定義

決定資料效能探究程式收集的資料及它如何收集資料的參數和條件，是使用效能探究程式定義來配置和儲存的。本主題說明如何使用這些定義並提供說明簡單定義的範例。

效能探究程式資料庫檔案

效能探究程式收集的資料儲存在效能探究程式資料庫檔案。

效能探究程式優點

效能探究程式包含可協助您收集和分析詳細的效能資訊的各種功能。本主題提供那些功能的概觀。

效能探究程式定義： 如果要收集效能探究程式資料，您必須告訴效能探究程式要收集什麼資料。要達成此目的，請使用新增效能探究程式定義 (ADDPEXDFN) 指令來建立效能探究程式定義。完成及儲存定義之後，您可以繼續工作週期中的下一項作業。

建立新定義之前，請考慮想要什麼種類資訊以及需要多少明細資料數量。效能探究程式提供下列類型的資料收集：

統計值類型定義

識別消耗太多 CPU 用量或執行大量磁碟 I/O 作業的應用程式和 IBM 程式或模組。通常您使用此統計類型識別那些應該進一步調查可能導致潛在效能瓶頸的程式。

- 適合 OS/400 程式、程序和 MI 複合指令的第一次序分析。
 - 提供一些呼叫
 - 以微秒為單位提供內含和累計的 CPU 用量
 - 提供同步和非同步 I/O 的內含及累計數字
 - 提供一些發出的呼叫
- 適合短期或長期執行
- 對於所有執行來說收集的資料大小很小而且固定
- ILE 程序的執行時間資料收集額外執行時間可能是由於呼叫頻率所導致的問題。雖然執行時間有下降，但是收集的統計值仍然正確，因為「效能探究程式」從資料移除大部份資料收集額外執行時間。
- 使用合併或分隔的資料區。ADDPEXDFN 指令上的 MRGJOB 參數指定是在一個資料區累計所有程式統計值或個別保留每一個程式統計值 (例如，一項工作一個資料區)。

可以採取階層式或平面方式建構統計值。

- 階層式結構將統計值組成一個呼叫樹狀表，該樹狀表中的每一個節點代表工作或作業執行的程式程序。
- 平面式結構將統計值組成一個程式或程序的簡單清單，每一個程式或程序有自己的統計值集。

下面是名稱是 MYSTATS 的效能探究程式統計值定義範例，該範例會在每一個程式或程序層次顯示 CPU 和磁碟資源使用量。

```
ADDPEXDFN DFN(MYSTATS) /* 定義的名稱。*/
TYPE(*STATS) /* 定義的類型 */
JOB(*ALL) /*所有工作 */
TASKS(*ALL) /*所有作業 */
MRGJOB(*YES) /* 合併記錄以減少資料收集額外執行時間 */
DTAORG(*FLAT) /* 不持續追蹤誰呼叫誰 */
```

設定檔類型定義

根據原始程式陳述式數目識別消耗過多 CPU 使用率的高階語言 (HLL) 程式。您也可以識別在程式開頭與程式尾端的次常式之間不斷分支的程式。如果程式很大，那麼這個連續來回跳換會在主記憶體有限的系統上造成過多尋頁錯失率。

- 程式設定檔 (在 ADDPEXDFN 指令指定 TYPE(*PROFILE) 和 PRFTYPE(*PGM))
 - 詳細分析您把時間花在特定工作內的一組程式內的什麼地方。
 - 可以按程式、模組、程序、陳述式或指令彙總資料。
 - 無論執行長度是多少，資料收集大小都很小而且固定。
 - 16 MI 程式限制表示您應該使用這個上限作為第二次序分析工具。
 - 可以變更取樣間隔使額外執行時間不同。2 毫秒的間隔似乎是基準性能測試的第一優先選擇。
 - 由於指定的程式數目或指定的程式大小而沒有限制窗格大小。

下面是名稱是 PGMPROF 的效能探究程式設定檔範例，該範例會顯示特定程序的用法。

```
ADDPEXDFN DFN(PGMPROF) /* 定義的名稱。*/
TYPE(*PROFILE) /* 定義的類型 */
JOB(*ALL) /*所有工作 */
PGM((MYLIB/MYPGM MYMODULE MYPROCEDURE)) /* 要監督的程式名稱。*/
INTERVAL(1) /* 會採用一個 1 毫秒的取樣。*/
```

- 工作設定檔 (在 ADDPEXDFN 指令指定下列參數：TYPE(*PROFILE) 和 PRFTYPE(*JOB))
 - 詳細分析您把時間花在資料收集的一組工作或作業的什麼地方。
 - 資料收集的大小很小但不固定。大小隨著執行長度增加而增加。
 - 可以剪影系統上的所有工作和作業，或者將收集的資料範圍縮小到只有感興趣的少數工作或作業。
 - 可以變更取樣間隔使額外執行時間不同。2 毫秒的間隔似乎是基準性能測試的第一優先選擇。

下面是名稱是 ALLJOBPROF 的效能探究程式工作設定檔範例，該範例會顯示所有工作的用法。

```
ADDPEXDFN DFN(ALLJOBPROF) /* 定義的名稱。*/
TYPE(*PROFILE) /* 定義的類型 */
PRFTYPE(*JOB) /* 會監督工作設定檔類型。*/
JOB(*ALL) /*所有工作 */
TASKS(*ALL) /*所有作業 */
INTERVAL(1) /* 會採用一個 1 毫秒的範例。*/
```

追蹤定義

收集由系統上的一項或多項工作產生的效能活動的歷程追蹤。追蹤類型收集關於發生事件的時間和次序的特定資訊。追蹤類型收集關於程式、授權內碼 (LIC) 作業、OS/400 工作和物件參照資訊的詳細參考資訊。

- 部份一般追蹤事件：
 - 程式和程序呼叫及傳回
 - 儲存體，例如，配置和取消配置。
 - 磁碟 I/O，例如讀取作業和寫入作業。
 - Java 方法，例如登錄和跳出。
 - Java，例如物件建立和垃圾收集。
 - 異動日誌，例如開始確定和結束確定。
 - 同步化，例如互斥鎖定和解除鎖定或號誌等待。
 - 通信，例如 TCP、IP 或 UDP。
- 執行愈久收集愈多資料。

下面是名稱是 DISKTRACE 的效能探究程式追蹤定義範例，該範例會顯示所有磁碟事件的用法。

```
ADDPEXDFN DFN(DISKTRACE) /* 定義的名稱。*/
TYPE(*TRACE) /* 定義的類型 */
JOB(*ALL) /*所有工作 */
TASKS(*ALL) /*所有作業 */
```

TRCTYPE(*SLTEVT) /* 只有選取的個別事件和機器指令才會併入追蹤定義中 */
 SLTEVT(*YES) /* *除了可以與 TRCTYPE 參數搭配使用的事件種類以外，
 SLTEVT 可讓您指定個別機器指令和要指定的事件。*/
 DSKEVT((*ALL)) /* 要追蹤所有磁碟事件。*/


效能探究程式資料庫檔案： 下表顯示當使用資料收集指令時系統收集的效能探究程式資料檔。鍵入下列的「顯示檔案欄位說明 (DSPFFD)」指令檢視一個檔案的內容：

DSPFFD FILE(xxxxxxxxxx)

其中 xxxxxxxx 是要顯示的檔案名稱。

併入檔案的資訊類型	檔名
參考資料	QAYPEREF
一般資訊	QAYPERUNI
PMC 選項	QAYPEFQCFG
基本配置資訊	QAYPECFG
針對以下而收集的機器介面 (MI) 複合指令：	QAYPELCLPX
針對以下而收集的工作：	QAYPELJOB
在以下收集資料時使用的計量單位：	QAYPELMET
針對以下收集的機器介面 (MI) 程式、模組或程序：	QAYPELMI
針對以下收集資料時使用的授權內碼 (LIC) 模組：	QAYPELLIC
針對以下收集資料時使用的作業名稱：	QAYPELNAMT
針對以下收集資料時使用的作業號碼：	QAYPELNUMT
機器介面 (MI) 複合指令對映	QAYPEMICPX
事件類型和子類型對映	QAYPEEVENT
硬體對映資料	QAYPEHWMAP
授權內碼 (LIC) 位址解析對映	QAYPEPROCI
區段位址解析對映	QAYPESEGI
處理和作業解析對映	QAYPETASKI
所有事件的一般追蹤資料	QAYPETIDX
輔助儲存體管理事件資料	QAYPEASM
基本事件資料	QAYPEBASE
磁碟事件資料	QAYPEDASD
磁碟伺服器事件資料	QAYPEDSRV
尋頁錯失事件資料	QAYPEPGFLT
資源管理處理事件資料	QAYPERMPM
資源管理抓取鎖定事件資料	QAYPERMSL
進階 System/36 事件資料	QAYPES36
區段位址範圍 (SAR) 資料	QAYPESAR
不明事件資料	QAYPEUNKWN
基本統計值資料	QAYPESTATS
統計剪影摘要資料	QAYPEPSUM
授權內碼 (LIC) 括組資料	QAYPELBRKT
機器介面 (MI) 使用者事件資料	QAYPEMIUSR
機器介面 (MI) 程式括組資料	QAYPEMBRKT

併入檔案的資訊類型	檔名
機器介面 (MI) 指標的位址	QAYPEMIPTR
使用者定義的括組連結資料	QAYPEUSRDF
硬體監督資料	QAYPEHMON
硬體監督資料總計	QAYPEHTOT
版次、版本、修正層次	QRLVRM
效能探究程式層次指示器	QRLVL
效能探究程式 Java 事件資料	QAYPEJVA
效能探究程式 Java 類別資訊資料	QAYPEJVICI
效能探究程式 Java 方法資訊資料	QAYPEJVMCI
效能探究程式 Java 名稱資訊資料	QAYPEJVNI
同步化事件資料	QAYPESYNC
通信事件資料	QAYPECMN
「檔案儲存」事件資料	QAYPEFILSV
資料堆事件資料	QAYPEHEAP
PASE 事件資料	QAYEPASE
追蹤工作相等事件資料	QAYPETBRKT
作業切換事件資料	QAYPETSWSW
同步化事件資料	QAYPESYNC
程式設定檔資料	QAYPEPPANE

效能探究程式報告： 效能探究程式會收集程式或工作行為與效能的詳細資訊，並將此資訊儲存在效能探究程式資料庫檔案中。您可以使用 SQL 來查詢這些檔案，或執行其中一種報告。您可以使用效能探究程式產生四種不同的報告：「統計值」、「設定檔」、「追蹤」及「基本報告」。請參閱效能探究程式定義，以取得為何要使用使用特定定義來產生其中一種報告的相關資訊。如需每一種報告的詳細說明，請參閱 Performance Tools 。

您可以使用「列印效能探究程式報告 (PRTPEXRPT)」指令，以建立及列印效能探究程式報告。當您想要自訂「追蹤報告」時，請使用 OUTFILE 參數。下列指令是列印每一種效能探究程式資料報告的範例：

- 列印 *STATS 報告，並按使用的 CPU 時間排序

```
PRTPEXRPT MBR(MYSTATS) LIB(MYLIB) TYPE(*STATS) STATSOPT(*CPU)
```

- 列印按程序彙總的設定檔報告

```
PRTPEXRPT MBR(MYPROFILE) LIB(MYLIB) TYPE(*PROFILE) PROFILEOPT(*SAMPLECOUNT *PROCEDURE)
```

- 列印按作業 ID 排序的追蹤

```
PRTPEXRPT MBR(MYTRACE) LIB(MYLIB) TYPE(*TRACE) TRACEOPT(*TASK)
```

效能探究程式會將收集的資料儲存在 QAVPETRCI 檔案中，該檔案位於 QPFR 檔案庫。鍵入下列指令以檢視單一記錄的內容：

```
DSPFFD FILE(QPFR/QAVPETRCI)
```

效能探究程式優點： 對於在 iSeries 伺服器需要詳細效能分析的人士來說，效能探究程式有一些優點。使用效能探究程式可以執行下列作業：

- 判斷是使用者、工作、檔案、物件、緒、作業、程式、模組、程序、陳述式或指令位址層次導致系統發生效能問題。

- 收集關於使用者開發的軟體和系統軟體的效能資訊。
- 對一項工作執行詳細分析而不影響系統上的其他作業的效能。
- 在一個不是用來收集資料的另一個系統上分析資料。例如，如果在網路中的某個受管理的系統收集資料，您可以將它傳送到中央站系統以進行分析。

配置效能探究程式

如果要配置效能探究程式，請遵循下列步驟：

1. 通知 iSeries 伺服器您要收集哪一個效能資料的建立階段作業定義。在「新增效能探究程式定義 (ADDPEXDFN)」畫面，指定定義的資料收集類型和名稱。這個定義按該名稱以資料庫成員身份儲存在 QUSRSYS 檔案庫中的 QAPEXDFN 檔。在「啟動效能探究程式 (STRPEX)」指令使用您指定的名稱。
2. 新增過濾器 (ADDPEXFTR 指令)。效能探究程式過濾器識別要在效能探究程式階段作業期間收集的效能資料，而且透過指定特定事件的比較值來限制收集的資料量。
3. 開始收集資料 (STRPEX 指令)。如果沒有收集 *PMCO 事件，那麼一項工作可能會出現在數個效能探究程式集合。如果有收集 *PMCO 事件，那麼唯有所有的資料收集有相同間隔指定 (ADDPEXDFN INTERVAL() 參數) 情況下，同一項工作才可以出現在數個資料收集。
4. 為要分析的資料執行指令、程式或工作量。
5. 停止收集資料並將它儲存到資料庫檔案以進行分析。使用「結束效能探究程式 (ENDPEX)」指令停止集合。
6. 分析效能資料。列印效能探究程式報告 (PRTPEXRPT) 指令 (併入「效能工具」授權程式) 為每一種資料 (統計、設定檔、追蹤設定檔或追蹤) 提供唯一的報告。分析的另一項選擇是將自己的查詢寫入資料庫檔案集。

可以使用下列其中一種方法存取所有效能探究程式指令：

- 指令介面。從指令行鍵入指令。除了「列印效能探究程式報告 (PRTPEXRPT)」指令以外，所有指令都是 OS/400 作業系統的一部份。
- 「效能工具」功能表選項。

如果要瞭解效能探究程式的工作週期，請參閱效能探究程式概念。

結束效能探究程式

如果要結束效能探究程式階段作業，請使用「結束效能探究程式 (ENDPEX)」指令。ENDPEX 指令對選取的資料執行下列動作：

- 將收集的資料放在您指定的檔案庫中的 QAYPExxx 檔案。
使用 OPTION(*END) 和 DTAOPT(*LIB) 執行這項作業。除非您為 DTAMBR 參數指定名稱，否則所有 QAYPExxx 檔案的資料庫成員名稱使用階段作業名稱作為預設名稱。您可以指定 RPLDTA(*NO) 刪除使用這個階段作業名稱收集的資料，或指定 RPLDTA(*YES) 新增收集的資料到現存資料。除非您是很有經驗的使用者，否則請使用 RPLDTA(*NO)。
- 將收集的資料放在一個 IBM 定義的檔案。
使用 OPTION(*END) 和 DTAOPT(*MGTCOL) 執行這項作業。通常唯有在 IBM 客戶服務代表指示之下您才需要使用 *MGTCOL。在 DTAOPT 參數指定 *MGTCOL 值將資料收集資訊儲存到管理資料收集物件。唯有資料要傳給 IBM 時才應該使用管理資料收集物件選項。效能工具只能分析資料庫檔案。
- 捨棄收集的資料。
如果要儲存資料請使用 OPTION(*END)，或使用 DTAOPT(*DLT) 捨棄收集的資料。判斷無法使用收集的資料時您執行這項作業。例如，其中一個預期的工作沒有如期啟動。如果您選擇 *DLT 選項，就不會儲存階段作業已收集的效能資料。

- 暫停資料收集階段作業但沒有結束它。

使用 `OPTION(*SUSPEND)` 執行這項作業。以後您可以針對特定階段作業 ID 發出具有 `OPTION(*RESUME)` 的 `STRPEX` 指令來重新啟動資料收集。

註：如果忘記作用中資料收集階段作業名稱，請使用 `ENDPEX SSNID(*SELECT)` 指令。

iDoctor for iSeries

iDoctor for iSeries 是由三個元件組成的工具套件：「工作監測程式」、「物件探究程式」及「效能探究程式分析程式」。

「工作監測程式」和「效能探究程式分析程式」用於效能分析。這些元件的伺服器端部份是由各種資料收集及分析程式所組成，它們是設計成以更適用的格式來合併效能資料。「工作監測程式」和「分析程式」的從屬端元件是由一些圖形介面組成，這些圖形介面以彈性圖形及表格檢視畫面顯示伺服器資料。

「物件探究程式」是設計來使 iSeries 資料更容易透過圖形介面存取的一種工具。可列示、檢視及描述系統上任何物件類型。「物件探究程式」可讓您存取系統上任何實體檔案或邏輯檔案，以及使用易用的「查詢定義」和「圖形定義」介面，透過資料建立自己的查詢及圖形。

工作監測程式

「工作監測程式」以非常詳細的方式顯示執行的工作及不執行的原因之即時表格和圖形資料。「工作監測程式」提供了一些不同的報告，這些報告按間隔提供詳細的工作統計值。這些統計值可讓您判斷一些事情，如 CPU 使用率、DASD 計數器、等待、錯誤、呼叫堆疊資訊、衝突資訊等等。

物件探究程式

「物件探究程式」可讓您瀏覽及使用 iSeries 伺服器上的物件。「物件探究程式」的主要功能是「資料檢視器」，它可讓您顯示系統上任何實體檔案的內容並透過資料撰寫自己的查詢及圖形。「物件探究程式」中也併入了下列功能：

- 輕易又快速地過濾檔案庫及物件清單
- 透過快速易用的使用者介面來複製 (CRTDUPOBJ)、剪下 (MOV OBJ)、更名 (RENOBJ) 及刪除 (DLT*) 物件的功能
- 瀏覽邏輯或實體檔案成員
- 使用「查詢定義」介面定義自己的查詢並存檔供日後使用
- 使用「圖形定義」介面定義自己的圖形並存檔供日後使用

效能探究程式分析程式

「效能探究程式分析程式」評估系統整體效能並建立在「效能工具」授權程式所完成的基礎上。「分析程式」將追蹤資料量壓縮成可繪製或檢視的報告，以協助隔離效能問題及縮減整體問題判定時間。「分析程式」提供一個易用圖形介面來分析 CPU 使用率、實體磁碟作業、邏輯磁碟輸入/輸出、資料區及資料佇列。「分析程式」也可協助您找出應用程式執行速度減慢的原因。

詳細資訊，請探訪 [Doctor for iSeries](#)  網站。

Performance Trace Data Visualizer (PTDV)

Performance Trace Data Visualizer (PTDV) 是一個 Java 應用程式，您可以使用該應用程式分析在 iSeries 伺服器執行的應用程式的效能。PTDV 在 OS/400 基本作業系統中使用效能探究程式讓分析師能夠檢視程式流程，取得按追蹤、工作、緒和程序彙總的明細 (例如 CPU 時間、目前系統時間、循環數目和指令數目)。當視覺化

呈現 Java 應用程式追蹤時，可以顯示其他明細，例如建立的物件數目和類型以及關於 Java 鎖定行為的資訊。也支援「WebSphere 應用程式伺服器」產生的效能探究程式事件。PTDV 允許排序直欄、匯出資料以及許多層次的資料彙總。

關於詳細資訊，請前往 [Performance Trace Data Visualizer !\[\]\(d0a1791f26d167e866e44ebbf83efebe_img.jpg\) 網站。](#)

效能管理 API

效能管理 API 可讓您使用「資料收集服務」、效能收集器、效能探究程式和「效能管理 /400 (PM/400)」，收集和管理效能資料。

「效能管理 API」包含：

- 資料收集服務 API
- 效能收集器 API
- 效能探究程式 (PEX) API
- Performance Management/400 (PM/400) APIs

“使用” OS/400 效能的指令

OS/400 包含一些指令，它們可讓您透過字元型介面執行即時監督效能資料。您可以使用這些指令回答關於系統效能的問題並協助您調整系統。關於從「iSeries 領航員」執行即時監督的資訊，請參閱 iSeries 領航員監督程式。

指令	功能
處理作用中的工作 (WRKACTJOB)	可讓您複查和變更正在系統執行的工作的屬性和資源使用率。
處理磁碟狀態 (WRKDSKSTS)	顯示系統硬碟機的效能資訊和屬性。
查看系統狀態 (WRKSYSSTS)	提供目前系統活動概觀。尤其是它顯示系統上的工作數目和儲存體儲存區使用率資訊。
處理系統活動 (WRKSYSACT)	處理系統上的工作和作業。這個指令是「效能工具」授權程式 (PT1) 的一部份。
處理物件鎖定 (WRKOBJLCK)	在指定物件處理和顯示鎖定，包含等待要套用的鎖定。
使用共用儲存體儲存區 (WRKSHRPOOL)	顯示使用率資訊和變更共用儲存體儲存區的屬性，包含機器和基本儲存區。

延伸的適應性快速記憶體

註：特性碼 #4331 和 #6831 (CCIN #6731) 已從市場中撤回。這裡所提供的資訊供現有使用者參考。iSeries 儲存體 I/O 的這個快取功能之任何未來加強功能可透過這些頁面取得。

使用「延伸的適應性快速記憶體」來增進您的 iSeries 系統效能！「延伸的適應性快速記憶體」是一種進階大量讀取快取技術，它藉由縮減讀取自磁碟的實體 I/O 要求數目來改善 I/O 子系統及系統回應時間。「延伸的適應性快速記憶體」產生資料的統計資訊，然後使用混合的管理策略來決定所要快取的資料。「延伸的適應性快速記憶體」已證明可以提高許多工作量類型的效率。

IBM 提供一個創新的工具，讓您判定「延伸的適應性快速記憶體」可在 iSeries 運算環境中提供多少優點。**延伸的適應性快速記憶體模擬器**透過「iSeries 領航員」內的「資料收集服務」功能來啟動。此「模擬器」顯示對每個磁碟在歷經一段時間後的實際工作量模擬的效能結果。「延伸的適應性快速記憶體模擬器」在 I/O 儲存體配接卡層次上執行，並且使用管理「延伸的適應性快速記憶體」的相同演算法。

詳細資訊，請參閱：

- **延伸的適應性快速記憶體概念**

探索「延伸的適應性快速記憶體」。開始使用此工具之前，請參閱有關規劃、限制及重要注意事項的資訊。

- **延伸的適應性快速記憶體模擬器**

瞭解如何使用「延伸的適應性快速記憶體」來判斷「延伸的適應性快速記憶體」在您運算環境中對回應時間有多少改善。

- **取得延伸的適應性快速記憶體**

使用「延伸的適應性快速記憶體模擬器」來查看此工具可為您環境提供的優點之後，請瞭解如何取得「延伸的適應性快速記憶體」。

延伸的適應性快速記憶體概念

使用「延伸的適應性快速記憶體」來增進系統效能，這是一種進階讀取快取技術，它藉由縮減讀取自磁碟的實體 I/O 要求數目來改善 I/O 子系統及系統回應時間。「延伸的適應性快速記憶體」所增進的不只是資料庫讀取動作的效能，而是所有讀取動作的效能。這包括其它系統元件 (如整合型 xSeries 伺服器) 產生的讀取動作。它在具有裝置同位檢查保護或鏡映保護的儲存體子系統上也很有效。「延伸的適應性快速記憶體」已證明可以提高許多工作量類型的效率。

如何使用「延伸的適應性快速記憶體」

「延伸的適應性快速記憶體」已整合到 iSeries I/O 子系統中。它在磁碟子系統控制器層次上操作，不影響 iSeries 系統處理器。儲存體 I/O 配接卡藉由使用「讀取快取裝置」(如實體狀態磁碟) 提供快取記憶體，以管理「延伸的適應性快速記憶體」。

「延伸的適應性快速記憶體」產生資料的統計資訊，然後使用混合的管理策略來決定所要快取的資料。快取的管理在 I/O 配接卡內自動執行，其設計來使用預測演算法快取資料。此演算法會考量主電腦存取預定資料範圍的時間性及頻率。

「延伸的適應性快速記憶體」的設計是根據 iSeries 伺服器的特定資料管理策略。無論磁碟是裝置同位檢查保護、鏡映保護或不受保護，儲存於磁碟上的資料有形成頻帶的可能。這表示資料主動讀取的磁碟儲存體有實際連續區域、經常寫入的實際連續區域、主動讀取及寫入的實際連續區域，或不常存取的儲存體之實際連續區域。

在「延伸的適應性快速記憶體」設計中說明了資料的這個“頻帶”。其目標是要快取以讀取/寫入及唯讀為特性的頻帶。以唯寫為特性的頻帶於儲存體子系統寫入快取中被快取時，通常不受「延伸的適應性快速記憶體」影響。「延伸的適應性快速記憶體」在設計上亦不損及循序寫入或循序讀取的大型資料區塊效能。在這個案例中，磁碟的預先提取功能及系統中的其它快取確保了快速回應時間。

詳細資訊，請參閱：

- **延伸的適應性快速記憶體限制和注意事項**

瞭解「延伸的適應性快速記憶體」需要哪些元件，以及詳細瞭解其內容。

- **延伸的適應性快速記憶體模擬器**

瞭解如何使用「延伸的適應性快速記憶體模擬器」來判斷「延伸的適應性快速記憶體」在您的運算環境中所提升的回應時間。

- **啟動延伸的適應性快速記憶體**

瞭解如何啟動「延伸的適應性快速記憶體」。

延伸的適應性快速記憶體 限制和注意事項: 開始使用延伸的適應性快速記憶體之前，您應該做一些初始規劃來考量與您的運算環境有關的任何限制或注意事項。

限制

若要使用「延伸的適應性快速記憶體」，您的系統必須具備下列所示：

- 支援「延伸的適應性快速記憶體」的一或多個儲存體 I/O 配接卡 (適用執行 V4R4 或更新版本系統的 CCIN 2748，或適用執行 V5R1 或更新版本系統的 CCIN 2778，或適用執行最新 V5R2 版次系統的 CCIN 2757)。
- 要啟動「延伸的適應性快速記憶體」的每一個儲存體 I/O 配接卡的「讀取快取裝置 (RCD)」 (適用 V4R4 或更新版本系統的 CCIN 6731)
- Performance Tools for iSeries 授權程式

「延伸的適應性快速記憶體」透過 RCD 自動啟用。沒有控制性開關。透過並行維護，新增 RCD 不會岔斷系統。RCD 位於內部磁碟插槽中並使用其它所有磁碟類型及產能。注意：「延伸的適應性快速記憶體」中所有資料也必然位於磁碟上。在不太可能發生的 RCD 故障事件中，不會流失資料。

對於 I/O 配接卡下的其它磁碟而言，在使用「延伸的適應性快速記憶體」時並沒有關於裝置同位檢查保護及 鏡映保護方面的限制。不過，在相同 I/O 配接卡上，「延伸的適應性快速記憶體」無法與「整合硬碟壓縮」一起使用。最後，「延伸的適應性快速記憶體」是特別設計來補充 iSeries Expert Cache，可搭配或獨立使用。

注意事項

在大部份環境下，「延伸的適應性快速記憶體」可讓您達到 I/O 回應時間明顯減少，以及系統 I/O 產能增加的目標。如同快取的一般情況，系統配置與工作量會影響「延伸的適應性快速記憶體」的效能。「延伸的適應性快速記憶體」在儲存體子系統層次上執行。它會快取特定子系統內磁碟集的資料。因此，理論上會新增「延伸的適應性快速記憶體」至系統內大部份作用中且重視效能的儲存體子系統。「延伸的適應性快速記憶體」不被視為預先提供類型的快取，因此不會與磁碟中的先讀功能衝突。

主動接收 I/O 要求的磁碟儲存體區域越大，「延伸的適應性快速記憶體」在決定何時將新資料帶入快取的選擇性就越多。這個適應性功能可讓「延伸的適應性快速記憶體」在許多工作量類型及大小很有效。從這個觀點來看，最能夠看出使用「延伸的適應性快速記憶體模擬器」對整體快取的有效性。

另外，「延伸的適應性快速記憶體模擬器」和「延伸的適應性快速記憶體」不能同時使用於相同的儲存體 I/O 配接卡。

在清楚這些限制和注意事項之後，您就可以開始啟動延伸的適應性快速記憶體。

啟動延伸的適應性快速記憶體：若要啟動延伸的適應性快速記憶體並增加系統效能，請購買「讀取快取裝置」。將「讀取快取裝置」插入系統的磁碟槽後，便會啟動「延伸的適應性快速記憶體」。這裡並沒有使用者控制的開關。「延伸的適應性快速記憶體」監督資料流及移入「讀取快取裝置」，大約需要 1 小時的時間。執行「延伸的適應性快速記憶體」1 小時之後，您的系統效能應有所改善 (根據您目前的工作量) 且 I/O 產能增加。

欲得知您的 iSeries 系統能否使用「延伸的適應性快速記憶體」，請參閱「延伸的適應性快速記憶體」的限制和注意事項。

延伸的適應性快速記憶體模擬器

註：特性碼 #4331 和 #6831 (CCIN #6731) 已從市場中撤回。這裡所提供的資訊供現有使用者參照。iSeries 儲存體 I/O 的這個快取功能之任何未來加強功能可透過這些頁面取得。

使用「延伸的適應性快速記憶體模擬器」來預估使用「延伸的適應性快速記憶體」增進的系統效能。「延伸的適應性快速記憶體模擬器」是一種效能工具，它可判斷「延伸的適應性快速記憶體」為您系統改善了多少回應時間。這個判斷可視您的系統配置及資料工作量而定，而且是在您購買「讀取快取裝置」之前做下判斷。

「延伸的適應性快速記憶體模擬器」在「資料收集服務」內控制，適用於具有 CCIN 2757 I/O 配接卡的最新 V5R2 系統。(CCIN 2748、CCIN 2778 和 CCIN 2757 I/O 配接卡為支援「延伸的適應性快速記憶體」本身之相同儲存體 I/O 配接卡)。此「模擬器」具有彈性，可讓您模擬不同的快取功能，對於最適合您特定系統及工作量需求的產能做出更好的判斷。實際的「讀取快取裝置」產能為 1600 MB。

經由啟動「延伸的適應性快速記憶體模擬器」所收集的效能資訊，指出透過使用「延伸的適應性快速記憶體」可儲存的磁碟讀取作業數目。效能資料反映出磁碟存取時間可能改善多少。

取得延伸的適應性快速記憶體

從延伸的適應性快速記憶體模擬器取得效能資料並決定要「延伸的適應性快速記憶體」改進您的系統效能之後，您必須購買「讀取快取裝置 (RCD)」。「延伸的適應性快速記憶體」透過 RCD 自動啟用。


若要開始使用「延伸的適應性快速記憶體」，您必須具備下列所示：

- 支援「延伸的適應性快速記憶體」之一或多個儲存體 I/O 配接卡 (適用執行 V4R4 或更新版本系統的 CCIN 2748，或適用執行 V5R1 或更新版本系統的 CCIN 2778，或適用執行最新 V5R2 版次系統的 CCIN 2757)。
- 要啟動「延伸的適應性快速記憶體」的每一個儲存體 I/O 配接卡的「讀取快取裝置」(適用 V4R4 或更新版本系統的 CCIN 6731)。

由於「延伸的適應性快速記憶體」透過 RCD 自動啟用，所以沒有控制性開關。透過並行維護，新增 RCD 不會中斷系統。RCD 位於內部磁碟插槽中並使用其它所有磁碟類型及產能。注意：「延伸的適應性快速記憶體」中所有資料也必然位於磁碟上。在不太可能發生的 RCD 故障事件中，不會流失資料。

您可以在銷售 iSeries 硬體的地方購買到「讀取快取裝置」，或是聯絡當地 IBM 業務代表。

Workload Estimator for iSeries

Workload Estimator  可協助您依照特定工作量類型的預估工作量，調整系統大小需求。PM/400 是一種整合的 OS/400 功能，依照處理器保固或 IBM 維護合約，使用者可以啟動此功能而無需另外付費。相對的，您會收到容量與效能分析圖形，這在規劃及管理系統成長與效能時非常有用。


Workload Estimator 及 PM/400 已增強，以便能彼此使用。透過 Web 型應用程式，您可以調整必要 iSeries 系統的升級版大小，以配合 PM/400 所報告的現有系統使用率、效能與成長。作為額外選項，調整大小也可以併入新增特定應用程式的容量，如 Domino、Java 及 WebSphere，或在一個系統上合併多個 AS/400 或 iSeries 傳統式 OS/400 工作量。此功能可讓您依據來自系統的現有使用率資料，規劃未來的系統需求。

iSeries Navigator for Wireless

iSeries Navigator for Wireless 可讓您使用網際網路型電話、含有無線數據機的個人數位助理 (PDA) 或傳統 Web 瀏覽器，從遠端監督系統效能和狀態。透過無線裝置，您可以執行下列工作：

- 在多個系統之間執行指令
- 啟動和檢視系統、工作和訊息監督程式
- 從監督程式處理工作和訊息 (保留、釋放、結束、回答、取得明細)
- 管理整合 xSeries 伺服器

關於 iSeries Navigator for Wireless 如何協助您開始執行遠端監督的概觀，請參閱 iSeries Navigator for Wireless 這個主題。

關於遠端監督的完整和最新資訊，請參閱 iSeries Navigator for Wireless  首頁。

PATROL for iSeries (AS/400) - Predict

PATROL for iSeries (AS/400) - Predict 產品可協助您透過自動化許多效能管理作業來管理 iSeries 效能。本產品提供 iSeries 伺服器群組的現行和歷程系統資料的詳細畫面，可讓您往下探查像 CPU 和檔案使用率和狀態等特定明細。此外，Patrol 提供數個自動化選項，這些選項可讓它在 iSeries 伺服器尚未發生效能和可用性問題之前先解決這些問題。

本產品可透過提供詳細產能規劃資訊協助您規劃未來升級和因應 iSeries 環境的成長。

關於詳細資訊，請參閱 BMC 產品網站 。

實務範例：效能

瞭解效能管理的最佳方法之一，就是察看說明在範例企業環境中可以使用多少應用程式與功能的範例。使用下列實務範例及配置範例，以瞭解有關管理效能的其餘資訊。

實務範例：升級或移轉之後，增進系統效能

在此實務範例中，您剛剛升級或移轉系統，且它現在的執行速度似乎比之前更慢。此實務範例可協助您識別及修正效能問題。

實務範例：系統監督程式

請參閱範例系統監督程式，如果 CPU 使用率太高時，它會發出警示，並暫時保留優先順序較低的工作，直到有更多的資源可用為止。

實務範例：訊息監督程式

請參閱範例訊息監督程式，它會顯示在任一 iSeries 伺服器上所發生的訊息佇列中的任何查詢訊息。只要它偵測到訊息，監督程式即會開啓並顯示訊息。

實務範例：工作監測程式

請參閱範例工作監測程式，它會追蹤指定工作的 CPU 使用率，且如果 CPU 使用率太高時，它會警示工作擁有者。

相關資訊

下面列出的是與「效能」主題相關的 iSeries 手冊 (有時稱為「白皮書」) 和 IBM Redbooks^(TM)，它們使用 PDF 格式。您也可以檢視或列印下列 PDF：

- 手冊

Performance Tools for iSeries

本書提供程式設計師要收集關於系統、工作或程式效能的資料時需要的資訊。它也包含一些秘訣，供您列印和分析效能資料以識別和更正那些可能存在的無效率事例，以及關於「管理程式」和「代理程式」特性的資訊。

- 網站

iSeries Performance Capabilities Reference

這個參照提供關於對效能評比、產能規劃和規劃伺服器效能來說很有用的伺服器效能的高度技術性資訊。

- 紅皮書：

- **IBM eserver iSeries Universal Connection for Electronic Support and Services** 
本文件簡介廣用連線。它也說明如何使用不同支援工具，這些工具向 IBM 報告您機器上的軟硬體庫存，使您可以根據系統資料獲得個人化電子支援。
- **Lotus Domino for AS/400: Performance, Tuning, and Capacity Planning** 
本文件說明效能管理的方法。它包含設定效能目標、收集和複查效能資料、調整資源和產能規劃。也提供效能指引和應用程式設計秘訣。
- **AS/400 Performance Management** 
本文件說明效能管理的方法。它包含設定效能目標、收集和複查效能資料、調整資源和產能規劃。也提供效能指引和應用程式設計秘訣。
- **AS/400 HTTP Server Performance and Capacity Planning** 
網際網路和 Web 瀏覽器型應用程式深刻影響組織如何分送資訊、執行業務處理、服務客戶以及開發新市場。本書適合 iSeries 程式設計師、網路和系統管理專家以及其他負責設計、開發和部署 Web 型應用程式與資訊系統的資訊技術人員閱讀。
- **Java and WebSphere Performance on IBM eserver iSeries Servers** 
本文件提供一些秘訣、技術和方法學，以處理特別針對 iSeries 伺服器的 Java 和「WebSphere 應用程式伺服器」效能相關問題。
- **Management Central: A Smart Way to Manage AS/400 Systems** 
探索「管理中心」的優點。本書討論置換效能監視器的「資料收集服務」。它也說明「管理中心」如何透過提供即時效能監督功能，例如事件通知和自動回應事件，讓操作員和管理者能夠在網路中監督伺服器。
- **Managing AS/400 V4R4 with Operations Navigator** 
本書讓您深入瞭解許多可以透過「作業領航員」(現在稱為「iSeries 領航員」) 來使用的 iSeries 功能。除了完整討論這些功能以外，本書還特別討論關於「管理中心」和「資料收集服務」的監督系統效能。
- **AS/400 Performance Explorer Tips and Techniques** 
本文件提供 V3R6 可使用的效能探究程式功能的說明和詳細範例。提供特定的應用程式範例和報告。

關於 iSeries 效能的完整資訊，請參閱效能主題。

IBM