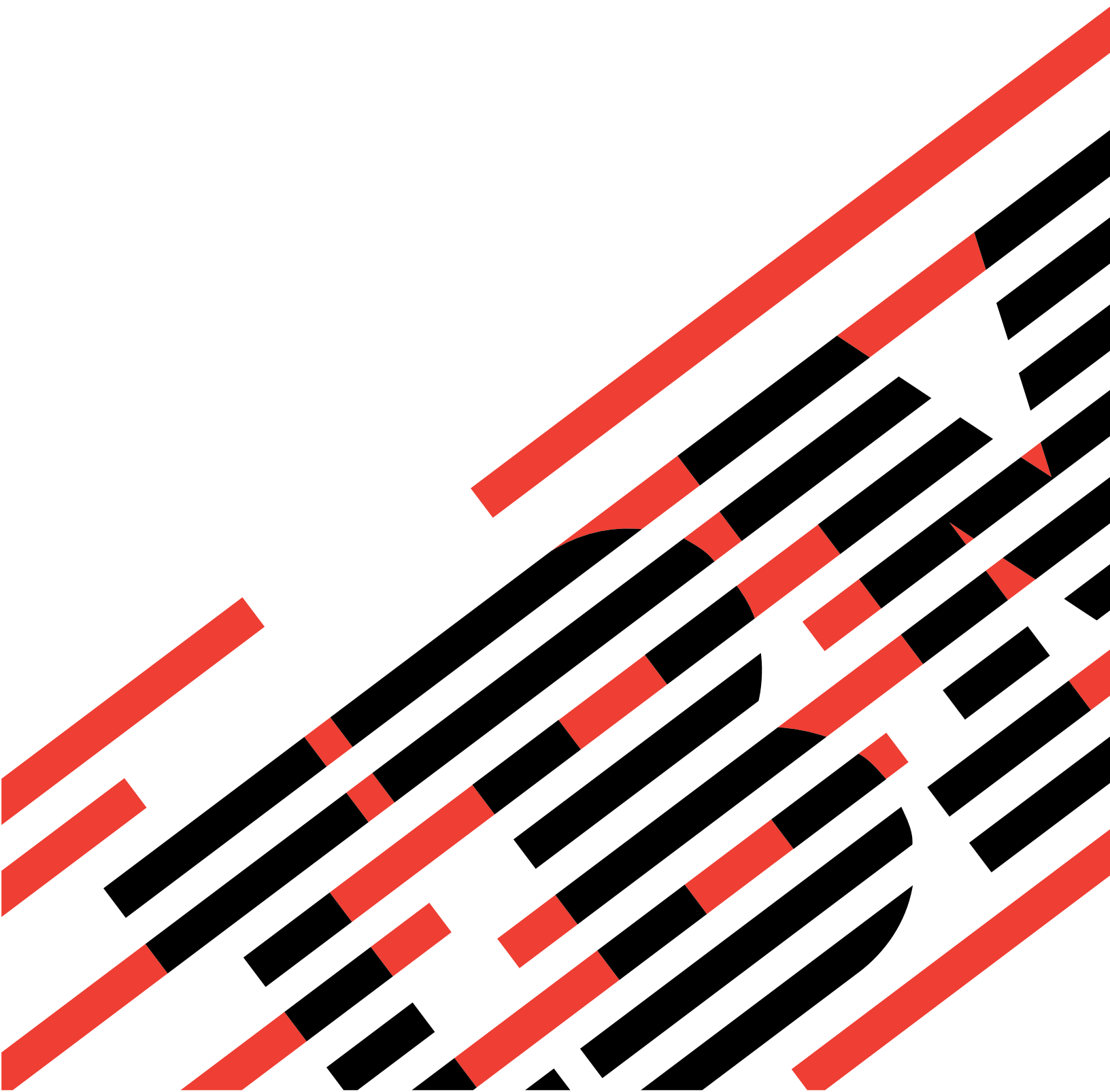


IBM

@server

iSeries

性能







@server

iSeries

性能



# 目录

性能	1
V5R2 中的新增内容	1
新增内容: 监控器	2
新增内容: 收集服务	2
新增内容: Performance Tools 许可程序	3
新增内容: Performance Management/400	5
新增内容: 性能资源管理器	5
打印本主题	7
计划性能	8
设置系统基准	9
确定何时和如何扩充系统	9
按需升级容量	10
按需升级容量概念	10
按需升级容量试用期	11
按需升级容量的软件发放许可证注意事项	11
按需处理器激活代码	12
Electronic Service Agent 需求	12
准备按需升级容量	12
按需升级容量的容量计划	13
确定何时激活处理器	13
为按需升级容量设置环境	13
订购按需升级容量	14
将重要产品数据发送至 IBM	14
激活备用处理器	15
在试用期内激活备用处理器	15
永久激活备用处理器	16
方案: 按需升级容量	17
选择性能管理策略	17
设置管理性能的环境	19
管理 iSeries 性能	19
跟踪性能	20
研究性能问题	20
标识性能问题	21
标识和解决公共性能问题	21
收集系统性能数据	22
收集有关系统资源利用率的信息	23
收集有关应用程序的性能的信息	23
转储跟踪数据	24
方案: 在升级或迁移后改进系统性能	24
显示性能数据	25
调整性能	25
基本性能调整	26
自动调整性能	27
管理电子商务性能	27
客户机性能	28
网络性能	28
OS/400 中的 Java 性能	29
HTTP Server 性能	29

WebSphere 性能 . . . . .	30
性能管理应用程序 . . . . .	31
收集服务 . . . . .	33
从收集服务数据创建数据库文件 . . . . .	36
从现有的收集对象创建数据库文件 . . . . .	37
定制数据收集 . . . . .	37
收集服务的时区注意事项 . . . . .	38
收集服务中的用户定义类别 . . . . .	38
收集程序建议和需求 . . . . .	39
示例: 实现用户定义类别 . . . . .	40
管理收集对象 . . . . .	47
用户定义的事务 . . . . .	47
C++ 示例: 将用户定义的事务集成到收集服务中 . . . . .	48
Java 示例: 将用户定义的事务集成到收集服务中 . . . . .	52
性能数据文件 . . . . .	55
包含时间间隔数据的性能数据文件 . . . . .	56
性能数据文件: 文件缩写 . . . . .	57
性能数据文件: 收集服务系统类别和文件关系 . . . . .	58
性能数据文件: 配置数据库文件的字段数据 . . . . .	59
性能数据库文件: 跟踪数据库文件的字段数据 . . . . .	60
iSeries 导航器监控器 . . . . .	60
监控器概念 . . . . .	61
配置监控器 . . . . .	61
监控器度量 . . . . .	61
方案: iSeries 导航器监控器 . . . . .	63
方案: 系统监控器 . . . . .	63
方案: 作业监控器 . . . . .	64
方案: 消息监控器 . . . . .	65
图形历史记录 . . . . .	66
图形历史概念 . . . . .	66
使用图形历史记录 . . . . .	67
Performance Management/400 . . . . .	67
PM/400 概念 . . . . .	68
PM/400 的益处 . . . . .	68
Operational Support Services for PM/400e 产品 . . . . .	68
PM/400 的数据收集注意事项 . . . . .	69
配置 PM/400 . . . . .	69
激活 Performance Management/400 . . . . .	70
定制 PM/400 . . . . .	76
管理 PM/400 . . . . .	78
取消激活 PM/400 . . . . .	78
更改 PM/400 联系信息 . . . . .	79
使用 PM/400 调度作业 . . . . .	79
省略 PM/400 分析中的项 . . . . .	80
立刻关闭 PM/400 . . . . .	80
显示 PM/400 状态 . . . . .	81
查看 PM/400 报告 . . . . .	81
Performance Tools . . . . .	81
Performance Tools 概念 . . . . .	82
Performance Tools 中提供的功能 . . . . .	82
管理器和代理程序功能部件比较 . . . . .	82
显示性能信息 . . . . .	83

安装和配置 Performance Tools . . . . .	84
Performance Tools 报告 . . . . .	84
性能资源管理器 . . . . .	86
性能资源管理器概念. . . . .	87
性能资源管理器定义. . . . .	88
性能资源管理器数据库文件 . . . . .	90
性能资源管理器报告. . . . .	91
性能资源管理器的益处. . . . .	92
配置性能资源管理器. . . . .	92
结束性能资源管理器. . . . .	92
iDoctor for iSeries . . . . .	93
Performance Trace Data Visualizer (PTDV) . . . . .	94
性能管理 API . . . . .	94
“使用” 有关 OS/400 性能的命令 . . . . .	94
扩展的自适应高速缓存 . . . . .	94
扩展的自适应高速缓存概念 . . . . .	95
扩展自适应高速缓存的限制和注意事项 . . . . .	96
启动扩展自适应高速缓存 . . . . .	96
扩展的自适应高速缓存模拟器 . . . . .	96
获取扩展的自适应高速缓存 . . . . .	97
Workload Estimator for iSeries . . . . .	97
iSeries 无线导航器 . . . . .	97
PATROL for iSeries (AS/400) — 预测. . . . .	98
方案: 性能 . . . . .	98
相关信息 . . . . .	98





---

# 性能

在管理系统的性能方面投资多少？商业更改的需要有时比预期的更早。为了有效地响应商业更改，系统也必须更改。管理您的系统乍看起来好像只是另一个消耗时间的作业。但是，由于系统更有效地运行，所以投资很快会得到回报，并且这反映在您的商务中。由于更改是有计划的并受到管理，所以它是高效的。

管理 iSeries 服务器的性能可能是一个复杂的任务，它要求彻底了解 iSeries 工作管理。了解影响系统性能的所有不同进程对于没有经验的用户可能是一个挑战。解决性能问题要求有效使用很大的一套工具，每个工具都具有它们自己唯一的一组需求和支持的功能。即使在您已收集并分析性能数据之后，了解要使用该信息干什么也可能使人感到沮丧。

本主题将指导您完成与性能管理关联的任务和工具。

## 『 V5R2 中的新增内容 』

此主题描述本发行版中哪些信息是新的或已显著更改的。

## 第 7 页的『打印本主题』

如果您更喜欢本信息的打印版本，转至此处来打印 PDF。

## 第 8 页的『计划性能』

设置 iSeries 服务器的性能目标将允许您具有可测量的性能基准来比较性能数据。此主题将解释如何设置那些基准和以后如何使用它们。

## 第 19 页的『设置管理性能的环境』

iSeries 服务器包括用于管理系统性能的功能强大的应用程序。然而，必须正确配置它们以满足独特商务环境的特定需要。了解如何配置应用程序来例行收集、监控和分析性能数据。

## 第 19 页的『管理 iSeries 性能』

要通过测量当前能力、识别趋势以及进行适当调整以满足最终用户和管理需求（如响应时间或作业吞吐量）来优化计算机系统的利用率，必须进行性能管理。需要维护商务效率并避免延长正常的商务活动的暂停。因此，对性能进行管理是日常操作的一部分。

## 第 31 页的『性能管理应用程序』

在 iSeries 系统中管理性能需要使用各种专门的应用程序。其中的每个应用程序可以提供对系统性能的特了解。此主题解释几个应用程序和每个应用程序的预期用途。

## 第 98 页的『方案：性能』

了解性能管理的最好方式之一是查看一些示例，这些示例说明了在您的商务环境中您如何可以使用这些应用程序或工具。找到这些示例。

## 第 98 页的『相关信息』

IBM<sup>(R)</sup> 相关信息包含技术、技术诀窍和“操作方式”信息。

注：此主题包含代码示例。阅读代码示例不保证声明以了解重要的法律信息。

---

## V5R2 中的新增内容

以下是此发行版的新增内容：

- 第 2 页的『新增内容：监控器』

查找有关新的监控器支持和对现有监控器的更改的信息。

- 『新增内容: 收集服务』  
阅读有关对性能数据库文件的更改和新功能的信息。
- 第 3 页的『新增内容: Performance Tools 许可程序』  
阅读关于对报告的更改的信息。
- 第 5 页的『新增内容: Performance Management/400』  
阅读关于 PM/400 如何自动收集数据的信息。
- 第 5 页的『新增内容: 性能资源管理器』  
阅读关于对性能资源管理器数据库文件的更改和新功能的信息。
- 第 10 页的『按需升级容量』  
阅知 Capacity Upgrade on Demand 的 iSeries 支持。
- 第 27 页的『管理电子商务性能』  
阅读关于可用于在电子商务环境中管理 iSeries 性能的任务、注意事项和资源的信息。
- 主存储器相似性  
阅读关于进程和线程如何能够获得内存和处理器资源的改进相似性的信息。

## 新增内容: 监控器

以下是此发行版的新增内容:

- 文件监控器: 一个使您能够管理多平台环境的功能。您将能够在 iSeries 导航器中运行命令、监控文件并管理多平台端点系统。
- B2B 活动监控器: 一种活动监控器, 它允许您查看指定的合并系统上具有详细信息的事务数据并自动运行 OS/400 命令来控制整个事务计数或事务的持续时间。
- 系统监控器度量现在包括与“点对点协议”相关联的信息。已将几个度量添加至 CPU 利用率属性。这些度量涉及文件系统、流文件、日志操作和计数器组。

## 新增内容: 收集服务

以下是此发行版的新增内容:

### 安装新发行版

关于当升级至新发行版时如何处理性能数据, 有多个选项。可以继续用“转换性能数据”(CVTPFRDTA) 命令转换收集库中的数据。可以使用新的自动数据收集支持。如果不转换性能数据库文件, 并且如果已启动“收集服务”且指定了自动创建数据库文件的选项, 则将创建先前发行版文件的性能库。将把性能数据库文件移至该库。此操作允许创建新文件并保留来自先前发行版级别的现有数据文件。最后, 如果不转换性能数据库文件且不运行“收集服务”, 可以使用“删除文件”(DLTF) 命令(DLTF library/QAPM\*) 删除 QAPMxxxx 文件。

### 用户定义类别

在 V5R2 之前, 预定义了所有性能收集类别, 且随系统一起交付了所有数据收集程序。从 V5R2 开始, 您可以定义自己的性能收集类别。用户定义的性能收集类别使您能够:

- 将新的性能收集类别添加至收集器。注册收集器数据类别(QypsRegCollectorDataCategory) 将用户定义的数据类别添加至“中央管理”的“收集服务”功能的一个或多个收集器定义。
- 为新类别提供数据收集程序以收集数据
- 将为该类别收集的数据存储在管理收集对象中
- 以与系统定义的类别相同的方式调度并运行该类别的数据收集
- 使用管理收集对象 API 访问管理收集对象中为新类别收集的数据。

Domino 和 HTTP Server (基于 Apache) 利用此新功能部件来将它们的性能数据集成到“收集服务”。

### 用户定义的事务

从 V5R2 开始,“收集服务”利用性能资源管理器事务边界 API 来给予应用程序用户定义它们自己事务的能力。“收集服务”数据类别是 USRTNS,与用户定义的事务相关联的性能数据库文件是 QAPMUSRTNS。这些 API 包括以下各项:

- 启动事务
- 结束事务

### 性能数据库文件

除 QAPMAPPN 以外,所有数据库文件现在都是交互式的,这意味着可以按字母顺序排序列中的文本并可以搜索表中的特定单字。

下表显示新的和已更改的数据库文件。

数据库文件	描述
QAPMJOBWT	包含作业、任务和线程等待情况的数据的新文件。
QAPMJOBWTD	包含文件 QAPMJOBWT 中的计数器组的描述的新文件。
QAPMHTTPB	包含 IBM HTTP Server (基于 Apache) 的基本数据的新文件。
QAPMHTTPD	包含 IBM HTTP Server (基于 Apache) 的详细数据的新文件。
QAPMDOMINO	包含 Domino for iSeries 的数据的新文件。
QAPMPPP	包含点对点协议的数据的新文件。
QAPMUSRTNS	包含用户定义的事务的数据的新文件。
QAPMSYSTEM	支持文件统计计数器和日志计数器的新字段。
QAPMSYS	提供一种方式来确定单个系统的不同分区上系统时钟之间的差别的新字段。
QAPMECL	报告已从不受支持的协议废弃的帧的数目的新字段。
QAPMETH	报告已已从不受支持的协议废弃的帧的数目的新字段。
QAPMCONF	新的记录密钥。
QAPMIOPD	支持 I/O 适配器数据的新字段。
QAPMMIOP	支持处理时间的新字段。
QAPMJOBMI	支持文件统计计数器和日志计数器的新字段。
QAPMDISK	支持独立的与 ASP 相关的更改的新的和已更改的字段。
QAPMJOBOS	支持文件统计计数器的新字段。

## 新增内容: Performance Tools 许可程序

以下是此发行版的新增内容:

- 将从此发行版中有效的 Performance Tools 中撤销 BEST/1。可以考虑用于确定容量需求的大小的替代工具包括 (但不限于) 用于分布式系统的 IBM 的 Workload Estimator (与 PM/400 集成在一起) 和 BMC 的第 98 页的『PATROL for iSeries (AS/400) — 预测』产品。将从 V5R2 中有效的 Performance Tools 许可程序中撤消下列命令或菜单选项的支持:
  - 启动 BEST/1 (启动 BEST/1)
  - 分析 BEST/1 型号 (ANZBESTMDL)
  - 创建 BEST/1 型号 (CRTBESTMDL)
  - 转换 MDLSYS 文件 (CVTMDLSYSF)


- 删除 BEST/1 型号 (DLTBESTMDL)
- 打印 BEST/1 文件 (PRTBESTF)
- 检查指定的性能数据成员的版本 (QCYCHKV)
- 将 System/36 文件从 MDLSYS 转换为 BEST/1 (QCYCVTBD)
- 查找指定的性能数据成员的第一个和最后一个日期 (QCYFLDT)
- 从先前层接收消息并重新发送至调用层 (QCYRSNDM)
- 启动 MDLSYS (MDLSYS)
- 已从 PERFORM 菜单除去容量计划 / 建模。
- 现在可以从作为 Performance Tools 插件的 iSeries 导航器获得与第 83 页的『显示性能信息』(DSPPFRDTA) 命令相关联的功能。该界面提供几乎等价于 DSPPFRDTA 命令的功能。它允许您生成和查看 Performance Tools 许可程序报告的子集。
- 已更新“打印组件报告”(PRTCPTRPT) 命令以反映几个增强，这些增强包括下列各项：
  - 新的选择类别允许您选择要打印的作业类型。此选择有助于分析服务器型号上交交互式工作。
  - 新的选择类别允许您选择要在报告中包括哪些作业优先级。当系统具有较高开销时，此选择有助于获取较高优先级工作的总量并查找所有优先级为 00 的工作。
  - 新的列报告写高速缓存过速 (overrun) 的百分比 (% 写高速缓存过速)。
  - “数据库日志记录摘要”部分包括新的子部分，它显示志日志计数器和操作相关的信息。新计数器的数据存储在 QAPMJOBMI 文件中。
- 已更新“打印系统报告”(PRTSYSRP) 命令来反映几个增强，这些增强包括下列各项：
  - “通信摘要”现在包括“点对点协议”(PPP) 信息。PPP 协议还显示在“资源报告”的“通信详细信息”部分中。“显示性能数据”(DSPPFRDTA) 命令还显示来自“显示通信线路详细信息”屏幕的“点对点协议”信息。
  - 添加了新的部分，该部分显示关于由 HTTP Server 作业处理的事务的信息。在报告的底部以摘要或平均行的形式对每个时间间隔提供了此信息。
  - 在“工作量”、“资源利用率”和“资源利用率扩充”部分中包括了新的类别，以显示有关非交互式服务器作业的更多详细信息。
- “使用系统活动”屏幕显示当前处理容量。此信息在共享处理器环境中很有用，特别是在分区的容量更改但虚拟处理器的数目未更改时更是如此。“使用系统活动”屏幕还显示处于等待状态的作业或任务所花费的时间的总百分比。可以使用选项 6 (等待详细信息) 查看等待类别的列表。将把与等待时间记帐数据相关的几个字段添加至 QAITMON 文件，该文件由 WRKSYSACT 命令创建。
- “事务报告”中的“对象的占用 / 锁定冲突摘要”部分最多可以显示 500 个对象。跟踪数据有可能具有 500 个以上的占有 / 锁定冲突。因此，将在此部分中显示 500 个最重要的对象。
- “启动性能跟踪”(STRPFRTRC) 命令现在支持更大的跟踪数据表。
- 已增大报告中某些字段的大小以显示更大的值。该更改影响以下报告。

报告	部分	字段
系统	资源利用率	每秒的磁盘 I/O
系统	资源利用率扩充	物理磁盘 I/O、逻辑数据基本 I/O、磁盘 I/O (同步和异步)、DIO / 秒 (同步和异步)
组件	组件时间间隔活动	每秒的磁盘 I/O
作业时间间隔	非交互式作业摘要	每秒的 I/O 数
作业时间间隔	非交互式作业详细信息	每秒的 I/O 数

## 新增内容: Performance Management/400

以下是此发行版的新增内容:

除非从先前发行版升级并关闭先前发行版上的 PM/400 性能数据收集, 否则 PM/400 自动收集性能数据。在您许可将该数据发送至 IBM 之前, 未传送该数据。自动收集性能数据的益处是当您需要数据时就可以获得它, 并且一激活 PM/400 就可以接收 PM/400 报告。

有关 PM/400e 报告的最新信息, 转至 PM/400 Web 站点 。

### 转换性能数据

因为在安装新发行版之后性能数据库文件是前一级别的, 所以需要执行操作以允许继续收集性能数据。如果使用 Performance Management/400、收集服务或 Performance Tools 许可程序, 且在安装新版之前收集了数据, 则您具有以下关于如何处理来自先前发行版的性能数据的选项:

- 转换数据

使用“转换性能数据”(CVTPFRDTA)命令来转换收集库中的数据。

- 

- 使用自动数据收集支持

如果不转换性能数据库文件, 并且已启动“收集服务”且指定了自动创建数据库文件的选项, 则将创建先前发行版文件的性能库。将把性能数据库文件移至该库。此操作允许创建新文件并保留来自先前发行版级别的现有数据文件。应了解下列各项:

- 创建的库名是 QPFRDvrmmn, 其中 vrm 是当前版本、发行版和修订版, nn 是从 01 开始的唯一序号, 例如 QPFRD52001。

- 

- 该库是用 \*EXCLUDE 公共权限创建的。QSYS 用户概要文件拥有该库, 并且原始库的所有者具有 \*ALL 权限。

- 

- 将移动所有 QAPMxxxx 文件。

- 

- 如果不想保存来自先前发行版的数据, 可以用“删除库”(DLTLIB)命令删除 QPFRDvrmmn 库 (DLTLIB qpfrdvrmmn)

- 

- 删除 QAPMxxxx 文件

如果未转换性能数据库文件且未运行“收集服务”, 则可以用“删除文件”(DLTF)命令删除 QAPMxxxx 文件 (DLTF library/QAPM\*).

## 新增内容: 性能资源管理器

以下是此发行版的新增内容:

- 性能资源管理器命令现在要求用户具有 \*SERVICE 权限。

- 新的 CL 命令允许您创建过滤器来减少捕获的数据量。

- 添加 PEX 过滤器 (ADDPEXFTR)

- 使用 PEX 过滤器 (WRKPEXFTR)

- 除去 PEX 过滤器 (RMVPEXFTR)

- 已增强 STRPEX 命令来支持过滤器。

- 性能资源管理器现在使用管理收集对象 (\*MGTCOL) 来将收集的数据转换为它的数据库文件。“创建 PEX 数据” (CRTPEXDTA) 命令执行转换。
- 已将以下事件添加至“添加 PEX 定义” (ADDPEXDFN) 命令:
  - 应用程序事件 (APPEVT)
  - 可移植应用程序解决方案环境 (PASE) 事件 (PASEEVT)
  - 日志事件 (JRNEVT)
  - iSeries NetServer、文件服务器以及网络文件系统服务器和客户机事件 (FILSVREVT)
  - 同步事件 (SYNCEVT)
  - 高级高速缓存事件 (EXPCCHEVT)
- 已将一个新值 (\*HOSTSVRCNN) 添加至 ADDPEXDFN 命令上的“操作系统”事件 (OSEVT) 参数。此值与 \*DBSVRCNN 值相同。\*HOSTSVRCNN 是建议的值。
- 提供了一个新命令 (“使用 PEX 定义” (WRKPEXDFN) 命令), 该命令允许您使用性能资源管理器定义。
- 当结束性能资源管理器会话时, 可以显示新的未完成收集状态。此状态意味着收集意外结束。
- 此发行版的新数据库文件包括下列各项:

新文件	描述
QAYPEPROCI	此文件包含与收集数据中的每个指令地址相关联的过程信息。它包含先前在 QAYPEMII、QAYPELICI、QAYPENMI 和 QAYPENLIC 文件中的数据。
QAYPECFG	此文件包含收集定义信息。它包含先前在 QAYPECOFG、QAYPECICFG、QAYPESTCFG、QAYPETRCFG 和 QAYPEHWCFG 文件中的数据。
QAYPEFTRI	此文件包含收集过滤器定义信息。
QAYPELTASK	此文件包含任务名和定义信息的编号。
QAYPERINF	此文件包含各种解析信息, 例如, 每个 IP 地址的字符串版本。
QAYPETSWSW	此文件包含特定于任务切换事件的信息。它包含先前在 QAYPEBASE 文件中的数据。

- 此发行版的已删除的数据库文件包括下列各项:

已删除的文件	移动数据的位置
QAYPECICFG	QAYPECFG
QAYPECOFG	QAYPECFG
QAYPEHWCFG	QAYPEPROCI
QAYPELICI	QAYPECFG
QAYPELNAMT	QAYPELTASK
QAYPELNUMT	QAYPELTASK
QAYPEMII	QAYPEPROCI
QAYPENLIC	QAYPEPROCI
QAYPENMIC	QAYPEPROCI
QAYPEPERD	因为不再支持周期方式, 所以不再需要表。
QAYPEPSUM	现在可以从 QAYPEPPANE 和 QAYPETASKI 数据计算概要文件摘要数据。
QAYPEPWDW	此文件包含未使用的概要文件摘要数据。

已删除的文件	移动数据的位置
QAYPESTCFG	QAYPECFG
QAYPES36	System/36 事件不再受支持。
QAYPETRCFG	QAYPECFGII
QAYPETRCPT	QAYPETIDX
QAYPEUNKWN	QAYPEUSRDF

- 此发行版的数据库文件更改包括下列各项：
  - 字符数据字段现在使用缺省值 CCSID 65535。
  - 八字节地址、指令和段现在存储在 8 字节十六进制字段中，而不是将每个字节转换为 2 个十六进制字符并存储在 16 字节字符字段或 16 字节十六进制字段。
  - 存储在压缩十进制数字段中的真数字数据已更改为整数字段：
    - 5 位压缩十进制数字段已更改为 9B 整数字段。
    - 10 位压缩十进制数字段已更改为 18B 整数字段。
    - 20 位压缩十进制数字段已更改为 18B 整数字段，除非值是大于 8 字节的带符号整数。

---

## 打印本主题

要查看或下载性能主题的 PDF 版本，选择性能（大约 350 KB 或 95 页）。此 PDF 不包括性能数据库表信息。

要查看或下载性能数据库表信息的 PDF 版本，选择性能数据库表（大约 350 KB 或 125 页）。

可以查看或下载以下相关主题：








- 中央管理（大约 250 KB 或 55 页）包括有关如何设置“中央管理”端点系统和系统组的信息以及“中央管理”可以帮助您简化服务器管理任务的所有方式的信息，如：
  - 管理用户和组
  - 封装和发送数据
  - 运行命令
  - 用“中央管理”调度程序或“高级作业调度程序”调度任务或作业。
- 工作管理（大约 173 KB 或 40 页）描述以下工作管理概念：
  - 作业的使用期限
  - 日常工作管理
  - 系统的结构
  - 如何完成工作

还可以查看或打印下列任何 PDF：

- 手册：
  - Performance Tools for iSeries （大约 400 页）
  - iSeries Performance Capabilities Reference   
链接至包含此书的最新版本的 Web 站点。此参考提供有关对性能基准测试、容量计划和服务器性能计划有用的服务器性能的高技术含量的信息。


此参考提供有关对性能基准测试、容量计划和服务器性能计划有用的服务器性能的高技术含量的信息。

- 红皮书:

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

要将 PDF 保存在工作站上以便进行查看或打印, 请遵循下列步骤:

1. 在浏览器中打开 PDF (单击以上链接)。
2. 在浏览器的菜单中, 单击**文件**。
3. 单击**另存为...**
4. 浏览至希望在其中保存 PDF 的目录。
5. 单击**保存**。

如果您需要使用 Adobe Acrobat Reader 来查看或打印这些 PDF, 则可以从以下网址来下载其副本: Adobe Web 站点 ([www.adobe.com/prodindex/acrobat/readstep.html](http://www.adobe.com/prodindex/acrobat/readstep.html)) 。

---

## 计划性能

计划系统的性能需要设置性能目标, 基于那些目标创建基准, 并计划系统的发展。本节指导您完成计划系统性能的必要步骤。

当计划系统性能时, 您需要完全了解系统要解决的企业需求, 并能够将这些企业需要转换为性能目标。记住: 因为企业需要发展, 所以性能目标也必须发展。

也许最好的开始方式是估计计算机系统在高峰营业时间段所需要的每小时和每天的最大交互式事务吞吐量。之后, 您可以决定您的本地和远程工作站可接受的平均响应时间。应考虑正常批处理进程将花费多长时间以及如何调度它们以便它们及时完成以满足商务需求。

然后您可以建立一组基本统计信息, 应将其归档于包含以下内容的性能目标计划中:

- 每小时的高峰事务数
- 每天的高峰事务数
- 本地工作站可接受的平均响应时间
- 高峰交互式事务数
- 主要的已调度批处理作业的列表, 包含将运行这些作业的时间和期望的持续运行时间
- 可能需要的其它未调度批处理作业的列表



要计划性能，完成下列任务：

#### 『设置系统基准』

设置良好的系统基准将使您获得正确调整的系统的性能数据。来自系统更改之前和之后的这些性能基准提供故障诊断和计划的重要信息。

#### 『确定何时和如何扩充系统』

因为您的企业需要更改，您的系统也必须更改。要准备进行任何更改，您将要建立当前系统的模型，然后查看当系统、配置或工作量更改时将发生什么情况。

#### 第 10 页的『按需升级容量』

Capacity Upgrade on demand 允许您在不断更改的企业需求需要更多资源时将处理器添加至 iSeries 服务器。您可以订购并激活处理器，或者临时或永久试用。

#### 第 17 页的『选择性能管理策略』

不同的企业需求需要不同的性能管理策略。以下有三种基本的企业模型及其建议的性能管理策略。

## 设置系统基准

您应在对系统配置作出任何重要更改（例如，添加新的交互式应用程序或执行系统升级）之前建立系统基准。维护精确的基准信息可以提供必不可少的故障诊断信息。基准最少应包括来自第 33 页的『收集服务』的当前收集对象。取决于环境，您可能需要使用第 86 页的『性能资源管理器』维护更详细的信息。

设置基准要求：

- 正确的 iSeries 配置是可用的
- 应用程序和数据是具有代表性且有效的
- 所有要使用的程序和软件的正确版本是可用的
- 所需数目的用户和工作站可用于运行测试
- 对每个用户适当地定义了事务

如果没有允许您在工作站仿真用户的专用设备，对交互式工作量运行有意义的基准测试几乎是不可能的。当然，运行批处理基准测试不会像测试交互式应用程序的性能一样复杂，上述前面三点对于此测试类型仍有效。然而，在当前批处理和交互式工作（这往往是实际的客户环境）中设置系统基准测试还需要适当数目的用户和工作站。

## 确定何时和如何扩充系统

因为您的企业需要发展，所以系统需要扩充。要计划未来系统的需要和增长，您必需确定当系统、配置或工作量更改时将发生的事情。此过程称为趋势分析，应该每月执行一次。当您的系统接近资源容量准则时，您可能要更为频繁地收集此数据。

应对交互式 and 批处理环境分别进行趋势分析。如果公司广泛地使用某一应用程序，您可能要对该应用程序执行趋势分析。可能对跟踪重要的另一环境将是月底处理。持续收集趋势分析数据是重要的。如果系统的高峰工作量时间在 10:00 AM 和 2:00 PM 之间，且您在此时间段收集了趋势分析数据，则不要将此数据与其它时间段收集的数据进行比较。

要正确进行容量计划和性能分析作业，您必须收集、分析、维护和归档性能数据。IBM 提供了几种工具来帮助您进行容量计划、资源估计和大小估计：

#### 第 67 页的『Performance Management/400』

PM/400 使收集数据、分析数据和归档数据完全自动化，并给您提供容易理解的性能和容量摘要信息。

PM/400 通过不断对关键性能指标进行分析来帮助您计划和管理系统资源。此功能随 OS/400 许可程序一起交付。除了激活该功能和定期检查是否正在收集数据和正在将数据传送给 IBM，您不必再做其它事情。所有的收集站点都是网络安全的，PM/400e 服务只传送非专利性能数据给 IBM。传送时间完全在您的控制下。

#### 第 97 页的『Workload Estimator for iSeries』

工作量估计器是一个工具，它帮助您基于特定工作量类型的估计工作量对系统需求进行大小估计。通过基于 Web 的应用程序，可以将升级缩放成需要的 iSeries 系统，该系统适应您现有的系统实用程序、性能和发展（由 PM/400 报告的）。作为附加选项，缩放还可以包括添加特定应用程序（例如，Domino、Java 和 WebSphere）的能力，或在一个系统上将多个 AS/400 或 iSeries 传统 OS/400 工作量合并的能力。此能力允许您基于来自自己系统的现有利用率数据来计划将来的系统需求。此应用程序不支持 Processor On-Demand 环境或具有逻辑分区的环境。

#### 第 98 页的『PATROL for iSeries (AS/400) — 预测』

此预测通过使高可用性和最佳性能所需的许多日常的管理任务自动化来帮助管理 iSeries 性能。另外，它还提供详细的容量计划信息来帮助您计划 iSeries 环境的成长。

有关创建和实现性能策略的更多信息，参见第 17 页的『选择性能管理策略』。

## 按需升级容量

iSeries 按需升级容量 (CUoD) 是一个功能部件，该它提供给您动态激活选择服务器型号的一个或多个中央处理器的能力，而不需要重新启动服务器或中断您的商业。可以在试验或永久基础上订购和激活已经安装在服务器上的备用处理器。

使用 CUoD，可以激活附加处理器且当您的需要增加时仅支付新的处理功率。可以增加处理器容量而不破坏任何一个当前操作。

使用本主题来了解 CUoD 如何工作以及需要执行什么操作来利用它。

#### 『按需升级容量概念』

此处开始概念上理解 CUoD。

#### 第 12 页的『准备按需升级容量』

考虑重要的计划和准备需求。

#### 第 14 页的『订购按需升级容量』

必须订购一个激活功能部件才能接收将允许您激活备用处理器的激活代码。

#### 第 15 页的『激活备用处理器』

使用此信息来激活 iSeries 服务器上的一个或多个备用处理器。

#### 第 17 页的『方案：按需升级容量』

使用此信息以获取管理员计划、订购和激活附加容量可能完成的步骤的示例。

## 按需升级容量概念

按需升级容量 (CUoD) 在免费的 14 天试用基础上或通过购买永久处理器激活提供了激活“选择 iSeries 型号”中的附加处理器的能力。此能力通过启用快速而经济的方式来为新的工作量添加容量，同时使您的服务器适应意外的性能要求而为 iSeries 服务器添加有意义的值。

选择 iSeries 型号附带许多**启动处理器**。启动处理器是在交付 iSeries 服务器时在服务器中已经活动的处理器。**备用处理器**是服务器附带的处理器，但在激活它们之后，才可供使用。可以在 14 天试用基础上临时激活备用处理器或通过购买激活功能部件并输入提供的激活代码永久地激活备用处理器。下表列示每个型号可用的启动和备用处理器数目。

型号	处理器功能部件	启动处理器数	备用处理器数	已安装的处理器数
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

\* 有限可用性

以下概念提供了要利用 CUoD 所需要的信息。

『**按需升级容量试用期**』

利用免费的 14 天试用期来测量附加处理器容量的好处。

『**按需升级容量的软件发放许可证注意事项**』

阅读有关激活 CUoD 如何影响软件层的信息。

第 12 页的『**按需处理器激活代码**』

一旦您已决定永久激活某些或所有的备用处理器，就将需要订购和购买一个或多个按需处理器激活功能部件。这将提供将允许您激活备用处理器的激活代码。

第 12 页的『**Electronic Service Agent 需求**』

使用 Electronic Service Agent 以便快速而准确地传送激活 CUoD 备用处理器所需要的重要产品数据。

**按需升级容量试用期:** 对于希望免费评估备用处理器的使用的客户，可获得按需升级容量的 14 天的试用期。一旦已激活，试用期就在初始启动服务器后可供一次性使用，并且在每次输入激活代码时也可供一次性使用。

一旦已启动，试用期就在 14 个开机日的时间段内可用。仅当服务器打开时，试用期时钟才开始滴答作响。在试用期期间，激活所有处理器（包括启动和备用处理器两者）。不可能停止和重新启动试用期。

参见第 15 页的『在试用期内激活备用处理器』以试验此功能部件。

**按需升级容量的软件发放许可证注意事项:** 许多商业伙伴使用服务器上可用的处理器功能部件代码系统值来设置软件发放许可证费用以获取软件。此系统值在具有备用处理器的服务器上继续可用，但应注意处理器功能部件代码系统值保持相同，而不论激活多少备用处理器。

将软件发放许可证费用基于服务器上的处理器数的软件供应商过去使用“软”兼容方法。对于每个处理器激活，通知和支付所需软件发放许可证费用仍然是客户的责任，这些费用与驻留在按处理器数发放了软件许可证的服务器上的任何软件关联。

激活处理器不会更改软件层。

**注：** 当通过其中一个 IBM 配置器订购按需要处理器激活功能部件时，对当前安装在按处理器发放许可证的客户服务器上的软件产品收取 IBM 软件发放许可证的附加费用。

**按需处理器激活代码：** 当您已决定永久激活某些或所有的备用处理器，就需要订购和购买一个或多个按需处理器激活功能部件。

在下了订单后，订单记录与从服务器收集的重要产品数据（VPD）组合在一起。此信息用来生成特定于服务器的 POD 激活代码。可以将它看作容量许可证密钥以便您用于激活备用处理器。


此激活代码将公布在 IBM Web 站点以用于快速访问。这通常发生在一个商业日（24 小时）内，假定此订单附带从服务器收集的所需的 VPD。一旦已经生成激活代码，就可以使用系统类型和序列号在 iSeries 按需要升级容量 Web 站点上访问它：

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

有关如何订购 POD 激活功能部件和接收 POD 激活代码的指示信息，参见第 14 页的『订购按需升级容量』。

**Electronic Service Agent 需求：** 当您下订单来购买按需要升级容量时，IBM 将订单信息用户服务器上的重要产品数据（VPD）组合在一起，以创建解锁和激活服务器中的备用处理器所需要的激活代码。

可以使用 Electronic Service Agent（它是超级支持的一部分，IBM 的全面技术服务）以电子方式将 VPD 信息发送至 IBM，该信息支持对 IBM iSeries 的主动排除。Electronic Service Agent 是驻留在服务器上的非收费许可程序（5798-RZG），并且是针对监控事件以及按周期性的客户可定义时间表来将服务器库存信息发送给 IBM 而设计的。

有关 Electronic Service Agent 的完整文档（包括安装信息），参见 Electronic Service Agent for iSeries User's Guide 。

## 准备按需升级容量

按需升级容量允许您添加处理器容量而不中断服务器操作。然而，要无缝集成新的容量，应该在下订单前准备服务器。

### 第 13 页的『按需升级容量的容量计划』

此主题包括对支持 CUoD 的服务器型号进行容量计划的注意事项和资源。

### 第 13 页的『确定何时激活处理器』

此主题讨论可以监控资源利用率方面的趋势的工具和何时可能需要附加容量的建议，以及工具如何报告具有备用处理器的服务器中的 CPU 利用率。

### 第 13 页的『为按需升级容量设置环境』

准备服务器以集成新的处理器容量，并就绪在需要附加容量时订购附加容量。

**按需升级容量的容量计划:** 具有备用处理器的服务器的容量计划本质上使用与用于缩放其它 iSeries 型号的相同过程和资源。已更新可用来帮助确定所需的服务器容量的一组综合的工具、资源和提供, 以支持具有备用处理器的服务器。

有关使用容量计划的帮助, 参考下列资源:

**第 9 页的『确定何时和如何扩充系统』**

本主题概述讨论几个可用来帮助您进行容量计划并标识资源利用率方面的趋势的工具。

**iSeries Benchmark Center** 

使用此 IBM Web 站点以获取基准测量应用程序环境的帮助。

**iSeries Solutions Center — Capacity Planning Services** 

此 IBM 咨询服务可以帮助您计划满足商业增长需要的服务器解决方案。

**注:** 工作量估计器支持 CUoD, 但仅返回对建议的活动处理器数的估计。在建议中未表示备用处理器提供的附加容量。

**确定何时激活处理器:** 按需升级容量提供了当工作量需要附加资源时将处理器添加至系统的能力。应该监控系统 CPU 利用率和 CPU 利用率方面的趋势, 以确定何时激活附加处理器, 以及将需要多少处理器。

有许多可用来报告 CPU 利用率信息的性能工具。特别地, 第 67 页的『Performance Management/400』可以标识资源利用率方面的趋势, 而第 60 页的『iSeries 导航器监控器』可以给出有关正在如何使用资源的更详细的信息, 并提醒您利用率何时达到预定义的临界级别。

### 测量具有备用处理器的服务器的 CPU 利用率

当计算所有可用处理器的平均使用情况时, 报告 CPU 利用率的系统功能在 CPU 容量的总量中不包括备用处理器。在报告 CPU 利用率百分比的各种系统功能中, 不认为备用处理器是活动的。已使用的 CPU 容量的百分比 (在 iSeries 导航器中是 CPU 利用率 CPU 百分比) 是以在过去的时间内处理器活动的时间量为基础计算的度量。这通常报告为百分比, 其中 100% 表示在整个过去的时间内处理器忙。当存在多个处理器时, 必须调整 CPU 时间以表示所有处理器的平均使用情况, 以便利用率始终报告为总的可用容量的百分比。

交互式能力是由购买的交互式功能部件确定的。此能力不受备用处理器数的影响并不更改备用处理器何时成为活动的。报告为交互式能力的百分比的交互式利用率不受按需升级容量技术的影响。在 iSeries 导航器中, 此度量称为 CPU 利用率 (交互式功能部件)。

交互式 CPU 利用率也报告为总的系统 CPU 的百分比。在中央管理中, 此度量称为 CPU 利用率 (交互式作业)。此度量影响具有 CUoD 的服务器的方式与以上描述的系统 CPU 利用率相同。

**为按需升级容量设置环境:** 在订购任何按需处理器 (POD) 激活代码前, 应该准备环境以便订购和集成附加容量。

### 准备订购

在购买 POD 激活代码时, 还必须提供服务器的重要产品数据 (VPD)。可以通过传真以手工方式或通过使用 Electronic Service Agent 以电子方式执行此操作。使用 Electronic Service Agent 来发送 VPD 数据避免了与手工方法关联的延迟, 并应该导致在接收到订单的 24 小时内公布 POD 激活代码。当根据试验正在使用附加处理器时, 应该允许在发送和处理订单所需要的时间, 以避免在性能方面的中断。

要使用发送 VPD 数据的电子方法, 第 14 页的『设置 Electronic Service Agent』。

## 准备附加容量

为了确保服务器能力完全利用已激活的处理器，可能需要进行下列准备：

- 执行任何 I/O 调节
- 执行内存升级
- 执行任何逻辑分区（LPAR）

对于使用 LPAR 的服务器，备用处理器始终与主分区关联。必须将所有处理器指定给一个分区。

设置 *Electronic Service Agent*： 可以使用 *Electronic Service Agent* 来发送 VPD 数据。要设置 *Electronic Service Agent*，执行下列步骤：

1. 安装 *Electronic Service Agent*。

参考 *Electronic Service Agent for iSeries User's Guide*  以获取设置和安装指示信息。

2. 确保已设置 TCP/IP 并已启动。

有关 *Electronic Service Agent* 的更多信息，参考 第 12 页的『*Electronic Service Agent* 需求』。

## 订购按需升级容量

可以为新服务器、型号升级或已安装的服务器订购按需处理器（POD）激活功能部件。对于新的服务器或型号升级，订单可以包含一个或多个 POD 激活功能部件。在此情况下，POD 激活代码是在将服务器交付给您之前输入的。

对于已安装的服务器，一旦您确定要永久激活某些或所有备用处理器，就将需要订购一个或多个 POD 激活功能部件，然后使用产生的 POD 激活代码来激活备用处理器。

### 注意事项：

1. 可能需要几天来处理订单并公布产生的 POD 激活代码。当正在完成附加容量的永久激活的订单时，可以第 15 页的『在试用期内激活备用处理器』（14 天）以满足工作量需求。
2. 如果订单中没有包括各种功能部件，则可以更快地处理 POD 激活功能部件的订单。

要订购一个或多个 POD 激活功能部件，执行以下步骤：

1. 确定要激活的备用处理器数。  
有关信息，参见第 13 页的『确定何时激活处理器』主题。
2. 与 IBM 商业伙伴或 IBM 销售代表联系以便下订单以获取一个或多个 POD 激活功能部件。位于美国的客户可以选择在 IBM Web 站点上下订单：

<http://www.ibm.com> 

3. 『将重要产品数据发送至 IBM』。  
在处理订单前，将订单与从服务器收集的重要产品数据（VPD）组合在一起。该信息然后用来生成特定于服务器的 POD 激活代码。将该 POD 激活代码邮寄给您并公布在 *iSeries* 按需升级容量 Web 站点上以便快速访问。
4. 在服务器上输入 POD 激活代码以第 16 页的『永久激活备用处理器』。

**将重要产品数据发送至 IBM：** 当订购一个或多个按需处理器（POD）激活功能部件时，必须向 IBM 提供您的 *iSeries* 服务器的重要产品数据（VPD）。可以通过使用 *Electronic Service Agent*<sup>(TM)</sup> 或通过打印并传真数据将您的 VPD 收集并发送至 IBM。

要通过使用 Electronic Service Agent 收集并发送 VPD，执行以下步骤：

1. 确保第 14 页的『设置 Electronic Service Agent』。
2. 启动 Electronic Service Agent 向导。
3. 选择**硬件**以收集和发送所需的 VPD 信息。

要打印和传真 VPD，执行下列步骤：

1. 从系统控制台，在命令行输入 STRSST 以启动系统服务工具（SST）。注册至 SST。

**注：**要使用系统服务工具，需要具有系统容量 — 管理员特权的有效服务工具用户标识。

2. 选择选项 6（使用系统容量）并按执行键。
3. 选择选项 1（显示系统容量信息）并按执行键。出现“显示永久系统容量”屏幕。
4. 按 F6 来打印信息。
5. 准备具有下列信息的传真文档：

- 传真目的地信息：
  - 发送给：按需容量管理员（507-253-7019）
  - 传真号码：507-253-4553
  - 位置：Rochester, Minnesota
- 传真源信息：
  - 客户名称：
  - 客户联系人名称：
  - 客户地址：
  - 客户电话号码：
  - 客户传真号码：

6. 将 VPD 传真至下列传真号码：  
507-253-4553

## 激活备用处理器

可以在试用或永久基础上激活已经安装在服务器上的一个或多个备用处理器。

### 『在试用期内激活备用处理器』

可以在 14 天的试用期内激活安装在服务器上的所有备用处理器。当将要完成订购附加容量的永久激活时，这允许您在决定是购买附加容量还是满足高峰工作量需求前，评估附加容量可以如何有助于您的环境。

### 第 16 页的『永久激活备用处理器』

可以永久激活指定数目的备用处理器。

**在试用期内激活备用处理器：**要在试用期内激活安装在服务器上的所有备用处理器，执行以下步骤：

1. 从系统控制台，在命令行输入 STRSST 以启动系统服务工具（SST）。注册至 SST。

**注：**要使用系统服务工具，需要具有系统容量 — 管理员特权的有效服务工具用户标识。

2. 选择选项 6（使用系统容量）并按执行键。出现“使用系统容量”屏幕。
3. 选择选项 3（使用临时系统容量）并按执行键。出现“确认启动临时系统容量激活”屏幕。

**注：**如果临时系统容量激活正在使用中或已经使用，则将不出现此选项。

4. 按“执行”键以确认系统容量激活。

5. 退出“使用系统容量”屏幕。
6. 如果服务器已分区，则必须将新的可用处理器分配给逻辑分区之后，才能使用新的容量。  
有关将激活的处理器分配给逻辑分区的信息，参见处理能力的动态移动。
7. 如果服务器未分区，则执行以下步骤：
  - a. 选择选项 5（使用系统分区）并按执行键。
  - b. 选择选项 3（使用分区配置）并按执行键。
  - c. 在 PRIMARY 旁边输入 2（更改分区处理资源）并按执行键。
  - d. 对新的处理器数输入一个值，它代表活动处理器的总数。  
**注：**屏幕底部的配置错误指示已变更缺省主分区（正在主动使用逻辑分区），并且必须参考您的逻辑分区计划以正确将激活的处理器分配给服务器。
  - e. 按“执行”键以确认更改。
8. 退出 SST。


现在可以开始使用新的容量。

**注：**

1. 在 14 天过去后或当输入新的按需要处理器（POD）激活代码时，试用自动结束。
2. 如果服务器已分区，则必须在试用期结束时将备用处理器返回至主分区。

**永久激活备用处理器：** 在购买了一个或多个按需要的处理器（POD）激活功能部件后，将接收到一个 POD 激活代码以激活备用处理器。

要永久激活某些或所有备用处理器，执行以下步骤：

1. 检索 POD 激活代码，如下所示：
  - a. 访问 iSeries 按需要容量升级 Web 站点：  
<http://www.ibm.com/servers/eserver/series/hardware/ondemand> 
  - b. 输入服务器的系统类型和序列号。
  - c. 记录显示在 Web 站点上的 POD 激活代码。
2. 在命令行输入 STRSST 以启动系统服务工具（SST）。注册至 SST。

**注：**要使用系统服务工具，需要具有系统容量 — 管理员特权的有效服务工具用户标识。

3. 选择选项 6（使用系统容量）并按执行键。
4. 选择选项 2（激活永久系统容量）并按执行键。
5. 在按需要处理器激活代码字段中输入您的 POD 激活代码并按执行键。出现“确认激活系统容量”屏幕。
6. 按“执行”键以确认系统容量激活。
7. 退出“使用系统容量”屏幕。
8. 如果服务器已分区，则必须将新的可用处理器分配给逻辑分区之后，才能使用新的容量。  
有关将激活的处理器分配给逻辑分区的信息，参见处理能力的动态移动。
9. 如果服务器未分区，则执行以下步骤：
  - a. 选择选项 5（使用系统分区）并按执行键。
  - b. 选择选项 3（使用分区配置）并按执行键。
  - c. 在 PRIMARY 旁边输入 2（更改分区处理资源）并按执行键。



- d. 对新的处理器数输入一个值，它代表活动处理器的总数。

**注：**屏幕底部的配置错误指示已变更缺省主分区（正在主动使用逻辑分区），并且必须参考您的逻辑分区计划以正确将激活的处理器分配给服务器。

- e. 按“执行”键以确认更改。

## 10. 退出 SST。

**注：**当输入 POD 激活代码后，备用处理器立即成为活动的。然而，服务器必须运行 15 分钟后才能存储 POD 激活代码。如果输入 POD 激活代码并在服务器运行了 15 分钟前关闭服务器，则在启动服务器时可能需要重新输入 POD 激活代码。

现在可以开始使用新的容量。

## 方案：按需升级容量

按需升级容量允许客户在工作量需要备用处理器时激活备用处理器。下列方案完成计划、订购和调节此功能部件的步骤。

1. 具有功能部件代码 2416 的型号 840 服务器正使用 8 个活动处理器和 4 个备用处理器操作。当服务器工作量增长时，可用的 CPU 资源利用率始终接近或超过可用容量的 70%。预期需要附加资源，管理员决定考虑激活某些备用处理器。
2. 在激活任何一个处理器之前，管理员第 12 页的『准备按需升级容量』。这包括执行趋势分析以了解将需要多少附加处理器，准备服务器以调节附加处理器，以及准备订购新的容量。
3. 要调查激活附加处理器的好处，管理员决定第 15 页的『在试用期内激活备用处理器』。试用期持续 14 天。
4. 在决定通过激活附加处理器获得的性能改进保证永久购买处理器之后，管理员与 IBM 销售代表或商业伙伴联系，或访问 [www.ibm.com](http://www.ibm.com) 来下订单以获取 4 个按需处理器（POD）激活代码。
5. IBM 销售代表在 IBM 配置器中下订单，并接收到提醒单以从具有订单的服务器第 14 页的『将重要产品数据发送至 IBM』。可以将 VPD 数据传真至 IBM 或使用 Electronic Service Agent 以电子方式发送至 IBM。
6. 管理员从 Web 接收 POD 激活代码，并第 16 页的『永久激活备用处理器』。这涉及在目标服务器中输入 POD 激活代码并将处理器指定一个逻辑分区。

型号 840 现在所有 8 个处理器都可供使用。

## 选择性能管理策略

开发合适的性能管理策略将有助于管理系统的性能。性能管理策略很大程度上取决于您能够花费在管理性能上的时间量。如果您正在管理一个小公司，您可能要管理企业的很多不同方面，而不能投入很多时间来管理性能。很多大公司雇用性能专家来不断调整系统并使之高效运行。

为了确定基本的性能管理策略和标识要使用哪些性能应用程序，以三种类别之一来分类您的公司：小企业、中等规模企业和大企业。企业资源随其规模而改变，您的管理策略将相应改变。

### 小企业

小企业很可能比大企业有较少的资源专用于管理性能。因为该原因，应尽可能使用更多的自动操作。使用 PM/400 将性能数据直接发送给 IBM，IBM 将编译该数据并为您生成报告。这不仅节省时间，当您的 iSeries 服务器需要升级时，IBM 还会给您提供建议。

下面是对于小企业的建议的性能应用程序的列表：

第 33 页的『收集服务』

以用户定义的时间间隔收集采样数据用于以后分析。

第 66 页的『图形历史记录』  
显示用“收集服务”收集的性能数据。

第 67 页的『Performance Management/400』  
使系统性能数据的收集、归档和分析自动化。

第 81 页的『Performance Tools』  
收集、分析和维护系统性能信息。

第 60 页的『iSeries 导航器监控器』  
观察 iSeries 系统性能的图形表示，自动响应预定义的事件或情况。

## 中等规模的企业

中等规模的企业与小企业相比可能将更多的资源专用于管理性能。您可能仍想尽可能使用更多的自动操作，也可以从使用 PM/400 中得益。

下面是对于中等规模企业的建议的性能应用程序的列表：

第 33 页的『收集服务』  
以用户定义的时间间隔收集采样数据用于以后分析。

第 66 页的『图形历史记录』  
显示用“收集服务”收集的性能数据。

第 67 页的『Performance Management/400』  
使系统性能数据的收集、归档和分析自动化。

第 81 页的『Performance Tools』  
收集、分析和维护系统性能信息。

第 60 页的『iSeries 导航器监控器』  
观察 iSeries 系统性能的图形表示，自动响应预定义的事件或情况。

第 86 页的『性能资源管理器』  
收集有关特定应用程序或系统资源的详细信息。

## 大企业

大企业将一些资源专用于管理性能。

下面是对于大企业的建议的性能应用程序的列表：

第 33 页的『收集服务』  
以用户定义的时间间隔收集采样数据用于以后分析。

第 66 页的『图形历史记录』  
显示用“收集服务”收集的性能数据。

第 67 页的『Performance Management/400』  
使系统性能数据的收集、归档和分析自动化。

第 81 页的『Performance Tools』  
收集、分析和维护系统性能信息。

第 60 页的『iSeries 导航器监控器』

观察 iSeries 系统性能的图形表示，自动响应预定义的事件或情况。

第 86 页的『性能资源管理器』

收集有关特定应用程序或系统资源的详细信息。

第 93 页的『iDoctor for iSeries』

分析跟踪数据以提高系统和应用程序的性能。

第 94 页的『Performance Trace Data Visualizer (PTDV)』

查看来自 Java 应用程序的跟踪数据。

---

## 设置管理性能的环境

iSeries 服务器包括若干工具，它们定期收集系统性能数据并监控系统的性能趋势和潜在问题。您的独特需求和环境将确定您选择投资的工具和应选择的配置。有效地设置系统将允许您在系统增长时进行精确的容量计划以及在发生性能问题时解决它们。

使用下列主题来了解和配置将收集、监控并分析系统性能的工具。

### 第 33 页的『收集服务』

“收集服务”管理系统性能数据的日常的收集。此工具定期收集数据并创建称为收集对象的归档。可以通过某些工具直接访问这些收集对象，也可以使用自己的定制查询或通过其它工具和报告将它们转换为数据库文件集进行分析。因为“收集服务”主要提供其它应用程序的数据，所以您正在使用的其它工具将对您的配置选择有重要的影响，包括收集数据的频率、收集的数据类型以及将数据保持在系统上的时间长度。

### 第 67 页的『Performance Management/400』

PM/400 使用“收集服务”来收集非专利性能数据，并将它发送至 IBM 以存储和进行高级分析。这使您不需要自己存储和维护该数据。然后可以使用 Web 浏览器来访问关于系统性能的详细报告和建议。

### 第 60 页的『iSeries 导航器监控器』

包括在 iSeries 导航器中的监控器使用“收集服务”数据来跟踪您特别关心的系统性能的元素。而且，当发生某些事件（如 CPU 利用率的百分比或作业的状态）发生时，它们可以执行指定的操作。使用此主题来了解如何使用这些监控器和如何在系统上设置它们。

---

## 管理 iSeries 性能

成功管理性能可确保系统有效使用资源并确保 iSeries 服务器为您的用户和商业需要提供可能的最佳服务。而且，有效的性能管理可以帮助您快速响应系统中的更改并可以通过推迟费用高的升级和服务费用节约资金。

了解影响系统性能的因子有助于您响应问题并作出更好的长期计划。有效的计划可以防止可能的性能问题发展，并确保您具有系统容量以处理当前和增长的工作量。

使用下列主题来了解如何维护系统的性能和解决性能问题。

### 第 20 页的『跟踪性能』

随时间跟踪系统性能允许您计划系统的增长并确保您具有数据以帮助找出和标识性能问题的原因。了解要使用哪些应用程序和如何定期收集性能数据。

### 『研究性能问题』

有许多选项可用来帮助您标识和解决性能问题。了解如何使用可以帮助您查找性能问题源的可用的工具和报告。

### 第 25 页的『显示性能数据』

在收集性能数据后，了解如何使用对于您的目的最合适的工具来显示数据。

### 第 25 页的『调整性能』

当已标识性能问题时，将需要调整系统以修正它。

### 第 27 页的『管理电子商务性能』

管理电子商务环境中的性能为 iSeries 管理员介绍了几个新问题。参考此主题以获取帮助您计划、跟踪和改进电子商务应用程序的性能的信息和资源。

## 跟踪性能

跟踪 iSeries 服务器的系统性能可以帮助您标识趋势，趋势可以帮助您调整系统配置并作出关于何时和如何升级系统的最佳选择。此外，当问题发生时，具有故障前后的性能数据以减少性能问题的原因并查找适当的分辨率是重要的。

iSeries 服务器包括几个应用程序，它们用于跟踪性能趋势和维护 iSeries 性能数据的历史记录。这些应用程序的大多数使用“收集服务”收集的数据。可以使用“收集服务”来监控以下方面的趋势：

- 系统资源利用率的趋势。可以使用此信息来计划系统配置更改和升级并特别定制它们。
- 配置的物理组件上的应力的标识。
- 高峰和正常使用期间，交互式作业和批处理作业对系统资源的使用之间的平衡。
- 配置更改。可以使用“收集服务”数据来准确地预测类似于添加用户组、增加的交互式作业的更改和其它更改的影响。
- 可能导致系统上其它活动出现问题的作业的标识

以下工具将帮助您实时监控系统性能：

### 第 33 页的『收集服务』

“收集服务”以用户定义的时间间隔收集性能数据，然后将此信息存储在系统上的收集对象中。许多其它工具（包括监控器、图形历史记录、PM/400 和 Performance Tools 许可程序中的许多功能）依赖这些收集对象来获取数据。

### 第 66 页的『图形历史记录』

图形历史记录显示用“收集服务”收集的的性能数据指定的时间段通过图形用户界面（GUI）。可用于显示的时间长度取决于您要保留收集对象多长时间以及您是否在使用 PM/400。

### 第 67 页的『Performance Management/400』

PM/400 使系统性能数据的收集、归档和分析自动化，并返回清楚的报告来帮助您管理系统资源和容量。

## 研究性能问题

收集或分析性能的大多数工具都使用跟踪或采样数据。“收集服务”定期地收集各种系统资源的采样数据。几个工具对此采样数据进行分析或报告，您可以使用此数据来获取对系统资源利用率的更广泛的了解并回答许多公共性能问题。有关更详细的性能信息，几个工具会生成跟踪级别数据。通常情况下，跟踪级别数据可以提供关于系统上作业和应用程序的行为和资源消耗的详细信息。“性能资源管理器”和 STRPFRTRC 命令是用于生成跟踪数据的两个公共工具。

例如，如果系统运行速度较慢，可以使用系统监控器来查找问题。如果您发现 CPU 利用率太高，可以标识似乎在使用异常多的资源的任何作业。然后，通过进行配置更改，也许能够更正问题。然而，某些问题将需要附加信息。要获取关于作业性能的详细信息，可以启动性能资源管理器会话，收集关于作业在 iSeries 系统上的行为的详细信息并对起源程序进行可能的更改。

要更多地了解如何收集性能数据，使用下列主题来了解如何和何时使用某些性能管理应用程序。

#### 『标识性能问题』

了解涉及标识性能问题的公共步骤。

#### 『标识和解决公共性能问题』

许多不同的性能问题经常影响 iSeries 系统的公共区域。了解如何研究和解决公共区域中的问题。

#### 第 22 页的『收集系统性能数据』

“收集服务”定期收集关于系统性能的信息。分析性能数据通常从此信息开始。

#### 第 23 页的『收集有关系统资源利用率的信息』

几个工具监控正在如何使用资源，如 CPU、磁盘空间、交互式容量和许多其它元素。可以使用这些工具来开始标识问题区域。

#### 第 23 页的『收集有关应用程序的性能的信息』

由于各种原因，应用程序的运行速度可能较慢。可以使用包括在 OS/400 的和其它许可程序中的几个工具来帮助您获取更多信息。

#### 第 24 页的『方案：在升级或迁移后改进系统性能』

在此方案中，您刚刚已升级或迁移系统，它现在似乎运行地比以前慢。此方案将帮助您标识并修正性能问题。

## 标识性能问题

当尝试标识性能问题时，评估硬件配置是否足够支持工作量非常重要。有足够的 CPU 容量吗？有足够的主存储器用于不同类型的应用程序吗？首先回答这些问题（也许通过容量建模技术）可以避免以后不必要的工作。

了解问题的症状和要达到的目标后，分析员可以用公式来表示可以解释问题原因的假设。分析员可以使用 OS/400 操作系统和 Performance Tools 许可程序提供的命令和工具来测量系统性能。

复查测量的数据可以帮助进一步定义问题并帮助验证或推翻假设。一旦找到明显的原因，就可以提出解决方案。当一次处理一个解决方案时，可以重新设计并测试程序。在许多情况下，分析员的工具可以再次测量解决方案的有效性并找出可能的副作用。

要达到最佳性能，必须考虑关键系统资源之间的关系，并尝试平衡这些资源（即 CPU、磁盘、主存储器和用于通信的远程线路）。这些资源的每一个都可以导致性能下降。

可以通过许多形式来提高系统性能（无论是交互式吞吐量、交互式响应时间、批处理吞吐量还是它们的某些组合），从仅仅调整活动级别或池大小至更改应用程序代码本身。在此实例中，活动级别是子系统的一个特征，它指定可以同时争用处理单元的最大作业数。

## 标识和解决公共性能问题

当 iSeries 服务器上发生性能问题时，它们经常首先影响系统的某些区域。参考下表以获取可用于研究这些系统区域上的性能的某些方法。其中许多区域可用作第 60 页的『iSeries 导航器监控器』。然而，有几种其它方法来访问关于这些区域的信息。

区域	描述	可用工具
处理器负载	确定系统上是否存在太多作业或某些作业是否在使用较大百分比的处理器时间。	<ul style="list-style-type: none"> <li>使用活动作业 (WRKACTJOB) 命令。</li> <li>使用系统活动 (WRKSYSACT) 命令, 它是 Performance Tools 许可程序的一部分。</li> <li>iSeries 导航器中的工作管理功能。</li> <li>iSeries 导航器系统监控器内的 CPU 利用率度量。</li> </ul>
主存储器	调查故障和“等待至不合格”转换。	<ul style="list-style-type: none"> <li>使用磁盘状态 (WRKDSKSTS) 命令</li> <li>iSeries 导航器系统监控器内的磁盘存储器度量</li> <li>使用系统状态 (WRKSYSSTS) 命令</li> <li>iSeries 导航器中“工作管理”下的“内存池”功能。</li> </ul>
磁盘	确定臂是否太少和臂是否太慢。	<ul style="list-style-type: none"> <li>使用磁盘状态 (WRKDSKSTS) 命令。</li> <li>iSeries 导航器系统监控器内的磁盘臂利用率度量。</li> <li>“Performance Tools 系统和组件”报告。</li> </ul>
通信	查找速度较慢的线路、线路上的错误或线路上是否有太多用户。	<ul style="list-style-type: none"> <li>Performance Tools 组件报告。</li> <li>iSeries 导航器系统监控器内的 LAN 利用率度量。</li> </ul>
IOP	确定是否有任何 IOP 不平衡或是否没有足够的 IOP。	<ul style="list-style-type: none"> <li>Performance Tools 组件报告。</li> <li>iSeries 导航器系统监控器内的 IOP 利用率度量。</li> </ul>
软件	调查锁定和互斥。	<ul style="list-style-type: none"> <li>Performance Tools 锁定报告</li> <li>Performance Tools 跟踪报告</li> <li>使用对象锁定 (WRKOBJLCK) 命令。</li> <li>在 iSeries 导航器中, 右键单击“作业管理”下的可疑作业, 并选择详细信息 → 已锁定的对象。</li> </ul>

## 收集系统性能数据

收集数据对于改善性能是一个重要步骤。在收集性能数据时, 收集有关服务器的信息, 该信息可以用来了解响应时间和吞吐量。一种方法是捕获在完成工作时涉及到的服务器或一组服务器的性能状态。数据收集为以后可以执行的任何比较和分析提供了一个上下文或一个起点。当使用第一个数据收集时, 具有一个将来改进的基准和一个今天改善性能的起点。可以使用收集的性能数据来进行调整、改善响应时间以及帮助系统获取最佳性能。性能问题分析常常从简单问题开始: “更改了什么?” 性能数据帮助您回答该问题。

可以使用收集服务来收集性能数据, 使用创建性能数据 (CRTPFRTA) 命令创建性能文件, 使用转换性能数据 (CVTPFRDTA) 命令或通过 iSeries 中的 Performance Tools 插件将它们转换为当前发行版, 然后通过使用性能数据库文件中的信息创建报告或创建您自己的查询。

有关性能数据的更多信息, 参见下列内容:

### 第 33 页的『收集服务』

查看如何收集性能数据以用于分析和如何定制收集。

### 第 55 页的『性能数据文件』

查找对您可用的性能数据库文件的概述并查看每个性能数据库文件的详细字段数据。

另外，可以使用性能管理 API 来启动、结束和循环收集，以及为已收集的数据更改和检索系统参数。有关其它 iSeries 性能工具和技术的更多信息，参见第 1 页的『性能』页面。

## 收集有关系统资源利用率的信息

许多工具可用来帮助您监控和跟踪 iSeries 服务器和您的应用程序正在使用可用资源的方式。可以使用此信息作为问题分析的起点并标识一些趋势，这些趋势将帮助您进行容量计划和管理系统增长。

参见下列主题以了解如何使用这些工具和何时使用它们：

### 第 60 页的『iSeries 导航器监控器』

包括在 iSeries 导航器中的监控器以广泛的各种度量提供当前和最近的数据。另外，可以将它们配置为在某些事件发生时执行指定的操作。

### 第 94 页的『“使用”有关 OS/400 性能的命令』

OS/400 包括几个重要功能以帮助您管理和维护系统性能。

### 第 67 页的『Performance Management/400』

PM/400 使用收集服务来收集非专有的性能数据并将它发送至 IBM 进行存储和专家分析。这就消除了需要您自己存储和维护它。然后可以使用 web 浏览器访问有关系统性能和趋势分析的详细报告和建议。

## 收集有关应用程序的性能的信息

收集有关应用程序的性能的信息与收集有关系统性能的信息有很大不同。收集应用程序信息只能使用某些性能应用程序完成，如 Performance Explorer、Performance Trace Data Visualizer (PTDV) 和 iDoctor。或者，可以通过使用第 60 页的『iSeries 导航器监控器』跟踪个别服务器性能以及使用第 81 页的『Performance Tools』跟踪和分析服务器作业来获取应用程序性能的概述。

**注：**收集应用程序的性能数据可能会显著影响系统的性能。在开始收集前，确保已尝试所有其它收集选项。

### 第 86 页的『性能资源管理器』

此工具帮助查找通过使用执行一般性能监控的工具不能标识的性能问题的原因。当计算机环境在大小和复杂性两方面都增长时，最好在复杂性方面获取性能分析是合理的。性能资源管理器通过对复杂的性能问题收集数据来解决复杂性方面的增长。

性能资源管理器是为对了解或改进程序性能感兴趣的应用程序开发者设计的。它还对于很了解性能管理的用户有用，以帮助标识和隔离复杂的性能问题。

### 第 94 页的『Performance Trace Data Visualizer (PTDV)』

此工具是一个 Java 应用程序，它可以用于正在 iSeries 中运行的应用程序的性能分析。PTDV 与 OS/400 操作系统中的性能资源管理器一起工作，以允许分析员查看程序流并获取由跟踪、作业、线程和过程总结的详细信息（如 CPU 时间、当前系统时间、循环次数和指令数）。当目视 Java 应用程序跟踪时，可以显示附加详细信息（如创建的对象的数量和类型）和有关 Java 锁定行为的信息。还支持由 WebSphere Application Server 生成的性能资源管理器事件。PTDV 允许排序列、导出数据和将许多级别的数据进行汇总。

有关更多信息，请转至 Performance Trace Data Visualizer  Web 站点。

### 第 93 页的『iDoctor for iSeries』

iDoctor 中的性能资源管理器分析器功能包括一个特定适合于分析跟踪数据以改进系统和应用程序性能的软件工具。此详细分析给出有关磁盘操作、CPU 利用率、文件打开操作、机器接口 (MI) 程序、等待状态、磁盘空间消耗等等的低级别摘要。客户机组件是一个 iSeries 导航器插件，它允许用户精简并以图形方式显示 iSeries 跟踪数据。

## 启动性能跟踪 (STRPFRTRC) 命令

OS/400 包括一个收集多个编程和事务数据的命令。此命令收集在先前发行版中 STRPFRMON 收集的数据。在运行此命令后，可以使用『转储跟踪数据』将数据导出至数据库文件。

**转储跟踪数据：** 因为转储会影响系统性能，所以决定何时转储跟踪数据是一个重要的决定。转储跟踪 (DMPTRC) 命令将信息从内部跟踪表放入数据库文件中。在装入的系统中或在高优先级 (交互式) 作业内，在高峰活动期间转储跟踪数据不太好。可以延迟跟踪转储，但在忘记数据存在之前要转储数据。如果由于任何原因清除了跟踪表，则会丢失跟踪数据。然而，稍微延迟转储然后使用 DMPTRC 命令来在批处理作业中转储跟踪可以保持用户的性能。

要转储跟踪数据，发出下列命令：

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

必须指定要存储数据的成员名和库名。可以在收集跟踪信息的同时，使用第 33 页的『收集服务』收集基于采样的数据。当像这样将采样数据和跟踪数据收集在一起时，应该将数据放入一致命名的成员中。换句话说，在 CRTPFRTA TOMBR 和 TOLIB 参数中提供的名称应该与在 DMPTRC MBR 和 LIB 参数中提供的名称相同。

## 方案：在升级或迁移后改进系统性能

最近已将 iSeries 服务器升级为最新发行版。在完成升级并恢复正常操作后，系统性能已显著降低。您希望标识性能问题的原因并将系统恢复到正常的性能级别。

### 隔离更改

在升级操作系统后，几个问题可能会导致性能降低。可以使用包括在 OS/400 中的性能管理工具和 Performance Tools 许可程序 (5722-PT1) 来获取有关性能问题的更多信息并将怀疑的问题缩小至可能的原因。

1. 检查 CPU 利用率。有时候，在升级后作业将不能访问某些需要的资源。这可能导致单个作业消耗不可接受的 CPU 资源量。
  - 使用 WRKSYSACT、WRKSYSSTS、WRKACTJOB 或 iSeries 导航器系统监控器来查找总的 CPU 利用率。
  - 如果 CPU 利用率高 (例如，大于 90%)，则检查活动作业利用的 CPU 量。如果单个作业正在消耗大于 30% 的 CPU 资源，则它可能会丢失文件调用或对象。然后可以打听供应商 (对于供应商提供的程序)，或者打听作业的所有者或程序员以获取附加支持。
2. 使用 STRPFRTRC 命令启动性能跟踪，然后使用第 84 页的『Performance Tools 报告』来标识和更正下列可能的问题：
  - 如果对于机器池的缺页故障率高于 10 个故障 / 秒，则给予机器池更多内存，直到故障率降低到此级别以下。
  - 如果磁盘利用率大于 40%，则查看等待和服务时间。如果这些值可以接受，则可能需要减少工作量以管理优先级。
  - 如果 IOP 利用率大于 60%，则添加附加 IOP 并将某些磁盘资源指定给它。
  - 如果用户池中的缺页故障高得不可接受，则参考主题第 27 页的『自动调整性能』。
3. 运行第 84 页的『Performance Tools 报告』并参考**捕获锁冲突报告**。如果捕获或锁冲突数高，则确保将访问路径大小设置为 1TB。如果捕获或锁冲突位于用户概要文件中，并且如果引用的用户概要文件拥有许多对象，则减少该概要文件拥有的对象数。
4. 使用**任务开关**选项运行第 93 页的『iDoctor for iSeries』5 分钟。然后使用任务开关监控器分析得出的跟踪数据。标识并解决下列任何一个：
  - 等待 CPU 的作业



- 发生故障的作业
- 捕获冲突

## 显示性能数据

显示性能数据帮助您更准确地分析系统的性能。可以使用许多不同方式显示性能数据；然而，可能发现在某些情况下某些性能应用程序更合适。大多数应用程序显示使用第 33 页的『收集服务』或从性能跟踪收集到的数据。访问该数据的最佳方式取决于您是正尝试解决性能问题，正监控系统性能以便计划未来增长，还是正在标识趋势。

### 显示接近实时性能数据

使用下列工具来显示当前或很近的性能信息：

#### 第 94 页的『“使用”有关 OS/400 性能的命令』

在基本操作系统中有许多命令，它们将使您查看有关系统性能的特定区域的当前信息。

#### 第 83 页的『显示性能信息』

Performance Tools 许可程序包括 iSeries 导航器的一个插件，它显示来自收集服务收集对象的性能数据。也可以查看有关系统中作业的详细信息并打印 Performance Tools 报告。

#### 第 60 页的『iSeries 导航器监控器』

这些监控器显示许多系统元素的性能数据。监控器数据是以收集对象为基础的，并将根据收集服务中的收集时间间隔在收集数据时显示数据。

### 显示历史性能数据

使用下列工具来查看存储在系统中的数据：

#### 第 67 页的『Performance Management/400』

PM/400 自动收集、归档和分析系统性能数据并返回清晰的报告以帮助您管理系统资源和容量。

#### 第 66 页的『图形历史记录』

图形历史根据收集服务中的保留周期提供在最多一个星期内的性能数据的图形显示。使用 PM/400，图形历史可以显示更长周期内的数据收集。

## 调整性能

性能调整的主要目标是允许服务器最好地利用系统资源，并允许工作量尽可能高效地运行。性能调整是手工或自动调整系统的性能的一种方式。有许多用于调整系统的选项。每个系统环境都是唯一的，这些环境要求您观察性能并对环境作出最佳的调整；换句话说，要求您进行日常的性能监控。有关必须在性能调整之前执行的性能监控步骤的更多信息，参见第 19 页的『管理 iSeries 性能』。

IBM 还提供了这样一个工具，该工具通过减少从磁盘读取的物理 I/O 请求的数目来降低 I/O 子系统和系统响应次数。了解如何才能第 94 页的『扩展的自适应高速缓存』。

有关性能调整的更多信息，从下列主题进行选择：

#### 第 26 页的『基本性能调整』

要调整系统的性能，需要设置初始调整值，观察系统性能，复查这些值并确定要调整的内容。

#### 第 27 页的『自动调整性能』

大多数用户都将系统设置为自动进行性能调整。当交付新系统时，已将它们配置为自动调整。

## 基本性能调整

要开始调整性能，必须首先通过确定初始机器和用户池大小来设置初始调整值。然后，可以开始观察系统性能。

### 设置初始调整值

设置初始调整值包括最初配置系统池大小和活动级别以有效地调整系统所执行的步骤。初始值基于估计值的；因此，估计值可能要求当系统是活动的时候进行进一步调整。下列步骤设置初始调整值：

- 确定初始机器池大小
- 确定初始用户池大小

### 观察系统性能

要观察系统性能，可以使用“使用系统状态”（WRKSYSSTS）、“使用磁盘状态”（WRKDSKSTS）和“使用活动作业”（WRKACTJOB）命令。对于每个观察周期，应对照性能目标检查并评估系统性能的测量值。

1. 除去任何无规律的系统活动。例如，可能导致严重的性能下降的无规律的活动是交互式程序编译、通信错误恢复过程（ERP）、打开查询文件（OPNQRYF）、应用程序错误和注销活动。
2. 使用 WRKSYSSTS、WRKDSKSTS 和 WRKACTJOB 命令来显示性能数据。也可以使用 Performance Tools 命令、“使用系统活动”（WRKSYSACT）来显示性能数据。
3. 允许系统最少收集 5 分钟的数据。
4. 对照性能目标评估性能的测量值。典型的测量值包括：
  - 交互式吞吐量和响应时间，可从 WRKACTJOB 屏幕获得它们。
  - 批处理吞吐量。观察活动批处理作业的辅助输入 / 输出（AuxIO）和 CPU 百分比（CPU%）值。
  - 假脱机吞吐量。观察活动写程序的辅助输入 / 输出（AuxIO）和 CPU 百分比（CPU%）值。
5. 如果观察到不满足期望的性能数据，则基于新数据调整系统。确保执行下列操作：
  - 测量并比较所有关键的性能测量值。
  - 一次执行并评估调整一个。

### 复查性能

一旦已设置正确的调整值，应定期复查它们以确保系统继续运行良好。正在进行的调整包括观察系统性能的各方面和根据建议的准则进行调整。

要收集有意义的统计信息，应在典型级别的活动期间观察系统性能。例如，当没有作业在系统中运行时收集的统计信息对于评估系统性能没有多少价值。如果您作出了最大的努力，性能仍不满意，则应评估配置的能力。

要满足目标，考虑下列各项：

- 处理器升级
- 附加的存储设备和控制器
- 附加的主存储器
- 应用程序修订

通过应用其中一个或多个方法，应达到您的目标。如果在作出适当的努力之后，仍无法达到目标，则您应确定您的目标对于您正在进行的工作类型是否现实。

### 确定要调整的内容

如果系统性能已下降且需要调整，则参考第 20 页的『研究性能问题』来标识性能问题的根源并进行相应的更正。

## 自动调整性能

系统可以自动设置性能值来提供系统资源的有效使用。通过下列操作，可以将系统设置为自动调整系统性能：

- 调整存储池大小和活动级别
- 调整存储池调页

### 调整存储池大小和活动级别

使用 QPFRADJ 系统值来控制存储池和活动级别的自动调整。此值指示系统应在系统重新启动（IPL）时还是在重新启动之后定期调整值。

可以将系统设置为在 IPL 时调整性能和 / 或动态调整性能。

- 要将系统设置为仅在系统重新启动（IPL）时调整，在 iSeries 导航器中选择“配置和服务” —> 系统值 —> 性能。单击“内存池”选项卡并选择**自动调整内存池和活动级别下的在系统重新启动时**。这等价于将 QPFRADJ 系统值设置为 1。
- 要将系统设置为在系统重新启动（IPL）时调整存储池和在重新启动之后定期调整存储池，在 iSeries 导航器中选择“配置和服务” —> 系统值 —> 性能。单击“内存池”选项卡并选择**自动调整内存池和活动级别下的在系统重新启动时和在重新启动之后定期**。这等价于将 QPFRADJ 系统值设置为 2。
- 要将系统设置为在重新启动后定期调整存储池而不是在系统重新启动（IPL）时调整，在 iSeries 导航器中选择“配置和服务” —> 系统值 —> 性能。单击“内存池”选项卡并选择**自动调整内存池和活动级别下的在重新启动后定期**。这等价于将 QPFRADJ 系统值设置为 3。

在系统重新启动时（IPL），存储池值未复位为初始值。

### 调整存储池调页

系统提供的动态调整支持自动调整共享池的池大小和活动级别以提高系统的性能。此调整将存储器从最少使用的存储池移至将受益于更多存储器的池。此调整还设置活动级别来使池中线程数与对池分配的存储量平衡。为了调整系统，调整程序使用基于线程数计算的准则。

当动态调整生效时，自动将下列性能值更改为适当的设置：

- 机器（\*MACHINE）内存池大小（QMCHPOOL 系统值）
- 基本（\*BASE）内存池活动级别（QBASACTLVL 系统值）
- 共享池 \*INTERACT 的池大小和活动级别
- 共享池 \*SHRPOOL 的池大小和活动级别
- 共享池 \*SHRPOOL1-\*SHRPOOL60 的池大小和活动级别

当动态调整生效时（QPFRADJ 系统值设置为 2 或 3），将认为在概要文件 PSYS 下运行的作业 QPFRADJ 在系统上是活动的。

有关内存池的更多信息，参见内存池。

## 管理电子商务性能

电子商务环境中的性能对于 iSeries 系统管理员存在几个复杂问题。除对 iSeries 服务器的日常调整外，管理员还必须监控和优化硬件和服务，以支持电子商务事物。

以下主题可以帮助您熟悉优化服务器的电子商务性能的某些重要注意事项，并将提供至附加资源的链接以获取详细建议和示例。

### 『客户机性能』

尽管 iSeries 系统管理员经常对电子商务网络的客户机端具有很小控制，但您可以使用这些建议来确保对于电子商务环境优化客户机设备。

### 『网络性能』

网络设计、硬件资源和流量压力经常对电子商务应用程序的性能具有显著影响。可以使用此主题以获取有关如何优化网络性能和调整 iSeries 通信资源的信息。

### 第 29 页的『OS/400 中的 Java 性能』

OS/400 提供了几个配置选项和资源以用于优化 iSeries 服务器中的 Java 应用程序或服务的性能。使用此主题来了解 OS/400 中的 Java 环境，以及如果从基于 Java 应用程序获取可能的最佳性能。


### 第 29 页的『HTTP Server 性能』

HTTP Server 在 iSeries 服务器中经常是电子商务性能的一个重要部分。IBM 提供了几个选项和配置选项以允许您从此服务器获取最佳性能。

### 第 30 页的『WebSphere 性能』

WebSphere Application Server 是 iSeries 服务器选项的电子商务应用程序部署环境。使用此主题来了解如何在 WebSphere 环境中计划和优化性能。

除这些特定的建议外，管理员还应该熟悉下列主题：

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

## 客户机性能

由一个 PC 和 web 浏览器组成的客户机常常代表管理员具有最小直接控制的电子商务组件。然而，这些组件对 Web 应用程序的端到端响应时间仍具有显著影响。

为了确保高端性能，客户机 PC 应该：

- 具有足够内存。资源密集的 applet 以及使用复杂表单和图形的界面也可能对客户机的处理器具有要求。
- 使用高速和优化网络连接。客户机 PC 中的许多通信适配器可能在未对网络环境优化它们时工作。有关更多信息，参考您的通信硬件的文档。
- 使用完全支持所需技术的浏览器。然而，在设计 Web 接口时，浏览器支持和性能应该是主要关心的内容。

## 网络性能

网络常常在 Web 应用程序的响应时间方面起主要作用。而且，网络组件的性能影响常常是复杂而难于测量，因为网络流量和可用的带宽可能不断更改，且受系统管理员可能不具有直接控制的影响而影响。然而，有几个资源可用来帮助您监控和调整 iSeries 服务器中的通信资源。

有关更多信息，参考下列主题：

### 第 33 页的『收集服务』

收集服务定期收集通信资源的性能数据。特别关心的是，性能数据文件 QAPMTCP 和 QAPMTCPIFC 存储有关 TCP 服务器的信息。可以通过直接查询这些文件，或通过使用包括在 Performance Tools 许可程序中的报告来引用此数据。

### 第 60 页的『iSeries 导航器监控器』

可以使用系统监控器来提供有关在 iSeries 服务器中正如何使用系统资源（包括通信硬件）的信息。特别地，系统监控器内的线路利用率和 IOP 度量可以提供有关网络性能的有价值的信息。

### 第 20 页的『跟踪性能』

几个应用程序和工具允许您定期收集 iSeries 服务器中的通信资源的性能数据并随时间监控它们的性能。

### iSeries Performance Capabilities Reference

Performance Capabilities Reference 提供了详细信息、报告和示例，它可以帮助您配置或调整 iSeries 服务器以获取最佳性能。特别地，参见第 5 章：Communications Performance 以帮助您计划和管理 iSeries 通信资源。

### iSeries Network.com

此 Web 站点主管用于优化网络计划和资源的许多资源。特别地，参考文章“Cultivate your AS/400 Networks”和“8 tools for better network performance”。

## OS/400 中的 Java 性能

Java 常常是基于 Web 的应用程序选择的语言。然而，Java 应用程序可能同时需要 OS/400 执行环境和 Java 应用程序的某些优化，以获取最佳性能。

使用下列资源来了解 OS/400 中的 Java 环境，以及用于分析和改进 Java 性能的可用技巧和工具。

### Java 性能

有几个重要的配置选项和工具以帮助您从基于 Java 的应用程序获取最佳性能。

### 第 23 页的『收集有关应用程序的性能的信息』

有几个工具可用来帮助您监控和调整 OS/400 中的应用程序性能。使用此主题来了解如何使用性能跟踪、性能资源管理器（PEX）和类似的工具来帮助您测量和改进应用程序性能。

### iSeries Performance Capabilities Reference

Performance Capabilities Reference 提供了详细信息、报告和示例，它可以帮助您配置或调整 iSeries 服务器以获取最佳性能。特别地，参见第 7 章：Java Performance 以帮助您优化 Java 应用程序的性能，并了解使用 Java 编程的性能技巧。

### Java and WebSphere performance in OS/400

使用此红皮书来了解如何计划和配置操作环境以优化 Java 和 WebSphere 性能，并帮助您收集和分析性能数据。

### WebSphere J2EE application development for the IBM eServer iSeries server

此红皮书提供了 J2EE 介绍，并提供建议和示例以帮助您在 iSeries 服务器上成功实现 J2EE 应用程序。

除性能信息外，Java 主题还提供了在 iSeries 服务器上开发和部署 Java 应用程序的资源。

## HTTP Server 性能

HTTP Server 可以在基于 Web 的应用程序的端到端性能中起重要作用，并且几个新的改进允许您有效地监控和改进 Web 服务器性能。特别地，新的快速响应高速缓存加速键（FRCA）可允许您显著改进 HTTP Server 性能，特别是在主要静态环境中。

有关如何最大化 HTTP Server 性能的信息，参考下列资源。

### 第 33 页的『收集服务』

可以使用收集服务来收集 HTTP Server 性能数据并基于时间监控结果。性能数据文件 QAPMHTTPB 和 QAPMHTTPD 存储每个收集时间间隔内的 HTTP Server 数据。QAPMHHTB 提供基本信息，而 QAPMHTTPD 提供更详细的统计信息。可以直接查询这些数据文件，或在 Performance Tools 许可程序中参考系统和组件报告。

### IBM HTTP Server for iSeries

参考此主题以获取有关在 iSeries 中设置、配置和管理 HTTP Server 的信息。此主题还包括此产品的最新功能增强（如快速响应高速缓存加速键（FRCA））的描述。

### iSeries Performance Capabilities Reference

Performance Capabilities Reference 提供了详细信息、报告和示例，它可以帮助您配置或调整 iSeries 服务器以获取最佳性能。特别地，参见第 6 章：Web Server and Web Commerce 以获取 HTTP Server 性能规范、计划信息和性能技巧。

### HTTP server (Powered by Apache)

使用此红皮书来获取 OS/400 中的 HTTP Server（基于 Apache）的深入描述，其中包括在公共使用方案中配置 HTTP Server 的示例。

### AS/400 HTTP Server Performance and Capacity Planning

使用此红皮书来了解 HTTP Server 对性能调整 and 计划的影响。此出版物还包括使用 iSeries 性能管理工具收集、解释和响应 Web 服务器性能数据的建议。

## WebSphere 性能

在 WebSphere 环境中管理 iSeries 服务器性能对 iSeries 管理员存在几个挑战。基于 Web 的事务可能消耗许多资源，且以与传统通信工作量不同的方式消耗它们。

参考下列主题和资源以了解如何计划最佳性能以及在 WebSphere 环境中调整服务器资源。

### WebSphere Application Server performance considerations

此 Web 站点提供了 iSeries 服务器上的每个版本的 WebSphere Application Server 的资源，包括许多有用的性能技巧和建议。此资源对于使用 servlets、Java Server Pages (JSP) 和 Enterprise Java Beans (EJB) 的环境特别有价值。

### DB2 UDB/WebSphere Performance Tuning Guide

此红皮书提供了 WebSphere 和 DB2 环境介绍，并提供了建议、示例和对公共性能问题的解决方案，可以帮助您优化 WebSphere 和 DB2 性能。

### Java and WebSphere performance in OS/400

使用此红皮书来了解如何计划和配置操作环境以优化 Java 和 WebSphere 性能，并帮助您收集和分析性能数据。

### WebSphere V3 Performance Tuning Guide

此红皮书提供了详细建议和示例，以用于优化 iSeries 服务器中的 WebSphere V3 性能。

## iSeries Performance Capabilities Reference

Performance Capabilities Reference 提供了详细信息、报告和示例，它可以帮助您配置或调整 iSeries 服务器以获取最佳性能。特别地，参见第 6 章：Web Server and Web Commerce 以获取特定于 WebSphere 的性能技巧。

有关其它 WebSphere 和电子商务信息资源，参考主题 WebSphere 电子商务管理。

---

## 性能管理应用程序

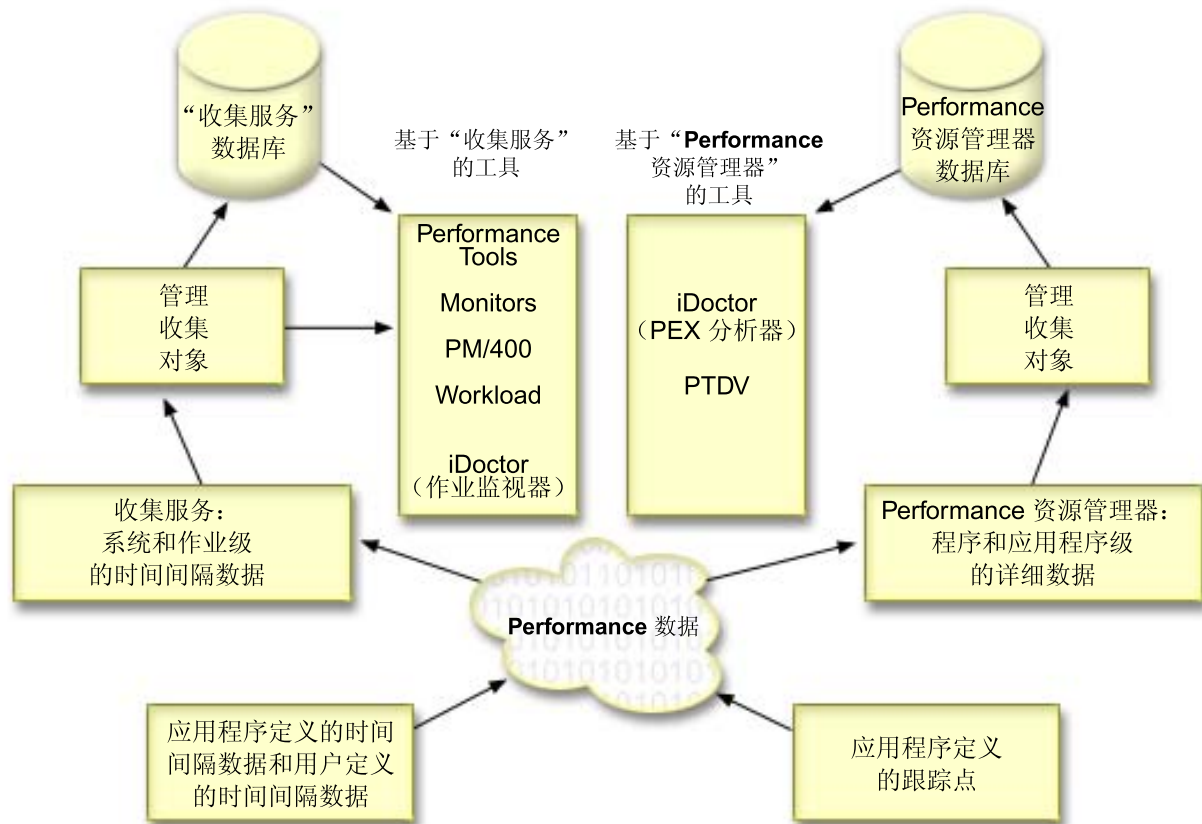
性能管理的许多应用程序都具有几种功能。精确了解可用的一组应用程序的哪个组件最适合给定的情况可能是复杂的。下列主题提供有关每个性能管理应用程序的详细信息，其中包括选择、使用和配置。

如下图所示，在 iSeries 服务器中基本上有两种性能收集功能：

- 收集服务，它收集系统和作业级别的间隔数据。可以连续运行它来了解系统中正在发生什么。收集的间隔数据是应用程序定义的或用户定义的。
- 性能资源管理器收集程序和应用程序级别的详细数据。它也跟踪应用程序中的工作流量并可以用来诊断困难的性能问题。收集的数据是由应用程序定义的性能资源管理器跟踪点，如使用 Domino、NetServer 或 WebSphere 定义的性能资源管理器跟踪点。

两种收集功能都将它们的数据存储在管理收集对象中。可以通过使用收集服务的创建性能数据（CRTPFRTA）命令或通过使用性能资源管理器数据的创建性能资源管理器数据（CRTPEXDTA）命令从管理收集对象转换数据。

本主题介绍可用来使用收集服务数据或性能资源管理器数据的性能管理应用程序。



### 第 33 页的『收集服务』

收集服务按用户定义的时间间隔收集性能数据，然后将此信息存储在系统中的收集对象中。许多其它工具（包括监控器、图形历史、PM/400 和 Performance Tools 许可程序中的许多功能）依赖于它们的数据的这些收集对象。

### 第 55 页的『性能数据文件』

可以从收集服务维护的收集对象生成数据库文件。使用本主题来查找这些数据库文件的名称、描述和属性。

### 第 60 页的『iSeries 导航器监控器』

监控器显示有关系统的性能当前信息。另外，可以使用它们来在特定事件发生时执行预定义的操作。可以使用系统、消息、作业、文件和 B2B 事务监控器来显示和监控有关系统的信息。系统和作业监控器使用收集服务所收集的性能数据。

### 第 66 页的『图形历史记录』

图形历史提供在指定的时间段内收集服务所收集的性能数据的图形显示。

### 第 67 页的『Performance Management/400』

PM/400 自动收集、归档和分析系统性能数据并返回报告以帮助您管理系统资源和容量。PM/400 使用收集服务所收集的性能数据。

### 第 81 页的『Performance Tools』

Performance Tools 许可程序包括许多功能部件以帮助您收集、分析和维护系统性能信息。这包括帮助管理



基于分布式网络的性能、收集和报告摘要和跟踪数据两者以及容量计划。Performance Tools 使用收集服务所收集的性能数据（采样数据）以及从启动性能跟踪（STRPFRTTC）命令和结束性能跟踪（ENDPFRTTC）命令获得的跟踪数据。

#### 第 86 页的『性能资源管理器』

性能资源管理器收集有关特定应用程序、程序或系统资源的更详细的信息，并提供详细洞察特定性能问题。这同时包括执行几种类型和级别的跟踪以及运行详细报告的能力。

#### 第 93 页的『iDoctor for iSeries』

iDoctor for iSeries 插件由用于管理性能的三个软件工具组成，即用于详细的跟踪数据分析的性能资源管理器分析器、有关作业的行为的跟踪级别信息的作业观察器以及帮助您查询和管理系统中的对象的对象资源管理器。

#### 第 94 页的『Performance Trace Data Visualizer (PTDV)』

Performance Trace Data Visualizer for iSeries (PTDV) 是一个 Java 应用程序，它可以用于正在 iSeries 中运行的应用程序的性能分析。

#### 第 94 页的『性能管理 API』

性能管理 API 提供管理收集的服务。这些 API 启动、结束和循环收集，并且它们为已收集的数据更改和检索系统参数。许多性能管理 API 使用收集服务所收集的性能数据。

#### 第 94 页的『“使用”有关 OS/400 性能的命令』

OS/400 包括几个重要功能以帮助您管理和维护系统性能。

#### 第 94 页的『扩展的自适应高速缓存』

扩展的自适应高速缓存可以通过收集磁盘使用情况数据，然后使用那些统计信息来创建大的高速缓存，有效地减少对磁盘的物理 I/O 请求来改善系统性能。

#### 第 97 页的『Workload Estimator for iSeries』

工作量估计器帮助您计划下次升级的大小和时间需求。此工具经常与 PM/400 一起使用来分析系统性能中的趋势并帮助您有效地管理 iSeries 服务器的成长和扩充。

#### 第 97 页的『iSeries 无线导航器』

iSeries 无线导航器允许您通过无线连接、使用个人数字助手（PDA）、因特网就绪电话或传统的 Web 浏览器来监控性能数据。iSeries 无线导航器使用收集服务所收集的性能数据。

#### 第 98 页的『PATROL for iSeries (AS/400) — 预测』

PATROL for iSeries (AS/400) — 预测通过自动执行高能力和优化性能所需要的许多日常管理任务来帮助管理 iSeries 性能。另外，此产品提供详细的容量计划信息来帮助您计划 iSeries 环境的成长。

## 收集服务

使用收集服务来收集性能数据，供以后由 Performance Tools for iSeries 许可程序或其它性能报告应用程序、iSeries 导航器监控器和图形历史功能进行分析。（如果您更喜欢查看实时性能数据，系统监控器提供了易于使用的图形界面，以用于监控系统性能。）收集服务收集数据，这些数据标识系统的不同区域所使用的系统资源的相对量。使用收集服务：

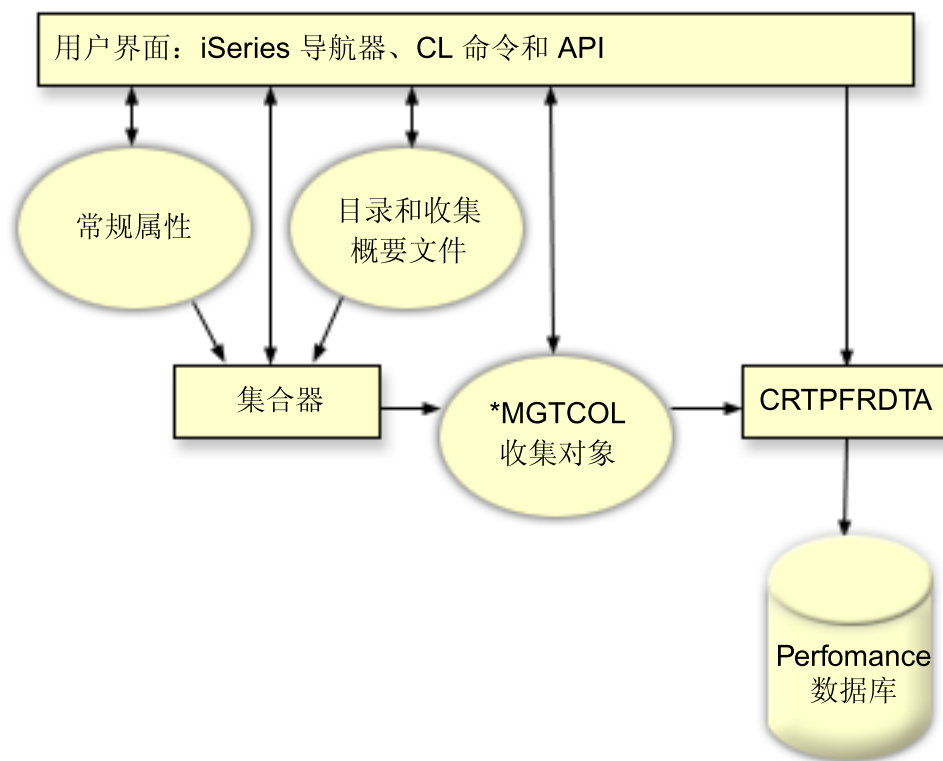
- 容易管理收集对象
- 使用最小的系统开销连续地自动收集性能数据
- 控制收集哪些数据和如何使用数据
- 在发行版之间移动性能数据而不必转换数据

- 创建由 Performance Tools 使用的性能数据文件
- 将您自己的用于收集用户定义的性能数据的程序集成到收集服务。

### 收集服务的工作方式

收集服务替换 OS/400 性能监控器，该性能监控器过去由“启动性能监控器”（STRPFRMON）命令调用。性能监控器（STRPFRMON 命令）自 V4R5 以来不再可用。当使用 OS/400 性能监控器时，您的数据被收集到多达 30 个数据库文件中。

收集服务能力引入一个新的进程来收集性能数据。收集服务将每个收集的数据存储在单个收集对象中，您可以从该对象创建需要的那么多组不同的数据库文件。这表示收集性能数据时较低的系统开销。即使您选择在收集期间创建数据库文件，也仍经历超过 OS/400 性能监控器的性能优点，因为收集服务使用较低优先级（50）批处理作业来更新这些文件。收集开销方面的减少使更详细地和以较短时间间隔连续地收集性能数据比较实际。收集服务使您能够建立用于收集和保存性能数据的网络范围的系统策略并自动实现该策略。对于在保持管理收集对象时，如果需要出现，则您有能力来回想和分析与性能有关的事件，向下直到收集的详细信息级别。



收集服务允许您收集性能数据，并对系统性能具有很小或没有可观察到的影响。可以使用 iSeries 导航器来配置收集服务，以按照要收集数据的频率收集需要的数据。收集对象 \*MGTCOL 充当有效的存储媒体，以保存大量性能数据。一旦配置和启动了收集服务，就会连续收集性能数据。当需要使用性能数据时，可以将需要的数据复制到一组性能数据库文件中。

上图提供了下列收集服务元素的概述:

#### 用户界面

存在几种方法，允许您访问收集服务的不同元素。例如，可以使用 CL 命令、API 和 iSeries 导航器界面。

## 一般属性

一般属性定义应该如何完成收集并且它们控制自动收集属性。

## 数据类别

数据类别标识要收集的数据类型。可以独立配置类别以控制收集什么数据和收集数据的频率。

## 收集概要文件

收集概要文件提供了一种保存和激活特定类别配置的方法。

## 性能收集器

性能收集器使用一般属性和类别信息来控制性能数据的收集。您可以启动和停止性能收集器，或者将它配置为自动运行。

## 收集对象

收集对象 \*MGTCOL 充当大量性能数据的有效存储媒体。

## 创建性能数据 (CRTPRFDTA) 命令


CRTPRFDTA 命令处理存储在管理收集对象中的数据并生成性能数据库文件。

## 性能数据库

数据库文件存储 CRTPRFDTA 命令处理的数据。这些文件可以分成以下类别：包含时间间隔数据的性能数据文件、配置数据文件和跟踪数据文件。

## 如何启动收集服务

可以通过使用下列任何一个方法来启动收集服务。然而，性能主题中的信息集中在 iSeries 导航器方法上。

启动方法	描述
iSeries 导航器	下面的部分显示如何使用 iSeries 导航器执行各种收集服务任务。
性能管理 API	可以使用性能管理 API 来启动、定制、结束和循环收集。另外，可以使用 API 来使用管理收集对象或定义您自己的事务。
传统的菜单选项	在基于字符的界面中输入 <b>GO PERFORM</b> 并从 Performance Tools 主菜单中选择选项 2 (收集性能数据)。有关附加信息，请转至 Performance Tools  一书。
Performance Management/400	可以激活第 70 页的『激活 Performance Management/400』，它自动启动收集服务然后在收集期间创建数据库文件。

## 收集服务任务

可以使用收集服务和 iSeries 导航器来执行下表中显示的各种数据收集任务。

任务	描述
以各种方式启动收集服务	在单个系统或几组系统中使用特定的性能度量创建定制的性能数据收集。也可以在启动程序中使用启动收集器 API 来自动启动性能数据收集。有关如何执行这些任务的更多信息，参考联机帮助。详细任务帮助可从 iSeries 导航器窗口中获得。从菜单栏中仅单击帮助并选择帮助主题即可。选择使用 . . . ? 来找出可以执行什么操作以及需要位于 iSeries 导航器窗口中的什么位置才能使它发生。

任务	描述
创建数据库文件	使用收集服务来自动创建性能数据库文件。也可以从收集对象创建数据库文件，其中数据是在收集数据之后存储的。可以将这些数据库文件与 PM/400 或 Performance Tools 许可程序一起使用，或可以创建您自己的查询来对这些文件运行。当『从收集服务数据创建数据库文件』时，了解如何控制收集什么数据。  参见第 55 页的『性能数据文件』以找出哪些数据库文件对您可用，以及在每个文件中包括什么字段级别的数据。
定制数据收集	第 37 页的『定制数据收集』查找有关控制收集什么性能数据和收集该数据的频率的信息。也可以查找有关重要的第 38 页的『收集服务的时区注意事项』的信息。
将用户定义类别添加至收集服务	可以通过编写出口程序并将它集成到收集服务中来从用户应用程序收集性能数据。然后可以在日常收集时间间隔期间收集此数据并将它存储在收集对象中。有关如何实现它的信息，参见第 38 页的『收集服务中的用户定义的类别』。
管理收集对象	查找需要第 47 页的『管理收集对象』的信息，包括收集对象的内容、多长时间保存收集对象以及使用收集对象可以执行什么操作。
收集跟踪数据	收集服务收集采样数据。然而，它不收集跟踪数据。查找如何第 23 页的『收集有关应用程序的性能的信息』。
收集用户定义的事务的性能数据	收集服务提供允许您定义您自己的事务的 API。查找如何使用第 47 页的『用户定义的事务』执行此任务。

## 从收集服务数据创建数据库文件

收集服务将已收集的数据放入管理收集对象中。要使用此数据，必须首先将此数据放入一组特殊的数据库文件中。要在收集数据时自动创建数据库文件，只需在启动收集服务对话框中选择创建数据库文件。当要将数据从现有的管理收集对象导出至数据库文件时，也可以第 37 页的『从现有的收集对象创建数据库文件』。

有许多允许您创建数据库文件的选项。

- 当使用收集服务来收集性能数据时，可以在收集数据时自动创建数据库文件。
- 可以从管理收集对象创建数据库文件，数据在被收集之后将存储在这些文件中。可以使用创建性能数据 (CRTPFRTDA) 命令从存储在管理收集 (\*MGTCOL) 对象中的性能信息创建一组性能数据库文件。可以使用 iSeries 导航器界面或 CRTPFRTDA 命令。
- 可以激活 Performance Management/400，它在收集期间自动启动收集服务然后创建数据库文件。

可以使用通过 Performance Tools for iSeries 许可程序或其它应用程序创建的数据库文件来生成性能报告。可以在一个系统中收集性能数据，然后将管理收集对象 (\*MGTCOL) 移至另一个系统来生成性能数据文件并运行 Performance Tools 报告。此操作允许您在另一个系统中分析性能数据，而不会影响源系统的性能。有关

Performance Tools 的更多信息，参见 Performance Tools  一书。

## 将数据存储在管理收集对象中而不是存储在数据库文件中

为什么应该将数据存储在管理收集对象中而不是存储在运行报告所需要的数据库文件中？因为您可以将管理收集对象与数据库文件分开进行管理，所以可以使用较小的收集时间间隔（如 5 分钟的时间间隔）收集性能数据，然后使用较长的采样时间间隔（如 15 分钟的时间间隔）创建数据库文件。

从单个管理收集对象中，可以通过指定不同的数据类别、不同的时间范围和不同的采样时间间隔为不同的目的创建许多组不同的数据库文件。

例如，可以在 24 小时内使用 5 分钟收集时间间隔对整个一组类别（所有数据，或 **Standard plus** 协议概要文件）收集性能数据。从管理收集对象中，可以为不同的目的创建不同的几组数据库文件。可以创建一组数据

库文件来运行正常的日常性能报告。这些文件可以包含使用 15 分钟的采样时间间隔的所有类别的数据。然后，为了分析特定的性能问题，可以创建另一组数据库文件。这些文件可以仅包含需要分析的单个类别、在 24 小时内的特定时间段和更小的 5 分钟采样时间间隔内的数据。

另外，单个管理收集对象允许您作为单个对象而不是作为许多文件对数据进行管理。单个收集对象允许您在发行版之间移动性能数据而不转换数据。只要保留收集对象，就能重新查看并分析与性能相关的事件，下至已收集的详细信息的级别。

## 导出已收集的数据

要将性能数据从管理收集对象导出至数据库文件，执行以下步骤：

1. 在 iSeries 导航器中，选择**中央管理**下的端点系统或选择**我的连接**（或您的活动环境）下具有直接连接的系统。
2. 展开**配置和服务**。
3. 单击**收集服务**。
4. 右键单击要导出至数据库文件的管理收集对象并选择**创建数据库文件**。
5. 在**创建数据库文件**对话框中，从收集对象中选择要包括在数据库文件中的类别。也可以选择不同的时间段和采样时间间隔，只要收集对象包含支持您的选择的数据。
6. 单击**确定**。

**从现有的收集对象创建数据库文件：** 可以将性能数据从现有的管理收集对象导出至数据库文件。请执行以下步骤：

1. 展开正在收集性能数据的系统的**配置与服务**。
2. 选择**收集服务**。
3. 右键单击要将数据导出至数据库文件的管理收集对象。
4. 可以首先选择**属性**来显示收集对象中的数据的特征。在数据属性页面中，可以看到在此收集对象中收集的数据类别以及收集的时间间隔。可以在选择要导出的数据时使用此信息。在复查了此信息后，单击**确定**。
5. 再次右键单击管理收集对象并选择**创建数据库文件**。使用联机帮助完成字段。
6. 单击**确定**。

在数据库文件中转换数据之后，可以使用第 81 页的『Performance Tools』或其它应用程序来生成性能报告。

## 定制数据收集

当使用收集服务来第 33 页的『收集服务』时，您控制收集什么数据和收集数据的频率。可以从提供的收集概要文件中进行选择。**标准**概要文件对应于先前发行版中“启动性能监控器”（STRPFRMON）命令提供的 OS/400 性能监控器功能中的系统数据的设置。**Standard plus 协议**概要文件对应于所有数据的 STRPFRMON 命令设置。或者，可以选择**定制**来创建您自己的定制概要文件。也有几种其它的概要文件可用，参考联机帮助以获取详细的描述。对于您的定制概要文件，可以从可用的数据类别（如系统 CPU、本地响应时间、磁盘存储器和 IOP（输入/输出处理器））列表中选择。

对于收集的每种类别的数据，可以指定将收集数据的频率。对于许多类别，将要选择缺省收集时间间隔，您可以从 15 秒和 60 分钟之间的预定义设置中设置。（建议的设置为 15 分钟。）

**注：**当将缺省值设置为任何指定的时间时，除那些具有显式时间间隔的类别（如磁盘存储器、输入/输出处理器和与通信相关的类别）外的所有类别都使用指定的时间。

收集的数据存储在称为收集的管理收集对象（类型为 \*MGTCOL）中。为了防止这些管理收集对象太大，必须定期循环收集。循环收集表示创建新的收集对象，并在数据收集在原始收集对象中停止的同时开始将数据存储在新的收集对象中。可以指定 1 小时到 24 小时范围内的任何时间间隔，这取决于您计划如何使用数据。

要在系统中定制收集服务，执行以下步骤：

1. 在 iSeries 导航器中，选择**中央管理**下的端点系统或**我的连接**下具有直接连接的系统（或您的活动环境）。
2. 展开**配置和服务**。
3. 右键单击**收集服务**并选择**属性**。
4. 在**常规**页面中，可能需要指定比缺省值 1 天长的保持周期。收集服务可以在保持周期到期后的任何时间从系统中删除它们包含的管理收集对象和数据。当创建管理收集对象时，将到期日期指定给它。即使将收集对象移动至另一个库，收集服务仍将在对象到期后删除对象。如果不希望收集服务将到期日期指定给新的收集对象，则可以指定**永久**。然后您将必须手工删除这些收集对象。  
要查看第 66 页的『图形历史记录』，必须指定图形或摘要的收集保持周期。在指定这些选项后，可以利用历史报告能力，它允许您对扩展时间段内的多个系统进行度量比较。  
也可以指定要存储收集的位置路径、要循环收集的频率以及缺省收集时间间隔。可以选择在收集期间自动创建数据库文件。
5. 单击**要收集的数据**选项卡。
6. 对于**要使用的收集概要文件**，选择**定制**。可以指定您为定制列表选择的每个类别的收集时间间隔。
7. 单击**确定**以保存定制的值。

一旦已将收集服务定制为您喜欢的设置，就可以再次右键单击**收集服务**并选择**启动收集服务**来开始收集性能数据。

## 收集服务的时区注意事项

当复查和分析性能数据时，收集的实际本地时间非常重要。例如，您可能需要知道在一天中最忙的时间段内收集了哪些数据，以便它表示复查中系统遇到的最重的工作量。如果收集性能数据的某些系统位于不同的时区，则应该知道以下注意事项：

- 当对系统组启动收集服务时，将对组中的所有系统同时启动收集服务。不考虑由于某些系统位于不同时间区域中而造成的系统时间与日期设置中的任何差别。
- 如果使用中央管理调度程序启动收集服务，则调度程序启动任务的时间是以中央管理中的中央系统的系统时间与日期为基础的。
- 每个端点系统的管理收集对象反映基于该端点系统和您的中央系统的 QTIME 和 QUTCOffset（世界协调时偏移）系统值的启动和结束时间。如果端点系统位于与中央系统不同的时间区域，且在两个系统中都正确设置这些系统值，则为收集对象报告的启动和结束时间是端点系统中的实际时间。换句话说，启动和结束时间反映端点系统中的 QTIME 的值，因为当那些事件发生时，QTIME 位于实际的时间点上。
- 性能收集的调度可能跨过从标准时间到夏令时或从夏令时到标准时间的边界。如果这样，则应该在调度启动时间时考虑此时间差别。否则，实际启动和结束时间可能比期望的时间晚或早 1 小时。另外，报告的管理收集对象的启动和结束时间受此差别的影响，除非每次更改为夏令时和从夏令时更改生效时调整 QUTCOffset 系统值。

有关使用收集服务来收集性能数据的更多信息，参见第 33 页的『收集服务』。

## 收集服务中的用户定义类别

第 33 页的『收集服务』中的用户定义的类别功能使应用程序能够将性能数据收集集成到收集服务中。这允许您通过编写数据收集程序、注册它并将它集成到收集服务来从应用程序收集数据。收集服务然后将在每个收集时间间隔调用数据收集程序，并将数据存储到收集对象中。应该使用以下列示的收集服务 API 来访问存储在收集对象中的数据。可以在正在收集数据时或在保持收集对象期间实时访问数据。

要实现此功能，需要：

1. 开发一个收集“收集服务”中的新类别的性能数据的应用程序。参考『收集程序建议和需求』以获取更多信息。
2. 创建收集程序的作业描述。QGPL 中的作业描述 QPMUSRCAT 提供了一个示例，但并不代表缺省值或建议。
3. 注册新类别并指定数据收集程序。有关更多信息，参见 API 描述：
  - 注册：QypsRegCollectorDataCategory
  - 取消注册：QypsDeregCollectorDataCategory在注册类别后，收集服务将它包括在可用收集类别的列表中。
4. 将类别添加至收集服务概要文件，然后循环收集服务
5. 开发一个查询收集对象的程序。有关更多信息，参见 API 描述：
  - 检索活动管理收集对象名称：QpmRtvActiveMgtcolName（仅用于实时查询收集对象。）
  - 检索管理收集对象属性：QpmRtvMgtcolAttrs
  - 打开管理收集对象：QpmOpenMgtcol
  - 关闭管理收集对象：QpmCloseMgtcol
  - 打开管理收集对象库：QpmOpenMgtcolRepo
  - 关闭管理收集对象库：QpmCloseMgtcolRepo
  - 读取管理收集对象数据：QpmReadMgtcolData

定制的收集程序立即在每个收集时间间隔运行，并且在收集对象中归档收集的数据。

也可以实现这些 API 的 Java 版本。所需要的 Java 类包括在 IFS 目录 QIBM/ProdData/OS400/CollectionServices/lib 中的 ColSrv.jar 中。Java 应用程序应该将此文件包括在它们的类路径中。有关 Java 实现的更多信息，参考 javadocs。

有关示例实现，参见第 40 页的『示例：实现用户定义的类别』。

## 实时查询收集对象

如果您的应用程序需要实时查询收集对象，则它将需要使查询与收集服务同步。为此，应用程序应该创建一个数据队列并使用收集服务注册它。一旦注册后，收集器就在每个收集时间间隔内和在收集循环结束时发送一个通知。应用程序应该维护数据队列，包括在完成后除去数据队列以及处理异常终止。要注册和取消注册数据队列，参考下列 API 描述：

- 添加收集器通知：QypsAddCollectorNotification
- 除去收集器通知：QypsRmvCollectorNotification

**收集程序建议和需求：** 收集服务在收集循环开始期间调用一次数据收集程序，为每个收集时间间隔调用一次，并在收集循环结束时再次调用。数据收集程序必须任何数据收集并将该数据返回至收集服务提供的数据缓冲区。除提供数据缓冲区外，收集服务还提供一个工作区，该工作区允许数据收集程序维护收集时间间隔之间的某些状态信息。

数据收集程序应该仅可能快地收集数据并应该执行最小格式化。该程序不应该执行任何数据处理或排序。尽管不将来自用户定义的类别的数据转换为数据库文件，但是，收集服务可以自动运行 CRTPFRTA 命令并将来自收集对象的数据在每个收集时间间隔结束时添加至数据库文件。如果数据收集程序在收集时间间隔内不能完成其任务，则 CRTPFRTA 命令不会正确运行。

可以在几种环境中创建数据收集程序：

- 用于 OPM 语言的 \*PGM。此环境可能不用来查询收集对象并可能导致较差性能。然而，它受较早的编程语言支持。
- \*SRVPGM，服务程序中的一个入口点。它用于 ILE 语言。
- \*JVAPGM，需要的 Java 类包括在 ColSrv.jar 中。此文件包括在 IFS 中的 QIBM/ProdData/OS400/CollectionServices/lib 中。下载 javadocs .zip 文件并打开 index.html 以获取 API 的 Java 实现的描述。

收集服务将下列请求发送至数据收集程序：

请求	描述
启动收集	数据收集程序应该初始化在数据收集期间使用的任何接口或资源。可选地，它还可以初始化收集服务提供的工作区，该工作区保存收集时间间隔之间的状态信息。如果要在收集的数据前包括控制记录，则数据收集程序还可以将少量数据写入数据缓冲区。一般情况下，此控制记录将在数据处理期间用来帮助解释数据。
收集时间间隔	收集服务为每个收集时间间隔发送一个时间间隔请求。数据收集程序应该收集数据并将它返回到数据缓冲区中。收集服务然后将该数据写入收集对象中的时间间隔记录。如果数据量对于数据缓冲区太大，则数据收集程序应该设置一个“更多数据”标志。此操作导致收集服务使用修饰符发送另一个时间间隔请求，以指示它是一个延续部分。收集服务在每次调用前复位更多数据标志。重复此进程，直到将所有数据移动到收集对象。
结束收集	当对于包含数据收集程序的类别的收集结束时，收集服务发送此请求。数据收集程序应该执行任何清除并可以有选择地返回收集控制记录。数据收集程序还应该发送一个返回码，以指示收集的结果。
清除并终止（关机）	如果有必要异常终止，则收集服务发送此请求。当数据收集程序作业结束时，自动释放操作系统资源，但任何其它关闭操作应该由数据收集程序执行。数据收集程序可以随时接收此请求。

有关这些参数、工作区、数据缓冲区和返回码的详细描述，参考头文件 QPMD CPRM，该文件位于 QSYSINC 中。

### 收集对象中的数据存储

收集对象对于每个数据收集类别具有一个库。此库是在启动该类别的收集时由收集服务创建的。每个库由下列记录组成：

记录	描述
控制	此可选记录可以是数据收集程序产生的第一个或最后一个记录，并可能在两个位置发生。一般情况下，它应该包含解释记录数据所需要的任何信息。
时间间隔	每个收集时间间隔创建一个时间间隔记录，即使它是空的。时间间隔记录包含在收集时间间隔期间写入数据缓冲区的数据。它的大小不得超过 4 GB。
停止	收集服务自动创建此记录，以指示数据收集会话的结束。如果重新启动了收集用户定义的类别而没有结束或循环收集服务，则可以在停止记录后选择性地包括后面跟有附加时间间隔记录的控制记录。

**示例：实现用户定义的类别：** 下列样本程序说明可以如何使用提供的 API 来将定制的数据收集集成到收集服务中。

- 第 41 页的『示例：数据收集程序』
- 第 43 页的『示例：注册数据收集对象程序的程序』
- 第 44 页的『示例：查询收集对象的程序』



## 代码示例不保证声明

IBM 授予您使用所有编程代码示例的非专有版权许可证，您可以由此生成相似的定制功能以满足您特定的需要。

IBM 提供所有样本代码只是出于解释的目的。并未在所有环境下完全测试这些示例。因此，IBM 不保证或暗示这些程序的可靠性、可服务性或功能。

本文档中包含的所有程序是以“按现状”的基础提供的，不附有任何形式的保证。明示的不保证声明包括非侵权性、适销性和适用于某特定用途的默示保证。

示例:数据收集程序: 下列程序收集某些测试数据并将它存储在数据缓冲区中，收集服务将该数据复制到收集对象。有关收集程序的更多信息，参考第 39 页的『收集程序建议和需求』

注: 请阅读代码示例不保证声明以了解重要的法律信息。

### C++ 样本代码

```
#include "string.h"           // memcpy(), memset(), strlen()
#include "stdio.h"           // printf()
#include "qpmcprm.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 + 4) = 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 + 8) = time( NULL );
            printf( "doCollect() complete in %d secs (%d bytes transferred)\n",
                *(time_t*)(workArea + 8) - *(time_t*)(workArea + 4), 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() */

```

示例: 注册数据收集对象程序的程序: 以下程序使用收集服务从先前示例中注册数据收集程序。在运行后, 收集服务在数据收集类别列表中显示数据收集程序。

### C++ 样本代码

```

#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, "DCPTTEST 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,
           "12345678901234567890123456789012345678901234567890", 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() */

```

示例: 查询收集对象的程序: 以下样本程序说明如何使用 QIBM/ProdData/OS400/CollectionServices/lib 中的 ColSrv.jar 文件中提供的 Java 类查询存储在收集对象中的数据。

### Java 样本代码

```

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 */

```

## 管理收集对象

当使用“收集服务”来第 33 页的『收集服务』时，将把每个收集存储在单个对象中。可以通过执行以下步骤来查看任何一个管理收集对象中的数据摘要：

1. 在 iSeries 导航器中，选择**中央管理**下的端点系统或**我的连接**（或您的活动环境）下您具有直接连接的系统。
2. 展开**配置和服务**。
3. 选择**收集服务**。
4. 右键单击列表中的任何一个管理收集对象并选择**属性**来查看有关该收集的一般信息和它包含的数据摘要。

可以右键单击任何一个收集对象并选择第 36 页的『从收集服务数据创建数据库文件』来指定数据类别、在收集周期内的时间范围和要包括在数据库文件中的采样时间间隔。

可以右键单击任何一个收集对象并选择第 66 页的『图形历史记录』来以图形方式查看管理收集对象中的数据。

### 删除或保存旧的管理收集对象

可以通过右键单击对象并选择**删除**从系统中删除收集对象。如果未手工删除这些对象，则收集服务将在到期日期和时间后自动删除它们。

收集服务仅删除**已循环**的管理收集对象。**已循环**的状态表示收集服务已停止收集数据和将数据存储在对​​象中。当您展开**配置和服务**并选择**收集服务**时，每个管理收集对象的状态显示在收集对象列表中。

收集服务在它下次启动或循环收集时删除已达到到期日期和时间的已循环的收集对象。到期日期与管理收集对象关联。即使将收集对象移动至另一个库，收集服务仍将在该对象到期后删除它。

每个管理收集对象的到期日期显示在该收集对象的属性中。要将对象在系统中保持更长时间，只需在属性页面中更改日期。右键单击列表中的任何管理收集对象并选择**属性**来查看有关该收集的信息。如果不希望收集服务为您删除管理收集对象，可以指定**永久**。

## 用户定义的事务

收集服务和性能资源管理器可以立即收集在应用程序中定义的性能数据。通过提供的 API，可以使用收集服务将事务数据集成到正常调度的采样数据收集中，并通过运行性能资源管理器获取有关事务的跟踪级别数据。

有关详细描述和用法注意事项，参考下列 API 描述：

- 启动事务：QYPESTRT, qypeStartTransaction
- 结束事务：QYPEENDT, qypeEndTransaction
- 作业记录事务：QYPELOGT, qypeLogTransaction（仅由性能资源管理器使用）

- 添加跟踪点: QYPEADDT, qypeAddTracePoint (仅由性能资源管理器使用)

注: 只需要检测一次应用程序。收集服务和性能资源管理器使用相同的 API 调用来收集不同类型的性能数据。

## 将用户定义的事务数据集成到收集服务中

可以选择用户定义的事务作为收集服务配置中的一种收集类别。收集服务然后在每个收集时间间隔收集事务数据并将该数据存储在收集对象中。CRTPFRTA 命令将此数据导出至用户定义的事务性能数据库文件中 QAPMUSRTNS。收集服务按事务类型组织数据。可以指定需要的那么多事务类型; 然而, 收集服务将仅报告前 15 个事务类型。另外的事务类型的数据组合在一起并存储为 \*OTHER 事务类型。在每个收集时间间隔, 收集服务为每个唯一作业的每个类型的事务创建一个记录。有关详细描述, 参考启动事务 API 中的用法注意事项。

收集服务收集一般的事务数据, 如事务响应时间。还可以包括多达 16 个可选的应用程序定义的计数器, 它们可以跟踪特定于应用程序的数据 (如用于事务的 SQL 语句数) 或其它增量测量。您的应用程序应该使用启动事务 API 来指示新的事务的开始, 并应该包括对应的结束事务 API 以将事务数据传送至收集服务。有关更多信息, 参考 QAMUSRTNS 文件描述和 API 描述。

有关样本实现, 参考『C++ 示例: 将用户定义的事务集成到收集服务中』或第 52 页的『Java 示例: 将用户定义的事务集成到收集服务中』中的示例。

注: 请阅读代码示例不保证声明以了解重要的法律信息。

## 使用性能资源管理器收集用户定义的事务的跟踪信息

可以在性能资源过滤器会话期间使用启动、结束和记录事务 API 来创建跟踪记录。性能资源过滤器将当前线程的系统资源利用率 (如 CPU 利用率、I/O 和捕获 / 锁定活动) 存储在这些跟踪记录中。另外, 可以选择包括特定于应用程序的性能数据, 然后将它发送至这些 API 中的每个 API 中的性能资源管理器。还可以使用添加跟踪点 API 来标识性能资源管理器应该收集跟踪数据的特定于应用程序的事件。

要为事务启动性能资源管理器会话, 在第 86 页的『性能资源管理器』定义的 (OSEVT) 参数中指定 \*USRTRNS。在输入 ENDPEX 命令后, 性能资源管理器将应用程序提供的数据写入 QAYPEMIUSR 性能资源数据库文件中的 QMUDTA 字段。系统提供的对于启动、结束和任何作业记录的性能数据存储在 QAYPEMIUSR 和 QAYPETIDX 数据库文件中。

有关详细描述, 参考启动事务 API 描述中的 API 描述和用法注意事项。

**C++ 示例: 将用户定义的事务集成到收集服务中:** 以下 C++ 示例程序显示如何使用启动事务和结束事务 API 来将用户定义的事务性能数据集成到收集服务中。

注: 请阅读代码示例不保证声明以了解重要的法律信息。

```
//*****  
// 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; // Used for 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 示例: 将用户定义的事务集成到收集服务中:** 以下 Java 示例程序显示如何使用启动事务和结束事务 API 来将用户定义的事务性能集成到收集服务中。

注: 请阅读代码示例不保证声明以了解重要的法律信息。

```

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 */

```

## 性能数据文件

性能数据是一组有关系统（或系统网络）的操作的信息，可以使用此信息来了解响应时间和吞吐量。可以使用性能数据来调整程序、系统属性和操作。这些调整可以降低响应时间和增大吞吐量。调整也可以帮助您预测对系统、操作或程序进行某些更改的效果。

第 33 页的『收集服务』将性能数据收集到管理收集对象（\*MGTCOL）中。“创建性能数据”（CRTPFRTA）命令处理来自该收集对象的数据并将结果存储到性能数据库文件中。数据库文件被划分成以下类别：

### 包含时间间隔数据的性能数据文件

这些文件包含每个时间间隔内收集的性能数据。参见『包含时间间隔数据的性能数据文件』数据以获取这些文件的列表（带有每个文件的简单描述和完整信息的链接）。要了解这些文件中的数据来自何处，参考第 58 页的『性能数据文件：收集服务系统类别和文件关系』。当查看这些文件时，还可能发现第 57 页的『性能数据文件：文件缩写』会有帮助。

### 配置数据文件

每个会话收集一次配置数据。要了解这些文件中的数据来自何处，参考第 58 页的『性能数据文件：收集服务系统类别和文件关系』。可以在第 59 页的『性能数据文件：配置数据库文件的字段数据』中查找 QAPMCONF、QAPMHDWR 和 QAPMSBSD 文件。

### 跟踪数据文件

仅当选择操作时，才收集跟踪数据。在第 60 页的『性能数据库文件：跟踪数据库文件的字段数据』中可以查找 QAPMDMPT 文件。

通过使用系统上可用的“显示器文件字段描述”（DSPFFD）命令，则如字节数和缓冲区位置之类的附加字段信息可用。例如，在任何命令行输入以下命令：

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

有关 iSeries 性能的更多信息，参见第 1 页的『性能』

### 包含时间间隔数据的性能数据文件

要查看有关性能数据文件的完成信息，从下面列表（按字母顺序显示）中选择要查看的文件。

文件	描述
QAPMAPPN	APPN 数据
QAPMASYN	异步统计信息（每个链接一个）
QAPMBSC	二进制同步统计信息（每个链接一个）
QAPMBUS	总线计数器器（每个总线一个）
QAPMCIOP	通信 IOP 数据（每个 IOP 一个）
QAPMDDI	分布式数字接口（DDI）数据（每个链接一个）
QAPMDIOP	存储设备 IOP 数据（每个 IOP 一个）
QAPMDISK	磁盘存储数据（每个读 / 写头一个）
QAPMDOMINO	Domino for iSeries 数据（每个 Domino Server 一条记录）
QAPMECL	令牌环文件项（每个链接一个）
QAPMETH	以太网统计信息（每个链接一个）
QAPMFRLY	帧中继数据（每个链接一个）
QAPMHDLC	HDLC 统计信息（每个链接一个）
QAPMHTTPB	IBM HTTP Server（基于 Apache）的基本数据（每个服务器一个）
QAPMHTTPD	HTTP Server（基于 Apache）的详细数据（每个服务器组件一个）
QAPMIDLC	综合业务数字网数据链接控制文件项（每个链接一个）
QAPMIOPD	扩展的 IOP 数据（为特定的 IOP 附加的数据）（每个 IOP 一个）
QAPMJOBMI	MI 作业数据（每个作业、任务或线程一个记录）当使用此文档时，您可以查到更多有关任务类型扩展器的有用的信息
QAPMJOBOS	作业操作系统数据（每个作业一个）
QAPMJOBS 和 QAPMJOBL	作业数据（每个作业、任务或线程一个）



文件	描述
QAPMJOBWT	作业、任务和线程等待调节
QAPMJOBWTD	在文件 QAPMJOBWT 查找计数器设置描述。
QAPMJSUM	由作业组处理的作业汇总数据（每个作业组一个记录）
QAPMLAPD	综合业务数字网 LAPD 文件项（每个链接一个）
QAPMLIOP	双轴线工作站控制器数据（每个物理控制器一个）
QAPMMIOP	多功能 IOP（每个 IOP 一个）
QAPMPOOL 和 QAPMPOOLL	主存储器数据（每个存储池一个）
QAPMPOOLB	存储池数据（每个池一个）
QAPMPOOLT	存储池调整数据（每个存储池一个）
QAPMPPP	点到点协议数据（每个链接一个）
QAPMRESP	本地工作站响应时间（每个工作站一个）
QAPMRWS	远程工作站响应时间
QAPMSAP	TRLAN、以太网、DDI 和帧中继 SAP 文件项（每个 SAP 项一个）
QAPMSNA	SNA 数据
QAPMSNADS	SNADS 数据（每个 SNADS 作业一个）
QAPMSTND	DDI 站数据
QAPMSTNE	以太网站文件项
QAPMSTNL	令牌环站文件项
QAPMSTNY	帧中继站文件项
QAPMSYS 和 QAPMSYSL	系统性能数据
QAPMSYSCPU	系统 CPU 使用率数据
QAPMSYSTEM	系统级别性能数据
QAPMTCP	TCP/IP 数据
QAPMTCPIFC	个别 TCP/IP 接口的 TCP/IP 数据
QAPMUSRTNS	用户定义的事务数据（对于每种事物类型每个作业有一个记录）
QAPMX25	X.25 统计信息（每个链接一个）

## 性能数据文件：文件缩写

第 55 页的『性能数据文件』在字段和字节数据表中使用缩写。这些缩写包括：

缩写	描述
主文件	这些文件与该类别有关并且是从该类别生成的。
<b>C</b>	属性列中的字符。
<b>PD</b>	属性列中的压缩十进制数。
<b>Z</b>	属性列中区位十进制数。
<b>IOP</b>	输入 / 输出处理器或 I/O 处理器。控制主机系统和其它设备（如磁盘、显示站和通信线路）之间的活动的处理器。
<b>DCE</b>	数据电路终端设备。
<b>MAC</b>	媒体访问控制。通信 IOP 中的实体。
<b>LLC</b>	逻辑链路控制。通信 IOP 中的实体。

信标帧	当振铃不工作时发送的帧。
类型 II 帧	系统网络体系结构 (SNA) 使用的面向连接的帧 (信息帧)。
I 帧	信息帧。

## 性能数据文件：收集服务系统类别和文件关系

当使用第 33 页的『收集服务』收集性能数据时，数据存储在管理收集 (\*MGTCOL) 对象中。CRTPFDRDTA 命令将数据从该管理收集对象中导出，然后将数据写入第 55 页的『性能数据文件』。收集服务可以独立控制和收集的每种数据类型通过一个数据类别表示。每个数据类别包含或提供写入一个或多个性能数据文件的数据。对于要创建的数据库文件或成员，该文件或成员所依赖的类别 (或一组类别) 必须存在且由 CRTPFDRDTA 处理。下表标识类别与文件的关系。有三种类型的关系：

关系	描述
主文件	这些文件与该类别有关并且是从该类别生成的。
兼容性文件	这些文件是逻辑文件，它们连接主文件以提供与先前文件结构的性能数据库兼容性。如果系统生成所有参与文件 (主类别)，则也会生成兼容性文件。
辅助文件	这些文件与从包含在该类别或主文件中的数据派生的某些数据有关并包含这些数据。然而，它们不由该类别控制。

用户应该注意以下几点：

1. 仅当数据库文件是选择的类别的主文件时，CRTPFDRDTA 命令才生成该数据库文件。
2. 如果为多个类别列示了一个主文件，则必须选择其中的每个类别才能生成该文件。
3. 如果一个类别的主文件列示为另一个类别的辅助文件，则必须选择第二个类别以确保生成的数据库文件中的信息是完整的。例如，如下表中所示，要生成 QAPMECL 的完整数据库文件，必须同时选择 \*CMNBASE 和 \*CMNSTN。
4. 仅当系统生成所有关联的主文件时，系统才会生成兼容性文件。

下表说明系统类别和性能数据库文件之间的关系。

类别	主文件	兼容性文件	辅助文件
*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 无		QAPMDDI QAPMETH QAPMECL QAPMFRLY QAPMX25
*CMNSAP	QAPMSAP		
*LCLRSP	QAPMRESP		
*APPN	QAPMAPPN		
*SNA	QAPMSNA		
*EACACHE	无		QAPMDISK (参见“注”)
*TCPBASE	QAPMTCP		
*TCPIFC	QAPMTCPIFC		
*DOMINO	QAPMDOMINO		
*HTTP	QAPMHTTPB QAPMHTTPD		
*USRTNS	QAPMUSRTNS		

注:

此类别不可以通过 CRTPFRTA 来选择。然而, 它会导致通过 \*DISK 类别来报告附加数据。

### 性能数据文件: 配置数据库文件的字段数据

每个会话收集一次配置数据。下列性能数据文件显示文件名、简要描述以及对系统配置数据、子系统数据和硬件配置数据的字段数据详细信息(当提供时)的引用。有关第 33 页的『收集服务』如何生成此文件和数据来自何处的信息, 参考第 58 页的『性能数据文件: 收集服务系统类别和文件关系』。

字段名称	描述
QAPMCONF	系统配置数据。
QAPMHDWR	系统硬件配置。
QAPMSBSD	子系统数据。没有字段和字节数据。

## 性能数据库文件: 跟踪数据库文件的字段数据

跟踪数据包括内部系统跟踪数据。这是详细的数据, 您收集该数据以获取关于特定作业和事务的附加信息。除非使用 Performance Tools 许可程序来分析此类型的数据, 否则不应收集它。以下是当使用“启动性能跟踪”(STRPFTRC) 命令时系统所支持的性能数据文件。

文件名	描述
QAPMDMPT	系统跟踪数据 (没有字段或字节详细信息)。

## iSeries 导航器监控器

包含在 iSeries 导航器中的监控器使用第 33 页的『收集服务』数据来跟踪您特别感兴趣的系统性能的元素。而且, 当某些事件 (如 CPU 利用率的百分比或作业的状态) 发生时, 它们可以执行指定的操作。当系统性能在多个系统和多组系统中发生时, 可以使用监控器来查看和管理系统性能。

对于监控器, 您可以启动监控器, 然后转至服务器上、iSeries 导航器中或 PC 上的其它任务。事实上, 甚至可以关闭 PC。iSeries 导航器继续监控和执行已指定的任何阈值命令或操作。监控器运行, 直到您停止它。也可以通过使用第 97 页的『iSeries 无线导航器』访问监控器来使用监控器以远程方式管理性能。

iSeries 导航器提供下列类型的监控器:

### 系统监控器

当性能数据发生或长达 1 小时, 收集和显示性能数据。详细图形有助于查看它发生时服务器正在继续执行什么操作。从各种度量 (性能评估) 中选择以查明系统性能的特定方面。例如, 如果正在监控服务器上的平均 CPU 利用率, 则可以单击图形上的任何一个收集点, 以查看显示具有最高 CPU 利用率的 20 个作业的详细图表。然后, 可以右键单击其中任何一个作业以直接使用该作业。

### 作业监控器

根据作业名称、作业用户、作业类型、子系统或服务器类型监控一个作业或作业列表。从各种度量中选择以监控作业的性能、状态或错误消息。要直接使用作业, 只需从显示在作业监控器窗口中的列表中右键单击该作业。

### 消息监控器

查看应用程序是否已成功完成, 或者监控对商业需要关键的特定消息。从消息监控器窗口, 可以查看消息的详细信息、回答消息、发送消息以及删除消息。

### B2B 活动监控器

如果已配置象 Connect for iSeries 的应用程序, 则可以使用 B2B 活动监控器来监控 B2B 事务。可以查看过去的时间内活动的活动的事务图, 并可以在触发阈值时自动运行命令。可以搜索和显示特定事务, 以及查看该特定事务的详细步骤的条形图。

### 文件监控器

监控一个或多个选择的文件, 以获取指定的文本字符串、指定的大小或对文件的任何修改。

要了解关于监控器的更多信息, 参见下列主题:

#### 第 61 页的『监控器概念』

监控器可以显示实时性能数据。另外, 它们可以连续监控系统, 以便在达到指定的阈值时运行选择的命令。了解监控器如何工作, 它们可以监控什么, 以及它们可以对给定的性能情况如何作出响应。

## 『配置监控器』

可以在 iSeries 导航器中配置监控器。使用此主题来了解如何安装监控器以及如何配置它，以更好地利用可用选项。

## 第 63 页的『方案: iSeries 导航器监控器』

此主题提供了一些方案，它们显示可以如何使用某些不同类型的监控器来查看系统性能的特定方面。

## 监控器概念

系统监控器显示存储在由第 33 页的『收集服务』生成和维护的收集对象中的数据。系统监控器在收集数据时显示数据，最多显示 1 小时。要查看更长时间段的数据，应该使用第 66 页的『图形历史记录』。可以更改监控器属性中的数据收集的频率，该频率覆盖收集服务中的设置。

可以使用监控器来跟踪和研究系统性能的许多不同元素，并可以同时运行许多不同的监控器。当一起使用时，监控器提供一种用于观察和管理系统性能的复杂工具。例如，当实现新的交互式应用程序时，可以使用系统监控器按优先级排列作业的资源利用率，使用作业监控器来观察和处理任何有问题的作业，以及使用消息监控器来提醒您在任何一个系统中是否发生了指定的消息。

## 设置阈值和操作

当创建新的监控器时，可以指定当系统度量达到指定的阈值级别或发生事件时希望发生的操作。当阈值级别或事件发生时，可以选择在端点系统中运行 OS/400 命令，如发送消息或暂挂作业队列。另外，可以选择使监控器执行几个预定义的操作，如更新事件作业记录和通过在 PC 上发出声音警报或启动监控器来提醒您。最后，可以通过指定当达到阈值时导致监控器恢复正常活动的第二个阈值级别来自动复位监控器。

## 配置监控器

系统监控器是高度交互式工具，您可以使用它们来从端点系统收集和显示实时性能数据。创建新的监控器是从新建监控器窗口开始的一种快速且容易的进程：

1. 在 iSeries 导航器中，展开中央管理，选择**监控器**，右键单击**系统**，然后选择**新建监控器**。
2. 指定监控器名称。从**新建监控器** — **常规**页面中为监控器指定一个名称。提供一个简要描述以便您可以在监控器列表中查找该监控器。
3. 选择度量。使用**新建监控器** — **度量**页面来选择您的度量。可以对任何数目的端点系统或系统组监控任何数目的度量。
4. 查看和更改度量信息。使用**新建监控器** — **度量**页面来编辑每个度量的属性。可以为选择的每个度量编辑收集时间间隔、最大绘图值和显示时间。
5. 设置阈值命令。使用**度量**页面中的**阈值**选项卡来启用阈值并指定当触发或复位阈值时在端点系统中运行的命令。
6. 设置阈值操作。使用**新建监控器** — **操作**页面来指定当触发或复位度量阈值时希望发生的操作。
7. 选择您的系统和组。使用**新建监控器** — **系统和组**页面来选择要启动监控器的端点系统或系统组。

在创建了监控器之后，右键单击该监控器名称并选择**启动**，以运行该监控器并开始使用监控器图形。

## 监控器度量

要有效地监控系统性能，必须决定要监控系统性能的哪些方面。中央管理提供了各种性能测量（称为**度量**）以帮助您查明系统性能的不同方面。

**新建监控器**窗口中的**度量**页面允许您查看和更改要监控的度量。要访问此页面，选择**监控器**，右键单击**系统**，然后选择**新建监控器**。填写所需字段，然后单击**度量**选项卡。

当第 61 页的『配置监控器』时，可以使用要包括在监控器中的列表中的任何一个度量、一组度量或所有度量。在监控器中可以使用的度量类型包括以下几个：

度量组:	度量描述:
<b>CPU 利用率</b>	由系统中的作业消耗的可用处理单元时间的百分比。从以下类型的 CPU 利用率度量中选择以用于您的监控器中： <ul style="list-style-type: none"><li>• CPU 利用率（平均值）</li><li>• CPU 利用率（交互式作业数）</li><li>• CPU 利用率（交互式功能部件）</li><li>• CPU 利用率（数据库能力）</li><li>• CPU 利用率（辅助工作量）</li><li>• CPU 利用率基本（平均值）</li></ul> 要了解有关这些度量以及如何使用它们的更多信息，参见在 iSeries 导航器中的 <b>新建监控器</b> 窗口或 <b>监控器属性</b> 窗口中的 <b>常规</b> 选项卡中可获得的联机帮助。
<b>交互式响应时间</b> (平均值和最大值)	交互式作业在系统中经历的响应时间。
<b>事务率</b> (平均值)	由系统中的所有作业每秒完成的事务数。
<b>事务率</b> (交互式)	由系统中的下列类型的作业每秒完成的事务数： <ul style="list-style-type: none"><li>• 交互式</li><li>• 多个请求者终端（MRT）</li><li>• System/36 环境交互式</li><li>• 传递</li></ul>
<b>批处理逻辑数据库 I/O</b>	当前由系统中的批处理作业执行的逻辑数据库输入 / 输出（I/O）操作的平均数。
<b>磁盘臂利用率</b> (平均值和最大值)	在收集数据期间当前在系统中使用的磁盘臂容量的百分比。
<b>磁盘存储量</b> (平均值和最大值)	在收集数据期间在系统中已满的磁盘臂存储器的百分比。
<b>磁盘 IOP 利用率</b> (平均值和最大值)	在收集数据期间在系统中磁盘输入 / 输出处理器（IOP）忙的程度。
<b>通信 IOP 利用率</b> (最大值和平均值)	在收集数据期间在系统中通信输入 / 输出处理器（IOP）忙的程度。
<b>通信线路利用率</b> (平均值和最大值)	在所有系统通信线路上实际发送和接收的数据量。
<b>LAN 利用率</b> (最大值和平均值)	在所有局域网（LAN）通信线路上实际发送和接收的数据量。
<b>机器池故障数</b>	系统的机器池中每秒发生的故障数。
<b>用户池故障数</b> (最大值和平均值)	系统的所有用户池中每秒发生的故障数。

如果需要更多帮助，单击**新建监控器** — **度量**窗口中的**帮助**按钮。一旦已熟悉中央管理度量，选择哪些度量就将取决于计算环境的信息需求。在已选择把正尝试查看的信息作为目标的度量后，查看和更改为监控器选择的每个度量的详细度量信息已准备就绪。

## 方案: iSeries 导航器监控器

包括在 iSeries 导航器中的监控器提供了用于研究和管理系统性能的一组功能强大的工具。有关 iSeries 导航器提供的监控器类型概述，参见第 60 页的『iSeries 导航器监控器』。

有关详细用法示例和样本配置，参见下列方案：

### 『方案: 系统监控器』

参见示例系统监控器，如果 CPU 利用率太高，该监控器提醒您，并临时暂挂任何较低优先级的作业直到更多资源可用。

### 第 65 页的『方案: 消息监控器』

参见示例消息监控器，该监控器显示发生在任何一个 iSeries 服务器中的消息队列的任何查询消息。一检测到消息，监控器就打开并显示消息。

### 第 64 页的『方案: 作业监控器』

参见示例作业监控器，该监控器跟踪已指定的作业的 CPU 利用率并且如果 CPU 利用率太高，提醒作业的所有者。

## 方案: 系统监控器:

### 情况

作为系统管理员，需要确保 iSeries 系统具有足够的资源以满足用户的当前要求和商业需求。对于您的系统，CPU 利用率是特别重要的。如果 CPU 利用率太高，希望系统提醒您，并临时暂挂任何较低优先级的作业直到更多资源可用。

要实现它，可以设置系统监控器，如果 CPU 利用率超过 80%，该监控器发送给您一条消息。而且，它还可以暂挂 QBATCH 作业队列中的所有作业，直到 CPU 利用率降低到 60%，在此点上释放作业，并且正常操作继续进行。

### 配置示例

要设置系统监控器，需要定义要跟踪哪些度量以及当度量达到指定的级别时要监控器执行哪些操作。要定义实现此目的的系统监控器，完成以下步骤：

1. 在 iSeries 导航器中，展开**中央管理 > 监控器**，右键单击**系统监控器**，并选择**新建监控器...**。
2. 在**常规**页面中，为此监控器输入名称和描述。
3. 单击**度量**选项卡，并输入下列值：
  - a. 从“可用度量”列表中选择 **CPU 利用率基本（平均值）**，并单击**添加**。CPU 利用率基本（平均值）现在列示在要监控的度量之下，并且窗口的底部显示此度量的设置。
  - b. 对于**收集时间间隔**，指定希望收集此数据的频率。它将覆盖“收集服务”设置。对于本示例，指定 **30 秒**。

- c. 要更改此度量的监控器图的垂直轴标度, 更改**最大绘图值**。要更改此度量的图的水平轴标度, 更改**显示时间的值**。
- d. 单击度量设置的**阈值 1** 选项卡, 并输入下列值以在 CPU 利用率大于或等于 80% 时发送一条查询消息:
  - 1) 选择**启用阈值**。
  - 2) 对于阈值触发器值, 指定 **>= 80** (忙的程度大于或等于 80%)。
  - 3) 对于**持续时间**, 指定 **1** 个时间间隔。
  - 4) 对于 **OS/400 命令**, 指定以下内容:
 

```
SNDMSG MSG('Warning,CPU...') TOUSR(*SYSOPR) MSGTYPE(*INQ)
```
  - 5) 对于阈值复位值, 指定 **< 60** (忙的程度小于 60%)。当 CPU 利用率降低到 60% 以下时, 它将复位监控器。
- e. 单击**阈值 2** 选项卡, 并输入下列值以在 CPU 利用率保持在 80% 以上达 5 个收集时间间隔时暂挂 QBATCH 作业队列中的所有作业。
  - 1) 选择**启用阈值**。
  - 2) 对于阈值触发器值, 指定 **>= 80** (大于或等于 80%)。
  - 3) 对于**持续时间**, 指定 **5** 个间隔。
  - 4) 对于 **OS/400 命令**, 指定以下内容:
 

```
HLDJOBQ JOBQ(QBATCH)
```
  - 5) 对于阈值复位值, 指定 **< 60** (忙的程度小于 60%)。当 CPU 利用率降低到 60% 以下时, 它将复位监控器。
  - 6) 对于**持续时间**, 指定 **5** 个间隔。
  - 7) 对于 **OS/400 命令**, 指定以下内容:
 

```
RLSJOBQ JOBQ(QBATCH)
```

此命令在 CPU 利用率保持在 60% 以下达 5 个收集时间间隔时释放 QBATCH 作业队列。

4. 单击**操作**选项卡, 并同时在**触发器**和**复位**列中选择**记录事件**。当触发和复位阈值时, 此操作在事件作业记录中创建一个项。
5. 单击**系统和组**选项卡以指定要监控的系统和组。
6. 单击**确定**以保存监控器。
7. 从系统监控器列表中, 右键单击新监控器并选择**启动**。

## 结果

新监控器显示 CPU 利用率, 并根据指定的收集时间间隔, 每 30 秒添加一次新的数据点。每当 CPU 利用率达到 80% 时, 即使 PC 已关闭, 监控器也自动执行指定的阈值操作。

**注:** 此监控器仅跟踪 CPU 利用率。然而, 可以在同一监控器中包括任何数目的可用度量, 并且每个度量可以具有它自己的阈值和操作。还可以具有同时运行的多个系统监控器。

## 方案: 作业监控器:

### 情况

当前正在 iSeries 服务器上运行新的应用程序, 并且担心一些新的交互式作业将消耗不可接受的资源量。如果有问题的作业曾消耗过多 CPU 容量, 将希望通知这些作业的所有者。



可以设置一个作业监控器来观察来自新的应用程序的作业，并且如果作业消耗的 CPU 容量大于 30%，则发送一条消息。

### 配置示例

要设置作业监控器，需要定义要观察哪些作业，要观察什么作业属性，以及当检测到指定的作业属性时，监控器应该执行什么操作。要设置达到此目的的作业监控器，完成以下步骤：

1. 在 iSeries 导航器中，展开**中央管理 > 监控器**，右键单击**作业监控器**，并选择**新建监控器...**。
2. 在**常规**页面中，输入下列值：
  - a. 为此监控器指定名称和描述。
  - b. 在**要监控的作业**选项卡中，输入下列值：
    - 1) 对于**作业名称**，指定要观察的作業的名称（例如 MKWIDGET）。
    - 2) 对于**子系统**，指定 QINTER。
    - 3) 单击**添加**。
3. 单击**度量**选项卡，并输入下列信息：
  - a. 在**可用度量**列表中，展开**摘要数字值**，选择 **CPU 百分比利用率**，并单击**添加**。
  - b. 在度量设置的**阈值 1**选项卡中，输入下列值：
    - 1) 选择**启用触发器**。
    - 2) 对于**阈值触发器值**，指定 **>= 30**（忙的程度大于或等于 30%）。
    - 3) 对于**持续时间**，指定 **1** 个间隔。
    - 4) 对于 **OS/400 触发器命令**，指定以下内容：  
`SNDMSG MSG('Your job is exceeding 30% CPU capacity') TOUSR(&OWNER)`
    - 5) 单击**启用复位**。
    - 6) 对于**阈值复位值**，指定 **< 20**（忙的程度小于 20%）。
4. 单击**收集时间间隔**选项卡，并选择 **15 秒**。这将覆盖收集服务设置。
5. 单击**操作**选项卡，并同时在**触发器**和**复位**列中选择**记录事件**。
6. 单击**服务器和组**选项卡，并选择要监控此作业的服务器和组。
7. 单击**确定**以保存新的监控器。
8. 从作业监控器列表中，右键单击新监控器并选择**启动**。

### 结果

新监控器每隔 15 秒检查一次 QINTER 子系统，并且，如果作业 MKWIDGET 正消耗大于 30% 的 CPU，则监控器将一条消息发送至作业的所有者。当作业使用的 CPU 容量小于 20% 时，监控器复位。

### 方案：消息监控器：

#### 情况

您的公司正在运行几个 iSeries 服务器，并且检查每个系统的消息队列是耗时的。作为系统管理员，当查询消息在系统中发生时，需要了解它们。

可以设置一个消息监控器来显示发生在任何一个 iSeries 系统中的消息队列的任何查询消息。一检测到消息，监控器就打开并显示消息。

## 配置示例

要设置消息监控器，需要定义希望观察的消息类型，以及当这些消息发生时，希望监控器执行什么操作。要设置达到此目的的消息监控器，完成以下步骤：

1. 在 iSeries 导航器中，展开**中央管理 > 监控器**，右键单击**消息监控器**，并选择**新建监控器...**。
2. 在**常规**页面中，为此监控器输入名称和描述。
3. 单击**消息**选项卡，并输入下列值：
  - a. 对于**要监控的消息队列**，指定 **QSYSOPR**。
  - b. 在**消息组 1** 选项卡中，为**类型**选择**查询**，并单击**添加**。
  - c. 选择按**下列消息计数触发**，并指定 **1** 条消息。
4. 单击**收集时间间隔**选项卡，并选择 **15 秒**。
5. 单击**操作**选项卡，并选择**打开监控器**。
6. 单击**系统和组**选项卡，并选择要监控查询消息的系统和组。
7. 单击**确定**以保存新的监控器。
8. 从消息监控器列表中，右键单击新监控器并选择**启动**。

## 结果

新消息监控器显示在监控的任何 iSeries 服务器中发送至 QSYSOPR 的任何查询消息。

**注：**此监控器仅响应发送至 QSYSOPR 的查询消息。然而，可以在单个监控器中包括两组不同的消息，并且可以具有几个同时运行的消息监控器。消息监控器也可以在接收到指定的消息时执行 OS/400 命令。

## 图形历史记录

图形历史提供了使用第 33 页的『收集服务』在过去的几天、几个星期、几个月或几年收集的性能数据的图形视图。不需要运行系统监控器才可查看性能数据。只要使用收集服务收集数据，就可以查看图形历史窗口。

### • 『图形历史概念』

可用于图形历史的历史数据量大大取决于收集服务中的收集保持值，并取决于是否启用第 67 页的『Performance Management/400』。参见此主题以获取用于管理和显示性能数据记录的可用选项的描述。

### • 第 67 页的『使用图形历史记录』

通过 iSeries 导航器可以访问图形历史。参见此主题以获取逐步指示信息。

有关监控系统性能的更多信息，参见第 20 页的『跟踪性能』主题。

## 图形历史概念

图形历史显示包含在由第 33 页的『收集服务』创建的收集对象中的数据。因此，可用的数据类型和数据量取决于收集服务配置。

可用于绘图的数据量是按从收集服务属性选择的设置确定的，特别是按收集保持周期确定的。使用 iSeries 导航器来对多个系统第 70 页的『激活 Performance Management/400』。当激活 PM/400 时，可以使用图形历史功能来查看几天前、几个星期前或几个月前收集的数据。您超出了实时监控能力，并具有对摘要或详细数据的访问。如果不启用 PM/400，则图形数据字段支持 1 到 7 天。启用 PM/400 时，可以定义管理收集对象在系统中保持多长时间：

- **详细数据**

在删除管理收集对象前它们保留在文件系统中的时间长度。可以选择以小时或天计的特定时间段，或可以选择**永久**。如果选择**永久**，则将不会自动删除管理收集对象。

- **图形数据**

在删除显示在图形历史窗口中的详细信息和属性数据之前它保留在系统中的时间长度。如果不启动 PM/400，则可以指定 1 到 7 天。如果启动 PM/400，则可以指定 1 到 30 天。缺省值为 1 小时。

- **摘要数据**

在删除图形的数据收集点之前它们可以显示在图形历史窗口中或保留在系统中的时间长度。没有详细信息或属性数据可用。必须启动 PM/400 才能启用摘要数据字段。缺省值为 1 个月。

## 使用图形历史记录

图形历史包括在 iSeries 导航器中。要查看正在使用第 33 页的『收集服务』监控的数据的图形历史，执行以下步骤：

1. 遵循 iSeries 导航器联机帮助，以在单个系统或系统组中启动收集服务。
2. 从**启动收集服务** — 常规页面中，选择启动 **Performance Management/400**（如果需要）。
3. 对收集保持周期的其它值进行更改。
4. 单击**确定**。
5. 可以通过右键单击系统监控器或收集服务对象并选择**图形历史**来查看图形历史。
6. 单击**刷新**以查看图形视图。

一旦已启动图形历史，一个窗口就会显示一系列绘制的收集点。图形线上的这些收集点是由与三个可用的数据级别对应的三个不同的图形标识的：

- 正方形收集点代表同时包括详细信息和属性信息的数据。
- 三角形收集点代表包含详细信息的摘要数据。
- 圆形收集点代表不包含详细信息或属性信息的数据。

## Performance Management/400

Performance Management/400 (PM/400) 已自动化且是自动管理的，从而容易使用。PM/400 自动触发“收集服务”以从服务器收集非专利性能和容量数据，并将它们发送至 IBM。所有收集站点都是网络安全的，传送时间完全在您的控制下。当发送数据至 IBM 时，您不需要自己存储所有趋势数据。IBM 会为您存储这些数据并提供一系列显示您的服务器的增长和性能的报告和图形。可以使用传统浏览器以电子方式访问报告。通过对关键性能指标不断进行分析，这些报告可以帮助您计划和管理系统资源。

IBM Operational Support Services for PM/400e 提供包括一组报告、图形和概要文件，以帮助您最大化当前应用程序和硬件的性能（通过使用性能趋势分析）。此产品也帮助您更好地理解（通过使用容量计划）您的企业趋势与所需的硬件升级（如 CPU 或磁盘）的时间安排之间的关系。通过对系统资源利用率和吞吐量数据进行趋势分析可以提供容量计划信息，可以认为这是服务器的早期警告系统。可以把 PM/400e 认作将系统的“健康”情况通知您的虚拟资源。

PM/400 占用不到百分之一的中央处理单元（CPU）。它使用大约 58 MB 的磁盘空间，这取决于硬件型号和收集时间间隔的大小。

### 第 68 页的『PM/400 概念』

了解 PM/400 可以提供的功能和益处以及重要的实现注意事项。

### 第 69 页的『配置 PM/400』

要开始使用 PM/400，必须激活它，设置允许发送数据和接收报告的传送方法，最后定制数据收集和存储。

## 第 78 页的『管理 PM/400』

既然已设置网络，可以用 PM/400 执行各种任务。

### PM/400 报告

可以将 iSeries 服务器配置为用 PM/400 直接将“收集服务”数据发送给 IBM。然后，IBM 生成几个报告，您可以在 Web 上查看这些报告，也可以由 IBM 直接发送给您。激活 PM/400 以自动生成报告不仅节省时间和资源，而且还允许您通过预测将来的增长需要来提前进行计划。

### PM/400 概念

PM/400 使用收集服务来从服务器收集非专利性能和容量数据，然后将其发送至 IBM。此信息可以包括 CPU 利用率和磁盘容量、响应时间、吞吐量、应用程序和用户使用情况。当发送数据至 IBM 时，您不需要自己存储所有趋势数据。IBM 会为您存储这些数据并提供一系列显示服务器的增长和性能的报告和图形。可以使用传统浏览器以电子方式访问报告。

#### 『PM/400 的益处』

PM/400 有助于使管理系统资源和容量计划非常容易。了解利用 PM/400 的更具体的方法。

#### 『Operational Support Services for PM/400e 产品』

PM/400 提供了范围很广的选项。使用此信息决定哪个服务组合最适合您的需要。

#### 第 69 页的『PM/400 的数据收集注意事项』

PM/400 使用第 33 页的『收集服务』来收集性能数据。了解 PM/400 和收集服务如何一起工作来提供您所需的数据。


**PM/400 的益处：** 当您使用 PM/400 服务时，您可以获得下列益处：

- **帮助您避免不幸的意外。**  
您可以避免令人失望的意外。您控制如何管理系统的增长和性能，这意味着您管理系统。而不是系统管理您。
- **节省时间。**  
通过使收集和报告性能数据的高强度且费时的任务自动执行，您就不必再执行这些任务了。此益处有利于将资源集中在管理系统和应用程序上。
- **允许您为获取最大效率提前进行计划。**  
您可以对财政需求提前进行计划以保持系统以最高效率运行。
- **提供容易理解的信息。**  
理解了信息后，当您面对高级管理被问到问题“为什么需要升级？”时，您就会很轻松
- **允许您对将来进行预测。**  
您可以基于实际的趋势信息来预测数据处理的增长。
- **允许您确定系统问题。**  
PM/400 数据使您可以确定性能瓶颈。
- **允许您帮助估计下一个升级的大小。**

您可以将 PM/400 数据上载至 Workload Estimator for iSeries  来估计下一个升级的大小。

参见 第 67 页的『Performance Management/400』以了解有关在使用 PM/400 之前需要做什么的更多信息。

**Operational Support Services for PM/400e 产品：** 您可以电子方式或以打印格式接收图形和报告。您可以每月接收到电子图形。您会每月或每季度接收到打印的图形。PM/400e 服务费用随您选择接收性能信息的频

率和选择的格式（电子方式或打印格式）而定。这些报告选项中的某些是免费的，另一些则不是免费的。每个国家或地区的营销和服务组织可以为您提供关于可用的支持机构的详细信息。有关免费和收费选项的信息，访问 PM/400e Web 站点 。

参见 第 67 页的『Performance Management/400』以了解有关在使用 PM/400 之前需要做什么的更多信息。

**PM/400 的数据收集注意事项：** 建立系统利用率、工作量和性能测量的精确趋势最重要的要求是一致性。理论上，每天应收集 24 个小时的性能数据。因为 PM/400 和“收集服务”之间的关系，您必须了解当使用 PM/400 时可能发生的隐含问题。

以下是帮助您在使用 PM/400 时定义收集的某些准则：

- **选择 QMPGDATA 库来存储数据。**

当 PM/400 是活动的时候，**存储收集的位置**字段使用缺省值 /QSYS.LIB/QMPGDATA.LIB。如果用某些其它值替换 QMPGDATA，PM/400 准时循环收集并将它更改回 QMPGDATA。如果要将数据收集至不同库，更改 PM/400 查找数据的位置。从命令行输入 **GO PM400**，选择选项 3（使用定制），并更改库名。

- **使用“收集服务”不断收集数据。**

PM/400 通过使用“收集服务”每天收集 24 个小时的数据来满足此要求。PM/400 以 15 分钟的时间间隔收集性能数据。PM/400 使用 15 分钟的时间间隔缺省值，但不更改时间间隔已设置的值。15 分钟的时间间隔是建议的时间间隔。

- **选择 Standard plus 协议概要文件。**

Standard plus 协议是收集概要文件的缺省值。收集概要文件指示收集什么数据。Standard plus 协议概要文件中的数据类别对应于“启动性能监控器”（STRPFRMON）命令上的 DATA 参数的 \*ALL 值。如果将它更改为任何其它值，PM/400 准时将它更改回。即使选择“定制”并包括所有类别，也是如此。更改立即生效。收集不会循环（除非因为其它原因需要这样做）。执行此操作以收集用于 PM/400 报告的足够信息。

- **避免在 PM/400 是活动的时候对收集参数进行临时更改。**

例如，当激活 PM/400 时，已选择**收集期间创建数据库文件**字段作为缺省值。如果更改它，则 PM/400 准时将它更改回缺省值。更改立即生效。收集不会循环（除非因为其它原因需要这样做）。

- **结束收集服务**

任何时候都可以从 iSeries 导航器结束“收集服务”。如果结束“收集服务”，当 PM/400 在运行时，以下注意事项适用：

- PM/400 调度程序在下一个小时开始时启动“收集服务”。
- 趋势计算中未包括只收集到很少数据或没有收集到数据的天数。因此，不应经常中断“收集服务”。

如果不想启动“收集服务”，可以第 80 页的『立刻关闭 PM/400』。

## 配置 PM/400

PM/400 通过收集服务使性能数据的收集自动执行。只要库驻留在基本辅助存储池（ASP）中，您就可以指定将数据放置在哪个库中。不应该将库移至独立辅助存储池中，因为可以使独立辅助存储池脱机，这会停止 PM/400 收集进程。如果库尚不存在，PM/400 会在激活期间创建库。

要开始使用 PM/400，需要执行以下任务：

### 第 70 页的『激活 Performance Management/400』

PM/400 是随 OS/400 一起交付的，但必须激活它才能使用其收集能力。

### 第 70 页的『确定要使用哪种 PM/400 传送方法』

确定您要如何发送数据。您可以使用“中央管理”来收集数据，而使用 Electronic Service Agent（超级支持）来发送数据，也可以让 PM/400 收集数据并基于 SNA 协议发送数据。

## 第 76 页的『定制 PM/400』

既然已设置网络，可能需要定制 PM/400 以满足您的需要。

**激活 Performance Management/400:** 必须启动 PM/400 才能利用它的数据收集能力。可以使用下列其中一种方法来启动 PM/400:

### 使用 iSeries 导航器

使用 iSeries 导航器来在多个系统上激活 PM/400。当激活 PM/400 时，可以使用第 66 页的『图形历史记录』功能来查看几天前、几周前或几月前收集的数据。您超出实时监控器能力。您可以访问摘要数据或详细数据。如果未启用 PM/400，则图形数据字段只支持 1 至 7 天。如果启用了 PM/400，则可以选择保持数据的时间长度。

要从 iSeries 导航器启动 PM/400，执行下列步骤:

1. 在 iSeries 导航器中，展开要启动 PM/400 的系统。
2. 展开**配置和服务**。
3. 右键单击**收集服务**。
4. 选择 **Performance Management/400**。
5. 选择**启动**。
6. 选择要启动 PM/400 的系统。
7. 单击**确定**。

### 应答 QSYSOPR 消息队列中的消息 CPAB02A

当 QSYSWRK 子系统启动时，此消息询问您是否要激活 PM/400。

1. 从基于字符的界面，用 G 应答 QSYSOPR 中的消息“要激活 PM/400 吗？(I G C)”。QSYSOPR 消息队列将接收到消息“已激活 PM/400”。
2. 更新联系信息。发出 **GO PM400** 命令并指定选项 1。

### 发出“配置 PM/400” (CFGPM400) 命令

从基于字符的界面，可以发出“配置 PM/400” (CFGPM400) 命令。

可以继续执行安装过程中的下一个步骤，它是『确定要使用哪种 PM/400 传送方法』。

有关 iSeries 性能主题的概述，参见第 1 页的『性能』。

**确定要使用哪种 PM/400 传送方法:** 自 V5R1 以来，PM/400 传送进程已利用通过中央管理执行的网络配置来设置中央系统和端点系统。然而，仍可以使用基于字符的界面来配置 PM/400。选择要使用哪种传送方法:

- 『使用基于超级支持的服务代理程序 (全球连接) 发送 PM/400 数据』  
如果选择此传送方法，需要将 PM/400 配置为通过“中央管理”来收集数据。如果服务器安装了 V4R5 或更高版本的操作系统，则对 PM/400 执行此配置 (还必须应用 Universal Connection 修订)。如果使用超级支持，应选择此方法。
- 第 71 页的『使用 SNA 协议发送数据』  
如果选择此传送方法，需要使用基于字符的界面来配置 PM/400。PM/400 收集数据并使用 SNA 来传送它。如果服务器安装了 V4R5 或更早版本的 OS/400，则对 PM/400 执行此配置。

一旦已实现要使用的传送方法，您就可以随时执行其它任务来第 78 页的『管理 PM/400』。

**使用基于超级支持的服务代理程序 (全球连接) 发送 PM/400 数据:** PM/400 使用收集服务来从服务器收集非专有的性能和容量数据。在收集了此数据之后，可以使用基于超级支持的服务代理程序来将数据发送至 IBM。

要利用这些能力，必须在服务器上安装了 V5R1 或 V5R2，或者安装了 V4R5 并应用了全球连接修订。以下是配置 PM/400 要执行的步骤：

1. 第 70 页的『激活 Performance Management/400』  
必须启动 PM/400 才能利用它的数据收集能力。
2. 设置中央管理网络。  
定义哪个服务器是中央系统和哪些服务器是端点系统。可以使用此网络层次结构来在将数据发送至 IBM 之前，将数据从端点系统发送至中央位置。
3. 与 IBM 连接以使用全球连接传送数据。  
这是中央管理要将 PM/400 数据传送至 IBM 将使用的连接。在先前发行版中，使用了基于 SNA 运行的电子客户支持（ECS）连接。当使用全球连接时，可以通过 TCP/IP 传送数据。
4. 『收集 PM/400 性能数据』  
使用中央管理库存功能来收集数据。
5. 将数据发送至 IBM。  
使用 Electronic Service Agent 来将数据发送至 IBM，该代理程序可以在中央管理层次结构中的超级支持下获得。Electronic Service Agent 使用全球连接。

也可以『使用 SNA 协议发送数据』。

一旦配置了 PM/400，就可以执行其它任务来第 78 页的『管理 PM/400』。

**收集 PM/400 性能数据：** 如果您已完成下列任务，您可以使用“中央管理”来收集 PM/400 性能数据：

1. 第 70 页的『激活 Performance Management/400』
2. 已配置通用连接
3. 已设置“中央管理”网络
4. 已验证系统上是否安装了 Electronic Service Agent。

要收集端点系统或系统组上的 PM/400 性能数据，执行下列步骤：

1. 在 iSeries 导航器中，展开**中央管理**。
2. 展开**端点系统或系统组**。
3. 右键单击端点系统或系统组，并选择**库存**。
4. 选择**收集**。
5. 选择一个或更多要收集的库存。在这种情况下，您将选择 **PM/400 性能数据**。
6. 如果当收集完成时要某个操作在中央系统上运行，从列表中选择该操作。
7. 单击**确定**以立即开始收集数据，或单击**调度**以指定何时收集数据。

一旦配置了服务器，您就可以执行其它任务来第 78 页的『管理 PM/400』。

**使用 SNA 协议发送数据：** 如果选择不利用第 70 页的『使用基于超级支持的服务代理程序（全球连接）发送 PM/400 数据』，则仍可以使用基于字符的界面来传送数据。PM/400 询问您有关配置和使用服务器的一系列问题。配置 PM/400 屏幕询问您有关您希望服务器如何发送和接收 PM/400 性能数据。进程的第一部分涉及设置您的网络。第二部分询问您希望如何传送数据。当您使用基于字符的界面时，可以使用直接拨号线路来传送数据。

执行以下步骤来使用 SNA 发送数据：

1. 第 70 页的『激活 Performance Management/400』  
必须启动 PM/400 才能利用它的数据收集能力。

## 2. 选择要使用哪个网络配置。

确定将使用哪个网络配置来传送数据。选择如何与 IBM 连接，通过使用第 77 页的『设置 PM/400 的直接拨号线路』、现有的因特网服务供应商 (ISP) 或虚拟专用网络 (VPN)。如果要使用 ISP 或 VPN，则必须配置全球连接。

如果决定使用直接拨号线路来向 IBM 报告数据，则关于如何配置网络，您有几个选择。选择哪个配置适合于您的网络，并从配置 PM/400 屏幕执行为该特定配置概括的步骤：

- 作为将数据直接发送至 IBM 的『单个服务器的 PM/400 网络』。
- 作为『主机服务器的 PM/400 网络』，这表示希望服务器来从其它服务器 (远程服务器) 接收性能数据，然后将数据转发至 IBM。主机服务器不能处于比其它服务器早的发行版级别。换句话说，主机服务器必须处于相同发行版级别或比其它服务器高的级别。
- 作为第 73 页的『远程服务器的 PM/400 网络』，这表示可以将性能数据发送至主机服务器。在配置 PM/400 屏幕上标识您需要远程服务器，然后从 PM/400 菜单中使用选项 5 (使用远程 iSeries 系统) 来定义远程服务器。

## 3. 第 73 页的『使用远程服务器』

如果选择为主机服务器设置您的网络，则需要标识那些将要把它们的数据发送至您的主机服务器的服务器。如果正在使用单个服务器或远程服务器，则可以忽略此步骤。

## 4. 第 76 页的『定制 PM/400』

在配置网络后，需要为 PM/400 软件的操作建立全局参数。如果愿意使用直接拨号线路与 IBM 连接，则需要定义 PM/400 数据电话号码。

一旦配置了服务器，就可以执行其它任务来第 78 页的『管理 PM/400』。

**单个服务器的 PM/400 网络：** 单个服务器将数据直接发送给 IBM。以下是一些步骤，仅当 PM/400 收集数据并基于 SNA 传送数据时，您才需要遵循这些步骤配置单个服务器的 PM/400。从服务器上的“配置 PM/400” (CFGPM400) 屏幕：

1. 从命令行输入 **CFGPM400**。
2. 对**将性能数据发送至 IBM** 字段指定 \*YES。
3. 对**接收性能数据**字段指定 \*NO。
4. 接受缺省库 QMPGDATA。
5. 如果对“将性能数据发送给 IBM”指定 \*YES，则会看到指示适当的通信对象是否存在的附加信息。如果这些对象不存在，PM/400 会为您创建通信对象以进行传送。对附加屏幕作出适当响应。
6. 在“使用联系信息”屏幕上输入公司的联系信息。

如果您决定单个服务器设置不是您想要的，您可以选择另一个 第 71 页的『使用 SNA 协议发送数据』。

一旦配置了服务器，您就可以执行其它任务来第 78 页的『管理 PM/400』。

**主机服务器的 PM/400 网络：** 主机服务器从其它服务器接收性能数据然后将该数据转发至 IBM。以下是一些步骤，仅当 PM/400 收集数据并基于 SNA 传送数据时才需要执行这些步骤来为主机服务器配置 PM/400。

1. 从主机服务器上的“配置 PM/400”屏幕
  - 从命令行输入 **CFGPM400**。
  - 对**将性能数据发送至 IBM** 字段指定 \*YES。
  - 对**接收性能数据**字段指定 \*NO。
  - 接受缺省库 QMPGDATA。
2. 从主机服务器上的“使用远程 iSeries 系统”屏幕



- 按 F6（创建）来标识哪些服务器将把它们的数据发送至主机服务器。
- 完成这些字段并按“执行”键。

**注：**仅当 PM/400 收集数据并基于 SNA 传送数据时，才会发生以下情况。如果具有系统网络，建议您使用 iSeries 导航器中的“全球连接”和“中央管理”来收集这些系统的数据然后传送它们。

PM/400 在从远程服务器接收到数据之后的当天就自动调度从主服务器至 IBM 的数据传送。如果自动调度不适合您的工作管理模式，可以手工调度从主服务器的数据传送。

以下是当调度数据的传送时应记住的技巧。在整个星期内，应均匀地调度至主服务器的数据传送。此操作使对主服务器的性能影响最小化。例如，在十二个服务器的网络中，您可以具有三个组，每组四个系统。可以调度每个组在星期一、星期三和星期五发送它们的数据。这将均匀地分配发送至主服务器的数据量。

如果您决定主机服务器设置不是您想要的，您可以选择另一个第 71 页的『使用 SNA 协议发送数据』。

一旦配置了服务器，您就可以执行其它任务来第 78 页的『管理 PM/400』。

**远程服务器的 PM/400 网络：** 远程服务器将其性能数据发送给主机服务器。以下是一些步骤，仅当 PM/400 收集数据并基于 SNA 传送数据时，您才需要遵循这些步骤来配置远程服务器的 PM/400。从远程服务器上的“配置 PM/400”屏幕（CFGPM400），执行下列步骤：

1. 从命令行输入 **CFGPM400**。
2. 对**将性能数据发送至 IBM** 字段指定 \*NO。
3. 对**接收性能数据**字段指定 \*NO。
4. 接受缺省库 QMPGDATA。

**注：**如果具有系统网络，建议您使用 iSeries 导航器的库存功能来收集数据然后通过“全球连接”传送这些系统的数据。

如果您决定远程服务器设置不是您想要的，您可以选择另一个第 71 页的『使用 SNA 协议发送数据』。

一旦配置了服务器，您就可以执行其它任务来第 78 页的『管理 PM/400』。

**使用远程服务器：** 在某些站点中，网络中的主机服务器将需要的性能数据发送至 IBM 进行处理。当使用主机服务器网络时，将使网络中的其它服务器将它们的性能数据发送至此主机服务器以便传送至 IBM。要设置网络以使用主机服务器，必须标识其它远程服务器并为它们的数据传送设置调度。“使用远程 iSeries 系统”屏幕使您能够定义这些其它服务器。

**注：**

1. 如果要将网络设置为远程服务器或单个服务器，则不必使用此屏幕。仅当 PM/400 收集数据并基于 SNA 传送数据时，才应执行此任务。
2. 如果具有系统网络，建议您使用 iSeries 导航器的库存功能来收集数据然后通过“全球连接”传送这些系统的数据。

遵循下列步骤来定义远程服务器：

1. 从命令行输入 **GO PM400**。
2. 从 PM/400 菜单输入 5（使用远程 iSeries 系统）并按“执行”键。您不会看到最初显示的远程服务器。必须创建新的远程位置。
3. 通过按 F6（创建）来创建新的远程位置。
4. 记录下列信息的值。使用“显示网络属性”（DSPNETA）命令来显示来自远程系统的这些值。

- 本地网络标识
- 缺省本地位置

“使用远程 iSeries 系统” 屏幕显示远程服务器的列表。此列表包括服务器的状态（活动或不活动）和每个服务器的描述。

5. 通过使用“PM/400 远程站点维护” 屏幕或“更改远程站点 iSeries” 屏幕创建或更改远程站点服务器的描述。远程位置名称在远程服务器之间必须是唯一的。

PM/400 在从远程服务器接收到数据之后的当天就自动调度从主服务器至 IBM 的数据传送。如果自动调度不适合您的工作管理模式，可以手工调度从主服务器的数据传送。要手工调度数据的传送，参见第 79 页的『使用 PM/400 调度作业』

PM/400 软件假定您已定义接收数据的服务器（主机服务器）和发送数据的服务器（远程服务器）之间的“高级对等联网”（APPN）链路。如果系统将系统值 QCRTAUT（创建缺省公共权限）设置为 \*EXCLUDE 或 \*USE，则参见第 75 页的『创建 PM/400 的设备描述』以获取关于如何定义控制器描述的信息。如果网络不满足这些假定，则参见『在非 APPN 网络中使用远程服务器』以获取关于创建设备对来支持与每个远程服务器的连接的信息。

一旦已定义远程服务器，您就可以第 76 页的『定制 PM/400』 来使用特定线路连接。

*在非 APPN 网络中使用远程服务器：* 主服务器从其它服务器接收 PM/400 数据然后将数据发送至 IBM。远程服务器将 PM/400 数据发送至主服务器。以下信息假定先前已定义所提到的控制器。

仅当 PM/400 收集数据并基于 SNA 传送数据时，才需要创建设备对以支持与每个远程服务器的连接。

1. 使用“创建设备描述（APPC）”（CRTDEVAPPC）命令。在远程服务器上，输入 CRTDEVAPPC。按 F4 键以获取对输入参数的提示，并用下列信息定义值：

#### 远程系统

DEVD(Q1PLOC)	指定设备描述的名称。
RMTLOCNAME(Q1PLOC)	指定远程位置的名称。
ONLINE(*YES)	指定当启动或重新启动系统时是否将此设备联机。
LCLLOCNAME(Q1PRMxxx)	指定本地位置名。Q1PRMxxx 与主服务器的 RMTLOCNAME 匹配，其中 xxx 对于每个远程位置是唯一的。
CTL(yyyyyy)	指定所连接的控制器的名称，其中 yyyyyy 是连接至主服务器的控制器。
MODE(Q1PMOD)	指定方式名。
APPN(*NO)	指定设备是否支持 APPN。

2. 指定主服务器的下列信息。在命令行，输入 CRTDEVAPPC。按 F4 键以获取对输入参数的提示，并用下列信息定义值：

#### 主服务器

DEV(D(Q1PRMxxx)	指定设备描述的名称。此处使用的名称与远程系统的设备描述名匹配。
RMTLOCNAME(Q1PRMxxx)	指定远程位置的名称。此处使用的名称与远程服务器的 LCLLOCNAME 值匹配，其中 xxx 对于每个远程位置是唯一的。
ONLINE(*YES)	指定当启动或重新启动系统时是否将此设备联机。
LCLLOCNAME(Q1PLOC)	指定本地位置名。此值与远程服务器的 RMTLOCNAME 匹配。
CTL(aaaaaa)	指定所连接的控制器名称，其中 aaaaaa 是连接至主服务器的控制器。
MODE(Q1PMOD)	指定方式名。
APPN(*NO)	指定设备是否支持 APPN。

3. 在定义 APPC 设备之后使设备联机（“转换配置”（VRYCFG）命令）。在远程服务器上，输入 VRYCFG。按 F4 键以获取对参数的提示。

#### 使远程系统联机

CFGOBJ(Q1PLOC)	指定配置对象。
CFGTYPE(*DEV)	指定配置对象的类型。
STATUS(*ON)	指定状态

4. 在 PM/400 菜单上输入选项 5 以添加 Q1PRMxxx 作为远程服务器。有关如何添加远程服务器的指示信息，参见第 73 页的『使用远程服务器』。

既然已完成配置 PM/400，参见第 78 页的『管理 PM/400』以了解可以用 PM/400 执行的其它任务。

**创建 PM/400 的设备描述：** 以下步骤在将“创建缺省公共权限”（QCRTAUT）系统值设置为 \*EXCLUDE 或 \*USE 的每个远程服务器上都是必需的。如果 QUSER 不具有对设备描述 Q1PLOC 的 \*CHANGE 权限，远程传送将失败。这些步骤确保不会自动创建或删除设备。

**注：** 仅当 PM/400 收集数据并基于 SNA 传送数据时，此任务才是必需的。

如果允许自动创建设备，则根据对 QCRTAUT 设置的值用 PUBLIC \*EXCLUDE 或 \*USE 权限创建设备描述。控制器控制能否自动创建或删除设备。参考下列命令来确定在系统上如何定义这些参数：

- 创建控制器描述（APPC）（CRTCTLAPPC）命令
- 更改控制器描述（APPC）（CHGCTLAPPC）命令
- 显示控制器描述（DSPCTLD）命令

对于未配置为使用 APPN 的系统，参见第 74 页的『在非 APPN 网络中使用远程服务器』以了解关于如何创建设备描述的信息。

以下信息假定远程服务器上先前已定义将用于与主机服务器通信的控制器。

在远程服务器上，重新创建设备描述 Q1PLOC:

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

DLTDEVD  DEVD(Q1PLOC)

CRTDEVAPP  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)
```

既然已完成配置 PM/400，参见第 78 页的『管理 PM/400』以了解可以用 PM/400 执行的其它任务。

**定制 PM/400:** “使用 PM/400 定制” 屏幕为您提供以下能力:

#### 建立运行 PM/400 软件的全局参数

全局参数允许您定制下列各项。有关这些字段的描述，参见联机帮助:

- 优先级限制
- 趋势与轮班调度
- 性能数据库
- 除去规范

#### 定义 PM/400 数据电话号码

在美国和加拿大以外，必须给 PM/400 提供将接收您的数据的 IBM 位置的电话号码。对于大多数位置，当您启动配置 PM/400 进程时，PM/400 会尝试为您的位置选择正确的电话号码。

#### 第 77 页的『使用 PM/400 断开与连接线路』

“PM/400 线路控制” 屏幕允许 PM/400 断开线路、传送 PM/400 数据然后使线路重新处于连接暂挂状态。

要定制全局参数，执行下列步骤:

1. 从命令行输入 **GO PM400**。
2. 从 PM/400 菜单输入 3 以显示“使用 PM/400 定制” 屏幕并按“执行” 键。

如果正在使用“收集服务”来收集 PM/400 数据，您应考虑第 69 页的『PM/400 的数据收集注意事项』。

有关可以用 PM/400 执行的其它任务，参见第 78 页的『管理 PM/400』。

**验证 PM/400 数据号码:** 如果服务器正在使用与 IBM 的直接拨号连接，则必须验证 PM/400 电话号码是否正确。电话号码还必须包含线路的正确前缀。

注：此号码仅用于 SNA 传送。

要检查电子客户支持线路的电话号码格式，执行下列步骤：

1. 输入

DSPDTAARA DTAARA(QUSRSYS/QESTELE)

并按“执行”键。

2. 确定在偏移 0 中找到的连接号码前缀。例如，如果偏移 0 是“T9:1800xxxxxxx”，则前缀是 T9:。

3. 输入

DSPDTAARA DTAARA(QUSRSYS/Q1PGTELE)

并按“执行”键。

4. 偏移 0 (零) 是将使用的拨号字符串。(将不使用其它号码。)

5. 如果使用 ECS 线路来订购 PTF，可以将偏移 0 (零) 中的格式与用于 ECS 线路的格式 CALL QESPHONE 进行比较，记下所使用的字符串，并将它与步骤 2 中找到的值进行比较。

电话号码将不同，但前缀必须相同(即，SST9:1800...、SST:1800... 等)。

如果需要更改电话号码，使用“更改数据区”(CHGDTAARA)命令：

输入 **CHGDTAARA**，其中 DTAARA 是 Q1PGTELE，LIB 是 QUSRSYS，子串开始位置是 \*ALL，新值是“SST:18005475497”

注：对于美国和加拿大，新值应是拨号前缀后跟 18005475497。

既然已完成 PM/400 配置，参见第 78 页的『管理 PM/400』以了解可以执行的任务。

**设置 PM/400 的直接拨号线路：**对于大多数位置，PM/400 会尝试为您的位置选择正确的数据电话号码。应始终第 76 页的『验证 PM/400 数据号码』是否正确。如果您具有的信息不包含 PM/400 数据电话号码和 PM/400 支持号码，与本地 IBM 支持人员联系。他们可以为您提供正确的电话号码。

注：如果正在通过“全球连接”传送数据，则不需要此电话号码。仅当使用直接拨号线路时，才需要此号码。

要定义 PM/400 数据电话号码或要更改该号码，执行下列步骤：

1. 从命令行输入 **GO PM400**。

2. 从 PM/400 菜单输入 3 以显示“使用 PM/400 定制”屏幕并按“执行”键。

3. 在此屏幕上，向前滚动直到看到屏幕中显示电话号码字段的部分。

4. 在 **IBM PM/400 电话号码** 字段中输入正确的拨号序列。对于许多 IBM 调制解调器，需要使用冒号(:) 字符表示拨号音。

**使用 PM/400 断开与连接线路：**PM/400 使用的线路有时会处于连接暂挂状态。此状态不允许 PM/400 访问线路以传送数据。“PM/400 线路控制”屏幕允许 PM/400 断开线路、传送数据然后使线路重新处于连接暂挂状态。当使用此屏幕时，可以将 PM/400 传送任务(Q1PCM1) 更改为检查线路状态和断开适当的线路。一旦完成传送，同一线路将处于连接暂挂状态。

注：仅当 PM/400 收集数据并基于 SNA 传送数据时，此任务才是必需的。

要断开或连接线路，执行下列步骤：

1. 通过从命令行输入 **PMLINMON**，启动 PM/400 线路监控功能。将会看到“PM/400 线路控制”屏幕。

2. 阅读第一个屏幕上显示的警告并按“执行”键。

3. 定义 PM/400 需要断开的线路、控制器和设备组合。
4. 使用提示要 **PM/400 自动线路控制活动吗?** 作为功能的主控制开关。如果指定是, 则该 PM/400 功能是活动的。如果指定否, 则禁用该功能。  
如果指定否, 则当指定是时不必再次定义线路控制列表。通过仅指定线路, 可以断开或连接线路。通过指定所有三个描述, 可以断开和连接线路、控制器和设备。
5. 验证所定义的线路、控制器和设备。按“执行”键来查看您的选择的摘要。
6. 按“执行”键确认您的选择, 或按 F12 键返回至先前屏幕以更改项。

也可以通过使用“配置 PM/400” (CFGPM400) 命令来设置 PM/400 线路控制。

有关可以用 PM/400 执行的附加任务, 参见『管理 PM/400』。

## 管理 PM/400

在将网络设置为使用 PM/400 之后, 可以执行下列任务:

### 『取消激活 PM/400』

了解如何能停止 PM/400。

### 第 79 页的『更改 PM/400 联系信息』

了解如何更改联系信息的初始设置。

### 第 79 页的『使用 PM/400 调度作业』

了解如何使用 PM/400 来调度作业。

### 第 80 页的『省略 PM/400 分析中的项』

了解当使用 PM/400 执行分析时如何省略作业、用户和通信线路。

### 第 80 页的『立刻关闭 PM/400』

了解如何能立即停止 PM/400。

### 第 81 页的『显示 PM/400 状态』

了解如何使用 iSeries 导航器或 PM/400 菜单来显示 PM/400 状态。

### 第 81 页的『查看 PM/400 报告』

参见 PM/400 报告的示例和如何解释那些报告的说明。

### 第 66 页的『图形历史记录』

图形历史记录提供在指定的时间段内收集的性能数据的图形视图。查明如何查看此数据。

**取消激活 PM/400:** 要使 PM/400 停止运行, 可以使用下列其中一个方法:

### 使用 iSeries 导航器

执行下列步骤:

1. 在 iSeries 导航器中, 展开 PM/400 正在其中运行的系统。
2. 展开配置和服务。
3. 右键单击收集服务。
4. 选择 **Performance Management/400**。
5. 选择停止。
6. 选择要停止 PM/400 的系统。

7. 单击**确定**。

## 使用 API

使用结束 PM/400 (Q1PENDPM) API 来取消激活 PM/400。

参见第 78 页的『管理 PM/400』以了解您可以执行的其它任务。

**更改 PM/400 联系信息:** 在 PM/400 软件的配置期间, 您已标识联系人并提供了组织的邮件信息。如果以后需要更新信息, 使用“使用联系信息”选项来更改此信息。要更改联系信息, 执行下列步骤:

1. 从命令行输入 **GO PM400**。
2. 从 PM/400 菜单输入 1, 并按“执行”键。出现“使用联系信息”屏幕。
3. 适当更改联系信息, 并按“执行”键。

有关可以执行的其它任务, 参见第 78 页的『管理 PM/400』。

**使用 PM/400 调度作业:** PM/400 软件在整体上是一个调度程序, 它自动启动支持 PM/400 性能数据收集和分析所需要的作业。

PM/400 软件激活过程的一部分包括启动 Q1PSCH 作业。此作业又启动其它作业, 如下表所示:

要访问 PM/400 已调度的作业, 执行下列操作:

1. 从命令行输入 **GO PM400**。
2. 从 PM/400 菜单输入 2 并按“执行”键。出现“使用自动调度的作业”屏幕。
3. 可以将每个作业的状态从活动更改为不活动。在要更改的作业旁边输入 2 (更改) 并按“执行”键。将显示“自动更改已调度作业”屏幕。

下表显示可能的 PM/400 作业的列表。

### PM/400 已调度的作业

作业	调度	功能
Q1PTEST	激活时	验证是否激活 PM/400 然后进入不活动状态。
Q1PCM1	每星期	将减少的性能数据传送至 IBM。仅当使用直接拨号线路时, 此作业才是活动的。
Q1PCM2	每天	使通信脱机。
Q1PPMSUB	每小时	确保“收集服务”正在收集数据。
Q1PDR	每天	执行数据简化并清除性能数据。
Q1PPG	每月	清除减少的性能数据。
Q1PCM3	需要时	在直接拨号传送未能断开线路之后, 使通信脱机。

Q1PCM4	需要时	从远程服务器访问 PM/400 数据。仅当已通过使用 PM/400 菜单的选项 5 添加远程系统时，才会启动此作业。
Q1PPMCHK	每 4 个小时一次	验证数据收集是否是活动的。
Q1PMONTH	每月	如果在 1 个月内需要附加传送，则允许每月传送。缺省值设置为不活动的。仅当使用直接拨号线路时，此作业才是可用的。

参见第 78 页的『管理 PM/400』以了解您可以执行的其它任务。

**省略 PM/400 分析中的项：** PM/400 软件应用程序摘要包括对批处理作业、用户和通信线路的前十项的分析。然而，某些作业、用户或通信线路不适合这样的分析。例如，可能要在运行时间类别中排除长于正常运行时间的作业，如自动启动作业。

通过使用类属省略功能，可以从前十个分析中省略多组批处理作业和用户。例如，要省略以 MYAPP 开始的所有作业，指定：MYAPP\*

要使用省略，执行下列步骤：

1. 从命令行输入 **GO PM400**。
2. 从 PM/400 菜单输入 4 并按“执行”键。出现“使用前十个省略”屏幕。
3. 根据要省略的项输入适当的选项号
  - 输入 1 以使用作业。
  - 输入 2 以使用用户。
  - 输入 3 以使用通信线路。
4. 在适当的字段中输入 1 以从特定类别中省略某个用户或作业。如果是通信线路，则输入线路名称然后在适当的字段中输入 1。

参见第 78 页的『管理 PM/400』以了解您可以执行的其它任务。

**立刻关闭 PM/400：** 如果需要使 PM/400 停止验证“收集服务”是否在收集数据，可以使用调度作业来将日期更改为 Q1PPMSUB 作业的将来日期。

1. 从命令行输入 **GO PM400**。
2. 输入 2（自动使用已调度的作业）。
3. 在 Q1PPMSUB 作业旁边输入 2（更改）。
4. 将日期或时间更改为将来的日期和时间。
5. 按“执行”键。此更改将立刻使 PM/400 停止验证“收集服务”是否在收集数据。必须结束当前正在收集的内容。

**注：**在达到设置 Q1PPMSUB 作业将执行的日期和时间之后，PM/400 才会启动、循环或更改“收集服务”。

参见第 79 页的『使用 PM/400 调度作业』以了解可以用调度程序调度的其它内容。

参见第 78 页的『管理 PM/400』以了解您可以执行的其它任务。



**显示 PM/400 状态:** 可以使用服务器上的 iSeries 导航器或 PM/400 菜单来显示 PM/400 的状态。使用“Performance Management/400 状态”对话框来查看一个或多个服务器或组上的 PM/400 的整体状态。例如,将显示 PM/400 是否是活动的详细信息。使用 PM/400 菜单来查看“收集服务”状态、PM/400 调度程序状态、性能数据发行版、上次传送尝试、性能数据成员和性能数据大小。

要从 iSeries 导航器查看 PM/400 的整体状态,执行下列步骤:


1. 在 iSeries 导航器中,展开端点系统或系统组。
2. 展开**配置和服务**。
3. 右键单击**收集服务**。
4. 选择 **Performance Management/400**。
5. 选择**状态**。

要从 PM/400 菜单查看 PM/400 的详细状态,执行下列步骤:

1. 从命令行输入 **GO PM400**。
2. 从命令行输入 6 并按执行键。参见联机帮助以获取每个字段的描述。

参见第 78 页的『管理 PM/400』以了解您可以执行的其它任务。

**查看 PM/400 报告:** PM/400e 产品的输出是一组每月或每季度的管理报告和图形。PM/400e 产品具有两个用于报告的第 68 页的『Operational Support Services for PM/400e 产品』。

报告和图形的用途是使管理对服务器的当前性能和准确的增长趋势有较清楚的了解。要详细地查看每个报告和图形并了解它们的某些益处和使用,访问 PM/400 Web 站点 。

参见第 78 页的『管理 PM/400』以了解您可以执行的其它任务。

## Performance Tools

Performance Tools for iSeries 许可程序允许以各种方式分析性能数据。Performance Tools 是进行查看、报告和图解性能数据的收集工具和命令。可以使用 Performance Tools for iSeries 查看使用第 33 页的『收集服务』收集的性能数据,或使用“启动性能跟踪”(STRPFTRTC)命令查看收集的跟踪数据。然后您可以将数据汇总到报告中研究系统上的性能问题。您也可以创建性能数据的图形来查看经过的时间内的资源利用率。

Performance Tools for iSeries 包括一个基本部件和两个功能部件(管理器和代理程序)。基本部件加上一个所需的功能部件。有关“Performance Tools 的管理器和代理程序功能部件”的更多信息,参见第 82 页的『管理器和代理程序功能部件比较』主题。

### 第 82 页的『Performance Tools 概念』


描述各种工具帮助您收集和分析性能信息。查找详细的关于哪种工具正好执行哪种功能和如何工作的信息。

### 第 84 页的『安装和配置 Performance Tools』

参见此主题以获取安装和设置指示信息。

### 第 84 页的『Performance Tools 报告』

Performance Tools 报告提供关于在经过时间里已收集的数据的信息。使用这些报告来获取有关系统资源性能和使用的附加信息。

关于如何使用 Performance Tools 收集有关系统、作业或程序性能的数据的更多详细的信息，查看 Performance Tools  一书。它也说明如何分析和打印数据来帮助标识和更正任何问题。参见 HVLPTASK 任务以查明 Performance Tools 如何显示此任务所消耗的 CPU 时间。

## Performance Tools 概念

Performance Tools for iSeries 许可程序分析性能数据的两个不同类型：采样数据和跟踪数据。第 33 页的『收集服务』收集采样数据，它是在规则的时间间隔里获取的总结数据。收集采样数据进行趋势分析和性能分析。与事情有关的数据象存储池和响应时间。然而，“收集服务”不支持跟踪数据的收集。跟踪数据是详细信息数据，您收集该数据以获取关于特定作业和事务的附加信息的。要收集跟踪数据，您可以使用启动性能跟踪（STRPFTRC）命令或性能资源管理器。

### 『 Performance Tools 中提供的功能 』

“性能工具”包括用于收集、分析和报告性能数据的各种应用程序。知道哪种功能可用和哪种最适合给定的任务的可能是机器群。参见此主题以获取在此许可程序中包括的功能的描述。




### 『 管理器和代理程序功能部件比较 』

您可以使用“管理器和代理程序”功能部件通过分布式环境有效地划分 Performance Tools 所需的功能。此主题包括这两个功能部件的描述、彼此包含的功能和关于如何最有效地使用的信息。

### 第 83 页的『 显示性能信息 』

可以在 iSeries 导航器中查看系统资源利用率数据。可以查看数据、将数据绘成图表和将数据汇总为报告。在此处查找有关如何访问此功能的信息。

**Performance Tools 中提供的功能:** Performance Tools 包括报告、交互式命令和其它功能。例如，Performance Tools 包括下列各项：

工具	描述
使用系统活动（WRKSYSACT）命令	“使用系统活动命令”（WRKSYSACT）操作允许交互地操作当前运行在系统上的作业、线程和任务。WRKSYSACT 命令报告系统资源利用率，包括在每一任务基本区间（使用共享的处理池）上的 CPU 利用率。参见 HVLPTASK 任务 要查到此命令如何显示此任务消耗的 CPU 时间。
第 83 页的『 显示性能信息 』	“显示性能数据”图示接口允许查看性能数据、将数据汇总到报告中、创建显示趋势的图形和分析所有来自 iSeries Navigator 中系统性能的详细性能数据。在 Performance Tools 报告中逻辑的和有用的格式中组织“收集服务”性能数据。在 Performance Tools  一书中详细论述此工具。
第 84 页的『 Performance Tools 报告 』	Performance Tools 图解功能允许在图解格式下操作性能数据。可以交互的显示图形、或可以打印、绘图或将数据保存到图解数据格式（GDF）文件中供其它应用程序使用。在 Performance Tools  一书中详细论述此工具。
图解功能	性能资源管理器是帮助标识性能问题原因的数据收集工具。用“收集服务”或进行常规趋势分析收集的采样数据不能标识这些问题。在程序、过程、模块或方法层中使用性能资源管理器进行详细的应用程序分析。例如，可以在单个程序或过程 CPU 和 I/O 统计信息或单个对象 I/O 特征中收集跟踪数据。在 Performance Tools  一书中详细论述此工具。
第 86 页的『 性能资源管理器 』	

**管理器和代理程序功能部件比较:** 带有两个独立安装的功能部件是可用的。此主题说明了这两个功能部件之间的不同以帮助决定哪个功能部件更适于应用程序。

### “管理器”功能部件

“性能工具管理器”功能部件是个全功能包，预期用在分布式环境中的中心站点系统上或单个系统上。如果需要跟踪数据的分析、图解查看数据、实时查看系统活动或管理和跟踪系统增长，Performance Tools 许可程序的“管理器”功能部件更有用。

### “代理程序”功能部件

带有“管理器”功能子集的“性能工具代理程序”功能部件是个带有更多基本功能的廉价软件包。在分布式环境中，“代理程序”功能部件对在网络中管理的系统运作良好，因为如果需要详细的分析，可以将数据发送到“管理器”。对于需要自供的适当层但没有可用的专业技术的站点，它也是个有效的工具。

Performance Tools 的“代理程序”功能部件提供简化收集、管理、在线显示、数据缩减和性能数据分析的功能。第 86 页的『性能资源管理器』报告功能及其相关命令包含在 iSeries 许可程序的“性能工具”中的基本选项中，对于“管理器”功能部件或“代理程序”功能部件都可用。未包含在“代理程序”功能部件中的 Performance Tools 的主要功能是性能和跟踪报告、性能应用程序（作业跟踪和选择文件和访问组），系统活动监控和性能图示。

**显示性能信息：** Performance Tools 可以显示来自 iSeries Navigator 的性能数据。从此图解接口可以查看性能数据、将数据汇总到报告中、创建显示趋势的图形和分析系统性能的信息。

### 度量

iSeries 导航器显示用选择的时间间隔进行的性能度量。在“显示器性能数据”窗口的图形窗格中可以查看的性能度量包括：

- 事务计数
- 事务响应时间
- 总 CPU 利用率
- 交互式 CPU 利用率
- 批处理 CPU 利用率
- 高磁盘利用率
- 机器池缺页故障/秒
- 用户池缺页故障/秒
- 异常

“详细信息”窗格允许用选择的时间间隔以不同的方式查看详细的性能数据。要分析系统性能，可以查看作业数据、子系统数据、池数据或磁盘单元数据。

### 报告

除了查看图形和详细的数据外，也可以打印来自“显示器性能数据”窗口的报告。性能报告允许您研究引起性能问题的系统的区域。可以运行不同的报告了解系统资源正在何处使用。仅当在中心系统上安装 Performance Tools for iSeries (5722-PT1) 的选项 1（管理器功能部件）时，Performance Tools 中的打印报告才可用。“管理器”功能部件上更多的信息参见第 82 页的『管理器和代理程序功能部件比较』主题。

来自“显示器性能数据”窗口的可以打印的报告包括：

- 系统
- 组件
- 作业
- 池
- 资源

## 通过 iSeries 导航器访问

通过 iSeries 导航器执行下面步骤可以访问“显示器性能数据”窗口。

1. 在 iSeries 导航器中，展开**我的连接**（或活动的环境）。
2. 展开包含要查看的性能数据的服务器。
3. 展开**配置和服务**。
4. 右键单击**收集服务**，选择 **Performance Tools**，然后选择**性能数据**。
5. 选择要显示的性能数据文件。
6. 单击**显示**。

有关如何使用 iSeries 导航器中的“显示性能数据”窗口的更多信息，参见 iSeries 导航器联机帮助。

## 安装和配置 Performance Tools

要安装 Performance Tools 需要用户概要文件来保存系统 (\*SAVSYS) 权限。可以使用系统操作员概要文件得到此权限。

Performance Tools 必须在名为 QPFR 的库中运行。如果系统上有此名的库，在安装 Performance Tools 之前，使用“重命名对象 (RNMOBJ)”命令来重命名此库名。此步骤将确保 Performance Tools 的适当操作。

使用下面命令将 Performance Tools 放入库 QPFR 中：

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

然后必须执行下面中的一步：

- 如果您已经购买了“管理器”功能部件，使用下面命令：  
RSTLICPGM LICPGM(5722PT1) DEV(*tape-device-name*) OPTION(1)
- 如果您已经购买了“代理程序”功能部件，使用下面命令：  
RSTLICPGM LICPGM(5722PT1) DEV(NAME) OPTION(2)

如果您要安装几个 CD-ROM，下面的情形可能会出现。安装第一个之后，可能接收告知许可程序已恢复但未恢复语言对象的消息。如果这样发生，插入下一个 CD-ROM 并如下输入：

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


安装 Performance Tools 程序的另一个方法是输入 GO LICPGM 并使用菜单选项。

Performance Tools 基于处理器的程序。使用类型是并发的，程序这置使用界限 \*NOMAX。

在 Performance Tools  一书中详细论述了此程序。

## Performance Tools 报告

Performance Tools 提供一种查看所收集数据的简易方法。第 33 页的『收集服务』为大多数具有“事务”异常的 Performance Tools 报告、锁定和跟踪报告提供数据。必须使用 STRPFTRC 和 ENDPFTRC 命令来收集第 23 页的『收集有关应用程序的性能的信息』以获取这三个报告。使用 Performance Tools 报告，可以隔离性能问题。在收集一段时间的性能数据之后，可以打印报告来查看正在使用系统资源的方式和位置。此报告可以将您定向至导致减慢全部响应时间的特定应用程序、用户或无效工作量。

以下列表描述了每个报告并给出了关于使用特定报告的原因的简要概述。Performance Tools  一书详细讨论了每个报告。

注：可以打印先前发行版中使用“启动性能监控器”（STRPFRMON）命令收集的数据的报告。

### Performance Tools 报告的概述

报告	描述	显示哪些内容	如何使用信息
系统	使用“收集服务”数据来提供系统正在运行的方式的概述。该报告包含关于工作量、资源使用、存储池利用率、磁盘利用率和通信的摘要信息。运行和打印此报告经常给予您系统使用的想法。	系统工作量。该报告包括数据库能力数据。	工作量预测
组件	使用“收集服务”数据来提供关于与“系统报告”相同的系统性能组件的信息，但详细信息的级别更大。此报告帮助您查找哪些作业正在消耗较大数量的系统资源，如 CPU、磁盘等等。	资源使用、通信、系统和用户作业。报告包括数据库能力数据和“交互式功能部件”利用率。	硬件增长和配置处理趋势
事务	使用跟踪数据来提供关于性能数据收集期间发生的事务的信息。	CPU 的工作量和利用率、磁盘、主存储器、事务工作量和对象争用	工作量预测、池配置、应用程序设计、文件争用和程序使用
锁定	使用跟踪数据来提供关于系统操作期间的锁定和占有冲突的信息。使用此信息，可以确定处理期间是否因为不满足的锁定请求或内部机器占有冲突而正在延迟作业。这些情况也称为等待。如果它们发生，您可以确定作业在等待哪些对象和等待的时间长度。	文件、记录或按时间的对象争用；挂起的作业或对象名称；发出请求的作业或对象名称	问题分析。减少或排除对象争用。
跟踪	使用跟踪数据来显示所跟踪的不同作业类型（例如，批处理作业）的时间进度。报告利用的资源、异常和状态过渡。	作业类时间片结束和跟踪数据	问题分析和批处理作业进度
作业	使用“收集服务”数据来显示关于全部或选择的时间间隔和作业的信息，包括交互式作业和非交互式作业的详细信息和摘要信息。因为报告可能较长，所以您可能要通过选择要包括的时间间隔和作业来限制输出。	按时间间隔的作业	作业数据

池	使用“收集服务”数据来提供关于子系统活动的部分和关于池的部分。将对每个采样时间间隔显示数据。因为报告可能较长，所以您可能要通过选择要包括的时间间隔和作业来限制输出。	按时间间隔的资源	池数据
资源时间间隔	使用“收集服务”数据来提供全部或选择的时间间隔的资源信息。因为报告可能较长，所以您可能要通过选择要包括的时间间隔来限制输出。	按时间间隔的资源	系统资源使用

性能资源管理器和“收集服务”是独立的收集代理程序。每一个都生成它自己的数据库文件集，这些文件包含已分组的收集数据集。您可以同时运行两个收集。

有关其它工具的报告列表，参见下列各项：

- 第 91 页的『性能资源管理器报告』
- Performance Management/400 报告

## 性能资源管理器

性能资源管理器是一种数据收集工具，它帮助用户确定不能通过使用“收集服务”来收集数据或进行一般趋势分析来确定的性能问题原因。使用性能资源管理器包括以下两个原因：

- 找出引起问题的系统资源、应用程序、过程或方法的性能问题
- 分析应用程序的性能

性能资源管理器的收集功能和相关命令是 OS/400 许可程序的一部分。报告功能及其相关命令是 Performance Tools for iSeries 许可程序中的基本选项的一部分，因此它们对于管理器功能部件和代理程序功能部件是可用的。

AS/400 Performance Explorer Tips and Techniques  一书提供了性能资源管理器功能的附加实例和增强型性能资源管理器跟踪支持的实例。

性能资源管理器是帮助查找性能问题原因的工具，这些原因不能使用进行一般性能监控的工具确定。当计算机环境在大小和复杂性两方面都增长时，性能分析从复杂性方面着手是比较合理的。性能资源管理器通过收集关于复杂的性能问题的数据来解决复杂性方面的增长。

**注：**性能资源管理器是在已尝试其它工具以后需要使用的工具。它收集特定格式的数据，这些数据可以容易找出与性能问题有关的因素；然而，当收集此数据时，可能会显著影响系统的性能。

此工具是为对了解或提高程序的性能感兴趣的应用程序开发者设计的。对性能管理很了解的用户也可以使用它来帮助标识和找出复杂的性能问题。

要了解有关性能资源管理器的更多信息，参考下列其中任何一个性能资源管理器主题：

### 第 87 页的『性能资源管理器概念』

性能资源管理器收集有关指定的系统进程或资源的详细信息。此主题解释性能资源管理器的工作方式和使用它的益处。

## 第 92 页的『配置性能资源管理器』

要收集详细的跟踪信息，您需要定制性能资源管理器，以便与执行跟踪的应用程序进程以最佳方式一起工作。


## 第 91 页的『性能资源管理器报告』

在您用性能资源管理器会话收集性能数据之后，可以通过运行包括的报告或直接查询数据库文件来查看它。

有关更详细的信息，参考 Performance Tools  一书。

## 性能资源管理器概念

与第 33 页的『收集服务』一样，性能资源管理器收集数据用于以后进行分析。然而，它们收集完全不同类型的的数据。“收集服务”以定期的调度时间间隔收集范围很广的系统数据，而消耗最少的系统资源。相反，性能资源管理器启动收集跟踪级别数据的会话。此跟踪生成大量关于应用程序、作业或线程所消耗的资源的信息。确定地说，可以使用“性能资源管理器”回答特定的关于区域的问题，象生成的系统磁盘 I/O，过程调

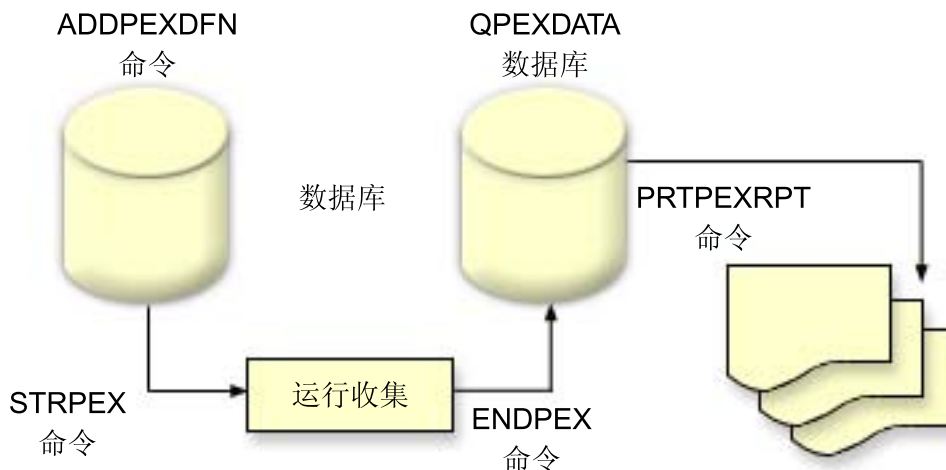
用，Java 方法，缺页故障和其它跟踪事件 。是选择完全特定的和非常详细的信息的能力使性能资源管理器在帮助隔离性能问题上更有效。例如，“收集服务”可能告知磁盘存储空间正迅速地消耗。可以使用性能资源管理器标识什么程序和对象在消耗太多的磁盘空间和为什么如此。

注：在同一时间里可以收集性能资源管理器数据和“收集服务”数据。

## 性能资源管理器如何工作

下面数字将帮助您通过性能资源管理器熟悉标准的路径。这些步骤的每步的详细信息，参见第 92 页的『配置性能资源管理器』。数字显示包含下列步骤的基本的工作环节：

1. 定义性能资源管理器数据收集。您也可以添加过滤器以限制通过为特定事件指定比较值来收集的数据量。
2. 启性能资源管理器收集基于您定义的数据。
3. 运行程序、命令或工作量。
4. 结束收集，这将把收集的数据保存到数据库文件集中。
5. 创建和打印数据库文件中的报告。



要了解有关性能资源管理器的更多信息，参考下列其中任何一个性能资源管理器主题：

#### 『性能资源管理器定义』

使用性能管理器定义配置和存储确定性能资源管理器收集何数据和如何收集的参数和条件。此主题说明如何使用这些定义并提供举例说明简单定义的样本。

#### 第 90 页的『性能资源管理器数据库文件』

将性能资源管理器收集的数据存储在性能资源管理器数据库文件中。

#### 第 92 页的『性能资源管理器的益处』

性能资源管理器包含可以帮助收集和分析详细的性能信息的各种功能。此主题提供对那些各种功能的一般概述。

**性能资源管理器定义：** 要收集性能资源管理器数据，您需要告知性能资源管理器要收集哪些数据。为此，使用添加性能资源管理器定义（ADDPEXDFN）命令来创建性能资源管理器定义。在完成并保存此定义后，您就可以继续执行工作循环中的下一个任务。

创建新定义之前，考虑您想要何种信息和需要的详细信息量。性能资源管理器提供下列类型的数据收集：

#### 统计信息类型定义

标识使用过多 CPU 或执行磁盘 I/O 操作太多的应用程序和 IBM 程序或模块。一般情况下，使用统计信息类型来标识因存在潜在的性能瓶颈而应进一步调查的程序。

- OS/400 程序、过程和 MI 机器群指令的第一个订单分析的益处。
  - 给出调用数
  - 给出直接插入和累积的 CPU 使用（毫秒）
  - 给出直接插入和累积的同步和异步 I/O 的数目
  - 给出已进行的调用次数
- 非常适合于短期或长期运行
- 收集的数据的大小相当小，对于所有运行是个常量
- 由于经常调用，ILE 过程的运行时收集开销可能是一个问题。尽管运行时间减少，但收集的统计信息仍是准确的，因为性能资源管理器从数据中除去了大多数收集开销。
- 使用组合的或单独的数据区。ADDPEXDFN 命令中的 MRGJOB 参数指定是在一个数据区累积所有程序的统计信息还是将它们单独保存（例如，每个作业一个数据区）。

可以将统计信息以分层或平铺方式进行组织。

- 分层结构将统计信息组织成树格式，树中的每个节点代表作业或任务运行的一个程序过程。
- 平铺结构将统计信息组织成程序或过程的简单列表，每个程序或过程使用自己的一组统计信息。

以下是性能资源管理器统计信息定义的示例，称为 MYSTATS，它将在每个程序或过程级别显示 CPU 和磁盘资源使用率。

```
ADDPEXDFN DFN(MYSTATS) /* The name of the definition. */
TYPE(*STATS) /* The type of definition */
JOB(*ALL) /*All Jobs */
TASKS(*ALL) /*All tasks */
MRGJOB(*YES) /* Merge records to reduce collection overhead */
DTAORG(*FLAT) /* Do not keep track of who calls who */
```



## 概要文件类型定义

基于源程序语句号标识消耗过多 CPU 利用率的高级语言 (HLL) 程序。也可以标识经常在程序的开始和程序结束处的子例程之间分支的程序。如果程序足够大, 经常进行这种前后转移会在具有受限制主存储器的系统中引起过高的缺页故障率。

- 程序概要文件 (在 ADDPEXDFN 命令中指定 TYPE(\*PROFILE) 和 PRFTYPE(\*PGM))
  - 给出详细的中断情况, 您在特定作业的一组程序中在该中断处正在花费很多时间。
  - 可以通过程序、模块、过程、语句或指令来总结数据。
  - 收集的大小相当小且经常与运行的长度无关。
  - 16 个 MI 程序的限制意味着您应使用它作为第二个订单分析工具。
  - 可以通过更改采样时间间隔来改变开销。2 毫秒的时间间隔可能是基准的较好的第一选择。
  - 由于指定的程序数目或指定的程序大小不同, 所以对窗格大小没有限制。

以下是性能资源管理器程序概要文件定义的示例, 称为 PGMPROF, 它将显示特定过程的使用。

```
ADDPEXDFN DFN(PGMPROF) /* The name of the definition. */
TYPE(*PROFILE) /* The type of definition */
JOB(*ALL) /*All Jobs */
PGM((MYLIB/MYPGM MYMODULE MYPROCEDURE)) /* The name of the program to monitor. */
INTERVAL(1) /* A 1-millisecond sample will be taken. */
```

- 作业概要文件 (在 ADDPEXDFN 命令上指定以下内容: TYPE(\*PROFILE) 和 PRFTYPE(\*JOB))
  - 给出详细的中断情况, 您在收集的一组作业或任务中在该中断处正在花费很多时间。
  - 收集大小相对较小但不是常量。大小随运行的时间长度增加而增加。
  - 可以将系统上的所有作业和任务记入概要文件, 也可以将收集数据的作用域减小为关心的一些作业或任务。
  - 可以通过更改采样时间间隔来改变开销。2 毫秒的时间间隔可能是基准测试的较好首选。

以下是性能资源管理器作业概要文件定义的示例, 称为 ALLJOBPROF, 它将显示所有作业的使用。

```
ADDPEXDFN DFN(ALLJOBPROF) /* The name of the definition. */
TYPE(*PROFILE) /* The type of definition */
PRFTYPE(*JOB) /* A job profile type will be monitored. */
JOB(*ALL) /*All Jobs */
TASKS(*ALL) /*All tasks */
INTERVAL(1) /* A 1-millisecond sample will be taken. */
```

## 跟踪定义

收集系统上由一个或多个作业生成的性能活动的历史记录跟踪。跟踪类型收集关于事件发生时间以及发生次序的特定信息。跟踪类型收集有关程序、许可内码 (LIC) 任务、OS/400 作业和对象参考信息的详细参考信息。

- 一些公共跟踪事件是:
  - 程序和过程的调用和返回
  - 存储, 例如, 分配和解除分配。
  - 磁盘 I/O, 例如, 读操作和写操作。
  - Java 方法, 例如, 入口和出口。
  - Java, 例如, 对象创建和垃圾收集。
  - 日志, 例如, 启动提交和结束提交。
  - 同步, 例如, 互斥锁定和解锁或信号等待。
  - 通信, 例如, TCP、IP 或 UDP。

- 长时间运行收集更多数据。

以下是性能资源管理器跟踪定义的示例，称为 DISKTRACE，它将显示所有磁盘事件的使用。

```
ADDPEXDFN DFN(DISKTRACE) /* The name of the definition. */
TYPE(*TRACE) /* The type of definition */
JOB(*ALL) /*All Jobs */
TASKS(*ALL) /*All tasks */
TRCTYPE(*SLTEVT) /* Only selected individual events and machine instructions
are included in the trace definition */
SLTEVT(*YES) /* *SLTEVT allows you to specify individual machine instructions
and events to be specified in addition to the categories of events
available with the TRCTYPE parameter. */
DSKEVT(*ALL) /* All disk events are to be traced. */
```


**性能资源管理器数据库文件：** 下表显示当使用数据收集命令时系统收集的性能资源管理器数据文件。接如下方式输入“显示文件字段描述”（DSPFFD）命令以查看单个文件的内容：

DSPFFD FILE(xxxxxxxx)

其中 xxxxxxxx 是要显示的文件名称。

包含在文件中的信息的类型	文件名
参考信息	QAYPEREF
一般信息	QAYPERUNI
PMC 选择	QAYPEFQCFG
基本配置信息	QAYPECFG I
收集所使用的机器接口（MI）机器群指令	QAYPELCPLX
收集所使用的作业	QAYPELJOB
收集数据所使用的度量	QAYPELMET
收集所使用的机器接口（MI）程序、模块或过程	QAYPELMI
收集数据所使用的许可内码（LIC）模块	QAYPELLIC
收集数据所使用的任务名	QAYPELNAMT
收集数据所使用的任务号	QAYPELNUMT
机器接口（MI）机器群指令映射	QAYPEMICPX
事件类型和子类型映射	QAYPEEVENT
硬件映射数据	QAYPEHWMAP
许可内码（LIC）地址解析映射	QAYPEPROCI
段地址解析映射	QAYPESEGI
进程和任务解析映射	QAYPETASKI
所有事件的公共跟踪数据	QAYPETIDX
辅助存储器管理事件数据	QAYPEASM
基本事件数据	QAYPEBASE
磁盘事件数据	QAYPEDASD
磁盘服务器事件数据	QAYPEDSRV
缺页故障事件数据	QAYPEPGFLT
资源管理进程事件数据	QAYPERMPM
资源管理占用锁定事件数据	QAYPERMSL
Advanced 36 事件数据	QAYPES36
段地址范围（SAR）数据	QAYPESAR

包含在文件中的信息的类型	文件名
未知事件数据	QAYPEUNKWN
基本统计信息数据	QAYPESTATS
统计信息概要分析摘要数据	QAYPEPSUM
许可内码 (LIC) 链组数据	QAYPELBRKT
机器接口 (MI) 用户事件数据	QAYPEMIUSR
机器接口 (MI) 程序链组数据	QAYPEMBRKT
机器接口 (MI) 指针的地址	QAYPEMIPTR
用户定义的链组 hook 数据	QAYPEUSRDF
硬件监控器数据	QAYPEHMON
硬件监控器总数据	QAYPEHTOT
发行版、版本和修订版级别	QRLVRM
性能资源管理器级别指示符	QRLVL
性能资源管理器 Java 事件数据	QAYPEJVA
性能资源管理器 Java 类信息数据	QAYPEJVCI
性能资源管理器 Java 方法信息数据	QAYPEJVMI
性能资源管理器 Java 名称信息数据	QAYPEJVNI
同步事件数据	QAYPESYNC
通信事件数据	QAYPECMN
文件服务事件数据	QAYPEFILSV
堆事件数据	QAYPEHEAP
PASE 事件数据	QAYEPASE
跟踪作业等价事件数据	QAYPETBRKT
任务切换事件数据	QAYPETSWSW
同步事件数据	QAYPESYNC
程序概要文件数据	QAYPEPPANE

**性能资源管理器报告：** 性能资源管理器收集关于程序或作业的行为和性能的详细信息，并将此信息存储在 90 页的『性能资源管理器数据库文件』中。可以使用 SQL 或通过运行几个报告之一来查询这些文件。可以使用性能资源管理器生成四种不同的报告：统计信息报告、概要文件报告、跟踪报告和基本报告。有关为何使用特定定义来生成这些报告之一的信息，参见第 88 页的『性能资源管理器定义』。Performance Tools  一书详细讨论了每个报告。

通过使用打印性能资源管理器报告 (PRTPEXRPT) 命令，可以创建并打印性能资源管理器报告。当要定制“跟踪报告”时，使用 OUTFILE 参数。以下命令是每个类型的性能资源管理器数据的打印报告的示例：

- 打印 \*STATS 报告，按使用的 CPU 时间排序

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

- 打印由过程总结的概要文件报告

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

- 打印按任务标识排序的跟踪

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

性能资源管理器将收集的数据存储在 QAVPETRCI 文件中，该文件位于 QPFR 库中。输入以下命令来查看单个记录的内容：

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

**性能资源管理器的益处：** 性能资源管理器有利于需要对 iSeries 服务器进行详细的性能分析的用户。使用性能资源管理器，可以：

- 确定系统上引起性能问题的原因，下至用户级别、作业、文件、对象、线程、任务、程序、模块、过程、语句或指令地址。
- 收集关于用户开发的软件和系统软件的性能信息。
- 对一个作业进行详细的分析而不会影响系统上其它操作的性能。
- 在收集数据所在的系统以外的系统上分析数据。例如，如果在网络中的某个受管系统上收集数据，则可以将数据发送到中央站点系统进行分析。

## 配置性能资源管理器

要配置性能资源管理器，遵循以下步骤：

1. 第 88 页的『性能资源管理器定义』在显示“添加性能资源管理器定义 (ADDPEXDFN) 上，为定义指定收集类型和名称。用此名将定义作为数据库成员存储在库 QUSRSYS 的 QAPEXDFN 文件中。指定的名称用在”启动性能资源管理器 (STRPEX) 命令中。
2. 添加过滤器 (ADDPEXFTR 命令)。性能资源管理器过滤器标识在性能资源管理器会话期间收集的性能数据，意思是通过为特定事件指定比较值来限制收集的数据数量。
3. 开始收集数据 (STRPEX 命令)。如果没在收集 \*PMCO 事件，一个作业可以在多于一个性能资源管理器中收集。如果正在收集 \*PMCO 事件，那么只要所有的收集有相同的时间间隔规范，作业就可以在多于一个性能资源管理器中收集。
4. 为想要分析的数据运行命令、程序或工作量。
5. 『结束性能资源管理器』将其保存到数据库文件中进行分析。使用“结束性能资源管理器 (ENDPEX) 命令停止收集。
6. 分析性能数据。打印性能资源管理器报告 (PRTPEXRPT) 命令，包含在 Performance Tools 许可程序中，为每种数据的类型提供唯一报告 (统计信息、概要文件、跟踪概要文件或跟踪)。用于分析的其它选项是基于此组数据库文件编写自己的查询。

所有的性能资源管理器命令都可以用下面方法之一访问：

- 命令接口。从命令行输入命令。所有命令都是 OS/400 操作系统的一部分，除了“打印性能资源管理系统”命令。
- Performance Tools 菜单选项。

要查看性能资源管理器工作循环，参见第 87 页的『性能资源管理器概念』。

## 结束性能资源管理器

要结束性能资源管理器会话，使用“结束性能资源管理器” (ENDPEX) 命令。ENDPEX 命令对收集的数据执行下列操作：

- 将收集的数据放入指定的库的 QAYPExxx 文件中。  
使用 OPTION(\*END) 和 DTAOPT(\*LIB) 来执行此操作。除非为 DTAMBR 参数指定了名称，否则所有 QAYPExxx 文件的数据库成员名都将使用会话名作为缺省值。您可以指定 RPLDTA(\*NO) 来擦除使用此会话名收集的数据，也可以指定 RPLDTA(\*YES) 来将收集的数据添加到现有的数据中。除非您是一个很有经验的用户，否则使用 RPLDTA(\*NO)。

- 将收集的数据放入 IBM 定义的单个文件中。  
使用 OPTION(\*END) 和 DTAOPT(\*MGTCOL) 来执行此操作。一般情况下，您只应在 IBM 服务代表的指导下使用 \*MGTCOL。在 DTAOPT 参数中指定 \*MGTCOL 值会将收集信息保存到管理收集对象中。仅当打算将数据交付给 IBM 时，才应该使用管理收集对象选项。Performance Tools 只能分析数据库文件。
- 废弃收集的数据。  
如果要保存数据则使用 OPTION(\*END)，要废弃任何已收集的数据则使用 DTAOPT(\*DLT)。当您确定不能使用收集的数据时，应执行此操作。例如，暂挂的作业之一不能如期启动。如果您选择 \*DLT 选项，将永远不会保存会话收集的性能数据。
- 暂挂收集会话，但不结束它。  
使用 OPTION(\*SUSPEND) 来执行此操作。通过对特定的会话标识发出带 OPTION (\*RESUME) 的 STRPEX 命令，您可以稍后再次启动数据收集。

注：如果您忘记活动的收集会话名，则使用 ENDPEX SSNID(\*SELECT) 命令。

## iDoctor for iSeries

iDoctor for iSeries 是一组由三个组件组成的工具：作业查看器、对象资源管理器和性能资源管理器分析器。

作业查看器和性能资源管理器分析器用于性能分析。这些组件的服务器端部分由各种数据收集和分析程序组成，这些程序设计成以更有用的格式合并性能数据。作业查看器和分析器的客户端组件由图形界面组成，以使用灵活的图形和表视图显示服务器数据。

对象资源管理器是一种设计成使 iSeries 数据更易于使用图形界面访问的工具。系统中的任何一个对象类型都可以列示、显示和描述。对象资源管理器允许您访问系统中的任何物理文件或逻辑文件，并使用易于使用的查询定义和图形定义接口针对数据创建您自己的查询和图形。

### 作业查看器

作业查看器显示实时表和图形数据，它们以很详细的方式表示作业正在执行的操作及其不运行的原因。作业查看器提供几个不同的报告，它们按时间间隔提供详细的作业统计信息。这些统计信息允许您确定一些情况，如 CPU 利用率、DASD 计数器数、等待数、故障数、调用堆栈信息、冲突信息等等。

### 对象资源管理器

对象资源管理器允许您浏览和使用 iSeries 服务器中的对象。对象资源管理器的主要功能部件是数据查看器，它让您显示系统中的任何物理文件的内容，并针对数据编写您自己的查询和图形。下列功能部件也包括在对象资源管理器中：

- 库和对象列表的方便而快速过滤
- 通过快速、易于使用的用户界面复制 (CRTDUPOBJ)、剪切 (MOV OBJ)、重命名 (RENOBJ) 和删除 (DLT\*) 对象的能力
- 浏览逻辑或物理文件成员
- 定义您自己的查询并保存它们以便以后通过查询定义接口使用
- 定义您自己的图形并保存它们以便以后通过图形定义接口使用

### 性能资源管理器分析器

性能资源管理器分析器评估系统的总体性能并使用 Performance Tools 许可程序对已执行的操作进行构建。分析器将跟踪数据卷浓缩为可以绘图或查看的报告，以帮助隔离性能问题并减少整个问题确定时间。分析器提供了一种易于使用的图形界面，以便分析 CPU 利用率、物理磁盘操作、逻辑磁盘输入/输出、数据区和数据队列。分析器还可以帮助您找出应用程序减速的原因。

访问 iDoctor for iSeries  Web 站点以获取更多信息。

## Performance Trace Data Visualizer (PTDV)

Performance Trace Data Visualizer (PTDV) 是一个 Java 应用程序，可能使用它对在 iSeries 服务器上运行的应用程序进行性能分析。PTDV 与 OS/400 基本操作系统中的性能资源管理器一起工作，以允许分析员查看程序流并获取由跟踪、作业、线程和过程总结的详细信息（如 CPU 时间、当前系统时间、循环次数和指令数）。当监视 Java 应用程序跟踪时，可以显示附加详细信息（如创建的对象数目和类型）和有关 Java 锁定行为的信息。还支持由 WebSphere Application Server 生成的性能资源管理器事件。PTDV 允许排序列、导出数据并将许多级别的数据进行汇总。

有关更多信息，请转至 [Performance Trace Data Visualizer !\[\]\(5eb1325dfdc3f1cad8426726c0db51cd\_img.jpg\) Web 站点](#)。

## 性能管理 API

性能管理 API 允许您使用收集服务、性能收集器、性能资源管理器和 Performance Management/400 (PM/400) 来收集和管理性能数据。

性能管理 API 包括：

- 收集服务 API
- 性能收集器 API
- 性能资源管理器 (PEX) API
- Performance Management/400 (PM/400) API

## “使用”有关 OS/400 性能的命令

OS/400 包括许多命令，它们可以允许您从基于字符的界面对性能数据执行实时监控。可以使用这些命令回答有关系统性能的特殊问题和帮助诸调系统。有关 iSeries 导航器实时监控的信息，参见第 60 页的『iSeries 导航器监控器』。

命令	功能
使用活动作业 (WRKACTJOB)	允许复查和更改在系统上运行的作业的属性 and 资源利用率。
使用磁盘状态 (WRKDSKSTS)	显示系统磁盘单元的性能信息和属性。
使用系统状态 (WRKSYSSTS)	提供当前系统活动的概述。确定地说，它显示系统中的作业和存储池利用率信息的数目。
使用系统活动 (WRKSYSACT)	使用系统中作业和任务。此命令是 Performance Tools 许可程序的一部分 (PT1)。
使用对象锁 (WRKOBJLCK)	使用对特定对象显示锁定，包括锁定等待应用。
使用共享存储池 (WRKSHRPOOL) 命令	显示利用率信息和更改共享存储池 (包括机器和基本部件池) 的属性。

## 扩展的自适应高速缓存

注：功能部件代码 #4331 和 #6831 (CCIN #6731) 已从市场上撤销。此处提供的信息是对于现有用户的参考。任何对 iSeries 存储 I/O 的此高速缓存功能部件的未来功能增强都将通过这些页面获得。

使用扩展自适应高速缓存改进 iSeries 系统性能！扩展自适应高速缓存是一种高级大量读高速缓存技术，它通过减少从磁盘读取的物理 I/O 请求数目来同时改进 I/O 子系统和系统响应时间。扩展自适应高速缓存生成数据的统计信息，然后使用管理策略的混合来确定哪些数据要进行高速缓存。扩展自适应高速缓存已证明对许多类型的工作量都非常有效。

IBM 提供了一种创新工具，它允许您确定扩展自适应高速缓存在您的 iSeries 计算环境中可以提供的好处。通过收集服务功能在 iSeries 导航器中激活**扩展自适应高速缓存模拟器**。模拟器显示在过去的时间里在每个磁盘的基础上对实际工作量模拟的性能结果。扩展自适应高速缓存模拟器在 I/O 存储适配器级别执行，并使用管理扩展自适应高速缓存的相同算法。

要了解更多信息，继续阅读：

- 『扩展的自适应高速缓存概念』  
探索扩展自适应高速缓存。在开始使用此工具前，查找有关计划、限制和重要注意事项的信息。
- 第 96 页的『扩展的自适应高速缓存模拟器』  
了解如何使用扩展自适应高速缓存来确定扩展自适应高速缓存在计算环境中可以提供的响应时间改进。
- 第 97 页的『获取扩展的自适应高速缓存』  
在已使用扩展自适应高速缓存模拟器查看此工具可以提供给您的环境的好处后，了解如何获取扩展自适应高速缓存。

## 扩展的自适应高速缓存概念

第 94 页的『扩展的自适应高速缓存』，它是一种通过减少从磁盘读取的物理 I/O 请求数目来同时改进 I/O 子系统和系统响应时间的高级读取高速缓存技术。扩展自适应高速缓存不仅改进数据库读取操作的性能，而且改进所有读取操作的性能。这包括由其它系统组件（如 Integrated xSeries Server）生成的读取操作。它还在具有设备奇偶性校验保护或 镜像保护的存储子系统中有效工作。扩展自适应高速缓存已证明对许多类型的工作量都非常有效。

## 扩展自适应高速缓存如何工作

扩展自适应高速缓存已集成到 iSeries I/O 子系统中。它在磁盘子系统控制器级别工作并不影响 iSeries 系统处理器。存储 I/O 适配器通过使用读高速缓存设置（如实体状态磁盘）以提供高速缓存内存来管理扩展自适应高速缓存。

扩展自适应高速缓存生成数据的统计信息，然后使用管理策略的混合来确定哪些数据要进行高速缓存。高速缓存管理通过使用预见性的算法在 I/O 适配器中自动执行并设计成高速缓存数据。该算法考虑主机访问预先确定的范围内的数据的时间和频率。

扩展自适应高速缓存的设计基于 iSeries 服务器的特定数据管理策略。不论磁盘是设备奇偶性校验保护、镜像还是不保护，存储在磁盘上的数据都具有在区域中发生的趋势。这表示存在主动读取数据的物理上连续的磁盘存储区域，不断写入的物理上连续的区域，同时主动读取和写入的物理上连续的区域，或不经常访问的物理上连续的存储区域。

在扩展自适应高速缓存设计中考虑此“区域”的数据。目标是高速缓存表现为具有读 / 写和只读特征的区域。表现为具有只写特征的区域（在存储子系统写高速缓存中高速缓存）仍大部分不受扩展自适应高速缓存的影响。扩展自适应高速缓存还设计成不损害按顺序写或按顺序读的大块数据的性能。在此实例中，磁盘的预取能力以及系统中的其它高速缓存确保快速响应时间。

要了解更多信息，继续阅读：

- 第 96 页的『扩展自适应高速缓存的限制和注意事项』  
查看扩展自适应高速缓存需要哪些组件并了解有关期望什么的更多信息。
- 第 96 页的『扩展的自适应高速缓存模拟器』  
了解如何使用扩展自适应高速缓存模拟器来确定扩展自适应高速缓存在计算环境中可以提供的响应时间改进。
- 第 96 页的『启动扩展自适应高速缓存』  
了解如何激活扩展自适应高速缓存。

**扩展自适应高速缓存的限制和注意事项:** 在开始使用第 95 页的『扩展的自适应高速缓存概念』之前, 应该进行某些初始计划以考虑可能与计算环境有关的任何限制或注意事项。

## 限制

要使用扩展自适应高速缓存, 系统必须具有下列几项:

- 支持扩展自适应高速缓存的一个或多个存储 I/O 适配器 (对于运行 V4R4 或更高版本的系统为 CCIN 2748, 对于运行 V5R1 或更高版本的系统为 CCIN 2778, 或对于运行最新发行版 V5R2 的系统为 CCIN 2757 (参见信息 APAR III13365))
- 要激活扩展自适应高速缓存的每个存储 I/O 适配器的读高速缓存设备 (RCD) (对于运行 V4R4 或更高版本的系统为 CCIN 6731)
- Performance Tools for iSeries 许可程序

扩展自适应高速缓存是通过 RCD 自动启用的。没有受控的打开或关闭开关。可以通过并行维护添加 RCD 而不中断系统。RCD 驻留在内部磁盘插槽中并与所有其它磁盘类型和容量一起工作。应了解扩展自适应高速缓存中的所有数据也保证位于磁盘中。在不太可能的 RCD 故障情况下, 将没有数据丢失。

关于 I/O 适配器下的其它磁盘的设备奇偶性校验保护和镜像保护, 没有使用扩展自适应高速缓存的限制。然而, 扩展自适应高速缓存不能与同一 I/O 适配器中的集成硬件磁盘压缩一起使用。最后, 扩展自适应高速缓存明确设计成补充 iSeries 专家高速缓存, 并可以与它一起使用或不与它一起使用。

## 注意事项

使用扩展自适应高速缓存允许您达到显著降低 I/O 响应时间并在大多数环境中增加系统 I/O 吞吐量。在使用高速缓存的一般情况下, 系统配置和工作量影响扩展自适应高速缓存的有效性。扩展自适应高速缓存在存储子系统级别执行。它对于该特定的子系统内的一组磁盘的数据进行高速缓存。因此, 将扩展自适应高速缓存添加到系统内最活跃和性能关键的存储子系统是合理的。扩展自适应高速缓存不认为是预取类型高速缓存, 因此将不干涉磁盘中的提前读能力。

正在活动接收 I/O 请求的磁盘存储区域越大, 有关决定何时使新数据进入高速缓存的选择的扩展自适应高速缓存越多。此自适应能力允许扩展自适应高速缓存对许多工作量类型和大小都有效。通过使用扩展自适应高速缓存模拟器, 总体高速缓存有效性从此观点最好理解。

另外, 扩展高速缓存模拟器和扩展自适应高速缓存不能同时在同一存储 I/O 适配器中活动。

一旦已了解这些限制和注意事项, 就已就绪『启动扩展自适应高速缓存』。

**启动扩展自适应高速缓存:** 要启动第 95 页的『扩展的自适应高速缓存概念』并增加系统的性能, 购买读高速缓存设备。一旦已将读高速缓存设备插入子系统内的磁盘插槽中, 就将激活扩展自适应高速缓存。没有用户控制的打开或关闭开关。扩展自适应高速缓存监控数据流量并植入读高速缓存设备大约需要 1 小时。在运行扩展自适应高速缓存 1 小时后, 系统应该显示改进的性能 (取决于当前工作量) 和增加的 I/O 吞吐量。

要查出您的 iSeries 系统是否能够使用扩展自适应高速缓存, 参见『扩展自适应高速缓存的限制和注意事项』。

## 扩展的自适应高速缓存模拟器

**注:** 功能部件代码 #4331 和 #6831 (CCIN #6731) 已从市场上撤销。此处提供的信息是对于现有用户的参考。任何对 iSeries 存储 I/O 的此高速缓存功能部件的未来功能增强都将通过这些页面获得。

使用扩展自适应高速缓存模拟器来估计第 94 页的『扩展的自适应高速缓存』。扩展自适应高速缓存模拟器是一种性能工具, 它可以确定扩展自适应高速缓存可以在系统中提供的响应时间改进。此确定可以基于系统配置和数据工作量, 并且是在购买读高速缓存设备前进行的。



扩展自适应高速缓存模拟器在收集服务中受控制，且在运行最新发行版 V5R2 并具有 CCIN 2757 I/O 适配器的系统中可用。（CCIN 2748、CCIN 2778 和 CCIN 2757 I/O 适配器是本身支持扩展自适应高速缓存的相同存储 I/O 适配器。）模拟器是灵活的，允许您模拟不同的高速缓存容量，以便更好地确定最适合特定系统和工作量需要的容量。实际读高速缓存设备容量是 1600 MB。

通过激活扩展自适应高速缓存模拟器收集到的性能信息将给予通过使用扩展自适应高速缓存可以节约的磁盘读操作数的指示信息。性能数据反映磁盘访问时间的可能改善。

## 获取扩展的自适应高速缓存

在从第 96 页的『扩展的自适应高速缓存模拟器』获取性能数据并决定需要扩展自适应高速缓存来改进系统的性能后，必须购买读高速缓存设备（RCD）。扩展自适应高速缓存是通过 RCD 自动启用的。


要开始使用扩展自适应高速缓存，必须具有：

- 支持扩展自适应高速缓存的一个或多个存储 I/O 适配器（对于运行 V4R4 或更高版本的系统为 CCIN 2748，或对于运行 V5R1 或更高版本的系统为 CCIN 2778，或对于运行最新发行版 V5R2 的系统为 CCIN 2757。
- 要激活扩展自适应高速缓存的每个存储 I/O 适配器的读高速缓存设备（对于运行 V4R4 或更高版本的系统为 CCIN 6731）。

因为扩展自适应高速缓存是通过 RCD 自动启用的，所以没有受控的打开或关闭开关。可以通过并行维护添加 RCD 而不中断系统。RCD 驻留在内部磁盘插槽中并与所有其它磁盘类型和容量一起工作。应了解扩展自适应高速缓存中的所有数据也保证位于磁盘中。在不太可能的 RCD 故障情况下，将没有数据丢失。

读高速缓存设备可以在销售 iSeries 硬件的地方购买，或与本地 IBM 代表联系。

## Workload Estimator for iSeries

Workload Estimator  帮助您基于特定工作量类型的估计工作量来确定系统需要。PM/400 是集成的 OS/400 功能，具有处理器保修或 IBM 维护协议的用户可以激活该功能，而不必另外付费。作为回报，您会接收到容量和性能分析图形，这些图形对于计划和管理系统增长和性能很有用。

已增强工作量估计器和 PM/400，以便它们可以互相配合使用。通过基于 Web 的应用程序，可以将升级缩放成需要的 iSeries 系统，该系统适应您现有的系统实用程序、性能和发展（由 PM/400 报告的）。作为附加选项，缩放还可以包括添加特定应用程序（例如，Domino、Java 和 WebSphere）的能力，或在一个系统上将多个 AS/400 或 iSeries 传统 OS/400 工作量合并的能力。此能力允许您基于来自自己系统的现有利用率数据来计划将来的系统需求。

## iSeries 无线导航器

iSeries 无线导航器允许您使用因特网就绪电话、带有无线调制解调器的个人数字助手（PDA）或传统的 Web 浏览器以远程方式监控系统性能和状态。借助无线设备，可以：

- 在多个系统之间运行命令
- 启动和查看系统、作业和消息监控器
- 使用来自监控器的作业和消息（挂起、释放、结束、应答和获取详细信息）
- 管理 Integrated xSeries Server

有关 iSeries 无线导航器可以如何帮助您入门远程监控的概述，参见主题 iSeries 无线导航器。

有关远程监控的完整和最新信息，参见 iSeries Navigator for Wireless  主页。

## PATROL for iSeries ( AS/400 ) — 预测

PATROL for iSeries ( AS/400 ) — 预测产品通过使性能管理的很多管理任务自动执行可以帮助您管理 iSeries 性能。此产品提供了 iSeries 服务器组的当前和历史系统数据的深入视图，允许您深入了解特定的详细信息，如 CPU 和文件利用率和状态。另外，Patrol 还为自动操作提供了几个选项，这些选项将允许在 iSeries 服务器上发生性能和可用性之前提前解决这些问题。

此产品通过提供详细的容量计划信息可以帮助您计划将来升级和管理 iSeries 环境的生长。

有关更多信息，参考 BMC 产品 Web 站点。 

---

### 方案：性能

了解性能管理的最好的方法之一是查看示例，它们说明了在采样商业环境中可以使用多少应用程序和功能。使用下列方案和配置示例来更多地了解如何管理性能。

#### 第 24 页的『方案：在升级或迁移后改进系统性能』

在此方案中，刚刚升级或迁移了系统，且它的运行速度似乎比以前慢。此方案帮助您标识并改正性能问题。

#### 第 63 页的『方案：系统监控器』

参见系统监控器示例，该监控器在 CPU 利用率太高时会警告您，并暂时挂起任何较低优先级的作业，直到更多资源变为可用为止。

#### 第 65 页的『方案：消息监控器』

参见消息监控器示例，该监控器显示消息队列中在任何 iSeries 服务器上出现的任何查询消息。该监控器一检测到消息就打开并显示该消息。

#### 第 64 页的『方案：作业监控器』

参见作业监控器示例，该监控器跟踪指定作业的 CPU 利用率，并在 CPU 利用率太高时警告作业的所有者。

---

### 相关信息

以下列示了与“性能”主题相关的 PDF 格式的 iSeries 手册（有时称为“白皮书”）、和 IBM Redbooks<sup>(TM)</sup>。还可以查看或打印下列任何 PDF:

- 手册

#### Performance Tools for iSeries

本书为程序员提供收集关于系统、作业或程序性能的数据所需要的信息。它也包括关于打印和分析性能数据以标识和更正可能存在的低效率的技巧以及关于管理器和代理程序功能部件的信息。

- Web 站点


#### iSeries Performance Capabilities Reference

此参考提供有关对性能基准、容量计划和服务器性能计划有用的服务器性能的高技术含量的信息。

- 红皮书:

#### – IBM eserver iSeries Universal Connection for Electronic Support and Services

此文档介绍“全球连接”。它还说明如何使用各种支持工具，这些工具可以将机器上软件和硬件的库存报告给 IBM，以便您能基于系统数据获取个性化的电子支持。

- **Lotus Domino for AS/400: Performance, Tuning, and Capacity Planning**   
此文档描述性能管理的方法学。它包括设置性能目标、收集和复查性能数据、调整资源和进行容量计划。还提供了一些性能准则和应用程序设计技巧。
- **AS/400 Performance Management**   
此文档描述性能管理的方法学。它包括设置性能目标、收集和复查性能数据、调整资源和进行容量计划。还提供了一些性能准则和应用程序设计技巧。
- **AS/400 HTTP Server Performance and Capacity Planning**   
因特网和基于 Web 浏览器的应用程序对于组织如何分发信息、执行商务处理、为客户服务和扩展新市场有深远的影响。此书供 iSeries 程序员、网络和系统管理专业人员和负责设计、开发和部署基于 Web 的应用程序和信息系统其它信息技术人员使用。
- **Java and WebSphere Performance on IBM eserver iSeries Servers**   
此文档提供处理与 Java 和 WebSphere Application Server 的性能有关的问题的技巧、技术和方法学，着重介绍了 iSeries 服务器。
- **Management Central: A Smart Way to Manage AS/400 Systems**   
发现“中央管理”的益处。此书讨论收集服务，它替换性能监控器。它也解释“中央管理”如何允许操作员和管理员通过提供实时性能监控性能（如事件的通告和对事件的自动响应）监控网络中的服务。
- **Managing AS/400 V4R4 with Operations Navigator**   
此书让您了解 iSeries 功能的广阔范围，其功能通过操作导航器（现已知为 iSeries 导航器）可用。除了这些功能的全部讨论，此书特殊讨论了用“中央管理”和“收集服务”监控系统性能。
- **AS/400 Performance Explorer Tips and Techniques**   
此文档对可用于 V3R6 的性能资源管理器能力提供描述和详细的示例。提供特殊应用程序示例和报告。

有关 iSeries 性能的完整信息，确保参见第 1 页的『性能』主题。







中国印刷