

IBM

@server

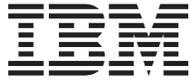
iSeries

パフォーマンス

バージョン 5 リリース 3







@server

iSeries

パフォーマンス

バージョン 5 リリース 3

**ご注意！**

本書および本書で紹介する製品をご使用になる前に、151 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM OS/400 (プロダクト番号 5722-SS1) および IBM Performance Tools for iSeries (プロダクト番号 5722-PT1) のバージョン 5、リリース 3、モディフィケーション 0 に適用されます。また、改訂版で断りがない限り、それ以降のすべてのリリースおよびモディフィケーションに適用されます。このバージョンは、すべての RISC モデルで稼働するとは限りません。また CISC モデルでは稼働しません。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： iSeries  
Performance  
Version 5 Release 3

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2005.8

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体\*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注\* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、  
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1998, 2005. All rights reserved.

© Copyright IBM Japan 2005

# 目次

パフォーマンス . . . . .	1	収集サービス . . . . .	36
V5R3 の新機能 . . . . .	2	インテリジェント・エージェント . . . . .	70
新機能: モニター . . . . .	2	パフォーマンス・データ・ファイル . . . . .	90
新機能: 収集サービス . . . . .	3	iSeries <sup>(TM)</sup> ナビゲーターのモニター . . . . .	95
新機能: Performance Tools ライセンス・プログラ ム . . . . .	6	グラフ・ヒストリー . . . . .	103
新機能: PM iSeries . . . . .	9	IBM Performance Management for eServer iSeries Performance Tools . . . . .	105 123
新機能: Performance Explorer . . . . .	10	Performance Explorer . . . . .	134
トピックの印刷 . . . . .	11	iDoctor for iSeries . . . . .	142
パフォーマンスの計画 . . . . .	11	Performance Trace Data Visualizer (PTDV) . . . . .	143
システム・ベンチマークの設定 . . . . .	12	Performance Management API . . . . .	143
システムの拡張時および拡張方法の決定 . . . . .	13	OS/400 パフォーマンス用の処理コマンド . . . . .	143
同時マルチスレッド化使用時の決定 . . . . .	14	拡張キャッシュ . . . . .	144
パフォーマンス管理ストラテジーの選択 . . . . .	14	Workload Estimator for iSeries . . . . .	146
パフォーマンスを管理するための環境のセットアッ プ . . . . .	16	iSeries <sup>(TM)</sup> ナビゲーター (ワイヤレス対応) . . . . .	147
iSeries <sup>(TM)</sup> パフォーマンスの管理 . . . . .	17	PATROL for iSeries (AS/400) - Predict . . . . .	147
パフォーマンスの追跡 . . . . .	18	シナリオ: パフォーマンス . . . . .	147
パフォーマンスの問題の調査 . . . . .	18	関連情報 . . . . .	148
パフォーマンス・データの表示 . . . . .	25		
パフォーマンスのチューニング . . . . .	25	<b>付録. 特記事項 . . . . .</b>	<b>151</b>
e-business パフォーマンスの管理 . . . . .	28	商標 . . . . .	153
パフォーマンス管理用のアプリケーション . . . . .	33	資料に関するご使用条件 . . . . .	153
		コードに関する特記事項 . . . . .	154



---

## パフォーマンス

システムのパフォーマンスを管理するためにどれだけの投資をするでしょうか。業務変更の必要性は、時には予想よりも早い場合があります。業務変更に対して有効な対応をするためには、システムも変更しなければなりません。システムの管理は、一見するとほんのありきたりの、時間ばかりかかる仕事のようにも見えるかもしれませんが、しかし、投資は、システムがより効率的に稼働して、それが業務に反映されることにより、速やかに効果を上げることになります。それが効率的なのは、変更が計画に基づき管理されているためです。

iSeries<sup>(TM)</sup> サーバーのパフォーマンスの管理は、iSeries 実行管理機能の完全な理解を必要とする複合タスクです。システム・パフォーマンスに影響のあるすべての異なるプロセスを理解することは、未経験のユーザーにとって挑戦となります。パフォーマンス上の問題を解決するには、それぞれ独自の要件およびサポートされる機能を使用して、大規模なツールを効率よく使用する必要があります。パフォーマンス・データを収集および分析した後でさえ、その情報を使用して行う事柄を知ると、気が遠くなるかもしれません。

このトピックでは、パフォーマンス管理と関連したタスクおよびツールを通じて説明します。

### 2 ページの『V5R3 の新機能』

このトピックでは、このリリースでの新規または変更点について説明します。

### 11 ページの『トピックの印刷』

この情報の印刷版が必要な場合には、PDF を印刷するためにここに進みます。

### 11 ページの『パフォーマンスの計画』

iSeries サーバーのパフォーマンス目標を設定すると、測定可能なパフォーマンス・ベンチマークによりパフォーマンス・データを比較することができます。このトピックでは、それらのベンチマークの設定方法および設定後のベンチマークの使用方法について説明します。

### 16 ページの『パフォーマンスを管理するための環境のセットアップ』

iSeries サーバーには、システム・パフォーマンスを管理するための強力なアプリケーションが含まれています。ただし、それらは、独自の業務環境の特定のニーズを満たすために正しく構成されていなければなりません。パフォーマンス・データを定期的に収集、モニター、および分析するようにアプリケーションを構成する方法について学習します。

### 17 ページの『iSeries<sup>(TM)</sup> パフォーマンスの管理』

パフォーマンス管理は、現在の機能を測定し、傾向を認識し、適当な調整を行うことによってご使用のコンピューター・システムの使用率を最適化し、エンド・ユーザーを満足させ、応答時間またはジョブ・スループットなどの管理要件を満たすために必要です。パフォーマンス管理は、業務の効率を保ち、通常の業務活動の延期を避けるために必要です。したがって、パフォーマンスの管理は、日常の操作の一部です。

### 33 ページの『パフォーマンス管理用のアプリケーション』

iSeries システムにおけるパフォーマンスの管理では、さまざまな特殊アプリケーションを使用する必要があります。これらのアプリケーションのそれぞれは、システム・パフォーマンスに特定の洞察を提供します。このトピックでは、いくつかのアプリケーションおよびそれぞれのアプリケーションの使用方法について説明します。

### 147 ページの『シナリオ: パフォーマンス』

パフォーマンス管理について学習する最善の方法の 1 つは、ご使用の業務環境でこれらのアプリケーションまたはツールを使用できる方法を示す例をご覧ください。これらの例について調べてください。

### 148 ページの『関連情報』

IBM<sup>(R)</sup> の関連情報には、技術、ノウハウ、および「方法」に関する情報が含まれています。

注: このトピックには、コード例が含まれています。法律上の重要な情報に関しては、『154 ページの『コードに関する特記事項』』をお読みください。

---

## V5R3 の新機能

ここでは、このリリースで新しくなった部分を紹介します。

- 『**新機能: モニター**』  
サポートされる新規モニターおよび既存のモニターへの変更について調べてください。
- **3 ページの『新機能: 収集サービス』**  
パフォーマンス・データベース・ファイルへの変更および新しい収集サービス機能についてお読みください。
- **6 ページの『新機能: Performance Tools ライセンス・プログラム』**  
報告書への変更についてお読みください。
- **9 ページの『新機能: PM iSeries』**  
PM eServer<sup>(TM)</sup> iSeries がデータを自動収集する方法についてお読みください。
- **10 ページの『新機能: Performance Explorer』**  
Performance Explorer データベース・ファイルへの変更および新機能についてお読みください。
- **Capacity on Demand**  
Capacity on Demand トピックは、「システム管理」→「Capacity on Demand」に移動しました。
- **70 ページの『インテリジェント・エージェント』**  
iSeries ナビゲーターは、システム管理者に、単一のシステム上で、またはさまざまなシステムすべてで実行する 1 つ以上の ABLE (Agent Building and Learning Environment) エージェントを管理する簡単な方法を提供します。

## 新機能: モニター

ここでは、このリリースで新しくなった部分を紹介します。

新規の時間帯システム値 (QTIMZON) を使用して、特定の時間帯にシステム時刻を設定できます。マネージメント・セントラル・モニターは、(エンドポイント・システムの) ローカル・システム時刻が正しく構成されているということを前提とします。これが構成されていないと、いくつかのメトリックは起動しません。マネージメント・セントラル・モニターが確実に正しいローカル時刻を処理するために、新規の時間帯システム値 (QTIMZON) を使用して、ご自分のエンドポイント・システムがローカル時間帯に合っていることを確認してください。

これで、ファイル・モニターおよびメッセージ・モニター用のしきい値が、起動コマンドの実行時に自動的にリセットされるように選択できます。しきい値を定義し、しきい値の起動時にコマンドが実行するように指定する場合は、「**起動コマンド実行後に自動的にリセット (Automatically reset after trigger command has run)**」を選択するだけで済みます。

CPU 使用率 (平均) メトリックおよび CPU 使用率 (対話式フィーチャー) メトリックの計算メソッドが、CPU カウントに基づく現行の計算を使用する代わりに、CPU 使用率値を使用するように変更されました。さらに、CPU 使用率 (平均) メトリックは、上限なしのプロセッサをサポートする区画の場合、100% を超えることが可能になったので、グラフの関連範囲にも 100% より上のパーセントを入れることができます。

システム・モニター・メトリックに新規フィールド、ディスク・アーム使用率 (平均)、ディスク・アーム使用率 (最大)、ディスク記憶域 (平均)、およびディスク記憶域 (最大) 追加されました。これらの新規フィールドは、マルチパス・ディスク装置、つまりシステムからディスク装置までに複数の冗長パスがある装置についての情報を提供します。各パスに固有の資源名があります。

- **マルチパス装置**は、資源がマルチパス・ディスク装置を表すことを示します。
- **マルチパス装置の初期パス**は、資源がマルチパス・ディスク装置の初期パスを表すことを示します。
- **リモートにミラー保護された独立 ASP の実動コピー**は、ディスク装置がリモートにミラー保護された独立 ASP の実動コピーにあることを示します。
- **リモートにミラー保護された独立 ASP のミラー・コピー**は、ディスク装置がリモートにミラー保護された独立 ASP のミラー・コピーにあることを示します。

### ジョブ通知

Advanced Job Scheduler の通知機能を使用して、CPU 使用率が指定したしきい値に達した場合に、自分、または指定した他の人に通知することができます。この機能の作動方法について詳しくは、101 ページの『シナリオ: Advanced Job Scheduler 通知を使ったジョブ・モニター』を参照してください。

## 新機能: 収集サービス

ここでは、このリリースで新しくなった部分を紹介합니다。

### CL コマンド

CL コマンドを使って、パフォーマンス収集を開始したり終了したりする機能が加えられました。また、CFGPFRCOL コマンドで収集サービスのプロパティを変更することもできます。

CHKPFRCOL コマンドを使用すると、収集サービス・サーバー・ジョブの現行状況を判別できます。

- パフォーマンス収集開始 (STRPFRCOL)
- パフォーマンス収集終了 (ENDPFRCOL)
- パフォーマンス収集の構成 (CFGPFRCOL)
- パフォーマンス収集の検査 (CHKPFRCOL)

### CPU 使用率計算のための新データおよび方法

部分的なプロセッサ区画および動的構成変更と関連した、127 ページの『CPU 使用率の報告』の新しい方法についてお読みください。このトピックでは次のことを説明しています。

- 今後、HVLPTASK は、QAPMJOBx ファイルを含め、どのシステム・インターフェースによっても報告されません。
- 収集サービスは、区画の構成が変化しても循環しなくなりました。
- QAPMSYSCPU データは拡大縮小されません。
- データを V5R3 \*MGTCOL 管理収集オブジェクトから作成した場合に、以前のリリースで表示される内容。

### 構成済みキャパシティー

論理区画の新規サポートにより、他の区画がその構成済みキャパシティーをすべて使用しているわけではない状況では、129 ページの『構成済みキャパシティーのレポート』を超えることができます。

## データ・ポート・サービス

収集サービスは、データ・ポート・サービスから獲得するパフォーマンス・データを報告します。このデータは、リモート独立 ASP ミラーリングなどの、データ・ポート・サービスのクライアントのパフォーマンスを理解するのに役立ちます。独立 ASP がリモートにミラー保護される場合、データはローカル独立 ASP に書き込まれます。データは、データ・ポート・サービスを使用することによって、同時にリモート・システム上の対応する独立 ASP に送信されます。ローカル・システム上のパフォーマンスは、リモート独立 ASP へのパスが遅い場合、悪影響を受けることがあります。データ・ポート・サービスのパフォーマンス・データは、このパスのパフォーマンスに関する情報を提供します。データは QAPMDPS ファイルに記録されます。

iSeries<sup>(TM)</sup> ナビゲーターから、データ・ポート・サービス・データの収集を指定できます。新規カテゴリ DPS - データ・ポート・サービスが Available categories/Categories に追加され、収集サービスのダイアログでフィールドを収集できるようになりました。カテゴリは、\*STANDARDP および \*ENHCPCPLN プロファイルに含まれます。カテゴリをカスタム・プロファイルに含めることもできます。

## 区画間パフォーマンス・データ

V5R3 では、収集サービスは、オペレーティング・システムに関係なく 64 ページの『区画を介したパフォーマンス・データの収集』フレームワークを提供します。たとえば、区画は OS/400<sup>(R)</sup>、AIX<sup>(R)</sup>、または Linux<sup>(TM)</sup> で実行できます。IBM<sup>(R)</sup> Director Multiplatform の IBM Director Server コンポーネントがインストールされていて、収集サービスの実行されている区画で実行されている必要があります。IBM<sup>(R)</sup> Director エージェントを、データを収集したい区画にインストールすることも必要です。収集サービスは各区画からデータを収集し、PM iSeries がデータを要約します。区画間パフォーマンス・データに使用できる Linux オペレーティング・システムについては、情報 APAR: II13986 を参照してください。

論理区画のデータは、iSeries ナビゲーターから収集できます。新規カテゴリの論理区画が”使用可能なカテゴリ”、”収集するカテゴリ”に追加され、収集サービスのダイアログでフィールドを収集できるようになりました。カテゴリは、\*STANDARD、\*STANDARDP、および \*ENHCPCPLN プロファイルに含まれます。カテゴリをカスタム・プロファイルに含めることもできます。

## グラフ・ヒストリー

V5R3 より前のバージョンでは、グラフ・データ獲得には 3 つの方法がありました。1 つは、特定の時刻範囲で使用可能な場合、そのグラフ・データが戻されました。2 つ目の方法は、使用可能なグラフ・データはないものの、ロー・データが使用可能で、ロー・データからグラフ・データを獲得するのに十分な保存期間がある場合、ロー・データが動的にグラフ・データに変換され、戻されました。3 つ目は、それでも使用可能なグラフ・データがない場合、要約されたデータが戻されました。V5R3 からは、ロー・データを動的にグラフ・データに変換する方法がなくなりました。「**グラフ・データを今作成する (Create graph data now)**」 オプションを使用してグラフ・データを作成することができます。

## パフォーマンス・データベース・ファイル

以下の表では、新規および変更データベース・ファイルを示しています。

データベース・ファイル	説明
QAPMCONF	<ul style="list-style-type: none"> <li>• 追加の区画構成情報を提供する新しいレコード・キー (SP)。</li> <li>• オンデマンド・プロセッサ、使用可能なオンデマンド・プロセッサ、オンデマンド・メモリー、および使用可能なオンデマンド・メモリーの数の物理システム・ビューを提供する、新しいレコードが追加されました。</li> <li>• 対話式キャパシティーの細分性が .1% から .01% に上がりました。この変更は、GKEY IT および IL のレコードに効果があります。</li> <li>• GKEY PC のレコードが、255 の区画をサポートするようになりました。以前のサポートでは 99 区画でした。</li> </ul>
QAPMDISK	<p>収集サービスが報告できるようになった新規ディスク・パフォーマンス・データは次のとおりです。ディスク装置がパリティ・セットにあるかどうかを示す (DSPS)、ディスク装置が高可用性パリティ・セットにあるかどうかを示す (DSHAPS)、資源がマルチパス・ディスク装置を表すことを示す (DSMU)、資源がマルチパス・ディスク装置の初期パスを表すことを示す (DSIP)、ディスク装置がリモートにミラー保護された独立 ASP の実動コピーにあることを示す (DSPC)、およびディスク装置がリモートにミラー保護された独立 ASP のミラー・コピーにあることを示す (DSMC)。これらの場合、マルチパス・ディスク装置は、システムからディスク装置までに複数の冗長パスがある装置です。各パスに固有の資源名があります。</p>
QAPMDPS	<p>データ・ポート・サービスのパフォーマンス・データを報告する新規ファイル。</p>
QAPMIOPD	<ul style="list-style-type: none"> <li>• ネットワーク・サーバー (*IPCS カテゴリー)、および入出力アダプター (*IOPBASE カテゴリー) に報告されるデータを記述するファイル記述への変更。</li> <li>• 新規のデータ・タイプが追加されました。</li> <li>• 新規フィールドが追加されました (XINWSD)。</li> <li>• INTSEC フィールドの記述に変更があります。</li> <li>• 仮想入出力データがファイルに追加されました。ホスティング区画 (物理資源を提供する区画) 上の仮想入出力アダプターの場合、ゲスト区画の代わりに提供される仮想装置サポートのためにこの区画内で発生する入出力活動についてのデータが提供されます。</li> </ul>
QAPMJOBMI	<p>割り振られたページ (JBPGA)、割り振り解除されたページ (JBPGD)、および現行ユーザー (JBCUSR) をサポートする新規フィールド。</p>

データベース・ファイル	説明
QAPMJOBS	割り振られたページ (JBPGA)、割り振り解除されたページ (JBPGD)、および現行ユーザー (JBCUSR) をサポートする新規フィールド。ファイル・システム・カウンターおよびジャーナル・カウンターをサポートする新規フィールド。(これらのフィールドは、前のリリースで QAPMJOBMI に追加されました。)
QAPMJSUM	新規ジョブ・タイプ、INF (対話式フィーチャー) が JSCBKT フィールドに追加されました。
QAPMLPAR	IBM Director Server ライセンス・プログラムがインストールされ、収集サービスを実行している区画で実行している場合に論理区画データを報告する新規ファイル。
QAPMSYS	区画の可用性および使用メトリックについてのデータを報告する、追加された新規フィールド。
QAPMSYSCPU	共用プロセッサ区画用に変更されたこのファイル (SCPU01...32) で報告される、個々の CPU データ。データは、上限ありまたは上限なしの区画両方に対して拡大縮小されなくなります。以前は、このデータは、フィールド SCTNUM で報告された構成済みの仮想プロセッサ全体に一致するように拡大縮小されていました。新規フィールドは、活動中のプロセッサの現行数 (SCTACT) を報告します。
QAPMSYSTEM	<ul style="list-style-type: none"> <li>SMXDU の説明は、最大ディスク使用率から、すべての単一パス・ディスク装置およびすべてのマルチパス・ディスク装置の最大使用率に変更されました。</li> <li>区画の可用性および使用メトリックについてのデータを報告する、追加された新規フィールド。</li> <li>キャッシュなしユーザー権限探索が実行された回数を報告する、追加された新規フィールド SYNUAL。</li> <li>SYAUTH の説明が、権限探索カウントから、オブジェクト権限検査に変更されました。</li> </ul>

## 新機能: Performance Tools ライセンス・プログラム

このトピックでは、Performance Tools ライセンス・プログラム V5R3 に加えられた変更点を示します。

### バッチおよび対話式 CPU 使用率

130 ページの『5250 オンライン・トランザクション処理 (OLTP)』のプロセスは、収集サービスが提供する QAPMJSUM の新規バケットに基づきます。iSeries<sup>TM</sup> アクセス・ジョブは、バッチとしても対話式としても実行できますが、前のリリースでは、すべてのジョブが CA4 バケットに組み込まれており、これは対話式 CPU 使用率の欄に示されていました。今回、システム報告書の非対話式作業負荷サブセクションに新規カテゴリー、**iSeries アクセス - バッチ**が追加されました。さらに、DDM サーバー・ジョブは対話式作業負荷を表していないので、対話式作業負荷サブセクションから非対話式作業負荷サブセクションに移動しました。

### CPU 使用率計算のための新しい方法

収集サービスは、共用プロセッサ・プールを使用することによって、127 ページの『CPU 使用率の報告』をよりよく実行するための新規メトリックを使用します。新規メトリックは、構成変更が区画に適用さ

れたときに調整されます。新しい方法では、使用可能な CPU 時間の計算タスクをなくします。共用プロセッサ環境での仮想プロセッサ全体に対する HVLPTASK および CPU の拡大縮小の概念はなくなりしました。Performance Tools で示される CPU 値は、処理装置の観点からは実際の使用率を表します。前のリリースで収集されたデータは、パフォーマンス・データの変換 (CVTPFRDTA) コマンドによって新しい方法に適應するように変換されます。

### 構成要素報告書

- QAPMCONF ファイルの仮想プロセッサ (GKEY 13)、処理装置 (GKEY PU)、および対話式しきい値 (GKEY IT) メトリックがヘッダー・セクションに追加されました。
- ディスク使用率セクションの Unit 欄の隣に正符号 (+) が追加され、マルチパス・ディスク装置を識別します。装置番号の後の文字 H は、そのディスク装置が高可用性パリティ・セットにあることを示します。文字が何も表示されていない場合は、ディスク装置は通常のパリティ・セットです。
- 対話式しきい値を超える時間の長さ を示す欄が構成要素間隔活動セクションに追加されました。これは、対話式使用量が構成済みの対話式しきい値を超えた時間を秒単位で示します。対話式しきい値を超える時間の長さは、QAPMSYSTEM ファイルにある新規フィールド **SYIFTE** です。対話式しきい値は、対話式作業の合計システム CPU のパーセントです。このリリースで新しくなった点として、対話式しきい値は、区画の再構成によってデータ収集期間中に変更することができ、QAPMSYSTEM ファイルの **SYIFTA** にあります。
- QAPMDISK ファイルで要求された間隔パフォーマンス・データが使用可能でない場合は、High Disk および Unit 欄にダッシュ記号 (-) を表示します。
- 構成要素間隔活動セクションに、システムに使用可能な上限なしの CPU 時間を示す新しい欄が追加されました。このデータは、上限なしの区画で収集されるパフォーマンス・データにのみ報告されます。

### システム報告書

- DDM サーバー・ジョブは、対話式作業負荷サブセクションから非対話式作業負荷サブセクションに移動しました。
- 非対話式作業負荷セクションに、新規カテゴリー iSeries アクセス - バッチ・ジョブが追加されました。
- QAPMCONF ファイルの仮想プロセッサ (GKEY 13)、処理装置 (GKEY PU)、および対話式しきい値 (GKEY IT) メトリックがヘッダー・セクションに追加されました。
- DDM サーバー・ジョブ統計が、資源使用率セクションの最初の部分に表示されなくなりました。資源使用率セクションの最初の部分には、対話式、システム/36 (System/36<sup>TM</sup>)、MRT、iSeries アクセス、およびパススルーなどの対話式ジョブの統計は今までどおり表示されます。
- DDM サーバー・ジョブ統計が、資源使用率拡張セクションの対話式資源使用率拡張サブセクションに表示されなくなりました。DDM サーバー・ジョブ統計は、今後非対話式資源使用率拡張サブセクションに表示されます。
- ディスク使用率セクションの Unit 欄の隣に正符号 (+) が追加され、マルチパス・ディスク装置を識別します。装置番号の後の文字 H は、そのディスク装置が高可用性パリティ・セットにあることを示します。文字が何も表示されていない場合は、ディスク装置は通常のパリティ・セットです。
- 作業負荷セクションに 2 つの行が追加されました。1 つは対話式しきい値を超える時間の長さについての情報、もう 1 つはパーセントの観点から見た共用プロセッサ・プール使用率を表示します。共用プロセッサ・プール使用率の行は、共用プロセッサ・プールを使用して区画で収集される、パフォーマンス・データについてのみ表示します。

### 各種報告書

- ディスク使用率セクションの Unit 欄の隣に正符号 (+) が追加され、マルチパス・ディスク装置を識別します。装置番号の後の文字 H は、そのディスク装置が高可用性パリティ・セットにあることを示します。文字が何も表示されていない場合は、ディスク装置は通常のパリティ・セットです。
- ジョブ間隔報告書の対話式ジョブ明細および非対話式ジョブ明細セクションに、QAPMJOBMI ファイルの収集サービスが収集するジョブの現行ユーザーを示す欄が追加されました。

### Performance Advisor

拡張には、より正確な推奨事項を提供するための既存のメトリックへの変更、および新機能に関連する推奨事項を分析し、提供するための新規メトリックを追加する変更が含まれます。

- 対話式フィーチャー使用率、Point-to-Point プロトコル (PPP) 活動、および TCP/IP 活動についての推奨事項を提供します。
- CPU 使用率、システム・メトリック、およびディスク・サービス時間についてのガイドラインを更新します。

#### 「パフォーマンス・データの表示」画面

- 「表示する時間間隔の選択 (Select Time Intervals to Display)」画面に、対話式フィーチャー使用率 (Int Feat Util) を表示する新規欄が追加されました。QAPMDISK ファイルで要求された間隔パフォーマンス・データが使用可能でない場合は、High Dsk および Unit 欄にダッシュ記号 (-) を表示します。
- 「パフォーマンス・データの表示 (Display Performance Data)」画面に、対話式作業に割り当てられる処理キャパシティーのパーセント (Int Threshold) を示すフィールドが追加されました。この情報は、QAPMCONF ファイルの IT フィールドから取られています。
- 「パフォーマンス・データの表示 (Display Performance Data)」画面に、QAPMCONF ファイルから仮想プロセッサ (GKEY 13) および処理装置 (GKEY PU) メトリックがヘッダーに追加されました。
- 「パフォーマンス・データの表示 (Display Performance Data)」画面に、システムが使用する対話式キャパシティーのパーセント (対話式フィーチャー使用率) を示す行が追加されました。
- 「パフォーマンス・データの表示 (Display Performance Data)」画面に、秒単位でしきい値を超える対話式 CPU 時間 (対話式 CPU しきい値を超える時間 (秒単位)) を示す行が追加されました。
- 「ジョブの表示 (Display Jobs)」画面に、要求されたジョブまたはタスクの待機時間統計を秒単位で示す、オプション 6 (待機詳細) が追加されました。

#### 「パフォーマンス・データの分析 (Analyze Performance Data)」画面

- 「分析する時間間隔の選択 (Select Time Intervals to Analyze)」画面に、対話式フィーチャー使用率 (Int Feat Util) を表示する新規欄が追加されました。QAPMDISK ファイルで要求された間隔パフォーマンス・データが使用可能でない場合は、High Dsk および Unit 欄にダッシュ記号を表示します。
- 「推奨事項の表示 (Display Recommendations)」画面に、対話式作業に割り当てられる処理キャパシティーのパーセント (Int Threshold) を示すフィールドが追加されました。この情報は、QAPMCONF ファイルの IT フィールドから取られています。
- 「推奨事項の表示 (Display Recommendations)」画面に、QAPMCONF ファイルから仮想プロセッサ (GKEY 13) および処理装置 (GKEY PU) メトリックがヘッダーに追加されました。

#### 「実行 (Perform)」メニュー

- 「実行 (Perform)」メニューの収集サービス・オプション、オプション 2 (パフォーマンス・データの収集) で、新しいパフォーマンス収集 CL コマンドを利用できるようになりました。以前は、これらのオプションはコレクター API を使用していました。今後、これらのオプションは次の CL コマンドを使用します。
  - オプション 1 (パフォーマンス収集の開始): パフォーマンス収集の開始 (STRPFCOL)

- オプション 2 (パフォーマンス収集の構成): パフォーマンス収集の構成 (CFGPFRCOL)
- オプション 3 (パフォーマンス収集の終了): パフォーマンス収集の終了 (ENDPFRCOL)
- 「時間間隔の選択 (Select Time Intervals)」画面に、対話式フィーチャー使用率 (Int Feat Util) を表示する新規欄が追加されました。QAPMDISK ファイルで要求された間隔パフォーマンス・データが使用可能でない場合は、High Disk および Unit 欄にダッシュ記号 (-) が表示されます。

### システム活動の処理 (WRKSYSACT) コマンド

このコマンドは、他の Performance Tools と一貫した値を計算し、表示するように変更されました。

WRKSYSACT の CPU 使用率は仮想プロセッサ全体に対して拡大縮小をしなくなりました。

HVLPTASK ジョブは CPU を消費するタスクのリストから除外されます。さらに、WRKSYSACT は、CPU 使用率を 99.9% で上限を設けずに、上限なしのプロセッサについては 100% を超える CPU 値を表示するように拡張されました。

### iSeries ナビゲーターでの Performance Tools のプラグイン

- 「高位ディスク使用率」グラフは、今後、ディスク情報のない間隔を除外します。
- 「ユーザー・プール不在/秒」グラフは、プールごとに 1 行を使用して、複数のプールをグラフ化するように拡張されました。「ユーザー・プール不在/秒」グラフには 2 つのオプションがあり、「グラフ (Graphs)」メニューから「設定 (Preferences)」を選択することによってアクセスできます。オプションは次のとおりです。
  - 選択する時間枠によって、最も上位の不在率を持つ上位 10 のプールをグラフ化する。
  - 選択する特定のプールをグラフ化する。

この拡張は、パフォーマンス分析者が、プールのサイズに加えられる変更が不在率に与える影響を表示する場合、さらに一般的なプール活動をよりよく理解するために役立ちます。

- 収集サービスが対話式フィーチャー使用率メトリックを提供する際、それを示す新しいグラフが追加されました。対話式フィーチャー使用率は、対話式 CPU しきい値メトリックを超える時間と共に、「要約 (Summary)」ウィンドウ、および収集で情報が使用可能な場合「データ・プロパティーに対する対話式しきい値 (Interactive Threshold to Data Properties)」ページに表示されます。
- 複数のウィンドウがオープンされる状況を受け入れるように、「パフォーマンス・データの表示」ウィンドウ・タイトルにライブラリーおよびメンバー名が追加されました。
- 「合計 CPU 使用率」グラフに、新規メトリック、使用可能な上限なし CPU% が追加されました。

## 新機能: PM iSeries

ここでは、このリリースで新しくなった部分を紹介します。

Performance Management/400 という名前が、IBM<sup>(R)</sup> Performance Management for **@server** iSeries<sup>(TM)</sup> (PM **@server** iSeries または PM iSeries) に変更されました。PM iSeries は、iSeries の新機能をサポートするために近年変更されていた、報告書およびツールの機能をよりよく反映するように選択されました。

ユニバーサル・コネクションが、データの伝送のため IBM Global Network<sup>(R)</sup> (IGN) サポートに置き換わりました。

V5R3 の前のバージョンでは、「上位 10 (Top Ten)」画面の「ジョブの省略 (Omit Jobs)」から、HvLp\* ジョブの省略を選択することができました。V5R3 では、アップグレード時に HvLp\* ジョブは組み込まれません。この変更が、マイグレーションを実行する場合にデータに影響することはありません。

### 区画間パフォーマンス・データ

V5R3 では、収集サービスは、オペレーティング・システムに関係なく区画からデータを収集するフレームワークを提供します。たとえば、区画は AIX<sup>(R)</sup> または Linux<sup>(TM)</sup> で実行できます。IBM Director Server ライセンス・プログラムをインストールし、収集サービスを実行している区画で実行する必要があります。区画に指示エージェントをインストールしておくことが必要です。収集サービスは各区画からデータを収集し、PM iSeries がデータを要約します。

iSeries ナビゲーターから論理区画データを収集するように指定できます。新規カテゴリー、論理区画が Available categories/Categories に追加され、収集サービスのダイアログでフィールドを収集できるようになりました。カテゴリーは、\*STANDARD、\*STANDARDP、および \*ENHCPCPLN プロファイルに含まれません。カテゴリーをカスタム・プロファイルに含めることもできます。

PM iSeries 報告書の最新情報については、PM iSeries Web サイト  に進んでください。

## 新機能: Performance Explorer

ここでは、このリリースで新しくなった部分を紹介します。

### CL コマンド

#### PEX 定義の追加 (ADDPEXDFN)

- スレッド/タスク・オプションの追加 (ADDTHDOPT)  
Performance Explorer セッションの開始時間に関連するスレッドおよびタスクの作成時に基づき、Performance Explorer セッションに含めるべきスレッドおよびタスクのタイプを指定します。
- Randomize エlementが INTERVAL パラメーターに追加されました。\*FIXED または \*VARY 値を指定できます。
- 基本イベント (BASEVT) パラメーターおよび通信イベント (CMNEVT) パラメーターに、イベント・フォーマット・Elementが追加されました。  
イベント・フォーマットは、このイベントに収集されるデータを記述します。\*FORMAT1 は、ほとんどのデータ分析に使用されるデータを提供します。他のフォーマットは、これらのイベントに関連した他のデータの収集を許可します。\*FORMAT1 以外の値は、\*PMCO および \*SWOQ イベントのみに有効です。その他のすべてのイベントについては、指定されるフォーマットに関係なく \*FORMAT1 が使用されます。
- 保管/復元イベント (SAVRSTEVT) パラメーターが追加されました。  
定義に含められる保管/復元イベントを指定します。

#### PEX フィルターの追加 (ADDPEXFTR)

- Java<sup>(TM)</sup> 起動パラメーター (JVATRQ) が追加されました。  
この起動仕様に一致する Java メソッド・エントリ・イベント (\*JVAENTRY) が発生する場合、Performance Explorer は、活動中の Performance Explorer セッションで使用される Performance Explorer 定義で指定されるすべてのイベントを収集します。イベントは、起動が発生するスレッドにのみ収集されます。
- Java クラス・フィルター・パラメーター (JVACLSFTR) が追加されました。  
Java クラス・フィルターで比較値として使用される Java パッケージおよびクラスを指定します。

---

## トピックの印刷

パフォーマンス・トピックの PDF 版を表示またはダウンロードするには、パフォーマンス (約 1973 KB) を選択します。この PDF には、パフォーマンス・データベース・テーブル情報、または Performance Tools 報告書のサンプルはありません。

パフォーマンス・データベース・テーブル情報の PDF 版を表示またはダウンロードするには、パフォーマンス・データ・ファイル (約 841 KB) を選択します。

Performance Tools 報告書情報の PDF 版を表示またはダウンロードするには、Performance Tools report (約 778 KB) を選択します。

関連するトピックを表示またはダウンロードすることもできます。

- また、マネージメント・セントラル (約 324 KB) には、エンドポイント・システムとシステム・グループをセットアップする方法と、マネージメント・セントラル機能が、以下のようなサーバー管理タスクを合理化するのに役立つさまざまな情報も含まれています。
  - ユーザーおよびグループの管理
  - データのパッケージおよび送信
  - コマンドの実行
- 実行管理 (870 KB) では、以下の作業管理の概念を説明しています。
  - 日次作業管理
  - システムの構造
  - 作業の実行方法
  - Advanced Job Scheduler を使ったタスクまたはジョブのスケジュール

148 ページの『関連情報』トピックから、PDF を表示または印刷することもできます。

### PDF ファイルの保存

表示または印刷のために PDF をワークステーションに保存するには、以下のようになります。

1. ブラウザーで PDF を右マウス・ボタン・クリックする (上部のリンクを右マウス・ボタン・クリック)。
2.  Internet Explorer を使用している場合は、「対象をファイルに保存」をクリックする。Netscape Communicator を使用している場合は、「リンクを名前を付けて保存」をクリックする。 
3. PDF を保存したいディレクトリーに進む。
4. 「保存」をクリックする。

### Adobe Acrobat Reader のダウンロード

 これらの PDF を表示または印刷するには、Adobe Acrobat Reader が必要です。このアプリケーションは Adobe Web サイト ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html))  からダウンロードできます。 

---

## パフォーマンスの計画

システムのパフォーマンスの計画には、パフォーマンス目標の設定、それらの目標に基づくベンチマークの作成、およびシステムの展開の計画が必要です。このセクションでは、ご使用のシステムのパフォーマンスの計画に必要なステップについて解説します。

ご使用のシステムのパフォーマンスを計画するとき、そのシステムが取り扱う業務要件を完全に理解し、これらの業務ニーズをパフォーマンス目標に変換できる必要があります。業務ニーズが変化するにつれて、パフォーマンス要求も変化しなければならないことに注意してください。

開始する最善の方法は、業務のピーク時に、ご使用のコンピューター・システムに必要な 1 時間ごとおよび 1 日ごとの最大対話式トランザクション・スループットを見積もることです。その後、ローカルおよびリモート・ワークステーションにアクセス可能な平均応答時間を決定することができます。業務要件を満たすために時間内に完了するようにするため、通常のバッチ処理にかかる時間、およびそのスケジュール方法について考慮する必要があります。

次に、統計の基本セットを確立することができ、その後、以下のものを含むパフォーマンス目標計画を文書化する必要があります。

- 1 時間ごとのピーク・トランザクション
- 1 日ごとのピーク・トランザクション
- ローカル・ワークステーションの許容平均応答時間
- ピーク対話式トランザクション
- 実行時および予期される所要時間を指定した、主なスケジュール済みバッチ・ジョブのリスト
- その他の必要なスケジュールされていないバッチ・ジョブのリスト

パフォーマンスを計画するには、以下のタスクを完了してください。

#### 『システム・ベンチマークの設定』

適切なシステム・ベンチマークを設定すると、正しく調整されたシステムに対してパフォーマンス・データを提供します。システム変更前およびシステム変更後の両方のパフォーマンス・ベンチマークは、トラブルシューティングおよび計画の両方のために重要な情報を提供します。

#### 13 ページの『システムの拡張時および拡張方法の決定』

業務ニーズが変化するにつれて、ご使用のシステムも変更しなければなりません。変更の準備を行うには、現行システムをモデル化し、システム、構成、またはワークロードが変更された場合に何が起こるかを知りたいと思われることでしょう。

#### ▶▶ 14 ページの『同時マルチスレッド化使用時の決定』

同時マルチスレッド化により、プロセスの共用機能が可能になり、2 つのアプリケーション、または同じアプリケーションの 2 つのスレッドを同時に実行できます。この概念について詳しく調べてください。◀◀

#### 14 ページの『パフォーマンス管理ストラテジーの選択』

異なる業務ニーズには、異なるパフォーマンス管理ストラテジーが必要です。以下に、3 つの基本ビジネス・モデルおよび提案されているパフォーマンス管理ストラテジーを示します。

## システム・ベンチマークの設定

新しい対話式アプリケーションの追加、またはシステムのアップグレードを実行するなど、システム構成に大きな変更を加える前には、システム・ベンチマークを設定するべきです。正確なベンチマーク情報を保守することは、トラブルシューティングに必要な不可欠な情報です。最低限、ベンチマークは 36 ページの『収集サービス』からの現行の収集オブジェクトを含んでいるべきです。ユーザーの環境によっては、134 ページの『Performance Explorer』を使用して、さらに詳細な情報を保守する必要があります。

ベンチマークの設定は、以下のものを必要とします。

- 正しい iSeries<sup>(TM)</sup> の構成が使用可能になっていること。
- アプリケーションおよびデータが典型的かつ有効なものであること。
- すべてのプログラムおよびソフトウェアの正しいバージョンが使用可能であること。
- テストを実行するために必要な数のユーザー、およびワークステーションが使用可能であること。
- 各ユーザー用のトランザクションが定義されていること。

ワークステーション上のユーザーをシミュレートすることのできる特殊な装置を使用しないと、対話式のワークロードに対する意味のあるベンチマークを実行することはほとんど不可能です。もちろん、バッチ・ベンチマークを実行することは、対話式アプリケーションのパフォーマンスをテストするタスクほど複雑ではありません。このタイプのテストにおいては、前述のポイントの最初の 3 つが有効です。しかし、実際の顧客の環境にあるような並行バッチおよび対話式作業のシステム・ベンチマークの設定では、やはり適切な数のユーザーおよびワークステーションが必要です。

▶ IBM<sup>(R)</sup> 開発のベンチマークは、Three-in-One Benchmark と呼ばれ、IT 関連会社が直面する実際の要求を反映しています。Three-In-One Benchmark  は、iSeries サーバーが現在の中小および中堅規模ビジネスにとって、パフォーマンスを心配せずに必要なアプリケーションを実行するのに役立つ優れた解決策であることを、明確に示します。◀

## システムの拡張時および拡張方法の決定

業務ニーズが発展するのに従って、システム・ニーズも発展します。将来のシステム・ニーズおよび成長を計画するには、システム、構成、またはワークロードが変更された場合に何が起きるかを判断する必要があります。このプロセスはトレンド分析と呼ばれ、毎月行うべきです。システムをリソース・キャパシティーの指針に近づけるにつれ、このデータをさらに頻繁に収集したいと思うかもしれません。

トレンド分析は、対話式およびバッチ環境で別々に行うべきです。会社でいくつかの大規模なアプリケーションが使用されているのであれば、そのアプリケーションのトレンド分析を行いたいと思うでしょう。追跡することが重要だと思われるその他の環境は、月末に処理することになります。トレンド分析データを一環して収集するのは重要なことです。システムのワークロードのピークが午前 10 時から午後 2 時の間で、この時間のトレンド分析データを収集した場合は、このデータを別の時間に取得したデータと比較してはなりません。

キャパシティー・プランニングおよびパフォーマンス分析というジョブを正しく実行するには、パフォーマンス・データの収集、分析、保守、および保存を行う必要があります。IBM<sup>(R)</sup> は、キャパシティー・プランニング、リソース見積もり、およびサイジングに役立ついくつかのツールを提供しています。

### 105 ページの『IBM Performance Management for eServer iSeries』

PM iSeries は、データの収集、データの分析、およびデータの保存を完全に自動化し、理解しやすい、要約されたパフォーマンスおよびキャパシティー情報を提供します。PM iSeries は、かぎとなるパフォーマンス標識の分析を続けることにより、システム・リソースを計画し、管理するのに役立ちます。この機能は、OS/400<sup>(R)</sup> ライセンス・プログラムに付属しています。機能を活動化し、データが収集され、IBM の送信されていることを定期的にチェックする以外は、何も行う必要はありません。すべてのコレクション・サイトはネットワーク保護されており、PM iSeries サービスは、所有権のないパフォーマンス・データだけを IBM に送信します。転送時間は、ユーザーによって完全に制御されています。

### 146 ページの『Workload Estimator for iSeries』

Workload Estimator は、特定のタイプのワークロードの見積もりに基づいて、システム・ニーズのサイジングを行うのに役立つツールです。Web ベース・アプリケーションを通して、既存のシステム

の使用状況、パフォーマンスおよび PM iSeries によって報告される成長に適合した、必要とされる iSeries システムへのアップグレードのサイジングが行えます。また追加のオプションとして、ドミノ (Domino<sup>®</sup>)、Java<sup>™</sup>、および WebSphere<sup>®</sup> のような特定のアプリケーションを追加するためのキャパシティーや、複数の AS/400<sup>®</sup> または iSeries の従来型の OS/400 ワークロードを 1 つのシステムに統合するためのキャパシティーも、見積もりに含めることができます。このようなキャパシティーを含めておこなうと、使用している独自のシステムから得られた既存の使用状況データに基づいて、将来のシステム要件を計画することが可能です。

#### 147 ページの『PATROL for iSeries (AS/400) - Predict』

この製品は、高可用性および最適なパフォーマンスのために必要な日常の管理タスクの多くを自動化することにより、iSeries のパフォーマンスを管理しやすくします。さらに、iSeries 環境の成長を計画するのに役立つ、詳細なキャパシティー・プランニング情報を提供します。

パフォーマンス戦略の作成およびインプリメントの詳細については、『『パフォーマンス管理ストラテジーの選択』』を参照してください。

## 同時マルチスレッド化使用時の決定

オペレーティング・システムが非常に多くの数のタスクを並行して実行しているような印象があっても、従来、対称マルチプロセッサ (SMP) 内の各プロセッサは、どんなときでも一度に 1 つのタスクの命令ストリームを実行します。QPRCMLTTSK システム値は、個々の SMP プロセッサが並行して複数の命令ストリームを実行できるかどうかを制御します。各命令ストリームは、個別のタスクまたはスレッドに所属します。可能な場合、個々のプロセッサは、同時に複数のタスクを実行します。この使用の効果として、システムのパフォーマンス・キャパシティーが増加したり、またはマルチスレッド化されたアプリケーションの応答が改善されたりすることがあります。複数の命令ストリームを同時に実行しても、指定されたタスクのパフォーマンスを改善することにはなりません。任意のパフォーマンスに関する推奨事項を適用した場合でも、結果は環境によって異なります。

マルチスレッド化が実行される方法はハードウェア・モデルに依存するので、パフォーマンス・キャパシティーの増加は異なります。@server 5xx モデルは、同時マルチスレッド化 (SMT) と呼ばれる概念を用いたこのアプローチをサポートします。このアプローチは、一部の Intel<sup>®</sup> プロセッサではハイパースレッド化と呼ばれ、プロセッサ機能を共用して各タスクの命令を同時に実行します。旧式のプロセッサは、ハードウェア・マルチスレッド化 (HMT) というアプローチを使用します。ハードウェア・マルチスレッド化アプローチでは、ハードウェアは任意の長期間の処理遅延イベント、たとえばキャッシュ・ミスなどの間で切り替えを行います。モデルの中には、マルチスレッド化のどの方法もサポートせず、したがって QPRCMLTTSK システム値にパフォーマンス効果がないものもあります。

QPRCMLTTSK システム値は共用プロセッサ・リソースの並列使用を可能にするので、パフォーマンスの増加はアプリケーションおよびモデルにかなり依存します。これを使用することでどのようなパフォーマンスの増加が見込めるかについてのガイドラインは、iSeries<sup>™</sup> Performance Capabilities Reference  を参照してください。場合によっては、このシステム値を使用不可にしたほうが効率の良いアプリケーションもあります。

## パフォーマンス管理ストラテジーの選択

よいパフォーマンス管理ストラテジーを開発することは、システムのパフォーマンスを管理するのに役立ちます。パフォーマンス管理ストラテジーは、ユーザーがパフォーマンスの管理に費やすことのできる時間的な余裕がどれくらいあるかに大きく依存します。小さな会社で働いている場合は、ユーザーはさまざまな異

なるビジネスの局面を管理することがあり、パフォーマンスの管理に多くの時間を費やすことができません。多くの大きな企業では、システムの調整を行い、効率的に動作するようにするため、パフォーマンスの専門家を採用します。

基本的なパフォーマンス管理ストラテジーを決定し、どのパフォーマンス・アプリケーションを使用するかを識別するには、ユーザーの企業を、小規模、中規模、大規模の 3 つのカテゴリーに分類します。ビジネス・リソースはそれぞれのサイズによってさまざまで、管理戦略もそれに依拠して異なります。

### 小規模ビジネス

小規模ビジネスでは多くの場合、パフォーマンスの管理に費やすことのできるリソースは、大規模ビジネスの場合よりも少なくなります。そのため、可能であればなるべく自動化します。パフォーマンス・データのコンパイルおよびレポートの生成を行う IBM にデータを直接送信するために、PM iSeries<sup>TM</sup> を使用します。これは時間の節約になるだけでなく、ユーザーの iSeries サーバーにアップグレードが必要な場合、IBM が提案することもできます。

小規模ビジネスで推奨されるパフォーマンス・アプリケーションを以下にリストします。

36 ページの『収集サービス』

今後の分析のために、ユーザーが定義した間隔でサンプル・データを収集します。

103 ページの『グラフ・ヒストリー』

収集サービスで収集されたパフォーマンス・データを表示します。

105 ページの『IBM Performance Management for eServer iSeries』

システム・パフォーマンス・データの収集、アーカイブ、および分析を自動化します。

123 ページの『Performance Tools』

システム・パフォーマンス情報の取得、分析、および保守を行います。

95 ページの『iSeries<sup>TM</sup> ナビゲーターのモニター』

iSeries システム・パフォーマンスのグラフィカル表現を監視し、事前定義されたイベントまたは状態に対して自動応答します。

### 中規模ビジネス

ほとんどの場合、中規模ビジネスは小規模ビジネスに比較して、多くのリソースをパフォーマンスの管理に費やすことができます。それでも、可能な限り自動化することが望ましく、PM iSeries を使用することによって利益が得られます。

中規模ビジネスで推奨されるパフォーマンス・アプリケーションを以下にリストします。

36 ページの『収集サービス』

今後の分析のために、ユーザーが定義した間隔でサンプル・データを収集します。

103 ページの『グラフ・ヒストリー』

収集サービスで収集されたパフォーマンス・データを表示します。

105 ページの『IBM Performance Management for eServer iSeries』

システム・パフォーマンス・データの収集、アーカイブ、および分析を自動化します。

123 ページの『Performance Tools』

システム・パフォーマンス情報の取得、分析、および保守を行います。

95 ページの『iSeries<sup>TM</sup> ナビゲーターのモニター』

iSeries システム・パフォーマンスのグラフィカル表現を監視し、事前定義されたイベントまたは状態に対して自動応答します。

134 ページの『Performance Explorer』

特定のアプリケーションまたはシステム・リソースに関する詳細な情報を収集します。

## 大規模ビジネス

大規模ビジネスでは、パフォーマンスの管理にリソースを費やすことができます。

大規模ビジネスで推奨されるパフォーマンス・アプリケーションを以下にリストします。

36 ページの『収集サービス』

今後の分析のために、ユーザーが定義した間隔でサンプル・データを収集します。

103 ページの『グラフ・ヒストリー』

収集サービスで収集されたパフォーマンス・データを表示します。

105 ページの『IBM Performance Management for eServer iSeries』

システム・パフォーマンス・データの収集、アーカイブ、および分析を自動化します。

123 ページの『Performance Tools』

システム・パフォーマンス情報の取得、分析、および保守を行います。

95 ページの『iSeries<sup>TM</sup> ナビゲーターのモニター』

iSeries システム・パフォーマンスのグラフィカル表現を監視し、事前定義されたイベントまたは状態に対して自動応答します。

134 ページの『Performance Explorer』

特定のアプリケーションまたはシステム・リソースに関する詳細な情報を収集します。

142 ページの『iDoctor for iSeries』

システムおよびアプリケーションのパフォーマンスを向上させるため、トレース・データを分析します。

143 ページの『Performance Trace Data Visualizer (PTDV)』

Java<sup>TM</sup> アプリケーションからトレース・データを表示します。

---

## パフォーマンスを管理するための環境のセットアップ

iSeries<sup>TM</sup> サーバーには、システム・パフォーマンス・データを定期的に収集し、システムのパフォーマンスの傾向および潜在的な問題をモニターするためのいくつかのツールが含まれています。ユーザーの個別の要件や環境によって、投入するために選択するツール、および設定する構成の選択が決定されます。システムを効果的に設定すると、システムの成長に合わせた正確なキャパシティー・プランニングが可能になり、パフォーマンス上の問題が発生したときにも解決することができます。

システム・パフォーマンスのデータを収集、モニター、および分析するためのツールとその構成について、下記のトピックを使用して確認してください。

### 36 ページの『収集サービス』

収集サービスは、システム・パフォーマンス・データの定期的な収集を管理します。このツールは定期的にデータを収集し、収集オブジェクトと呼ばれるアーカイブを作成します。これらの収集オブジェクトは、いくつかのツールから直接アクセスされたり、または、独自のカスタム照会または他のツ

ールおよびレポートによる分析のためのデータベース・ファイルのセットに変換されます。収集サービスは主に他のアプリケーションに対してデータを提供するため、使用するその他のツールは、データをどの程度の周期で収集するか、収集するデータのタイプ、およびシステム上にデータを保持する時間を含めた、構成の選択に大きく影響します。

#### **105 ページの『IBM Performance Management for eServer iSeries』**

PM iSeries は収集サービスを使用して所有権を主張されないパフォーマンス・データを集め、保存と専門家の分析のために IBM<sup>(R)</sup> に送信します。これでお客様が保存し保持する必要はなくなります。その後、Web ブラウザーを使用して、システム・パフォーマンスに関する詳細な報告書、および推奨にアクセスすることができます。

#### **95 ページの『iSeries<sup>(TM)</sup> ナビゲーターのモニター』**

iSeries ナビゲーターに組み込まれているモニターは、収集サービス・データを使用して、特定のシステム・パフォーマンスの要素を追跡します。さらに、CPU 使用率のパーセンテージやジョブの状況など、特定のイベントが生じた場合に、指定されたアクションを実行できます。このトピックを使用して、これらのモニターの使い方、およびシステムにセットアップする方法を確認してください。

---

## **iSeries<sup>(TM)</sup> パフォーマンスの管理**

パフォーマンスの管理を適切に行うことは、システムが効率的にリソースを使用し、サーバーがユーザー・ニーズや業務ニーズに対して最良のサービスを提供することを確実にします。さらに、効果的なパフォーマンス管理は、システム内の変更に迅速に対応することを可能にし、高価なアップグレードや保守費用を先送りすることにより、コストを節約することができます。

システムのパフォーマンスに影響のある要素を理解することは、問題に対応し、より良い長期計画を作成するのに役立ちます。効果的な計画により、開発によるパフォーマンスの問題の可能性を回避し、現在および増大するワークロードを処理するシステム能力を確保することができます。

システムのパフォーマンスの保守、およびパフォーマンス上の問題を解決する方法について、下記のトピックを使用して確認してください。

#### **18 ページの『パフォーマンスの追跡』**

時間をかけてシステム・パフォーマンスを追跡すると、ユーザーのシステムの発展を計画することが可能になり、パフォーマンス上の問題の原因を切り分けて、原因を識別するのに役立つデータを得ることができます。どのアプリケーションを使用するのか、また定期的にパフォーマンス・データを収集する方法を確認してください。

#### **18 ページの『パフォーマンスの問題の調査』**

パフォーマンス上の問題の識別、および解決に役立つさまざまな選択可能なオプションがあります。パフォーマンス上の問題の原因を見つけるのに役立つ、使用可能なツールおよびレポートの使用方法について確認してください。

#### **25 ページの『パフォーマンス・データの表示』**

パフォーマンス・データを収集した後に、ユーザーの目的に合った最も適切なツールを使用してデータを表示する方法を確認してください。

#### **25 ページの『パフォーマンスのチューニング』**

パフォーマンス上の問題を識別したなら、問題を修正するためにシステムを調整します。

### 28 ページの『e-business パフォーマンスの管理』

e-business 環境のパフォーマンスの管理には、OS/400<sup>(R)</sup> 管理者にとっての新しい問題がいくつか伴います。e-business アプリケーションのパフォーマンスの計画、追跡、および改善に役立つ情報と資源については、このトピックを参照してください。

## パフォーマンスの追跡

iSeries<sup>(TM)</sup> サーバーのシステム・パフォーマンスを追跡すると、傾向を見極めて、システム構成の調整とシステムのアップグレードの時期と方法について最良の選択をする手掛かりとすることができます。さらに、問題が発生したときには、パフォーマンス上の問題の原因の範囲を絞り込み、適切な解決策を見つけるために、その前後のパフォーマンス・データを手に入れることは不可欠です。

iSeries サーバーには、パフォーマンスの傾向を追跡し、iSeries パフォーマンス・データのヒストリー・レコードを保持するいくつかのアプリケーションがあります。こうしたアプリケーションのほとんどは収集サービスが収集したデータを使用します。収集サービスを使用して、以下の領域で傾向を監視できます。

- システム・リソースの使用状況の傾向。この情報を使用して、システム構成の変更やアップグレードを計画し、明確に調整することができます。
- 構成中の物理的構成要素に対するストレスの識別
- ピーク時と通常時、対話型ジョブとバッチ・ジョブのシステム・リソース使用量のバランス
- 構成変更。収集サービスのデータを使用して、ユーザー・グループの追加、対話型ジョブの増加、およびその他の変更の影響を正確に予測することができます。
- システム上の他の活動に問題を引き起こしている恐れのあるジョブの識別
-  使用可能な通信回線用の使用率レベルおよび傾向 

以下のツールがシステム・パフォーマンスをモニターするのに役立ちます。

### 36 ページの『収集サービス』

収集サービスは、ユーザー定義の時間間隔でパフォーマンス・データを収集し、この情報をシステム上の収集オブジェクトに保管します。モニター、グラフ・ヒストリー、PM iSeries などの他の多くのツールや、Performance Tools ライセンス・プログラムの多くの機能は、これらの収集オブジェクトからデータを入手します。

### 103 ページの『グラフ・ヒストリー』

グラフ・ヒストリーは指定された期間にわたって収集サービスが収集したパフォーマンス・データをグラフィカル・ユーザー・インターフェース (GUI) で表示します。表示できる期間の長さは収集オブジェクトをどれだけ長く保存しているかと PM iSeries を使用しているかどうかによります。

### 105 ページの『IBM Performance Management for eServer iSeries』

PM iSeries はシステム・パフォーマンス・データの収集、保存、および分析を自動化し、システム資源およびキャパシティーを管理するのに役立つクリア報告書を戻します。

## パフォーマンスの問題の調査

パフォーマンスを収集または分析するツールの多くはトレース・データとサンプル・データのいずれかを使用します。収集サービスはさまざまなシステム・リソース上で定期的にサンプル・データを収集します。いくつかのツールがこのサンプル・データを分析し、それに基づいて報告し、これを使用してシステム・リソースの使用状況の全体像をつかみ、多くの共通なパフォーマンスに関する質問に答えることができます。より詳細なパフォーマンス情報については、いくつかのツールがトレース・レベルのデータを生成します。し

ばしば、トレース・レベルのデータは、システム上のジョブとアプリケーションの動作とリソース使用量について詳細な情報を提供してくれます。 Performance Explorer および STRPFRTRC コマンドがトレース・データを生成する 2 つの共通ツールです。

たとえば、システムの実行が遅い場合、システム・モニターを使用して問題を探します。 CPU 使用率が高いことが分かれば、異常に大量のリソースを使用しているように見えるジョブを識別できるかもしれません。そうすれば、構成を変更することによって、問題を正せるかもしれません。しかしながら、問題によっては追加情報が必要です。そのジョブのパフォーマンスについて詳細な情報を得るため、Performance Explorer セッションを開始し、iSeries<sup>TM</sup> システム上でのそのジョブの動作について詳細な情報を集め、もしかすると問題を引き起こしているプログラムに変更を加えることができます。

パフォーマンス・データの収集についてもっと知るために、以下のトピックを使用してパフォーマンス管理アプリケーションをいつどのように使うか学んでください。

### 『パフォーマンスの問題の識別』

パフォーマンス上の問題の識別に関係した共通ステップを考えます。

#### 20 ページの『共通のパフォーマンスの問題の識別と解決』

多くのさまざまなパフォーマンス上の問題が iSeries システムの共通域にしばしば影響します。共通域の問題の調査解決方法を考えます。 ➤ たとえば、バックアップおよびリカバリーなどです。 ⬅

#### 21 ページの『システム・パフォーマンス・データの収集』

収集サービスは定期的にシステム・パフォーマンスに関する情報を収集します。しばしば、パフォーマンス・データの分析はこの情報から始まります。

#### 22 ページの『システム・リソース使用状況についての情報の収集』

いくつかのツールが CPU、ディスク・スペース、対話式能力、および多くのほかの要素のようなリソースがどう使用されているかをモニターします。問題のある領域を識別するのにこれらのツールを使用できます。

#### 22 ページの『アプリケーションのパフォーマンスについての情報の収集』

いくつかの理由でアプリケーションの実行が遅くなることがあります。 OS/400<sup>®</sup> に組み込まれているいくつかのツールや他のライセンス・プログラムを使用してもっと情報を得ることができます。

#### 24 ページの『シナリオ: アップグレードまたはマイグレーション後にシステム・パフォーマンスを改善する』

このシナリオは、システムをアップグレードまたは移行したところ、以前よりも実行速度が遅くなったように思える、というものです。このシナリオはパフォーマンス上の問題を識別して修正するのに役立ちます。

## パフォーマンスの問題の識別

パフォーマンスの問題を識別しようとする際には、ハードウェア構成がそのワークロードをサポートするのに適切かどうかについて評価することが重要になります。 CPU の能力は充分ですか？ 主記憶域はさまざまなアプリケーションを処理するのに充分ですか？ これらの疑問に対して、たとえばキャパシティー・モデル化技法を使用してまず回答を出しておく、あとで不必要な作業をしないで済みます。

問題の症状と達成目標を理解することにより、分析担当者は問題の原因を説明できる仮説を立てることができます。分析担当者は、システム・パフォーマンスを測定するために、OS/400<sup>®</sup> オペレーティング・システムおよび Performance Tools ライセンス・プログラムで使用可能なコマンドおよびツールを使用することができます。

測定したデータを検討することは、問題をさらに定義し、仮説が妥当であるか却下すべきものであるかを判断するのに役立ちます。1 つまたは複数の明白な原因が分離されると、ソリューションを提案することができます。一度に 1 つのソリューションを処理するようにすると、プログラムを再設計してテストすることが可能です。分析担当者のツールは多くの場合、ソリューションの効果を測定し、副次作用があるかどうかについても探し出すことができます。

最適なパフォーマンスを達成するには、重要なシステム・リソースの相互の関係を知り、それらのリソース（つまり CPU、ディスク、主記憶装置、および通信の場合のリモート回線）の間のバランスがとれるようにしなければなりません。これらのリソースは、それぞれがパフォーマンス低下の原因となる可能性があります。

システム・パフォーマンスを改善することは、それが対話式スループット、対話式応答時間、バッチ・スループット、あるいはそれらの組み合わせに対する改善であっても、単に活動レベルまたはプール・サイズを調整することからアプリケーション・コード自体を変更することまで、多くの形態を取ることが考えられます。この場合、活動レベルは、処理装置を獲得するために同時に競合し得るジョブの最大数を指定するサブシステムの 1 つの特性です。

## 共通のパフォーマンスの問題の識別と解決

iSeries<sup>(TM)</sup> サーバーでパフォーマンスの問題が発生すると、しばしば最初に影響を受けるのはシステムのある領域です。このようなシステム領域のパフォーマンスを調査するのに使用可能ないくつかの方法について次の表を参照してください。このような領域の多くはシステム・モニターのメトリックとして使用可能です。しかしながら、それらについての情報を見る他の方法もいくつかあります。

領域	説明	使用可能なツール
プロセッサ負荷	システム上にジョブが多すぎないかどうか、あるいは一部のジョブがプロセッサ時間を独占していないかどうかを判別します。	<ul style="list-style-type: none"> <li>活動ジョブの処理 (WRKACTJOB) コマンド</li> <li>システム活動の処理 (WRKSYSACT) コマンド (Performance Tools ライセンス・プログラムに含まれています)</li> <li>iSeries ナビゲーターの実行管理機能</li> <li>iSeries ナビゲーター・システム・モニター内の CPU 使用状況メトリック</li> </ul>
主記憶装置	ページ不在や待ち状態から不適格状態に移行しているトランザクションを調査します。	<ul style="list-style-type: none"> <li>ディスク状況の処理 (WRKDSKSTS) コマンド</li> <li>iSeries ナビゲーター・システム・モニター内のディスク装置メトリック</li> <li>システム状況の処理 (WRKSYSSTS) コマンド</li> <li>iSeries ナビゲーターの実行管理機能の下にあるメモリー・プール機能</li> </ul>
ディスク	アームが少なすぎないかどうか、あるいはアームが遅すぎないかどうかを判別します。	<ul style="list-style-type: none"> <li>ディスク状況の処理 (WRKDSKSTS) コマンド</li> <li>iSeries ナビゲーター・システム・モニター内のディスク・アーム使用状況メトリック</li> <li>Performance Tools のシステムと構成要素報告書</li> </ul>
通信	遅い回線、回線上のエラー、あるいは特定回線への過剰なユーザーの集中を検出します。	<ul style="list-style-type: none"> <li>Performance Tools の構成要素報告書</li> <li>iSeries ナビゲーター・システム・モニター内の LAN 使用状況メトリック</li> </ul>

領域	説明	使用可能なツール
IOP	使用率に偏りのある IOP がないかどうか、あるいは IOP が不足していないかどうかを判別します。	<ul style="list-style-type: none"> <li>Performance Tools の構成要素報告書</li> <li>iSeries ナビゲーター・システム・モニター内の IOP 使用状況メトリック</li> </ul>
ソフトウェア	ロックと相互排他 (mutex) を調べます。	<ul style="list-style-type: none"> <li>Performance Tools のロック報告書</li> <li>Performance Tools のトレース報告書</li> <li>オブジェクト・ロックの処理 (WRKOBJLCK) コマンド</li> <li>iSeries ナビゲーターの実行管理機能で疑わしいジョブを右マウス・ボタン・クリックし、「詳細」 → 「ロックされたオブジェクト」を選択。</li> </ul>
>> バックアップおよびリカバリー <<	>> バックアップおよびリカバリーに影響する領域を調査し、操作を保管および復元します。 <<	>> <ul style="list-style-type: none"> <li>iSeries Performance Capabilities Reference (Save/Restore Performance の章) </li> <li>サーバーを再始動した後、バックアップにこんなに時間がかかるのはなぜですか。</li> <li>新しいリリースにアップグレードした後、バックアップにかかる時間が前よりも長くなったのですが、それはなぜですか。</li> <li>サーバーのハードウェアを変更した後、バックアップにかかる時間が前よりも長くなったのですが、それはなぜですか。</li> </ul> <<

## システム・パフォーマンス・データの収集

データの収集は、パフォーマンスを改善するための重要なステップです。パフォーマンス・データを収集するとき、応答時間やスループットを理解するために使用できるサーバーに関する情報を収集します。データを収集することは、ユーザーの作業を行うために関係のあるサーバー、または一連のサーバーのパフォーマンス状況を把握する手段です。データ収集により、後に行われるすべての比較および分析に備えた、コンテキストや開始点が提供されます。初めてデータ収集を使用する時、ユーザーは将来の改善に対するベンチマークと、現在のパフォーマンスを改善する出発点を手にすることになります。調整を行い、応答時間を改善し、システムが最高のパフォーマンスを得るのを助けるために収集するパフォーマンス・データを使用することができます。パフォーマンス上の問題の分析は、多くの場合「何か変わったのか」という単純な質問から始まります。パフォーマンス・データはその質問に答えるのに役立ちます。

収集サービスを使用して、パフォーマンス・データの収集、パフォーマンス・データの作成 (CRTPFDRDTA) コマンドでパフォーマンス・ファイルの作成、パフォーマンス・データの変換 (CVTPFRDITA) コマンドまたは iSeries<sup>TM</sup> ナビゲーターの Performance Tools プラグインで現行リリースに交換、そしてパフォーマンス・データベース・ファイルの情報を使用して報告書作成または独自の照会を行うことができます。

パフォーマンス・データの詳細については、以下のものを参照してください。

### 36 ページの『収集サービス』

分析のためにパフォーマンス・データを収集する方法と収集をカスタマイズする方法を参照してください。

### 90 ページの『パフォーマンス・データ・ファイル』

使用可能なパフォーマンス・データベース・ファイルの概要を調べ、各パフォーマンス・データベース・ファイルの詳細なフィールド・データを参照してください。

さらに、Performance Management API または 3 ページの『新機能: 収集サービス』を使用して、収集を開始、終了したり、コレクションを循環させたりすることができます。また、収集されたデータと関連するシステム・パラメーターを変更したり、検索したりすることもできます。

## システム・リソース使用状況についての情報の収集

iSeries<sup>(TM)</sup> サーバーとアプリケーションが使用可能なリソースを使用している状態をモニターおよび追跡するのに役立つツールが数多くあります。この情報を問題分析の出発点として使用し、傾向を識別し、キャパシティー・プランニングとシステムの成長を管理するのに役立ててください。

以下のトピックを参照して、これらのツールをいつどのように使うかを調べてください。

### 95 ページの『iSeries<sup>(TM)</sup> ナビゲーターのモニター』

iSeries ナビゲーターに組み込まれたモニターはさまざまなメトリックの現在および最近のデータを提供します。さらにある種のイベントが発生したときに特定のアクションをとるように構成することができます。

### 143 ページの『OS/400 パフォーマンス用の処理コマンド』

OS/400 には、システム・パフォーマンスの管理および保守に役立つ複数の重要な機能があります。

### 105 ページの『IBM Performance Management for eServer iSeries』

PM iSeries は収集サービスを使用して所有権を主張されないパフォーマンス・データを集め、保存と専門家の分析のために IBM<sup>(R)</sup> に送信します。これでお客様が保存し保持する必要はなくなります。ご使用のシステムのパフォーマンスおよび傾向分析について詳細な報告書と推奨事項を Web ブラウザーでアクセスできます。

## アプリケーションのパフォーマンスについての情報の収集

アプリケーションのパフォーマンスについての情報の収集は、システム・パフォーマンスについての情報の収集と大きく異なります。アプリケーション情報の収集は Performance Explorer、Performance Trace Data Visualizer (PTDV)、および iDoctor のようなパフォーマンス・アプリケーションによってのみ行えます。別の方法として、95 ページの『iSeries<sup>(TM)</sup> ナビゲーターのモニター』を使用して個々のサーバーのパフォーマンスを追跡したり、123 ページの『Performance Tools』を使用してサーバーのジョブを追跡および分析することにより、アプリケーション・パフォーマンスの概要を取得することもできます。

注: アプリケーションのパフォーマンス・データを収集するとシステムのパフォーマンスに目立って影響が出る場合があります。収集を開始する前に、他の収集オプションをすべて試したか確認してください。

### 134 ページの『Performance Explorer』

このツールは、一般的なパフォーマンス・モニターを行うツールを使用していたのでは識別できないパフォーマンス問題の原因を見つけ出すのに役立ちます。コンピューター環境がサイズと複雑さの両面で拡大すると、当然パフォーマンス分析も同様に複雑になります。Performance Explorer は、複合したパフォーマンスの問題に関するデータを収集することにより、そのような複雑さの拡大に対処しています。

Performance Explorer はプログラムのパフォーマンスを理解し改善したいと思っているアプリケーション開発者のために設計されています。複雑なパフォーマンスの問題を識別し明確にできるようにすることによって、パフォーマンス管理の知識のあるユーザーにも役に立ちます。

### 143 ページの『Performance Trace Data Visualizer (PTDV)』

このツールは、iSeries で稼働するアプリケーションのパフォーマンス分析に使用できる Java<sup>TM</sup> アプリケーションです。PTDV は OS/400<sup>®</sup> オペレーティング・システムの Performance Explorer 機能と共に働き、分析者がプログラム・フローを見てトレース、ジョブ、スレッド、およびプロセスごとにまとめられた詳細 (CPU 時間、現在のシステム時刻、サイクル数、命令数など) を入手できるようにします。Java アプリケーション・トレースをビジュアル化するとき、Java ロック動作についての情報と共に作成されたオブジェクトの数とタイプなどのような追加の詳細も表示できます。また、WebSphere<sup>®</sup> Application Server が生成する Performance Explorer イベントもサポートします。PTDV では、カラムのソート、データのエクスポートおよびさまざまなレベルでのデータ要約が可能です。

詳しくは、Performance Trace Data Visualizer  Web サイトに進んでください。

### 142 ページの『iDoctor for iSeries』

iDoctor の PEX Analyzer 機能には、システムおよびアプリケーションのパフォーマンスを改善しようとトレース・データを分析するために特に調整されたソフトウェア・ツールが入っています。この詳細な分析により、ディスク操作、CPU 使用率、ファイル・オープン操作、マシン・インターフェース (MI) プログラム、待ち状態、ディスク・スペース使用量などの低レベル要約を取得できます。クライアント構成要素は、iSeries トレース・データを圧縮したりグラフィックに表示する機能を持つ iSeries ナビゲーター・プラグインです。

### パフォーマンス・トレース開始 (STRPFTRC) コマンド

OS/400 にはマルチプログラミングおよびトランザクション・データを収集するコマンドがあります。このコマンドは前のリリースで STRPFMON が収集していたデータを収集します。このコマンドを実行後、『トレース・データのダンプ』でデータをデータベース・ファイルにエクスポートできます。

**トレース・データのダンプ:** ダンプはシステムのパフォーマンスに影響を与えるため、いつトレース・データのダンプを行うかを決定するのは重要な問題です。トレースのダンプ (DMPTRC) コマンドは、内部トレース表内の情報をデータベース・ファイルに書き込みます。負荷のかかったシステムや、高優先順位の (対話式の) ジョブで、活動がピークに達しているときにトレース・データをダンプするのは好ましくありません。トレース・ダンプを後で行うこともできますが、データの存在を忘れないうちにダンプしておくたものです。もし、何らかの理由でトレース表がクリアされるようなことがあれば、トレース・データが失われてしまうからです。しかし、ダンプをわずかに遅らせて、追跡ダンプ (DMPTRC) コマンドを使用してバッチ・ジョブでトレースをダンプすれば、ユーザーのパフォーマンスを維持することが可能です。

トレース・データをダンプするには、次のコマンドを実行します。

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

コマンドを実行する際には、データの保管先としてメンバー名とライブラリー名を指定する必要があります。36 ページの『収集サービス』を使えば、トレース情報と同時に、サンプル・ベース・データの収集を行うことができます。これと同じように、サンプル・データとトレース・データを一緒に収集する場合には、それらのデータを保管するメンバーの名前が一致している必要があります。つまり、CRTPFRTA TOMBR および TOLIB パラメーターで指定した名前と、DMPTRC MBR および LIB パラメーターで指定した名前が一致していなければなりません。

## シナリオ: アップグレードまたはマイグレーション後にシステム・パフォーマンスを改善する

最近 iSeries<sup>(TM)</sup> サーバーを最新のリリースにアップグレードしたとします。アップグレードが完了し、通常操作を再開した後は、システム・パフォーマンスは著しく低下しています。パフォーマンス上の問題の原因を突き止め、システムを通常レベルに復元したいとします。

### 変更点を分離する

オペレーティング・システムのアップグレード後にパフォーマンスの低下を招く問題がいくつかあります。OS/400<sup>(R)</sup> および Performance Tools ライセンス・プログラム (5722-PT1) に組み込まれているパフォーマンス管理ツールを使用して、パフォーマンス上の問題についての詳細情報を入手し、疑わしい問題を可能性のある原因へと絞り込むことができます。

1. CPU 使用率を調べる。アップグレード後に、ジョブは、必要なリソースの一部にアクセスできなくなる場合があります。これは、許容できないほどの量の CPU 資源が 1 つのジョブで消費されるという結果を招く場合があります。
  - WRKSYSACT、WRKSYSSTS、WRKACTJOB、または iSeries ナビゲーターのシステム・モニターを使用して、CPU の合計使用率を検出する。
  - CPU 使用率が高い (例: 90% を超えている) 場合は、アクティブ・ジョブが使用している CPU の量を調べる。1 つのジョブで 30% を超える CPU 資源が消費されている場合は、欠落ファイルと呼び出しているか、オブジェクトが欠落していることが考えられます。その場合は、ベンダーに連絡してベンダー提供のプログラムを入手するか、ジョブの所有者またはプログラマーに連絡して追加プログラムを入手してください。
2. STRPFRTTC コマンドでパフォーマンス・トレースを開始してから、131 ページの『Performance Tools 報告書』を使用して、以下の考えられる問題を識別し、訂正します。
  - マシン・プールの 1 秒あたりのページ不在率が 10 より高い場合は、マシン・プールに割り当てるメモリーを増やして、不在率がこのレベルより低くなるようにする。
  - ディスク使用率が 40% を超えている場合は、待ち時間およびサービス時間を調べる。これらの値が許容範囲内の場合は、優先順位を管理するためのワークロードを削減しなければならない場合があります。
  - IOP 使用率が 60% を超えている場合は、IOP を追加し、いくらかのディスク資源を割り当てる。
  - ユーザー・プールのページ不在率が許容できないほど高い場合は、トピック『27 ページの『パフォーマンスの自動調整』』を参照してください。
3. 131 ページの『Performance Tools 報告書』を実行して、**占有ロック競合報告書**を参照する。占有またはロック競合数が高い場合は、公開アクセス・パス・サイズを 1TB に設定してください。占有またはロック競合がユーザー・プロファイルで起こっている場合で、参照されたユーザー・プロファイルが多くのオブジェクトを所有している場合は、そのプロファイルが所有するオブジェクトの数を減らしてください。
4. 「**タスク切り替え (Task switch)**」 オプションを指定して 142 ページの『iDoctor for iSeries』を 5 分間実行する。その後、タスク切り替えモニターを使用して結果トレース・データを分析します。以下を確認して解決します。
  - CPU 待ちのジョブ数
  - 不在ジョブ数
  - 占有競合数

主要なシステム変更後のパフォーマンス上の問題解決について詳しくは、 **iSeries Performance Capabilities**

**Reference**  を参照してください。

## パフォーマンス・データの表示

パフォーマンス・データを表示すると、システムのパフォーマンスをより正確に分析するのに役立ちます。パフォーマンス・データはさまざまな方法で表示できますが、状況によって特定のパフォーマンス・アプリケーションが適している場合があります。ほとんどのアプリケーションは、36 ページの『収集サービス』を使用して収集されたデータか、パフォーマンス・トレースから収集されたデータを表示します。データにアクセスする最善の方法は、パフォーマンス上の問題を解決しようとしているのか、今後の成長を考慮してシステム・パフォーマンスをモニターするのか、傾向を見分けるのかによって異なります。

### 近況リアルタイム・パフォーマンス・データを表示する

現在または最近のパフォーマンス情報を表示するには、以下のツールを使用します。

#### 143 ページの『OS/400 パフォーマンス用の処理コマンド』

基本オペレーティング・システムには、システム・パフォーマンスの特定の領域についての現行情報を表示できる多くのコマンドがあります。

#### 126 ページの『Performance Tools プラグイン』

Performance Tools ライセンス・プログラムには、収集サービスの収集オブジェクトのパフォーマンス・データを表示する、iSeries<sup>(TM)</sup> ナビゲーター用のプラグインが組み込まれています。システム上のジョブについての詳細情報を表示したり、Performance Tools の報告書を印刷することもできます。

#### 95 ページの『iSeries<sup>(TM)</sup> ナビゲーターのモニター』

これらのモニターは、多くのシステム要素のパフォーマンス・データを表示します。モニター・データは、収集オブジェクトに基づいており、収集サービスの収集間隔に従ってデータが収集されたときに表示されます。

### ヒストリー・パフォーマンス・データを表示する

システム上に保管されているデータを表示するには、以下のツールを使用します。

#### 105 ページの『IBM Performance Management for eServer iSeries』

PM iSeries はシステム・パフォーマンス・データの収集、保存、および分析を自動化し、システム資源およびキャパシティーを管理するのに役立つクリア報告書を戻します。

#### 103 ページの『グラフ・ヒストリー』

グラフ・ヒストリーは、収集サービスの保存期間に基づいて、最大で 1 週間相当のパフォーマンス・データをグラフィカルに表示します。PM iSeries を使用すると、グラフ・ヒストリーはより長い期間のデータ収集を表示できます。

## パフォーマンスのチューニング

パフォーマンス調整の主な目的は、サーバーがシステム資源を最大限に活用できるようにし、ワークロードを可能な限り効果的に管理することにあります。パフォーマンス調整は、システムのパフォーマンスを調整する 1 つの方法であり、手動で、あるいは自動的に行うことができます。システムを調整するためのオプションはたくさんあります。それぞれのシステム環境はどれも固有なものであるため、そのパフォーマンスを観察して、その環境にとって最良の調整を施すことが必要です。言い換えれば、パフォーマンス・モニタ

ーを定期的に行う必要があります。パフォーマンスを調整する前に実行すべきパフォーマンス・モニターのステップの詳細については、『17 ページの『iSeries<sup>TM</sup> パフォーマンスの管理』』を参照してください。

IBM<sup>®</sup> では、ディスクから読み取られる物理入出力要求の数を減らすことによって入出力サブシステムとシステム応答時間の両方を向上させるツールも提供しています。144 ページの『拡張キャッシュ』方法を学んでください。

さらに、プロセスおよびスレッドがメモリーおよびプロセッサ資源の類縁性を改善するための、いくつかのチューニング・オプションを考慮する必要があるかもしれません。詳しくは、スレッド類縁性システム値 (Thread affinity system value) を参照してください。同時マルチスレッド化については、プロセッサ・マルチタスキング・システム値 (Processor multitasking system value) を参照してください。

パフォーマンス調整の詳細については、以下のトピックを参照してください。

#### 『パフォーマンス調整の基本』

システムのパフォーマンスを調整するには、初期調整値をセットアップし、システム・パフォーマンスを監視し、値を検討し、調整対象を判別する必要があります。

#### 27 ページの『パフォーマンスの自動調整』

ほとんどのユーザーの場合は、自動的にパフォーマンス調整を行うようにシステムをセットアップできます。出荷される時点の新しいシステムには、あらかじめ、自動的に調整を行うように構成が行われています。

## パフォーマンス調整の基本

パフォーマンスの調整を開始するには、まず、初期マシン・プール・サイズと初期ユーザー・プール・サイズを決定することによって、初期調整値を設定する必要があります。その後、システム・パフォーマンスの監視を開始できます。

### 初期調整値を設定する

初期調整値の設定には、効果的にシステムを調整するために、最初にシステム・プール・サイズと活動レベルを構成するステップが含まれます。初期値は推定値に基づいています。したがって、システムが活動状態にある間に、推定値をさらに調整しなければならない場合があります。初期調整値は、以下のステップで設定します。

- 初期マシン・プール・サイズを決定する
- 初期ユーザー・プール・サイズを決定する

### システム・パフォーマンスを監視する

システム・パフォーマンスを監視するために、システム状況の処理 (WRKSYSSTS)、ディスク状況の処理 (WRKDSKSTS)、および活動ジョブの処理 (WRKACTJOB) コマンドを使用できます。各監視期間について、パフォーマンスの目標値に照らしてシステム・パフォーマンスの測定値を考察および評価する必要があります。

1. 不規則なシステム活動を除去する。重度のパフォーマンス低下の原因となりうる不規則な活動には、たとえば、対話式プログラム・コンパイル、通信エラー・リカバリー手順 (ERP)、QUERY ファイルのオープン (OPNQRYP)、サインオフ活動などがあります。
2. WRKSYSSTS、WRKDSKSTS、および WRKACTJOB コマンドを使用して、パフォーマンス・データを表示する。パフォーマンス・データは、Performance Tools コマンド (システム活動の処理 (WRKSYSACT)) を使用して表示することもできます。
3. システムのデータ収集を最低 5 分行う。

4. パフォーマンスの目標値に照らしてパフォーマンスの測定値を評価する。通常の測定値は以下のとおりです。
  - 対話式スループットおよび応答時間。これは WRKACTJOB 画面に表示されます。
  - バッチ・スループット。活動状態のバッチ・ジョブの補助入出力 (AuxIO) 値と CPU 使用率 (CPU%) 値を監視します。
  - スプール・スループット。活動状態の書き出しプログラムの補助入出力 (AuxIO) 値と CPU 使用率 (CPU%) 値を監視します。
5. 期待に沿わないパフォーマンス・データがある場合は、システムを新しいデータに基づいて調整する。以下の点に気を付けてください。
  - すべての重要なパフォーマンス測定値を測定および比較する。
  - 調整の実施と評価を同時に行う。

### パフォーマンスを検討する

適切な調整値を設定したなら、システムが順調に機能しつづけるようにするために、定期的に調整値を検討する必要があります。継続調整には、システム・パフォーマンスの各面の監視と、推奨されるガイドラインへの調整が含まれます。

意味のある統計を収集するには、標準的な活動レベルの時にシステム・パフォーマンスを監視する必要があります。たとえば、システムでジョブが実行されていない間に収集される統計は、システム・パフォーマンスを評価する点ではほとんど価値がありません。最大限の努力を払ってもパフォーマンスが納得のいくものではない場合は、現在の構成の能力を評価する必要があります。目標を達成するために、以下を考慮してください。

- プロセッサのアップグレード
- 記憶装置およびコントローラーの追加
- 主記憶装置の追加
- アプリケーションの修正

上記の 1 つ以上の方法を適用することによって、目標を達成できます。適切な方法を適用した後に、まだ目標を達成できない場合は、実行する作業の種類にとって現実的な目標を設定しているかどうかを判断する必要があります。

### 調整対象を判別する

システム・パフォーマンスが低下して調整が必要な場合は、パフォーマンス上の問題の原因を突き止めて具体的な訂正を行うために、『18 ページの『パフォーマンスの問題の調査』』を参照してください。

### パフォーマンスの自動調整

システム資源を効果的に使用するために、システムはパフォーマンス値を自動的に設定できます。以下を行うことによって、システム・パフォーマンスを自動的に調整するようにシステムをセットアップできます。

- 記憶域プール・サイズと活動レベルの調整
- 記憶域プール・ページングの調整

### 記憶域プール・サイズと活動レベルの調整

QPFRADJ システム値を使用して、記憶域プールと活動レベルの自動調整を制御します。この値は、システムがシステム再始動 (IPL) 時に値を調整するのか、再始動後に定期的に調整するのかを指示します。

パフォーマンスを IPL 時に調整する、動的に調整する、または IPL 時にも動的にも調整するようにシステムをセットアップできます。

- システム再始動 (IPL) 時にのみ調整を行うようシステムをセットアップするには、iSeries<sup>(TM)</sup> ナビゲーターで「構成およびサービス」->「システム値」->「パフォーマンス」を選択する。「メモリー・プール」タブをクリックして、「メモリー・プールと活動レベルを自動的に調整する (Automatically adjust memory pools and activity levels)」の下で「システム再始動時 (At system restart)」を選択します。これは、QPFRADJ システム値を 1 に設定することと同じです。
- 記憶域プールの調整をシステム再始動 (IPL) 時に行い、再始動後に記憶域プールの調整を定期的に行うようシステムをセットアップするには、iSeries ナビゲーターで「構成およびサービス」->「システム値」->「パフォーマンス」を選択する。「メモリー・プール」タブをクリックして、「メモリー・プールと活動レベルを自動的に調整する (Automatically adjust memory pools and activity levels)」の下で「システム再始動時 (At system restart)」と「再始動後定期的に (Periodically after restart)」を選択します。これは、QPFRADJ システム値を 2 に設定することと同じです。
- 記憶域プールの調整をシステム再始動 (IPL) 時に行わず、再始動後に定期的に行うようシステムをセットアップするには、iSeries ナビゲーターで「構成およびサービス」->「システム値」->「パフォーマンス」を選択する。「メモリー・プール」タブをクリックして、「メモリー・プールと活動レベルを自動的に調整する (Automatically adjust memory pools and activity levels)」の下で「再始動後定期的に (Periodically after restart)」を選択します。これは、QPFRADJ システム値を 3 に設定することと同じです。

記憶域プール値は、システム再始動 (IPL) 時に初期値にリセットされることはありません。

#### 記憶域プール・ページングの調整

システムによって提供される動的調整は、システムのパフォーマンスを改善するために、共用プールのプール・サイズと活動レベルを自動的に調整します。この調整は、使用率が最も低い記憶域プールから、より多くの記憶域があることで利点があるプールに記憶域を移動することによって行われます。この調整では、プール内のスレッドの数のバランスを取るために、プールに割り振られた記憶域を使用して、活動レベルの設定も行われます。システムを調整するために、調整プログラムは、スレッドの数に基づいて計算されるガイドラインを使用します。

動的調整が有効になっている場合は、以下のパフォーマンス値が適切な設定値に自動的に変更されます。

- マシン (\*MACHINE) メモリー・プール・サイズ (QMCHPOOL システム値)
- ベース (\*BASE) メモリー・プール・レベル (QBASACTLVL システム値)
- 共用プール \*INTERACT のプール・サイズと活動レベル
- 共用プール \*SPOOL のプール・サイズと活動レベル
- 共用プール \*SHRPOOL1-\*SHRPOOL60 のプール・サイズと活動レベル

動的調整が有効になっている (QPFRADJ システム値が 2 または 3 に設定されている) 場合は、プロファイル QSYS の下で実行されるジョブ QPFRADJ はシステム上で活動状態として表示されます。

メモリー・プールの詳細については、『メモリー・プール』を参照してください。

## e-business パフォーマンスの管理

e-business 環境におけるパフォーマンスには、iSeries<sup>(TM)</sup> システム管理者にとって複雑な問題がいくつか含まれます。iSeries サーバーの日常的な調整に加え、管理者は e-business トランザクションをサポートしているハードウェアとサービスもモニターして最適化する必要があります。

以下のトピックは、サーバーの e-business パフォーマンスを最大化するためのいくつかの重要な考慮事項を理解するのに役立つもので、詳細な推奨事項と例がある追加情報資源へのリンクが含まれています。

### 『クライアント・パフォーマンス』

多くの場合、iSeries システム管理者が e-business ネットワークのクライアント・サイドの制御を行うことはほとんどありませんが、ここでの推奨事項を基に、クライアント装置が e-business 環境に合わせて最適化されていることを確認することができます。

### 30 ページの『ネットワーク・パフォーマンス』

多くの場合、ネットワーク設計、ハードウェア資源、およびトラフィック・プレッシャーは、e-business アプリケーションのパフォーマンスに大きな影響を与えます。ネットワーク・パフォーマンスの最適化の方法と iSeries 通信資源の調整の方法については、このトピックを参照してください。

### 30 ページの『OS/400<sup>(R)</sup> における Java<sup>(TM)</sup> パフォーマンス』

OS/400 には、iSeries サーバー上の Java のアプリケーションやサービスのパフォーマンスを最適化するための構成オプションと資源がいくつかあります。OS/400 における Java 環境について、および Java ベースのアプリケーションで可能な限り最高のパフォーマンスを得る方法については、このトピックを参照してください。

### 31 ページの『IBM HTTP Server パフォーマンス』

IBM<sup>(R)</sup> HTTP サーバーは多くの場合、iSeries サーバーでの e-business パフォーマンスの重要な部分です。IBM は、このサーバーを最大限に生かすことができるオプションと構成の選択項目をいくつか用意しています。

### 32 ページの『WebSphere パフォーマンス』

WebSphere Application Server は、iSeries サーバーにとって最適な e-business アプリケーション配置環境です。WebSphere 環境でのパフォーマンスの計画と最適化の方法については、このトピックを参照してください。

管理者は、これらの特定の推奨トピックのほかに、次のトピックも理解する必要があります。

- 実行管理機能
- Java for iSeries
- HTTP サーバー
- Domino<sup>(R)</sup> for iSeries sizing and performance tuning 

## クライアント・パフォーマンス

Web ブラウザーを搭載している PC から成るクライアントは多くの場合、管理者が直接制御を行うことが最も少ない e-business 構成要素になります。しかし、この構成要素は Web アプリケーションのエンドユーザーの応答時間にも大きく影響します。

高パフォーマンスが得られるようにするには、クライアント PC を次のようにする必要があります。

- 十分なメモリー容量にします。資源集中アプレット、および複雑なフォームとグラフィックスを使用するインターフェースによって、クライアントのプロセッサに対しても要求が行われる場合があります。
- 高速で最適化されたネットワーク接続を使用します。クライアント PC 上の通信アダプターの多くが、それらのネットワーク環境に合わせて最適化されないまま機能していることがあります。詳しくは、ご使用の通信ハードウェアの資料を参照してください。
- 必要なテクノロジーを完全にサポートしているブラウザーを使用します。また、ブラウザーのサポートとパフォーマンスは、Web インターフェースを設計するときの主要な関心事にすべきことです。

## ネットワーク・パフォーマンス

多くの場合、ネットワークは Web アプリケーションの応答時間において主要な役割を果たします。その上、ネットワーク構成要素に対するパフォーマンスの影響はたいていは複雑で測定が困難です。ネットワーク・トラフィックと使用可能帯域幅はたびたび変化することがあり、システム管理者が直接制御できない作用の影響を受けるためです。しかし、ご使用の iSeries<sup>(TM)</sup> サーバーの通信資源のモニターと調整に使用できる資源がいくつかあります。

詳しくは、以下のトピックを参照してください。

### 36 ページの『収集サービス』

収集サービスは通信資源のパフォーマンス・データを定期的に収集します。特に関心がある TCP サーバーに関する情報は、パフォーマンス・データ・ファイルの QAPMTCP と QAPMTCPIFC に保管されます。このデータはそれらのファイルを直接照会するか、あるいは Performance Tools ライセンス・プログラムに含まれる報告書を使用して参照することができます。

### 95 ページの『iSeries<sup>(TM)</sup> ナビゲーターのモニター』

iSeries サーバーの通信ハードウェアを含むシステム資源がどのように使用されているかという情報を、システム・モニターを使用して得ることができます。システム・モニターでの回線使用率および IOP メトリックは、ネットワーク・パフォーマンスに関する特に有用なデータになります。

### 18 ページの『パフォーマンスの追跡』

いくつかのアプリケーションとツールを使用すると、サーバーの通信資源のパフォーマンス・データを定期的に収集して、そのパフォーマンスを長期間に渡ってモニターすることができます。

iSeries Performance Capabilities Reference



Performance Capabilities Reference には、ご使用のサーバーを最適なパフォーマンスに構成したり調整するとき役立つ詳細な情報、報告書、および例が記載されています。特に通信資源の計画と管理について、第 5 章『Communications Performance』を参照してください。

iSeries Network.com



この Web サイトには、ネットワーク計画と資源を最適化するための多数の情報資源があります。特に、『Cultivate your AS/400<sup>(R)</sup> Networks』と『8 tools for better network performance』を参照してください。

## OS/400<sup>(R)</sup> における Java<sup>(TM)</sup> パフォーマンス

Java は多くの場合、Web ベース・アプリケーションに最適な言語です。しかし Java アプリケーションで最適なパフォーマンスを得るには、OS/400 実行環境と Java アプリケーションの両方で幾らかの最適化が必要な場合があります。

OS/400 における Java 環境と、Java パフォーマンスの分析と改善のための有効なヒントとツールについては、以下の情報資源を利用してください。

### Java パフォーマンス

Java ベースのアプリケーションで最高のパフォーマンスを得るのに役立つ、いくつかの重要な構成の選択とツールがあります。

## 22 ページの『アプリケーションのパフォーマンスについての情報の収集』

OS/400 におけるアプリケーションのパフォーマンスのモニターと調整に使用できるツールがいくつかあります。アプリケーション・パフォーマンスの測定と改善に役立つパフォーマンスのトレース、Performance Explorer (PEX)、および類似ツールの使用方法については、このトピックをご覧ください。

### iSeries<sup>(TM)</sup> Performance Capabilities Reference



Performance Capabilities Reference には、ご使用のサーバーを最適なパフォーマンスに構成したり調整するときに役立つ詳細な情報、報告書、および例が記載されています。特に Java アプリケーションのパフォーマンスの最適化と Java プログラミングでのパフォーマンスのヒントについて、第 7 章『Java Performance』を参照してください。

### Java and WebSphere<sup>(R)</sup> performance in OS/400



Java および WebSphere のパフォーマンスを最高にするための、およびパフォーマンス・データの収集と分析を行うための稼働環境の計画および構成の方法については、この Redbook をお読みください。

### WebSphere J2EE application development for the IBM<sup>(R)</sup> eServer<sup>(TM)</sup> iSeries server



この Redbook には、J2EE の概要と、サーバーに J2EE アプリケーションを正しくインプリメントするのに役立つ提案と例が記載されています。

パフォーマンス情報のほかに、Java トピックには、サーバーに Java アプリケーションを開発して配置する際の情報資源が記載されています。

## IBM HTTP Server パフォーマンス

IBM<sup>(R)</sup> HTTP Server for iSeries<sup>(TM)</sup> は Web ベース・アプリケーションのエンドツーエンドのパフォーマンスで重要な役割を果たします。また新規改良点によって Web サーバーのパフォーマンスの効果的なモニターと改善が可能になりました。新規の Fast Response Caching Accelerator (FRCA) を使用することによって、特に大部分が静的な環境では、HTTP サーバーのパフォーマンスを大幅に向上させることができます。

HTTP サーバーのパフォーマンスを最大限に高める方法については、以下の情報資源を参照してください。

### 36 ページの『収集サービス』

収集サービスを使用して HTTP サーバーのパフォーマンス・データを収集し、その結果を長期間に渡ってモニターすることができます。HTTP サーバーのデータは、収集間隔ごとにパフォーマンス・データ・ファイル QAPMHTTPB および QAPMHTTPD に保管されます。QAPMHTTPB には基本情報があり、QAPMHTTPD にはより詳細な統計情報があります。これらのデータ・ファイルは直接照会することができますが、Performance Tools ライセンス・プログラムのシステム報告書と構成要素報告書を参照することもできます。

## IBM HTTP Server for iSeries

iSeries の HTTP サーバーのセットアップ、構成、および管理については、このトピックを参照してください。このトピックには、Fast Response Caching Accelerator (FRCA) のような本製品への最新の機能強化の説明も含まれています。

iSeries Performance Capabilities Reference



Performance Capabilities Reference には、ご使用の iSeries サーバーを最適なパフォーマンスに構成したり調整するとき役に立つ詳細な情報、報告書、および例が記載されています。特に、HTTP サーバーのパフォーマンスの仕様、計画に関する情報、およびパフォーマンスのヒントについて、第 6 章『Web Server and Web Commerce』を参照してください。

HTTP server (Powered by Apache)



OS/400<sup>(R)</sup> での HTTP Server (Powered by Apache) の詳細な解説は、この Redbook をお読みください。一般的な使用シナリオで HTTP Server を構成する例が含まれています。

## AS/400<sup>(R)</sup> HTTP Server Performance and Capacity Planning



パフォーマンスの調整と計画における HTTP サーバーの影響については、この Redbook を参照してください。この資料には、iSeries パフォーマンス管理ツールを使用して Web サーバーのパフォーマンス・データを収集、解釈、およびそれに対応する場合の提案も記載されています。

## WebSphere パフォーマンス

WebSphere<sup>(R)</sup> 環境の iSeries<sup>(TM)</sup> サーバーのパフォーマンスの管理には、iSeries 管理者にとってのいくつかの課題が伴います。Web ベースのトランザクションによって消費される資源は多くなることがあり、しかもそこで消費される資源はいつもの通信ワークロードとは異なる場合があります。

WebSphere 環境での最適パフォーマンスの計画方法およびサーバー資源の調整方法については、以下のトピックと情報資源を参照してください。

### WebSphere Application Server performance considerations



この Web サイトには、iSeries サーバー上の WebSphere Application Server の各バージョンの、パフォーマンスに関する多数の有用なヒントと推奨事項を含む情報資源があります。この情報資源は、サブレット、Java<sup>(TM)</sup> Server Pages (JSP)、および Enterprise Java Beans (EJB) を使用する環境では特に有用です。

### DB2<sup>(R)</sup> UDB/WebSphere Performance Tuning Guide



この Redbook には WebSphere 環境と DB2 環境のそれぞれの概要が記載されており、パフォーマンス上の共通の問題に対する提案、例、およびソリューションが示されています。これらを基に WebSphere と DB2 のパフォーマンスを最適化することができます。

## Java and WebSphere performance in OS/400<sup>(R)</sup>



Java および WebSphere のパフォーマンスを最高にするための、およびパフォーマンス・データの収集と分析を行うための稼働環境の計画および構成の方法については、この Redbook をお読みください。

### WebSphere V3 Performance Tuning Guide



この Redbook には、iSeries サーバー上の WebSphere V3 のパフォーマンスを最適化するための推奨事項と例が詳細に示されています。

### iSeries Performance Capabilities Reference



Performance Capabilities Reference には、ご使用のサーバーを最適なパフォーマンスに構成したり調整するときに役立つ詳細な情報、報告書、および例が記載されています。特に WebSphere 特有のパフォーマンス上のヒントについて、第 6 章『Web Server and Web Commerce』を参照してください。

WebSphere および e-business に関するその他の情報資源については、トピック『WebSphere e-business の管理』を参照してください。

---

## パフォーマンス管理用のアプリケーション

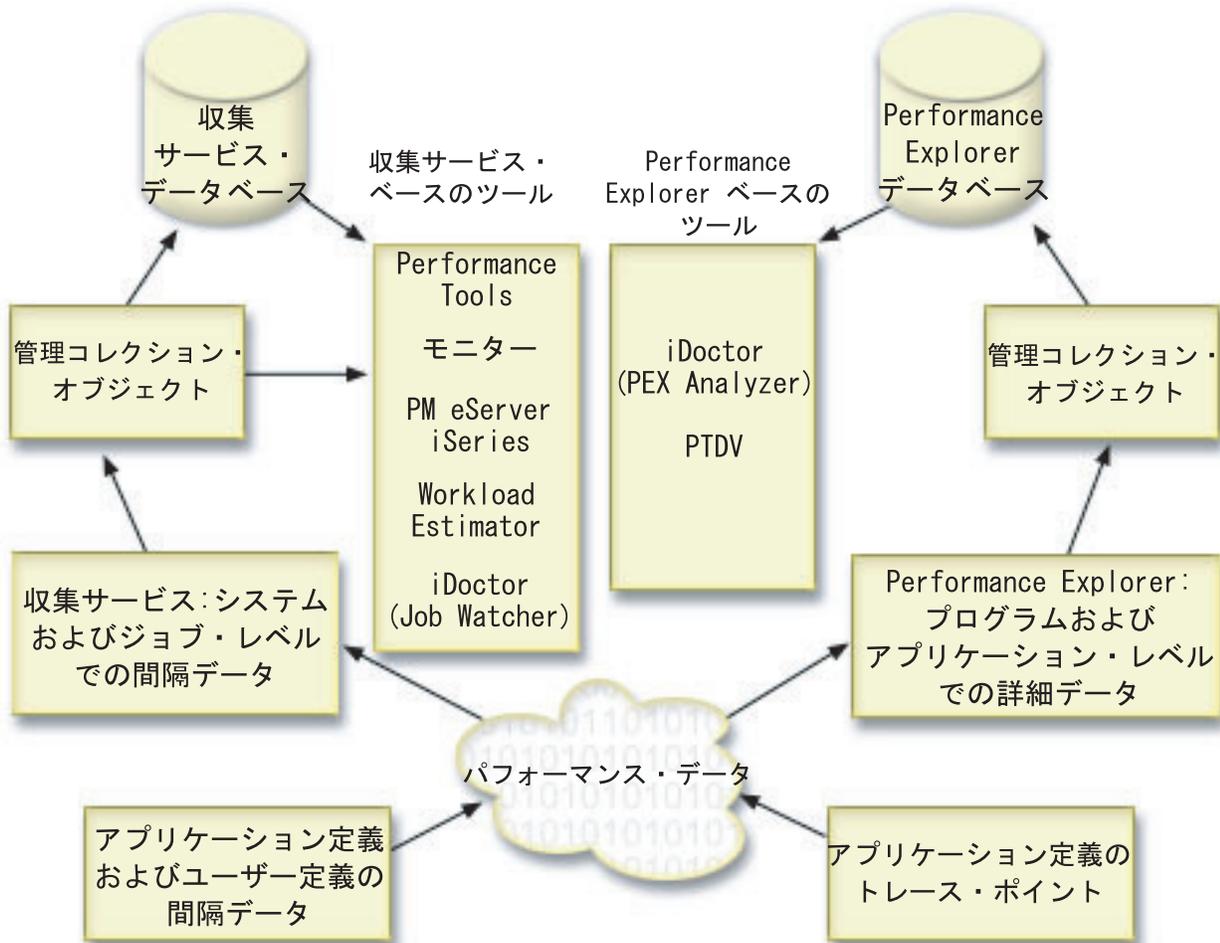
パフォーマンス管理用の多くのアプリケーションには、複数の機能があります。使用可能な一組のアプリケーションのどの構成要素が、特定の状況に最適であるかを正確に把握するのは困難です。以下のトピックでは、各パフォーマンス管理アプリケーションの選択、使用、および構成を含めた詳細が説明されています。

以下の図に示されているとおり、iSeries<sup>(TM)</sup> サーバーには、基本的に 2 つのパフォーマンス収集機能があります。

- 収集サービス。システム・レベルおよびジョブ・レベルで間隔データを収集します。システムの状態を把握するために、このサービスを継続的に実行できます。収集される間隔データは、アプリケーション定義データまたはユーザー定義データです。
- Performance Explorer。プログラム・レベルおよびアプリケーション・レベルで詳細データを収集します。アプリケーションでの作業の流れもトレースするので、パフォーマンス上の難しい問題を診断するために使用することもできます。データは、ドミノ (Domino<sup>(R)</sup>)、NetServer、または WebSphere<sup>(R)</sup> などの、アプリケーションで定義された Performance Explorer トレース・ポイントに基づいて収集されます。

これらの収集機能は両方とも、データを管理収集オブジェクトに保管します。収集サービス・データの場合はパフォーマンス・データの作成 (CRTPFDDTA) コマンド、Performance Explorer データの場合は Performance Explorer データの作成 (CRTPEXDDTA) コマンドを使用して、管理収集オブジェクトのデータを変換できます。

このトピックでは、収集サービス・データまたは Performance Explorer データに使用できるパフォーマンス管理アプリケーションを紹介します。



### 36 ページの『収集サービス』

収集サービスは、ユーザー定義の時間間隔でパフォーマンス・データを収集し、この情報をシステム上の収集オブジェクトに保管します。モニター、グラフ・ヒストリー、PM iSeries などの他の多くのツールや、Performance Tools ライセンス・プログラムの多くの機能は、これらの収集オブジェクトからデータを入手します。

### 70 ページの『インテリジェント・エージェント』

iSeries ナビゲーターのインテリジェント・エージェント・コンソールにより、システム管理者は、単一のシステム上、または複数のシステムをまたいで実行される 1 つ以上の ABLE (Agent Building and Learning Environment) エージェントを簡単に管理できます。

### 90 ページの『パフォーマンス・データ・ファイル』

収集サービスが管理する収集オブジェクトからデータベース・ファイルを生成できます。これらのデータベース・ファイルの名前、説明、および属性については、このトピックを参照してください。

### 95 ページの『iSeries<sup>TM</sup> ナビゲーターのモニター』

モニターは、システムのパフォーマンスについての現行情報を表示します。特定のイベントの発生時に事前定義アクションを実行するために、モニターを使用することもできます。システム、メッセージ、ジョブ、ファイル、および B2B トランザクション・モニターを使用して、システムについての情報を表示したりモニターすることができます。システム・モニターとジョブ・モニターは、収集サービスによって収集されたパフォーマンス・データを使用します。

### 103 ページの『グラフ・ヒストリー』

グラフ・ヒストリーは、指定した期間に渡って収集サービスによって収集されたパフォーマンス・データをグラフィカルに表示します。

### 105 ページの『IBM Performance Management for eServer iSeries』

PM iSeries はシステム・パフォーマンス・データの収集、保存、および分析を自動化し、システム資源およびキャパシティーを管理するのに役立つ報告書を戻します。PM iSeries は、収集サービスが収集するパフォーマンス・データを使用します。

### 123 ページの『Performance Tools』

Performance Tools ライセンス・プログラムには、システム・パフォーマンス情報を収集、分析、および保守するのに役立つ多くの機能があります。これには、分散ネットワーク上でのパフォーマンスの管理、要約データおよびトレース・データの収集と報告、およびキャパシティーの計画に役立つ機能が含まれています。Performance Tools は、収集サービスによって収集されるパフォーマンス・データ (サンプル・データ) と、パフォーマンス・トレースの開始 (STRPFTRTC) コマンドおよびパフォーマンス・トレースの終了 (ENDPFTRTC) コマンドから取得されるトレース・データを使用します。

### 134 ページの『Performance Explorer』

Performance Explorer は、特定のアプリケーション、プログラムまたはシステム資源に関するより詳細な情報を収集し、特定のパフォーマンス上の問題を詳しく洞察します。これには、複数のタイプおよびレベルのトレースを実行する機能と、明細報告書を実行する機能が含まれます。

### 142 ページの『iDoctor for iSeries』

iDoctor for iSeries プラグインは、パフォーマンスを管理するためのさまざまなソフトウェア・ツールで構成されます。たとえば、PEX Analyzer (トレース・データの詳細分析用)、Job Watcher (ジョブの動作についてのトレース・レベルの情報用) などです。

### 143 ページの『Performance Trace Data Visualizer (PTDV)』

Performance Trace Data Visualizer for iSeries (PTDV) は、iSeries 上で実行されるアプリケーションのパフォーマンス分析に使用できる Java<sup>(TM)</sup> アプリケーションです。

### 143 ページの『Performance Management API』

Performance Management API は、収集を管理するためのサービスを提供します。これらの API は、収集を開始、終了、および循環させ、収集されたデータのシステム・パラメーターを変更します。多くの Performance Management API は、収集サービスによって収集されたパフォーマンス・データを使用します。

### 143 ページの『OS/400 パフォーマンス用の処理コマンド』

OS/400 には、システム・パフォーマンスの管理および保守に役立つ複数の重要な機能があります。

### 144 ページの『拡張キャッシュ』

拡張キャッシュは、ディスク使用量データを収集し、それらの統計を使用して大規模なキャッシュを作成し、ディスクの物理入出力要求を効果的に削減することによって、システム・パフォーマンスを改善できます。

### 146 ページの『Workload Estimator for iSeries』

Workload Estimator は、次のアップグレードのサイズおよびタイミング要件を計画するのに役立ちます。このツールは、システム・パフォーマンスの傾向を分析するために PM iSeries と一緒に使用されることがよくあり、iSeries サーバーの成長と拡張を効果的に管理するのに役立ちます。

### 147 ページの『iSeries<sup>TM</sup> ナビゲーター (ワイヤレス対応)』

iSeries ナビゲーター (ワイヤレス対応) を使用すると、携帯情報端末 (PDA)、インターネット電話、または旧来の Web ブラウザーを使用した無線接続を介して、パフォーマンス・データをモニターすることができます。iSeries ナビゲーター (ワイヤレス対応) は、収集サービスによって収集されたパフォーマンス・データを使用します。

### 147 ページの『PATROL for iSeries (AS/400) - Predict』

PATROL for iSeries (AS/400) - Predict は、高可用性および最適なパフォーマンスに必要な多くの定期管理作業を自動化することにより、iSeries パフォーマンスの管理に役立ちます。さらに、この製品は、iSeries 環境の成長を計画するのに役立つ、詳細なキャパシティー・プランニング情報を提供します。

## 収集サービス

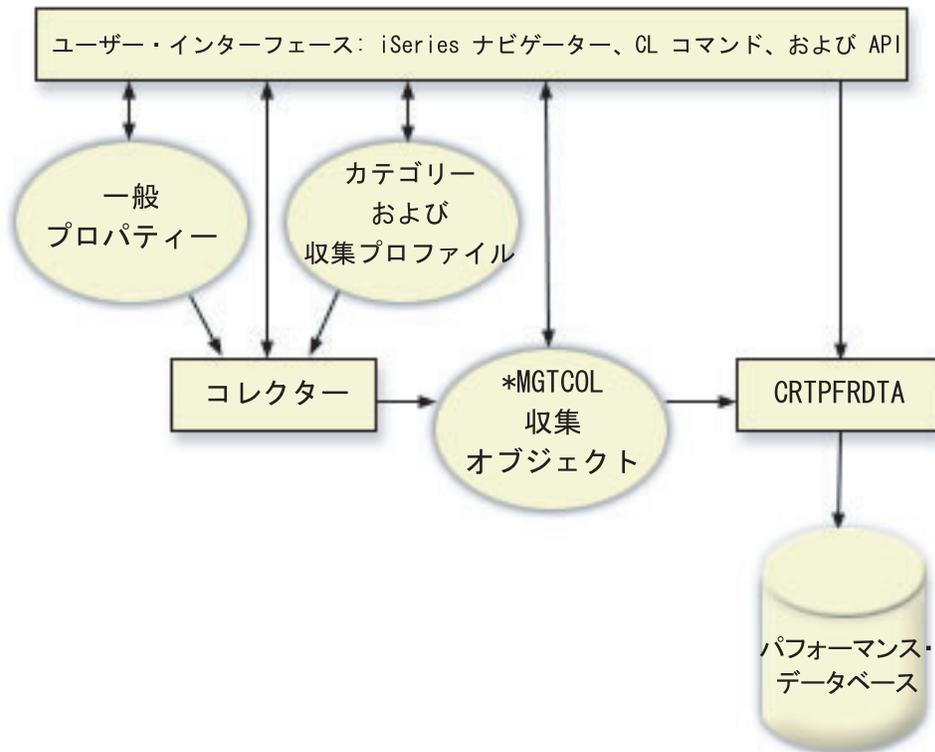
収集サービスを使用して、Performance Tools for iSeries<sup>TM</sup> ライセンス・プログラムまたはその他のパフォーマンス報告書アプリケーション、iSeries ナビゲーター・モニター、およびグラフ・ヒストリー機能による今後の分析のために、パフォーマンス・データを収集します。(リアルタイムにパフォーマンス・データを表示させたい場合、システム・モニターには、システム・パフォーマンスをモニターするための使いやすいグラフィカル・インターフェースもあります。) 収集サービスは、システムのさまざまな分野ごとに使用されるシステム・リソースの相対的な量を示すデータを収集します。収集サービスを使用して、以下のことを行うことができます。

- 収集オブジェクトを容易に管理する。
- 最小のシステム・オーバーヘッドを使用してパフォーマンス・データを継続および自動的に収集する。
- どのようなデータを収集するか、またデータをどのように使用するかを制御する。
- データを変換することなく、リリース間でパフォーマンス・データを移動する。
- Performance Tools によって使用されるパフォーマンス・データ・ファイルを作成する。
- ユーザー定義パフォーマンス・データを収集するためのプログラムを収集サービスに統合する。

### 収集サービスの動作方法

収集サービスは、OS/400<sup>®</sup> パフォーマンス・モニターの代わりとなります。これは、パフォーマンス・モニターの開始 (STRPFRMON) コマンドによって呼び出されました。パフォーマンス・モニター (STRPFRMON コマンド) は、V4R5 以降は使用可能ではありません。OS/400 パフォーマンス・モニターを使用したとき、収集されたデータは 30 個ものデータベース・ファイルになります。

収集サービス機能では、パフォーマンス・データを収集するための新しいプロセスを紹介します。収集サービスであれば、データは収集ごとに単一の収集オブジェクトに入れられ、そこからデータベース・ファイルのさまざまな集合を必要に応じて作成することができます。つまり、パフォーマンス・データを収集する間も、システムのオーバーヘッドを低く押さえることができます。また、収集の実行時にデータベース・ファイルを作成する場合でも、収集サービスは、低い優先順位 (50) のバッチ・ジョブを使用してこれらのファイルを更新するため、OS/400 パフォーマンス・モニターよりも高いパフォーマンスを維持することができます。このようにして、収集によるオーバーヘッドを減らすことにより、パフォーマンス・データをより詳細に、また、より短い間隔で継続的に収集することが可能になります。収集サービスを使用すれば、パフォーマンス・データの収集と保存に関してネットワーク規模のシステム・ポリシーを設け、そのポリシーを自動的に実装することができます。これらの管理収集オブジェクトが保管されている限り、必要が生じた場合でも、収集したデータから過去のパフォーマンスに関連したイベントを振り返り、詳細なレベルまでそれを分析することができます。



収集サービスを使用すると、システム・パフォーマンスにほとんど影響を与えることなく、またははっきり分かるほどの影響を与えることなく、パフォーマンス・データを収集できます。iSeries ナビゲーターを使用して、任意の頻度でデータを収集するよう収集サービスを構成することができます。収集オブジェクト \*MGTCOL は、大量のパフォーマンス・データを保持するための効果的なストレージ・メディアとしての機能を果たします。収集サービスを構成して開始すると、パフォーマンス・データは継続的に収集されます。パフォーマンス・データを処理する必要がある場合は、必要なデータを一連のパフォーマンス・データベース・ファイルにコピーできます。

上の図は、以下の収集サービス要素の概要を示しています。

#### ユーザー・インターフェース

収集サービスの異なる要素にアクセスできる複数の方法が用意されています。たとえば、CL コマンド、API、および iSeries ナビゲーター・インターフェースを使用できます。

#### 一般プロパティ

一般プロパティは、収集がどのように実行されるかを定義し、自動収集属性を制御します。

#### データ・カテゴリ

データ・カテゴリは、収集するデータのタイプを識別します。各カテゴリを別々に構成し、収集するデータと、データ収集の頻度を制御することができます。

#### 収集プロファイル

収集プロファイルは、特定のカテゴリ構成を保管および活動化するための手段を提供します。

## パフォーマンス・コレクター

パフォーマンス・コレクターは、一般プロパティとカテゴリ情報を使用して、パフォーマンス・データの収集を制御します。パフォーマンス・コレクターは手動で開始および停止できますし、自動的に実行されるよう構成することもできます。

## 収集オブジェクト

収集オブジェクト \*MGTCOL は、大量のパフォーマンス・データを保持するための効果的なストレージ・メディアとしての機能を果たします。

## パフォーマンス・データの作成 (CRTPFRDTA) コマンド

CRTPFRDTA コマンドは、管理収集オブジェクトに保管されているデータを処理し、パフォーマンス・データベース・ファイルを生成します。

## パフォーマンス・データベース

データベース・ファイルには、CRTPFRDTA コマンドによって処理されるデータが保管されます。このファイルは、時間間隔データが入っているパフォーマンス・データ・ファイル、構成データ・ファイル、トレース・データ・ファイルというカテゴリに分けることができます。

➤ 収集サービスおよびジョブ・モニターがシステムで連動する方法の図は、39 ページの『システムおよびジョブ・モニターと収集サービスの相互作用』を参照してください。◀

## 収集サービスの開始方法

以下の方法のいずれかを使用して、収集サービスを開始することができます。ただし、パフォーマンスのトピック内にある情報は、iSeries ナビゲーターの方法に焦点を当てています。

開始方法	説明
パフォーマンス収集開始 (STRPFCOL)	STRPFCOL コマンドは、収集サービスによってパフォーマンス・データのシステム・レベル収集を開始するのに使用します。
iSeries ナビゲーター	iSeries ナビゲーターを使用して、さまざまな収集サービス・タスクを実行します。以下の表に、これらのタスクと、それを実行するための情報へのリンクを示します。
Performance Management API	Performance Management API を使用して、収集の開始、カスタマイズ、終了、循環を行います。さらに、API を使用して、管理収集オブジェクトを処理したり独自のトランザクションを定義することができます。
従来のメニュー・オプション	文字ベースのインターフェースで <b>GO PERFORM</b> と入力して、Performance Tools のメインメニューでオプション 2 (パフォーマンス・データの収集) を選択します。追加情報については、Performance Tools for iSeries  に進んでください。
108 ページの『PM iSeries の活動化』	PM iSeries は、自動的に収集サービスを開始し、収集の際にデータベース・ファイルを作成します。

## 収集サービスのタスク

収集サービスおよび iSeries ナビゲーターを使用して、下の表に示されているように、さまざまなデータ収集タスクを実行することができます。

タスク	説明
さまざまな方法での収集サービスの開始	特定のパフォーマンス・メトリックを使用する個々のシステムまたはシステム・グループ上に、カスタマイズされたパフォーマンス・データ・コレクションを作成します。パフォーマンス・データ・コレクションを自動的に開始するために、始動プログラム内の Start Collector API を使用することもできます。これらのタスクを実行する方法の詳細については、オンライン・ヘルプを参照してください。作業の詳細ヘルプは、「iSeries ナビゲーター」ウィンドウから使用できます。メニュー・バーから「ヘルプ」をクリックして、「ヘルプ・トピック」を選択します。「...によって実行できる処理」を選択して、実行できる処理と、「iSeries ナビゲーター」ウィンドウのどこから実行するのかを調べます。
42 ページの『収集サービスのデータからのデータベース・ファイルの作成』	データベース・ファイルを作成するときに、どのデータを収集するかを制御する方法を学習します。たとえば、収集サービスを使ってパフォーマンス・データベース・ファイルの作成を自動化したり、収集された後のデータが保管されている収集オブジェクトからデータベース・ファイルを作成できます。さらに、PM iSeries、Performance Tools ライセンス・プログラムとともにこれらのデータベース・ファイルを使用するか、またはこれらのファイルに対して実行するための独自の照会を作成することができます。  『90 ページの『パフォーマンス・データ・ファイル』』を参照し、それぞれのファイルに含まれているフィールド・レベルのデータだけでなく、どのデータベース・ファイルが使用可能であるかについても調べます。
44 ページの『データ収集のカスタマイズ』	データ収集をカスタマイズします。収集するパフォーマンス・データおよびデータが収集される頻度の制御に関する情報を調べます。重要な 45 ページの『収集サービスに関連した時間帯の考慮事項』についての情報も検索できます。
46 ページの『収集サービスのユーザー定義カテゴリー』	出口プログラムを作成し、それを収集サービスに統合することによって、ユーザー・アプリケーションからパフォーマンス・データを収集できます。定期収集間隔中にこのデータを収集し、収集オブジェクトに保管することができます。
55 ページの『収集オブジェクトの管理』	収集オブジェクトの内容、収集オブジェクトの保管期間、および収集オブジェクトを使用して実行できる事柄を含め、収集オブジェクトの管理の必要がある情報を検索します。
22 ページの『アプリケーションのパフォーマンスについての情報の収集』	収集サービスでは、サンプル・データを収集します。しかし、トレース・データは収集しません。トレース・データの収集方法を学習するには、このトピックを参照してください。
56 ページの『ユーザー定義トランザクション』	収集サービスには、独自のトランザクションを定義できる API が用意されています。
64 ページの『区画を介したパフォーマンス・データの収集』	収集サービスは、区画で実行するオペレーティング・システムに関係なく、iSeries 区画からパフォーマンス・データを収集できます。その後、PM iSeries は、分析のために IBM <sup>®</sup> に送信する前にデータを集約します。
68 ページの『ジョブ、タスク、またはスレッドの待機統計の検索』	ジョブ、タスク、またはスレッドに発生した待機時間の原因と所要時間を識別します。
68 ページの『収集サービスによるディスク使用量の理解』	収集サービスが消費するディスク・スペースの量は、選択された収集間隔と保存期間によって大幅に異なります。収集サービスのディスク使用量の計画に役立つ、このトピックを参照してください。 <<

## システムおよびジョブ・モニターと収集サービスの相互作用

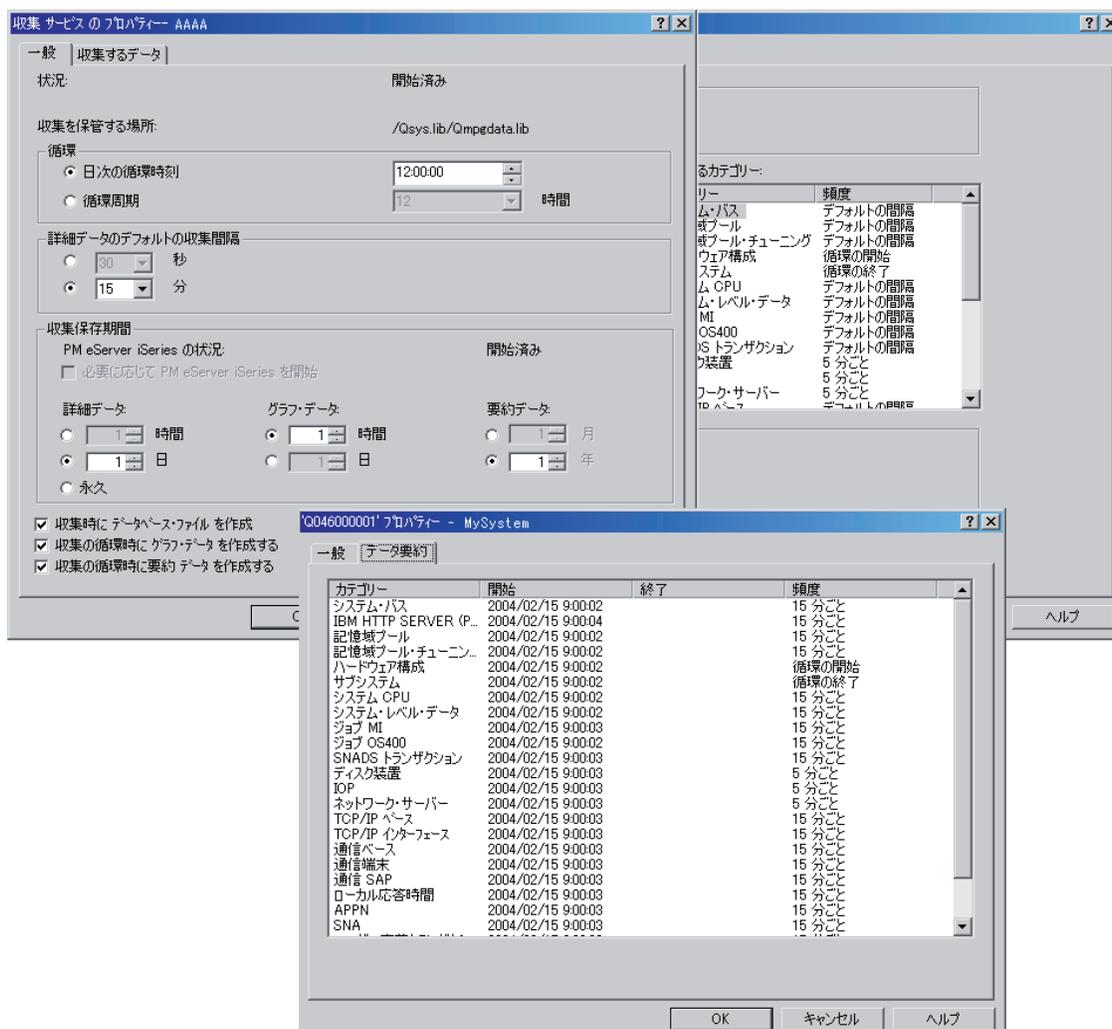
➤ 収集サービスは、スタンドアロン・アプリケーションとして、またパフォーマンス・データを収集するために他のアプリケーションが使用するユーティリティとして、パフォーマンス分析用の貴重なツールです。時々、システムで生じるアクティビティを担当するアプリケーションを判別しようとして、パフォー

マンス分析が混乱を起こすことがあります。この問題に関して覚えておくべき 1 つの簡単な規則は、それらの他のアプリケーションが使用中のように見えても、任意の指定時間にシステムで発生するデータ収集は 1 つだけであるということです。

次のシナリオは、システム・モニターとジョブ・モニターおよび収集サービス間の異なる組み合わせ、および収集サービスが表示するものを説明します。

### 収集サービスがデフォルト値を使用してデータを収集する場合

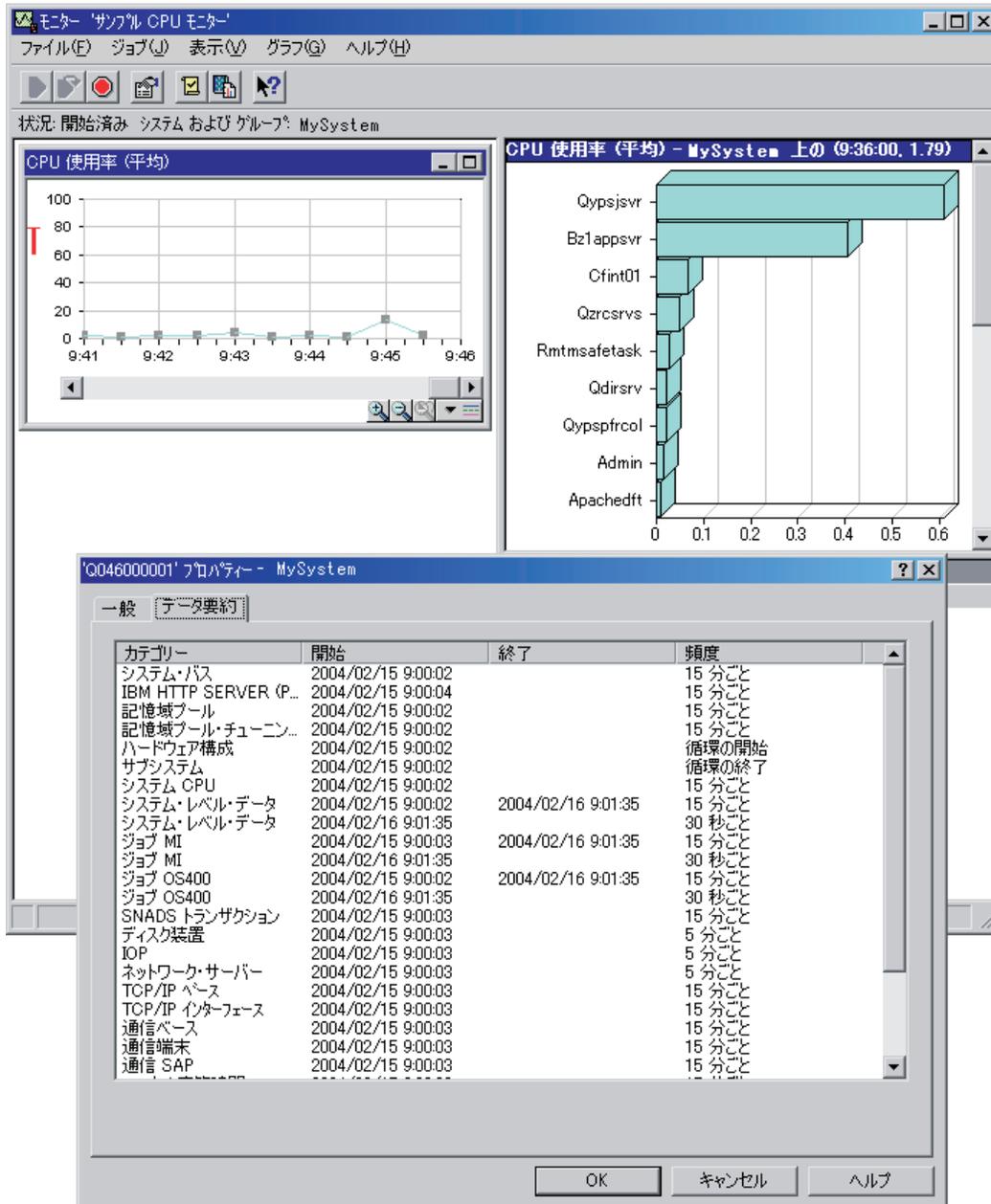
このシナリオでは、システム上で活動状態のシステム・モニターまたはジョブ・モニターはありません。収集サービスのプロパティ・ページおよび \*MGTCOL オブジェクトのプロパティ・ビューを表示すると、以下のようなものが表示されます。



### 収集サービスとシステム・モニターの両方が開始する場合

このシナリオでは、収集サービスがすでにある時点で開始されており、後で誰かがシステム・モニターを開始して 30 秒の間隔で CPU 使用率 (平均) メトリック・データを収集する場合は示します。\*MGTCOL オブジェクト・プロパティ・ビューで、システム・レベル・データ、ジョブ MI データ、およびジョブ OS データの категорииの収集間隔が 15 分から 30 秒に変化したことに注意してください。これは、同じ

\*MGTCOL オブジェクトが使用されていること、および指定されたメトリックの情報を計算するのに必要なこれらのカテゴリーだけが、新しい間隔で収集するように変更されたことを示しています。



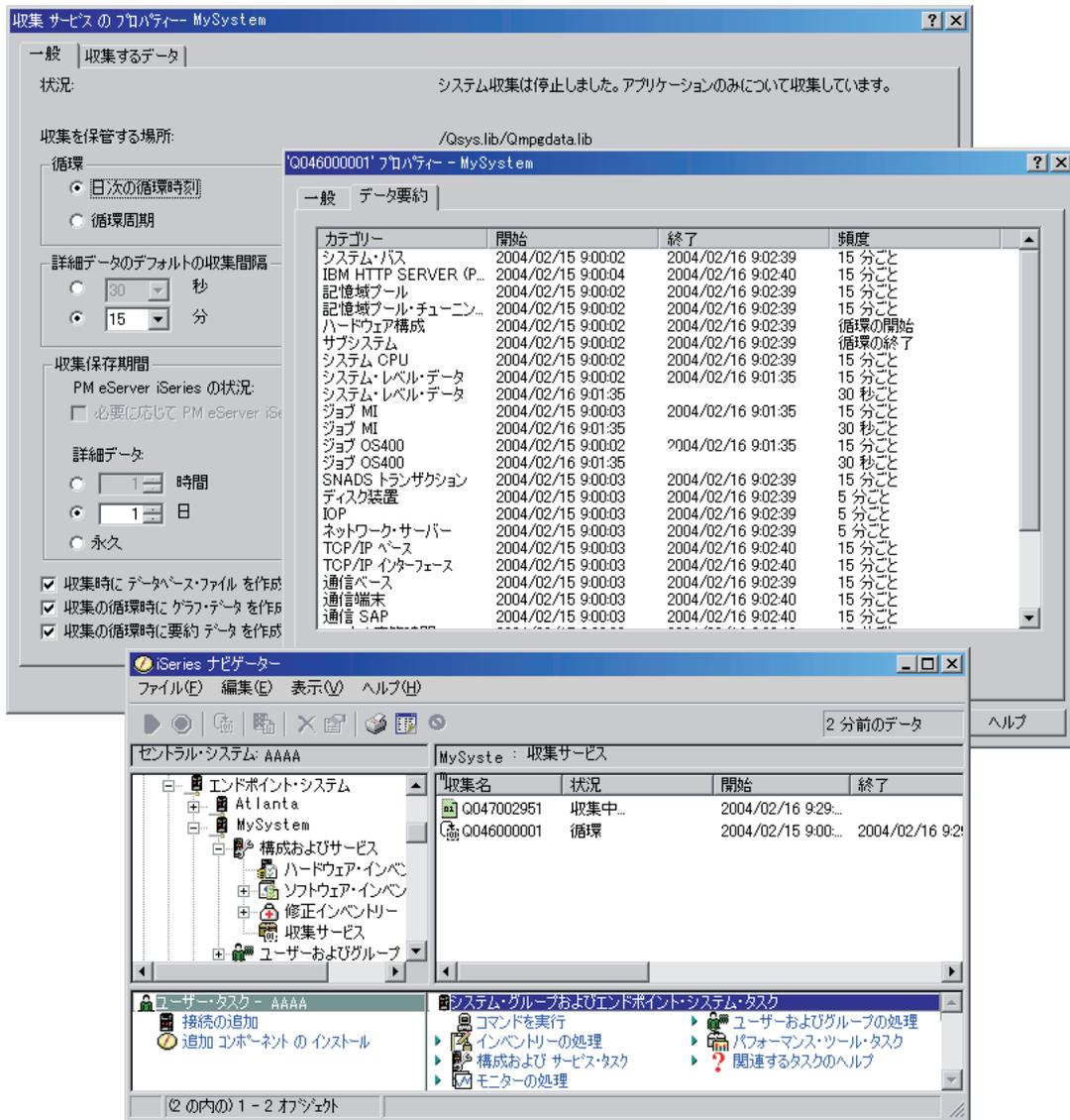
### 収集サービスは停止し、システム・モニターは開始されたままの場合

このシナリオでは、収集サービスは停止され、システム・モニターは開始されたまま、グラフ・メトリックを計算するのに必要なデータの収集を継続します。

以下をご覧ください。

- 収集サービスのプロパティ・ページは、「システム収集は停止しました。アプリケーションのみについて収集しています。」の状況を示します。
- \*MGTCOL オブジェクトのプロパティ・ページは、グラフ・メトリック・データを計算する必要があるものを除き、すべてのカテゴリーに関してデータ収集が終了したことを示します。

- 収集サービスのリスト・ビューは、\*MGTCOL オブジェクトが **収集中...** の状況であることを示します。これは混乱を招く場合があります。したがって、収集サービスの状況を知るには、収集サービスの「プロパティ」ページを見てください。



## 収集サービスのデータからのデータベース・ファイルの作成

収集サービスは、収集されたデータを管理収集オブジェクトに保管します。このデータを使用するには、まず、データをデータベース・ファイルの特別なセットの中に入れる必要があります。データ収集時に自動的にデータベース・ファイルが作成されるようにするには、ただ「**収集サービスの開始**」ダイアログから「**データベース・ファイルの作成**」を選択します。また、既存の管理収集オブジェクトからデータをエクスポートしたいときに、44 ページの『既存の収集オブジェクトからのデータベース・ファイルの作成』こともできます。

データベース・ファイルは、いろいろなオプションで作成できます。

- 収集サービスを使用してパフォーマンス・データを収集する場合、データが収集されるにつれて自動的にデータベース・ファイルが作成されるようにすることができます。

- 収集された後のデータが保管されている管理収集オブジェクトからデータベース・ファイルを作成できます。パフォーマンス・データの作成 (CRTPFRDTA) コマンドを使用すると、管理収集 (\*MGTCOL) オブジェクトに保管されているパフォーマンス情報から一連のパフォーマンス・データベース・ファイルを作成できます。これには、iSeries<sup>TM</sup> ナビゲーターのインターフェースを使用することもできますし、CRTPFRDTA コマンドを使用しても構いません。
- PM iSeries を活動化して、自動的に収集サービスを開始させ、収集の際にデータベース・ファイルを作成させることもできます。

作成したデータベース・ファイルは、Performance Tools for iSeries ライセンス・プログラムやパフォーマンス報告生成用の他のアプリケーションで使用できます。その際には、まず 1 つのシステムでパフォーマンス・データの収集を行い、それからその管理収集 (\*MGTCOL) オブジェクトを別のシステムに移して、パフォーマンス・データ・ファイルの生成と Performance Tools 報告書の実行を行います。このようにすることで、ソース・システムのパフォーマンスには影響を与えることなく、別のシステムでパフォーマンス・

データを分析できます。Performance Tools についての詳細は、Performance Tools for iSeries  を参照してください。

### データベース・ファイルではなく管理収集オブジェクトにデータを保管する理由

なぜ、報告書の実行に必要なデータベース・ファイルではなく管理収集オブジェクトにデータを保管する必要があるのでしょうか。それは、管理収集オブジェクトとデータベース・ファイルを別個に管理して、パフォーマンス・データの収集間隔は短く (例: 5 分間隔) しておき、データベース・ファイルはもっと長いサンプリング間隔 (例: 15 分間隔) で作成するということができるからです。

さまざまなデータ・カテゴリ、さまざまな時間範囲、さまざまなサンプリング間隔を指定することによって、1 つの管理収集オブジェクトから、さまざまな目的のための異なるデータベース・ファイルのセットを作成できます。

たとえば、全カテゴリのセット (すべてのデータ、または「標準 + プロトコル」プロファイル) について、5 分間隔で 24 時間、パフォーマンス・データを収集するようにできますが、その 1 つの管理収集オブジェクトから、さまざまな目的のためのさまざまなデータベース・ファイルのセットを作成することができます。通常の 1 日ごとのパフォーマンス報告を実行するためのデータベース・ファイルのセットを 1 つ作成できます。それらのファイルには、サンプリング間隔を 15 分とした全カテゴリのデータを含めることができます。特定のパフォーマンス上の問題を分析する場合には、データベース・ファイルの別のセットを作成できます。それらのファイルには、分析の対象となる単一のカテゴリで、24 時間以内の特定の期間、そしてサンプリング間隔を 5 分に細分したデータだけを含めることができます。

さらに、単一の管理収集オブジェクトにより、データは多数のファイルとしてではなく、単一のオブジェクトとして管理することができます。この単一の収集オブジェクトによって、リリース間でパフォーマンス・データをデータ変換せずに移動することができます。収集オブジェクトを保持していれば、パフォーマンスに関連したイベントを振り返って、収集できた範囲での詳細な分析が行えます。

### 収集したデータのエクスポート

パフォーマンス・データを管理収集オブジェクトからデータベース・ファイルにエクスポートするには、次のようにします。

1. iSeries ナビゲーターで、「マネージメント・セントラル」の下にあるエンドポイント・システムか、「現在の接続 (My Connections)」(または活動状態の環境) の下にある、直接接続されているシステムを選択します。
2. 「構成およびサービス」を展開します。

3. 「**収集サービス**」をクリックします。
4. データベース・ファイルにエクスポートする管理収集オブジェクトを右クリックして、「**データベース・ファイルの作成**」を選択します。
5. 「**データベース・ファイルの作成**」ダイアログで、収集オブジェクトのうちデータベース・ファイルに含めるカテゴリを選択します。また、収集オブジェクトに含まれているデータが、指定しようとしている期間、サンプリング間隔に対応するものである限り、異なる期間、そして異なるサンプリング間隔を選択することもできます。
6. 「**OK**」をクリックします。

**既存の収集オブジェクトからのデータベース・ファイルの作成:** 既存の管理収集オブジェクトからデータベース・ファイルにパフォーマンス・データをエクスポートできます。次のステップを実行します。

1. パフォーマンス・データが収集されるシステムの「**構成およびサービス**」を展開します。
2. 「**収集サービス**」を選択します。
3. データベース・ファイルにデータをエクスポートする管理収集オブジェクトを右クリックします。
4. 初めに、「**プロパティ**」を選択すると、収集オブジェクトに含まれているデータの特徴を表示することができます。「**データ・プロパティ**」ページには、この収集オブジェクトで収集されたデータのカテゴリや、データが収集された間隔が示されています。これらの情報を利用して、エクスポートするデータを選択することができます。この情報を検討した後、「**OK**」をクリックします。
5. 管理収集オブジェクトをもう一度右クリックして、「**データベース・ファイルの作成**」を選択します。オンライン・ヘルプを使用しながらフィールドに必要な情報を入力します。
6. 「**OK**」をクリックします。

データベース・ファイル内のデータを変換した後、123 ページの『Performance Tools』やパフォーマンス報告生成用の他のアプリケーションを使用できます。

## データ収集のカスタマイズ

収集サービスを使用して 36 ページの『収集サービス』ときは、どのようなデータを収集するか、またどれほどの頻度で収集するかを制御します。これは、提供されているコレクション・プロファイルから選択できます。「**標準 (Standard)**」プロファイルは、以前のリリースでパフォーマンス・モニター機能開始 (STRPFRMON) コマンドにてよって提供されていた OS/400<sup>(R)</sup> のパフォーマンス・モニター機能の設定に相当します。「**標準 + プロトコル**」プロファイルは、全データ用の STRPFRMON コマンドの設定に相当します。あるいは、「**カスタム**」を選択して、独自にカスタマイズしたプロファイルを作成することもできます。使用可能なプロファイルは他にもいくつかあります。詳細な説明については、オンライン・ヘルプを参照してください。プロファイルのカスタマイズにおいては、システム CPU、ローカル応答時間、ディスク装置、IOP (入出力処理装置) などを、使用できるデータ・カテゴリのリストから選択できます。

データを収集する頻度は、収集するデータの各カテゴリごとに指定できます。多くのカテゴリでは、デフォルトの収集間隔を選択できます。これは、15 秒から 60 分の範囲の事前設定値から設定できます。(推奨される設定値は 15 分です。)

**注:** デフォルト値に何らかの特定時間が設定されている場合、ディスク・ストレージ、入出力処理機構、および通信関連のカテゴリなど、明示的な時間間隔を持つカテゴリ以外のカテゴリはすべてこの特定時間を使用します。

収集されたデータは、コレクションと呼ばれる管理収集オブジェクト (タイプ \*MGTCOL) に保管されます。管理収集オブジェクトが大きくなりすぎるのを防ぐため、収集作業は一定の間隔でサイクルとして実行されるようにしてください。収集のサイクルとは、元の収集オブジェクトへのデータ収集が停止すると同時

に、新しい収集オブジェクトを作成してそこにデータを保管し始める、ということです。この間隔は、データの用途に応じて 1 から 24 時間の範囲で自由に指定できます。

システムに合わせて収集サービスをカスタマイズするには、次のようにします。

1. iSeries<sup>™</sup> ナビゲーターで、「マネージメント・セントラル」の下にあるエンドポイント・システムか、「現在の接続 (My Connections)」(または活動状態の環境) の下にある、直接接続されているシステムを選択します。
2. 「構成およびサービス」を展開します。
3. 「収集サービス」を右クリックして、「プロパティ」を選択します。
4. 保存期間をデフォルトの 1 日より長くする場合は、「一般」ページでそれを指定します。保存期間が過ぎると、収集サービスは、管理収集オブジェクトとその中のデータをシステムから削除することになります。管理収集オブジェクトが作成されると、それに有効期限が割り当てられます。それで、その管理収集オブジェクトが別のライブラリーに移動されていても、有効期限が満了すれば、収集サービスはそのオブジェクトを削除します。収集サービスによって新しい収集オブジェクトに有効期限が割り当てられることを望まない場合は、「永続」を指定してください。その後は、これらの収集オブジェクトの削除は、手動で行わなければなりません。  
103 ページの『グラフ・ヒストリー』場合は、「グラフ (Graph)」ないし「要約 (Summary)」のコレクション保存期間を指定する必要があります。これらのオプションを指定すると、ヒストリー報告の機能が利用できるようになり、この機能によって、より長い期間における複数のシステムのメトリックを比較できるようになります。  
さらに、コレクションの保管先のパス、収集のサイクルの頻度、デフォルトの収集間隔も指定できます。収集時に自動的にデータベース・ファイルが作成されるよう選択することもできます。
5. 「収集するデータ」タブをクリックします。
6. 「使用するコレクション・プロファイル」に「カスタム」を選択します。収集の間隔は、カスタマイズ・リストに選択するカテゴリーごとに指定できます。
7. 「OK」をクリックすると、カスタマイズの値が保管されます。

収集サービスの設定値のカスタマイズが完了したなら、もう一度「収集サービス」を右クリックして、「収集サービスの開始」を選択すると、パフォーマンス・データの収集が開始します。

**収集サービスに関連した時間帯の考慮事項:** パフォーマンス・データを検討して分析する際、収集が行われた現地の実時間は重要な要素となります。たとえば、1 日のピークとなる時にどのようなデータが収集されたのかを検討すれば、システムが処理した最も大きなワークロードを判断することができます。パフォーマンス・データを収集するシステムの中に、異なった時間帯に置かれているシステムがある場合は、これらの点について考慮する必要があります。

- システム・グループに対して収集サービスを開始するときは、グループ内のすべてのシステムで同時に収集サービスを開始する必要があります。一部のシステムが別の時間帯にあるために生じるシステム時間や日付の設定のずれは、一切考慮されません。
- マネージメント・セントラル・スケジューラーを使用して収集サービスを開始する場合、スケジューラーは、マネージメント・セントラルのセントラル・システムになっているマシンのシステム時間と日付に基づいてタスクを開始します。
- 各エンドポイント・システムの管理収集オブジェクトは、そのエンドポイント・システムと、使用するセントラル・システムの QTIME および QUTCOFFSET (協定世界時オフセット) システム値に基づいた開始および終了の時刻を反映します。このエンドポイント・システムとセントラル・システムが異なる時間帯にあり、両方のシステムでこれらのシステム値が正しく設定されている場合、収集オブジェクト

に報告される開始および終了の時刻は、エンドポイント・システムがある時間帯での実時間になります。つまり、開始および終了の時刻は、エンドポイント・システムの QTIME の値を、それらのイベントが起きた実際の時刻として反映します。

- パフォーマンスの収集をスケジューリングする場合には、標準時から夏時間へ、あるいは夏時間から標準時への境界を通過することが考えられます。そのような場合は、開始時刻をスケジューリングする際に、この時間のずれを考慮に入れる必要があります。このことを考慮に入れずにスケジューリングを行ってしまうと、実際の開始と終了の時刻が、期待していたよりも 1 時間早く、または 1 時間遅くなってしまう可能性があります。これらに加えて、管理収集オブジェクトに報告される開始と終了の時刻もこの時間のずれの影響を受けるため、夏時間の開始と終了に合わせて QUTCOFFSET システム値を調整する必要があります。

収集サービスを使用したパフォーマンス・データの収集についての詳細は、『36 ページの『収集サービス』』を参照してください。

## 収集サービスのユーザー定義カテゴリー

36 ページの『収集サービス』のユーザー定義カテゴリー機能によって、アプリケーションはパフォーマンス・データ収集を収集サービスに統合することができます。この機能では、データ収集プログラムを作成して登録し、それを収集サービスと統合することによって、データをアプリケーションから収集できます。そして、収集サービスは収集間隔ごとにデータ収集プログラムを呼び出し、収集オブジェクトにデータを保管します。収集オブジェクトに保管されているデータにアクセスするには、以下にリストする収集オブジェクト API を使用する必要があります。データには、そのデータが収集されている間にリアルタイムでアクセスでき、また、収集オブジェクトが保持されている限りアクセスできます。

この機能をインプリメントするには、以下のようにします。

1. 収集サービスで新しいカテゴリー用のパフォーマンス・データ収集プログラムを作成します。詳しくは、『47 ページの『収集プログラムに関する勧告事項と要件』』を参照してください。
2. 収集プログラムのジョブ記述を作成します。QGPL のジョブ記述 QPMUSRCAT は 1 つの例ですが、デフォルト値や勧告事項を示すものではありません。
3. 新しいカテゴリーを登録して、データ収集プログラムを指定します。詳しくは以下の API の説明を参照してください。

- 登録: QypsRegCollectorDataCategory
- 登録取り消し: QypsDeregCollectorDataCategory

カテゴリーを登録すると、それは収集サービスによって使用可能な収集カテゴリーのリストに含まれます。

4. カテゴリーを収集サービス・プロファイルに追加し、収集サービスを循環させます。
5. 収集オブジェクトを照会するプログラムを作成します。詳しくは以下の API の説明を参照してください。
  - 活動状態の収集オブジェクト名の検索: QpmRtvActiveMgtcolName (リアルタイムでの収集オブジェクトの照会にのみ使用されます。)
  - 管理収集オブジェクト属性の検索: QpmRtvMgtcolAttrs
  - 管理収集オブジェクトのオープン: QpmOpenMgtcol
  - 管理収集オブジェクトのクローズ: QpmCloseMgtcol
  - 管理収集オブジェクト・リポジトリのオープン: QpmOpenMgtcolRepo
  - 管理収集オブジェクト・リポジトリのクローズ: QpmCloseMgtcolRepo
  - 管理収集オブジェクト・データの読み取り: QpmReadMgtcolData

これで、カスタマイズした収集プログラムは各収集間隔ごとに実行し、収集されたデータがコレクション・オブジェクト内にアーカイブされるようになります。

さらに、これらの API の Java<sup>(TM)</sup> バージョンもインプリメントできます。必要な Java クラスは、Integrated File System (IFS) ディレクトリー QIBM/ProdData/OS400/CollectionServices/lib の ColSrv.jar に組み込まれています。Java アプリケーションには、そのクラスパスにこのファイルが含まれていなければなりません。Java インプリメンテーションについては、Information Center の javadocs を参照するか、または .zip ファイルの javadocs をダウンロードしてください。

インプリメンテーション例については、『48 ページの『例: ユーザー定義カテゴリーのインプリメント』』を参照してください。

## リアルタイムでの収集オブジェクトの照会

アプリケーションで、収集オブジェクトをリアルタイムで照会する必要がある場合は、照会を収集サービスと同期する必要があります。そうするには、アプリケーションでデータ・キューを作成し、それを収集サービスに登録する必要があります。一度登録されると、コレクターによって各収集間隔ごとと収集サイクルの終了に通知が送信されるようになります。アプリケーションは、データ・キューを保守する必要があります。保守には、終了時のデータ・キューの除去と異常終了の処理が含まれます。データ・キューを登録および登録取り消しするには、以下の API の説明を参照してください。

- コレクター通知の追加: QypsAddCollectorNotification
- コレクター通知の除去: QypsRmvCollectorNotification

**収集プログラムに関する勧告事項と要件:** 収集サービスはデータ収集プログラムを、収集サービスを開始するときに 1 回、各収集間隔ごとに 1 回、そして収集サイクルの終了にもう 1 回呼び出します。データ収集プログラムは任意のデータ収集を実行し、そのデータを収集サービスによって用意されているデータ・バッファーに戻さなければなりません。収集サービスはデータ・バッファーの提供に加えて、データ収集プログラムが収集間隔の間にいくつかの状態情報を保持することのできる作業域を用意しています。

データ収集プログラムはできるだけ速くデータを収集し、最小限の書式設定を実行します。このプログラムはデータ処理やソートは実行しません。ユーザー定義カテゴリーのデータはデータベース・ファイルには変換されませんが、収集サービスが各収集間隔の終了に自動的に CRTPFRTA コマンドを実行して、収集オブジェクトのデータをデータベース・ファイルに追加できます。データ収集プログラムが収集間隔以内にそのタスクを完了できない場合は、それに順応して CRTPFRTA コマンドは実行されません。

データ収集プログラムは以下のいくつかの環境で作成できます。

- OPM 言語用の \*PGM。この環境は、収集オブジェクトの照会には使用できず、パフォーマンスを低下させる可能性があります。しかし、これより古いプログラミング言語ではサポートされます。
- \*SRVPGM。サービス・プログラムのエントリー・ポイント。これは ILE 言語用です。
- \*JVAPGM。必要な Java<sup>(TM)</sup> クラスは ColSrv.jar に含まれています。このファイルは、QIBM/ProdData/OS400/CollectionServices/lib の IFS にあります。API の Java インプリメンテーションの説明については、javadocs.zip ファイルをダウンロードして index.html をオープンしてください。

収集サービスは、以下の要求をデータ収集プログラムに送信します。

要求	説明
収集の開始	データ収集プログラムは、データ収集時に使用する任意のインターフェースまたは資源を初期化します。また、任意で、収集サービスによって用意されていて、収集間隔の間に状態情報が保持される作業域も初期化します。収集データの前に制御レコードを含めたいなら、データ収集プログラムがデータ・バッファに少量のデータを書き込むことも可能です。通常、この制御レコードは、データ処理の際にデータを解釈するために使用されます。
収集間隔	収集サービスは、各収集間隔ごとに間隔要求を送信します。データ収集プログラムはデータを収集して、それをデータ・バッファに戻します。その後、収集サービスは、そのデータを収集オブジェクト内の間隔レコードに書き込みます。 データ・バッファに対してデータの量が大きすぎる場合、データ収集プログラムは「More data」フラグを設定します。この処置が取られると、収集サービスは、別の間隔要求を、それが続きであることを示す修飾子を含めて送信します。収集サービスは、各呼び出しの前に <code>more data</code> フラグをリセットします。このプロセスが、すべてのデータが収集オブジェクト内に移動するまで繰り返されます。
収集の終了	データ収集プログラムが帰属するカテゴリーの収集が終了すると、収集サービスはこの要求を送信します。データ収集プログラムは終結処理を実行します。そして任意で収集制御レコードに戻すことができます。また、データ収集プログラムは、収集の結果を表す戻りコードも送信します。
終結処理と終了 (シャットダウン)	収集サービスは、異常終了が必要な場合にこの要求を送信します。オペレーティング・システム資源は、データ収集プログラムのジョブが終了すると自動的に解放されますが、その他のシャットダウン操作は、データ収集プログラムによって実行されなければなりません。データ収集プログラムはいつでもこの要求を受け付けることができます。

これらのパラメーター、作業域、データ・バッファ、および戻りコードの詳細については、`QSYSINC`にあるヘッダー・ファイル `QPMDCPRM` を参照してください。

### 収集オブジェクト内のデータ・ストレージ

収集オブジェクトには、各データ収集カテゴリー用のリポジトリがあります。このリポジトリは、収集サービスによって、そのカテゴリーの収集が開始するときに作成されます。各リポジトリは、以下のレコードによって構成されます。

レコード	説明
制御	このオプションのレコードは、データ収集プログラムの結果として生じる最初または最後のレコードとなる可能性があり、その両方の位置に存在することがあります。普通、これにはレコード・データを解釈するために必要な任意の情報が含まれている必要があります。
間隔	各収集間隔には、間隔レコードが作成されます。これは空であっても作成されます。間隔レコードには、その収集間隔内にデータ・バッファに書き込まれたデータが含まれています。このサイズは 4 GB 以内でなければなりません。
停止	収集サービスは自動的にこのレコードを作成してデータ収集セッションの終了を示します。収集サービスを終了または循環させずにユーザー定義カテゴリーの収集を再開した場合は、任意で終了レコードの後、追加の間隔レコードの前に制御レコードを含めることができます。

**例: ユーザー定義カテゴリーのインプリメント:** 以下のサンプル・プログラムは、提供されている API を使用してカスタマイズしたデータ収集を収集サービスに統合する方法を示しています。

- 49 ページの『例: データ収集プログラム』
- 51 ページの『例: データ収集プログラムを登録するためのプログラム』

- 53 ページの『例: 収集オブジェクトを照会するためのプログラム』

## コードに関する特記事項

IBM<sup>(R)</sup> は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作使用权を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。

強行法規で除外を禁止されている場合を除き、IBM、そのプログラム開発者、および供給者は「プログラム」および「プログラム」に対する技術的サポートがある場合にはその技術的サポートについて、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

IBM、そのプログラム開発者、または供給者は、いかなる場合においてもその予見の有無を問わず、以下に対する責任を負いません。

1. データの喪失、または損傷。
2. 特別損害、付随的損害、間接損害、または経済上の結果的損害
3. 逸失した利益、ビジネス上の収益、あるいは節約すべかりし費用

国または地域によっては、法律の強行規定により、上記の責任の制限が適用されない場合があります。

**例: データ収集プログラム:** 以下のプログラムはいくつかのテスト・データを収集してデータ・バッファーに保管し、それを収集サービスは収集オブジェクトにコピーします。収集プログラムについて詳しくは、『47 ページの『収集プログラムに関する勧告事項と要件』』を参照してください。

注: 法律上の重要な情報に関しては、『154 ページの『コードに関する特記事項』』をお読みください。

## C++ サンプル・コード

```
#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() */

```

**例: データ収集プログラムを登録するためのプログラム:** 以下のプログラムは、前の例のデータ収集プログラムを収集サービスに登録します。実行すると、収集サービスのデータ収集カテゴリーのリストに、このデータ収集プログラムが表示されます。

注: 法律上の重要な情報に関しては、『154 ページの『コードに関する特記事項』』をお読みください。

### C++ サンプル・コード

```

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

```

```

#include "qypscoll.cleinc"

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

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

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

    if ( argc < 2 || *argv[1] == 'R' )
    {
        pgmAttr = (Qyps_USER_CAT_PROGRAM_ATTR *)malloc( 4096 );
        memset( pgmAttr, 0x00, sizeof(pgmAttr) );
        pgmAttr->fixedPortionSize = sizeof( Qyps_USER_CAT_PROGRAM_ATTR );
        memcpy( pgmAttr->programType,    "*SRVPGM    ", 10 );
        memcpy( pgmAttr->parameterFormat, "PMDC0100", 8 );
        memcpy( pgmAttr->ownerUserId,    "USERID    ", 10 );
        memcpy( pgmAttr->jobDescription, "QPMUSRCAT QGPL    ", 20 );
        memcpy( pgmAttr->qualPgmSrvpgmName, "DCPTEST  LIBRARY    ", 20 );
        pgmAttr->workAreaSize = 123;
        pgmAttr->srvpgmEntrypointOffset = pgmAttr->fixedPortionSize;
        pgmAttr->srvpgmEntrypointLength = 8;
        pgmAttr->categoryParameterOffset = pgmAttr->srvpgmEntrypointOffset +
                                           pgmAttr->srvpgmEntrypointLength;
        pgmAttr->categoryParameterLength = 10;
        /* Set entry point name */
        memcpy( (char *)pgmAttr + pgmAttr->srvpgmEntrypointOffset,
               "DCPentry", pgmAttr->srvpgmEntrypointLength ); /* Set parameter string */
        memcpy( (char *)pgmAttr + pgmAttr->categoryParameterOffset,
               "1234567890", pgmAttr->categoryParameterLength );

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

```

**例: 収集オブジェクトを照会するためのプログラム:** 以下のサンプル・プログラムは、QIBM/ProdData/OS400/CollectionServices/lib の ColSrv.jar ファイルに入って出荷されている Java<sup>TM</sup> クラスを使用して収集オブジェクト内に保管されているデータを照会する方法を示しています。

注: 法律上の重要な情報に関しては、『154 ページの『コードに関する特記事項』』をお読みください。

## Java サンプル・コード

```

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, "MCOD0100" );
} 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 */

```

## 収集オブジェクトの管理

収集サービスを使用して 36 ページの『収集サービス』場合、それぞれの収集の結果は 1 つのオブジェクトの中に保管されます。管理収集オブジェクトに入っているデータの要約は、次のようにして調べることができます。

1. iSeries<sup>(TM)</sup> ナビゲーターで、「**マネージメント・セントラル**」の下にあるエンドポイント・システムか、「**現在の接続 (My Connections)**」(または活動状態の環境)の下にある、直接接続されているシステムを選択します。
2. 「**構成およびサービス**」を展開します。
3. 「**収集サービス**」を選択します。
4. リストの中から任意の管理収集オブジェクトを右クリックし、「**プロパティ**」を選択します。そのコレクションについての一般的な情報と、そこに入っているデータの要約情報が表示されます。

任意の収集オブジェクトを右クリックして、「42 ページの『収集サービスのデータからのデータベース・ファイルの作成』」を選択すると、データベース・ファイルに含めるデータの категория、コレクション期間内の時間範囲、およびサンプリング間隔を指定することができます。

任意の収集オブジェクトを右クリックして「103 ページの『グラフ・ヒストリー』」を選択すると、管理対象オブジェクト内のデータをグラフで表示することができます。

### 古い管理収集オブジェクトの削除または保存

システムから収集オブジェクトを削除するには、オブジェクトを右クリックして「**削除**」を選択します。手動で削除されないオブジェクトは、有効期限の満了後、収集サービスによって自動的に削除されます。

収集サービスは、**循環済み**の管理収集オブジェクトしか削除しません。**循環済み**とは、収集サービスがデータ収集とそのオブジェクトへの保管を停止したことを意味しています。各管理収集オブジェクトの状況は、「**構成およびサービス**」を展開して「**収集サービス**」を選択すると表示される、収集オブジェクトのリストに示されます。

有効期限の満了日に達した循環済み収集オブジェクトは、次に収集サービスが開始された時点で削除されません。有効期限は、管理収集オブジェクトに関連付けられています。それで、その管理収集オブジェクトが別のライブラリーに移動されていても、有効期限が満了すれば、収集サービスはそのオブジェクトを削除しません。

各管理収集オブジェクトの有効期限の満了日は、そのコレクション・オブジェクトの「プロパティ」の中に示されます。その「プロパティ」ページで日付を変更するだけで、オブジェクトがシステムに存在する時間をもっと長くすることができます。リストの中から任意の管理収集オブジェクトを右クリックし、「**プロパティ**」を選択すると、そのコレクションについての情報が表示されます。収集サービスによってその管理収集オブジェクトが自動的に削除されることを望まない場合は、「**永続**」を指定してください。

### ユーザー定義トランザクション

収集サービスおよび Performance Explorer は、ユーザーのアプリケーションで定義するパフォーマンス・データを収集します。備えられている API を用いるならば、収集サービスを使用してトランザクション・データを定期的にスケジュールされているサンプル・データに統合することができ、さらに Performance Explorer を実行することによりトランザクションに関するトレース・レベルのデータを取得できます。

詳細記述および使用上の注意については、以下の API の説明を参照してください。

- トランザクション開始: QYPESTRT、qypeStartTransaction
- トランザクション終了: QYPEENDT、qypeEndTransaction
- トランザクション・ログ: QYPELOGT、qypeLogTransaction (Performance Explorer のみによる使用)
- トレース・ポイント追加: QYPEADDT、qypeAddTracePoint (Performance Explorer のみによる使用)

注: アプリケーションを一度装備するだけで構いません。収集サービスおよび Performance Explorer は同じ API 呼び出しを使用して、さまざまなタイプのパフォーマンス・データを収集します。

### ユーザー定義トランザクション・データを収集サービスに統合する

収集サービスの構成で、ユーザー定義トランザクションを収集用カテゴリーとして選択することができます。収集サービスはその後、すべての収集間隔においてトランザクション・データを収集し、そのデータを収集オブジェクトに保管します。CRTPFRTDTA コマンドは、このデータをユーザー定義のトランザクション・パフォーマンス・データベース・ファイル、QAPMUSRTNS にエクスポートします。収集サービスは、トランザクション・タイプに基づいてデータを編成します。必要な数のトランザクション・タイプを指定することができますが、収集サービスは、最初の 15 のトランザクション・タイプのみを報告します。それ以外のトランザクション・タイプのデータは、\*OTHER トランザクション・タイプとして結合および保

管されます。すべての収集間隔において、収集サービスは、それぞれの固有のジョブのトランザクション・タイプごとに 1 つのレコードを作成します。詳細記述については、『トランザクションの開始 API』にある使用上の注意を参照してください。

収集サービスは、トランザクション応答時間などの一般トランザクション・データを収集します。トランザクションのために使用される SQL ステートメントの数、またはその他の増分測定機能のような、アプリケーションの特定のデータを追跡することのできるオプションのアプリケーション定義によるカウンターを、最大 16 まで組み込むこともできます。アプリケーションは、トランザクション開始 API を使用して新規トランザクションの開始を示す必要があり、収集サービスにトランザクション・データを引き渡すために、対応するトランザクション終了 API を組み込む必要があります。詳しくは、QAPMUSRTNS ファイル記述および API 記述を参照してください。

実装例については、『C++ の例: ユーザー定義トランザクションを収集サービスに統合する』または『61 ページの『Java™ の例: ユーザー定義トランザクションを収集サービスに統合する』(TM)』の例を参照してください。

注: 法律上の重要な情報に関しては、『154 ページの『コードに関する特記事項』』をお読みください。

### Performance Explorer によるユーザー定義トランザクションのトレース情報の収集

Performance Explorer セッションの時に、トランザクションの開始、終了、およびログ API を使用して、トレース・レコードを作成することができます。Performance Explorer は、現行のスレッドのシステム・リソース使用状況 (CPU 使用率、入出力、捕獲/ロック活動など) を、それらのトレース・レコードに保管します。加えて、アプリケーション固有のパフォーマンス・データを組み込む選択をするならば、その後、それらの各 API にある Performance Explorer にそれを送信することができます。さらに、トレース・ポイント追加 API を使用して、Performance Explorer がトレース・データを収集する必要があるアプリケーション固有のイベントを識別します。

トランザクションのための Performance Explorer セッションを開始するには、134 ページの『Performance Explorer』定義の (OSEVT) パラメーターで \*USRTRNS を指定します。ENDPEX コマンドを入力した後、Performance Explorer は、アプリケーションによって提供されたデータを、QAYPEMIUSR Performance Explorer データベース・ファイルの QMUDTA フィールドに書き込みます。開始、終了、およびすべてのログ・レコードのためにシステムにより備えられたパフォーマンス・データは、QAYPEMIUSR および QAYPETIDX データベース・ファイルに保管されます。

詳細記述については、『トランザクションの開始 API』の記述にある API 記述および使用上の注意を参照してください。

**C++ の例: ユーザー定義トランザクションを収集サービスに統合する:** 以下の C++ のプログラム例では、トランザクション開始およびトランザクション終了 API を使用して、ユーザー定義のトランザクション・パフォーマンス・データを収集サービスに統合する方法を示しています。

注: 法律上の重要な情報に関しては、『154 ページの『コードに関する特記事項』』をお読みください。

```
//*****  
// 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<sup>TM</sup> の例: ユーザー定義トランザクションを収集サービスに統合する:** 以下の Java のプログラム例では、トランザクション開始およびトランザクション終了 API を使用して、ユーザー定義のトランザクション・パフォーマンス・データを収集サービスに統合する方法を示しています。

**注:** 法律上の重要な情報に関しては、『154 ページの『コードに関する特記事項』』をお読みください。

```

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|disablesecsToDelay");
        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.println("%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 */

```

## 区画を介したパフォーマンス・データの収集

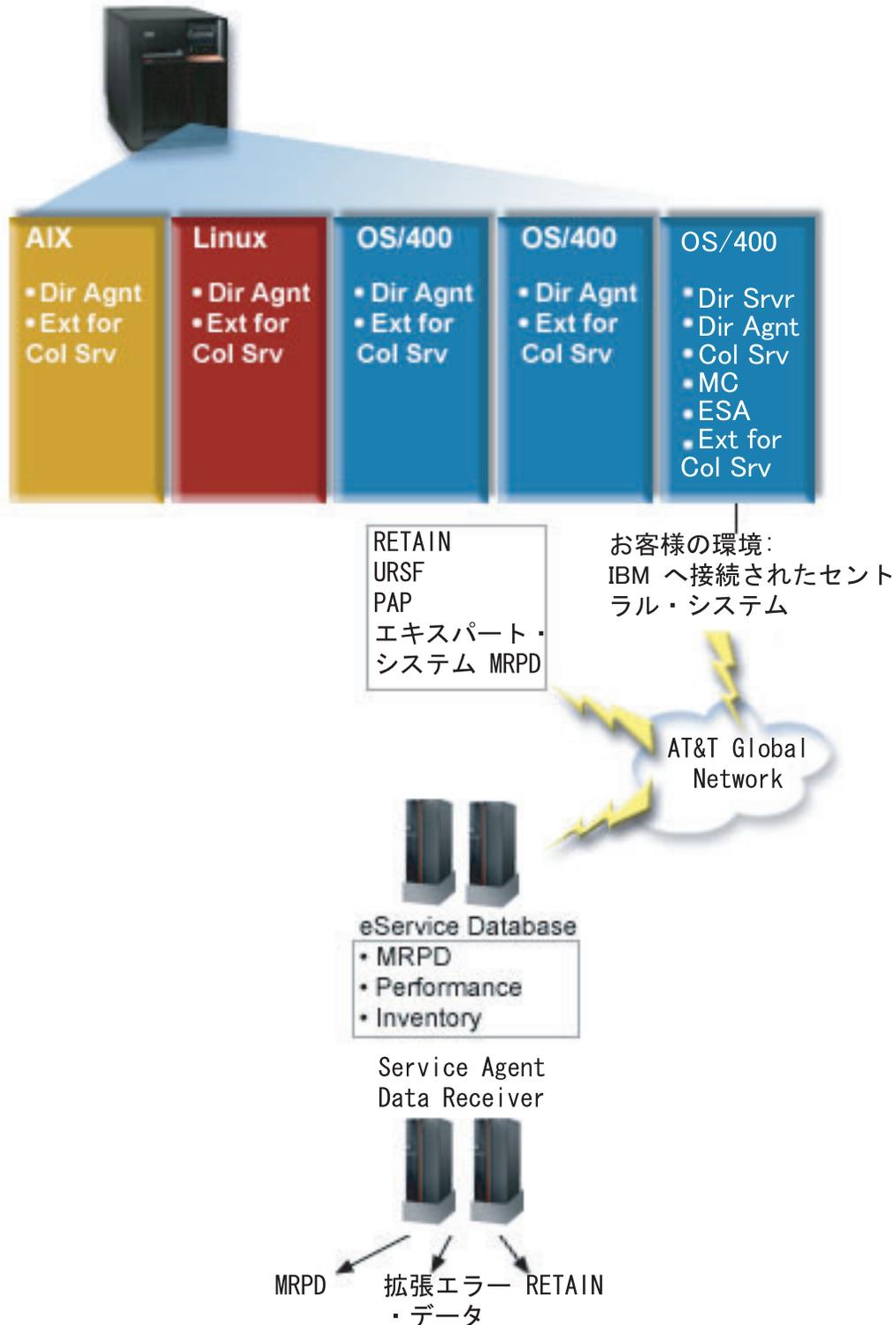
 IBM<sup>(R)</sup> Performance Management for  eServer iSeries<sup>(TM)</sup> (PM eServer<sup>(TM)</sup> iSeries または PM iSeries) は、収集サービスを自動的に起動して、所有権の付いていないパフォーマンス・データおよびキャパシティー・データをサーバーから収集してから、そのデータを分析するため IBM に送信します。PM iSeries が提供する分析の 1 つは、システムの成長を作図して、アップグレードが必要になる時期を判別することです。区画に分割されていないシステムの場合、これは非常に簡単なプロセスです。しかし、システムが複数の OS/400<sup>(R)</sup> 区画に分割されている場合、データは各区画から個別に IBM に届くため、システム・パフォーマンス全体の信頼できるビューを形成することがより難しくなります。区画が AIX<sup>(R)</sup> または Linux<sup>(TM)</sup> を実行している場合、または OS/400<sup>(R)</sup> 区画のいずれかで PM iSeries<sup>(R)</sup> がオフになっている場合、データは送信されず、システム全体のビューの形成はほとんど不可能になります。

これらの問題に対処するため、IBM Director Multiplatform を使用する収集サービスは、実行するオペレーティング・システムに関係なく、iSeries 区画から、使用可能なプロセッサの CPU 使用量および数についてのデータを受け取ることができるようになりました。PM iSeries は、IBM に配送される前にデータを要約します。CPU 使用率の区画をまたいだビューが提供されているため、ユーザーも IBM も、システム・リソースの管理作業をよりよく実行することができます。これには、プロセッサの現行セット全体でワークロードを平衡化すること、必要なときにプロセッサを追加購入したり、より高速なプロセッサを購入するためのプランを立てることが含まれます。

### 作業の方法

以下の図は、論理区画を介した CPU 使用率データの収集を行う方法を示します。「セントラル・システム」では、\*LPAR カテゴリーを選択した収集サービスを実行している OS/400 区画に、IBM Director Server がインストールされています。他の区画から IBM Director Server がパフォーマンス・データを収集するためには、それらの区画のそれぞれに IBM Director Agent がインストールおよび構成されていなければなりません。それぞれの区画には、収集サービスの Director Multiplatform 拡張機能がインストールされていなければなりません。IBM Director Server は、一定の間隔で、それ自体も含めた各区画ごとに CPU 使用率データを検索し、そのデータを収集サービスの \*MGTCOL オブジェクトに保管します。その後、データは処理され、QAPMLPAR データベース・ファイルに書き込まれます。最後に、PM iSeries はデータ

を収集および集約し、IBM に伝送する準備をします。この図では、データを送信するようセットアップしたマネージメント・セントラルと IBM Electronic Service Agent<sup>TM</sup> (ESA) を、IBM Director Server および収集サービスと同じ区画にあるものとして示されていますが、実際には、IBM への送信メカニズムを完全に別個のシステム上で実行し、なおかつ PM iSeries から区画をまたいだデータを収集し、通常の業務として IBM に送信するようにセットアップすることができます。



## 注

Dir Srvr = IBM Director Server  
Dir Agnt = IBM Director Agent  
Col Srv = 収集サービス  
MC = マネージメント・セントラル  
ESA = IBM Electronic Service Agent  
Ext for Col Srv = 収集サービスの Director Multiplatform 拡張機能  
RETAIN = リモート・テクニカル支援情報ネットワーク  
URSF = Universal Remote Support Facility  
MRPD = Machine Reported Product Data

## セットアップ

次のリストは、論理区画を介してパフォーマンス・データを収集するために完了する必要のステップの概要を示します。

1. 同じ物理システム上のすべての区画について、IP ネットワークが適切に構成されているか確認します。
2. パフォーマンス・データ収集の対象となる各区画で実行されているオペレーティング・システムが、サポートされているものであることを確認します。サポートされているオペレーティング・システムは、以下のとおりです。
  - OS/400 バージョン 5 リリース 3
  - AIX 5L<sup>(TM)</sup> バージョン 5.3
  - Red Hat Enterprise Linux<sup>(TM)</sup> AS V3.0 for IBM PowerPC<sup>(R)</sup>
  - SUSE LINUX<sup>(TM)</sup> Enterprise Server 8 for IBM pSeries<sup>(R)</sup> and IBM iSeries
  - SUSE LINUX<sup>(TM)</sup> Enterprise Server 9 for IBM pSeries<sup>(R)</sup> and IBM iSeries
3. 管理サーバーとして動作する区画に、以下の収集サービス PTF 修正が適用されていることを確認します。
  - SI12971
  - SI13838 (SI16328 に置き換えられています)
  - SI15131 (SI16499 に置き換えられています)
  - SI16328 (Linux サポート)
  - SI16499 (AIX サポート)

Linux オペレーティング・システム用収集サービス区画間サポートの最新の情報については、情報 APAR II13986 を参照してください。

最新の PTF 修正については、Fix Central をご覧ください。

4. Virtualization Engine<sup>(TM)</sup> を使用して、管理サーバーとして動作させる OS/400 区画上で、IBM Director Server をインストールします。管理サーバーを、管理対象システム、デバイス、および収集サービスと通信する、中央制御点と考えます。Virtualization Engine インストール・ウィザードが完了すると、IBM Director Server および IBM Director Agent が、管理サーバーとして動作させる OS/400 区画にインストールされます。
5. IBM Director Server で管理される区画に、IBM Director Agent をインストールします。これらの区画は、IBM Director Server のインストール先区画と同じ物理システム上になければなりません。
6. Director Multiplatform 管理コンソールとして動作させるシステム上に IBM Director Console をインストールします。

7. 必要な構成ステップとして、以下のことを実行します。
  - a. 管理区画上で OS/400 のユーザーに許可を与えます。
  - b. 各区画上で Director Multiplatform を開始します。
  - c. 管理コンソール上で IBM Director コンソールを開始します。
  - d. IBM Director コンソールの「Group Contents (グループ・コンテンツ)」ペインで「New (新規)」>「IBM Director Systems (IBM Director システム)」を選択することにより、パフォーマンスをモニターする区画をそれぞれ追加します。
  - e. 各区画を追加した後、区画を管理するためのアクセスを要求します。「Group Contents (グループ・コンテンツ)」ペインで区画を右クリックして、「Request Access (アクセスの要求)」を選択します。
8. OS/400 管理区画上で、収集サービスに必要なファイルを収集サービスのディレクトリーから該当する Director Multiplatform ディレクトリーにコピーすることにより、収集サービスの Director Multiplatform 拡張機能をインストールします。収集サービスのファイルは、ColSrvLparDataExt.TWGExt、ColSrvLparDataSubagt.TWGSubagent、および ColSrvDir.jar です。収集サービスのファイルをコピーするには、以下のコマンドを使用します。
 

```
CPY OBJ('/qibm/proddata/os400/collectionservices/lib/ColSrvLparDataExt.TWGExt')
  TODIR('/qibm/userdata/director/classes/extensions')
```

```
CPY OBJ('/qibm/proddata/os400/collectionservices/lib/ColSrvLparDataSubagt.TWGSubagent')
  TODIR('/qibm/userdata/director/classes/extensions')
```

```
CPY OBJ('/qibm/proddata/os400/collectionservices/lib/ColSrvDir.jar')
  TODIR('/qibm/userdata/director/classes')
```
9. 管理区画にある収集サービスのファイルを、パフォーマンス・データを収集する予定の OS/400 区画に配布します。これは、ファイル転送プロトコル (FTP) にバイナリー・オプションを指定することによって、またはドライブをマッピングしてファイルをファイル・システムにコピーすることによって、あるいはその他の配布メカニズムによって実行できます。OS/400 管理区画上のファイルには、/qibm/proddata/os400/collectionservices/lib ディレクトリー内でアクセスできます。
  - a. ColSrvLparDataExt.TWGExt を、管理対象の OS/400 区画上の Director Multiplatform の拡張機能のディレクトリー /qibm/userdata/director/classes/extensions に配布します。
  - b. ColSrvLparDataSubagt.TWGSubagent を、管理対象の OS/400 区画上の Director Multiplatform の拡張機能のディレクトリー /qibm/userdata/director/classes/extensions に配布します。
  - c. ColSrvDir.jar を、管理対象の OS/400 区画上の Director Multiplatform のクラス・ディレクトリー /qibm/userdata/director/classes に配布します。
10. 各 Linux<sup>(TM)</sup> 区画上に、収集サービスの Director Multiplatform 拡張機能をインストールするには、収集サービス RPM ファイルである ColSrvDirExt-5.3.0-1.ppc64.rpm をインストールします。
  - a. 管理区画にある収集サービスの RPM ファイルを、パフォーマンス・データを収集する予定の Linux<sup>(TM)</sup> 区画に配布します。これは、ファイル転送プロトコル (FTP) にバイナリー・オプションを指定することによって、またはドライブをマッピングしてファイルをファイル・システムにコピーすることによって、あるいはその他の配布メカニズムによって実行できます。OS/400 管理区画ディレクトリー内の RPM ファイル /qibm/proddata/os400/collectionservices/lib/ColSrvDirExt-5.3.0-1.ppc64.rpm にアクセスするには、Qshell を使用することができます。
  - b. 各 Linux<sup>(TM)</sup> 区画上で、RPM ファイルが存在するディレクトリーから以下のコマンドを実行します。
 

```
rpm -Uhv --force ColSrvDirExt-5.3.0-1.ppc64.rpm
```

11. 各 AIX<sup>(R)</sup> 区画に、収集サービスの Director Multiplatform 拡張機能をインストールするには、収集サービスのパッケージである aix-ColSrvDirExt-5.3.bff をインストールします。
  - a. 管理サーバーにある収集サービスのパッケージ・ファイルを、パフォーマンス・データを収集する予定の AIX<sup>(R)</sup> 区画に配布します。これは、ファイル転送プロトコル (FTP) にバイナリー・オプションを指定することによって、またはドライブをマッピングしてファイルをファイル・システムにコピーすることによって、あるいはその他の配布メカニズムによって実行できます。OS/400 管理区画ディレクトリー内のパッケージ・ファイル  
/qibm/proddata/os400/collectionservices/lib/aix-ColSrvDirExt-5.3.bff にアクセスするには、Qshell を使用することができます。
  - b. 各 AIX<sup>(R)</sup> 区画上で、BFF ファイルが存在するディレクトリーから以下のコマンドを実行します。

```
installp -Fac -d aix-ColSrvDirExt-5.3.bff ColSrvDirExt
```
12. IBM Director Console で、区画を右クリックして、「**Perform Inventory Collection (インベントリー収集の実行)**」を選択し、それぞれの区画上で収集インベントリーを更新します。
13. 108 ページの『PM iSeries の活動化』して、収集サービスの開始を自動化してから、収集中にデータベース・ファイルを作成します。PM iSeries が既に実行中の場合は、以下のパフォーマンス収集開始 (STRPFCOL) コマンドを実行します。

```
STRPFCOL CYCCOL(*YES)
```

## 関連情報

- IBM Director Multiplatform
- 105 ページの『IBM Performance Management for eServer iSeries』
- IBM Virtualization Engine
- マネージメント・セントラル
- サーバーの区分化 (Partitioning the server)
- 109 ページの『エクストリーム・サポートでの Service Agent による PM iSeries<sup>(TM)</sup> データの送信 (ユニバーサル・コネクション)』



## ジョブ、タスク、またはスレッドの待機統計の検索

▶ ジョブ、タスク、またはスレッドの実行中には、そのプロセスを待機させる条件が発生します (たとえば、システムがロックを解決している間や、必要なオブジェクトの保留中)。収集サービスは、プロセス待機の時間の原因と所要時間のデータを収集できます。

この情報の使用およびアクセスについては、パフォーマンス・データベース・ファイル QAPMJOBWT および QAPMJOBWTD を参照してください。

注: これらのファイルを照会するためには、システム CCSID ID を 1 次言語 (バイナリー・データの 65535 とは異なる) に設定する必要があります。◀

## 収集サービスによるディスク使用量の理解

▶ 収集サービスが消費するディスク・リソース量は、使用する設定によってかなり異なります。一例として、収集サービスが毎日使用され、真夜中に循環し、各 \*MGTCOL オブジェクトに 1 日分のデータ収集が入れるると想定します。次に、収集サービスのデフォルト・プロパティーを使用して、1 日分のデータ収集の基本サイズを確立します。標準に、15 分の間隔値を持つプロトコル・プロファイルを足すと、\*MGTCOL オブジェクトに 500 MB のデータを収集できます。デフォルト・プロパティーを使って実際に

1 日ごとに収集されるサイズは、システム・サイズおよび使用量によってかなり異なります。500 MB の例は、頻繁に使用されるハイエンド・システムを表すと考えられます。

間隔の頻度	収集ごとの間隔数	乗数	MB 単位のサイズ
15 分	96	1	500

1 日分のデータのサイズは、収集期間ごとに収集される間隔の数に対して、直接的に比例します。たとえば、間隔の頻度を 15 分から 5 分に変更すると、間隔の回数は 3 の係数分増え、サイズも同じ係数分増えます。

間隔の頻度	収集ごとの間隔数	乗数	MB 単位のサイズ
15 分	96	1	500
5 分	288	3	1500

この例を継続し、以下の表に、プロトコル・プロファイルを加えたデフォルト標準を使って、各間隔の頻度で収集サービスが 1 日ごとに作成する 1 つの \*MGTCOL オブジェクトのサイズを示します。

間隔の頻度	収集ごとの間隔数	乗数	MB 単位のサイズ
15 分	96	1	500
5 分	288	3	1500
1 分	1440	15	7500
30 秒	2880	30	15000
15 秒	5760	60	30000

この例で、\*MGTCOL オブジェクトのサイズは、収集の比率によって 500 MB から 30 GB まで変化します。15 分のデフォルト収集間隔およびプロトコル・プロファイルをベースとして追加した標準を使用してから、上記の表からの乗数を使用して他の収集間隔でのディスク使用量を判別すると、作成される \*MGTCOL オブジェクトを実際に観察することによって、1 日の収集間隔に応じた特定のシステムのディスク使用量を予測できます。たとえば、\*MGTCOL オブジェクト・サイズを観察した結果、1 日の収集に応じたオブジェクトのサイズが 15 分の間隔の場合 50 MB だと分かったら、15 秒の間隔でデータを収集する場合、収集サービスが 3 GB のサイズで \*MGTCOL オブジェクトを作成すると予想できます。

注: 15 秒程度の頻繁な収集間隔を考慮する場合には、注意が必要です。収集間隔が頻繁な場合、システム・パフォーマンスに悪影響を及ぼす恐れがあります。

## 保存期間

保存期間も、収集サービスが消費するディスク・リソースの量に重要な役割を果たします。デフォルト保存期間は 1 日です。しかし、このデフォルト値を指定した場合、実際には、\*MGTCOL オブジェクトは、作成日を過ぎてから、収集の 3 日目に削除されることとなります。したがって、収集の 3 日目には、以前に収集された 2 日分のデータと、当日のデータがシステム上に存在します。上記の表を使用すると、これは 3 日目以降、間隔が 15 分なら 1 GB から 1.5 GB のディスクが、間隔が 15 秒なら 60 から 90 GB のディスクが、システム上で消費されることに相当します。

保存期間を基にしたディスク使用量を計算するための公式は、次のとおりです。

(日数単位の保存期間 + 2.5) \* 1 日の収集のサイズ = 合計ディスク使用量

注: 2.5 は前の収集データの 2 日分、および当日の平均です (2 日 + 1/2 日)。

上記の表と公式を使用すると、2 週間の保存期間の場合、例のシステムでは、間隔が 15 分なら 8.25 GB のディスク使用量、間隔が 15 秒なら 495 GB です。

受け入れ可能な収集間隔および指定されたシステムの保存期間を知るため、収集サービスによるディスク使用量を理解することは重要です。これを知っておくことによって、ディスク使用量がシステム問題を起こさないようにすることができます。システム・モニターまたはジョブ・モニターが、モニター用のデータをグラフ化するためのカテゴリーの収集間隔をオーバーライドできることを忘れずに考慮してください。システム管理者は注意して、モニターのデータ収集の間隔が原因でデータ消費が過剰にならないようにすべきです。



## インテリジェント・エージェント

インテリジェント・エージェントは、複雑なオートノミック・アルゴリズムを介し、長期に渡って特定の動作を学習することが可能な、Java<sup>(TM)</sup> ベースのソフトウェア・コンポーネントです。インテリジェント・エージェントには、特定のイベントのモニターなどの単純なモニターから、ネットワーク問題の分析、計画外のシステム再開の防止、またはストレージの管理などのより複雑なアクションまで、さまざまな機能を持たせることができます。エージェントを使用する目的は、オートノミック・コンピューティングによってシステム管理者のタスクを単純化することですが、依然としてシステム管理者には、それらのエージェントのアクションの開始や停止、アクションへの応答、およびアクションのモニターの手段が必要です。

iSeries<sup>(TM)</sup> ナビゲーターのインテリジェント・エージェント・コンソールにより、システム管理者は、単一のシステム上、または複数のシステムをまたいで実行される 1 つ以上の ABLE (Agent Building and Learning Environment) エージェントを簡単に管理できます。エージェント・コンソールをドメインのいたるところに存在するエージェント・サービスへ接続したら、構成済みのエージェントをいくつでもモニターおよび処理できます。エージェントはドメインのどのシステム上で稼働していてもかまいません。

### 71 ページの『インテリジェント・エージェントの概念』

インテリジェント・エージェント・コンソールは、分散エージェント・プラットフォーム上で実行する ABLE エージェント、または分散エージェント・プラットフォームを介して実行する ABLE エージェントを使用します。分散プラットフォームを構成する ABLE エージェントとエージェント・サービスについて理解を深めてください。

### 73 ページの『エージェントの開発』

必要なタスクを実行するには、独自のエージェントを作成およびカスタマイズします。ABLE ツールキットおよびその関連資料には、作業開発環境と、独自のエージェントを開発するためのガイドとして使用できるテンプレート・エージェントが用意されています。

### 75 ページの『エージェント環境のセットアップ』

インテリジェント・エージェント・コンソールを使ってエージェントの管理を開始する前に、エージェントとエージェント・サービス (エージェント・プラットフォーム) を、環境内のシステム上、またはシステムを介して実行するように構成しなければなりません。セキュア環境では、Kerberos および追加のプラットフォーム構成が必要です。

### 86 ページの『エージェントの管理』

エージェント・コンソールを使ってドメインに接続し、エージェントの管理を開始します。エージェントに関連した自動化のレベルの制御の方法が分かると、容易に要求に応答したり、エージェント・ヒストリーを追跡したりすることができます。

## インテリジェント・エージェントの概念

概念:

### 『ABLE エージェント』

iSeries<sup>(TM)</sup> ナビゲーターのインテリジェント・エージェント・コンソールは、ABLE (Agent Building and Learning Environment) エージェントと連動します。ABLE は、マルチエージェント・インテリジェント・オートノミック・システムの構築に使用される、Java<sup>(TM)</sup> フレームワークおよびツールキットです。

### 『エージェント・プラットフォーム』

エージェント・サービスはご使用のシステム上、または分散プラットフォーム全体で作動し、エージェントのライフ・サイクル、セキュリティー、および動作を管轄します。

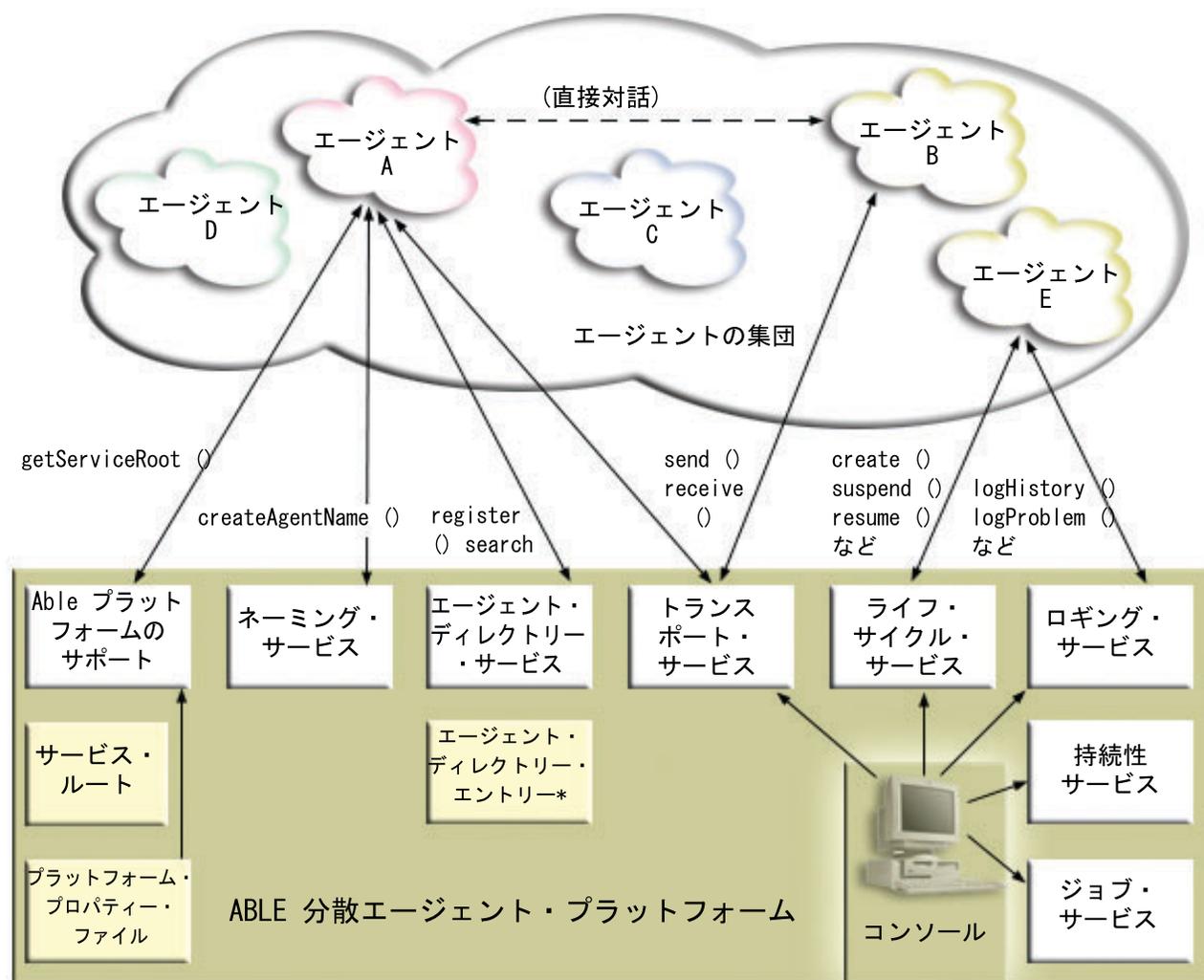
**ABLE エージェント:** iSeries<sup>(TM)</sup> ナビゲーター用のインテリジェント・エージェント・コンソールは、Agent Building and Learning Environment (ABLE) エージェントと連動します。ABLE エージェントは Java<sup>(TM)</sup> オブジェクトであり、ABLE コンポーネント・ライブラリー内のデータ・マイニング・アルゴリズムを使い、長期に渡ってルール・ベースで動作の推論と学習をすることにより、タスクを自動化できます。ABLE はマルチエージェント・エージェント・オートノミック・システムの構築に使用される Java フレームワークおよびツールキットであり、iSeries ナビゲーターのインテリジェント・エージェントのプラットフォームおよびコンソールと連動するエージェントを開発するための特定のサポートを提供します。ABLE を使って開発したインテリジェント・エージェントには次の機能があります。

- 経験から学習し、将来の状態を予測する。
- 分類およびクラスタリング・アルゴリズムを使用してメトリック・データを分析し、複雑な状態を検出したり問題を診断したりする。
- Web サービスを介して他のオートノミック・コンポーネントとの間のインターフェースとして機能する。
- ドメイン特有の Java アプリケーション・オブジェクトを使って推論する。
- 強力な機械推論を使用する (ブール前向きおよび後向き推論、述語論理 (Prolog)、Rete 型パターン・マッチング、およびファジー・システムを含む)。
- 自律的 (自発的) な動作とゴールを持つ。
- 事象を相関させて状態にし、推論し、処置を講じる。

ABLE ツールキットには独自のエージェントの設計方法の例がいくつか含まれており、独自のエージェントの開発時にモデルとして使用できる、iSeries のテンプレート・エージェントが含まれています。完全にコンソールから管理可能なエージェントを作成するには、エージェントは **AbleEServerDefaultAgent** を拡張しなければなりません。ABLE エージェントの開発、および ABLE ツールキットとそれに対応する資料を入手する方法については、73 ページの『エージェントの開発』を参照してください。

**エージェント・プラットフォーム:** iSeries<sup>(TM)</sup> ナビゲーターのインテリジェント・エージェント・コンソールでは、エージェント・プラットフォームをシステム上、または分散ネットワーク全体に構成する必要があります。エージェント・プラットフォームとは、プラットフォームのサービスまたはエージェントを実行する Java<sup>(TM)</sup> 仮想マシン、つまりエージェント・プールに過ぎません。プラットフォームは、`ableplatform.preferences` というプリファレンス・ファイルで定義されます。このファイルには、各エージェント・プール (JVM) のロケーション (システムおよびポート)、プラットフォーム上またはプラットフォーム全体で実行するサービス、およびプラットフォームでの実行が許可されるエージェントをリストします。セキュリティーが構成される場合、プリファレンス・ファイルには、プラットフォームを構成する各サービス、エージェント、およびユーザーを認証するために使用される Kerberos ユーザー・プリンシパルとサービス・プリンシパルもリストされます。

エージェント・サービスは、分散プラットフォームのどのシステムにでも存在でき、エージェントのライフ・サイクル、セキュリティー、および動作を管轄します。同一システムで実行されるエージェント、または異なる複数のシステムにまたがって実行される分散エージェントは、さまざまなタスク用に定義されたプラットフォーム・サービスのセットを使用します。たとえば、固有の名前を取得すること、ディレクトリーで他のエージェントを探索すること、ロギング、別のエージェントへメッセージを渡すことなどです。



プラットフォーム上またはプラットフォームを介して実行するエージェント、およびそのプラットフォームに接続されるユーザーは、次のサービスを使用可能です。

• **ネーミング・サービス**

このサービスは、分散プラットフォームの他の部分すべてに渡って、グローバルに固有の名前を作成します。またセキュリティーがオンの場合、ネーミング・サービスはプラットフォームにセキュリティーを提供します。プラットフォームを開始して、すべてのサービス、プール、およびユーザーを認証するには、Kerberos が使用されます。プラットフォームの持続期間中、このサービスは信頼のおける第三者機関として作動し、プラットフォームのエージェント、サービス、およびユーザー間のすべての対話を保護します。

• **ディレクトリー・サービス**

あるエージェントが、プラットフォーム全体の他のサービスおよびエージェントに対して自分自身につ

いて知らせる場合、このエージェントはエージェント記述を作成し、この記述をディレクトリー・サービスに登録します。エージェントに登録した後で、記述を変更したり除去したりすることができます。

- **ライフ・サイクル・サービス**

このサービスは、エージェントの管理に使用されます。このサービスを使って、エージェントの作成、開始、中断、再開、および破壊ができます。

- **トランスポート・サービス**

このサービスは、プラットフォームの各部分にロケーターを提供します。エージェント間の通信も、このサービスによって使用可能になります。

- **ロギング・サービス**

実行中のエージェントが問題に遭遇し、外部からの介入を必要とする場合があります。ロギング・サービスは要求を作成してログに記録し、要求に対して送り返される対応する応答を処理します。エージェントの進行状態も、他者から見られるよう、このサービスにログ記録されます。

- **eServer<sup>(TM)</sup> ジョブ・サービス**

プラットフォームのさまざまなサービスとジョブは、ジョブ項目をこのサービスに登録します。このサービスは、プラットフォームが iSeries で実行している間、プラットフォームについての重大な情報を提供します。

- **持続性サービス**

サービスおよびエージェントは、このサービスを使って有用な情報を持続させることができます。持続性サービスを構成すると、ネーミング、ディレクトリー、ライフ・サイクル、ロギング、およびジョブ・サービスをバックアップして、データベースに保管できます。

## エージェントの開発

Agent Building and Learning Environment (ABLE) ツールキットを使って、独自のハイブリッド・インテリジェント・エージェントを開発することができます。この Java<sup>(TM)</sup> フレームワークには独自のルール言語 (ARL) および独自の GUI ベースの対話式開発環境、Able Agent Editor があります。どちらも、ABLE エージェントの構築を援助するために提供されています。

### ABLE 2.0

ABLE ツールキットおよび ABLE の完全な資料を .zip パッケージの形でダウンロードできます。

iSeries<sup>(TM)</sup> ナビゲーターのインテリジェント・エージェント・コンソールにはテンプレート・エージェントが付属しており、コンソールと連動するエージェントを開発するためのガイドラインとして使用できます。

**AbleEserverTemplateAgent** のソース・コードは、**QIBM/ProdData/OS400/Able** にある **ableplatform.jar** に保管されています。

**AbleEserverTemplateAgent** は、ABLE フレームワークを使用してエージェントを開発する際に使用可能な多くの機能を利用しています。また、コンソールを介して管理可能な機能のセットを、エージェントによってどのようにして作成できるかを示しています。さらに、エージェント設定を変更するのに使用する「**カスタマイズ (Customize)**」パネル、およびエージェントに関する情報を表示する「**情報 (About)**」パネルが含まれています。また、エージェントがロギング・サービスを使用して要求と履歴項目をログ記録する方法も示しています。これらの要求と項目はコンソールで表示と応答が可能です。

### エージェントの機能

EserverTemplateAgent には以下の機能があります。

- **時刻モニター**

エージェントは、分や時を変更するイベントを監視して、アクションを実行します。設定されている機能、または要求をログ記録する場合のユーザーの応答の仕方に応じて、エージェントは次の 4 種類の対応をします。

1. 時刻を通知せずに変更をログ記録する。
2. 時刻を long の形式で通知して、変更をログ記録する。
3. 時間を MM/DD/YY 形式で通知して、変更をログ記録する。
4. 何もしない。

• **重複した重複**

エージェントは時分を変更する複数の要求を監視します。この機能に関し、重複があった場合にエージェントは次の 2 種類の対応をします。

1. 重複要求を作成する。
2. 重複要求を作成しない。

**カスタマイズ・パネル**

エージェントには、分または時間が変更されたかどうかをエージェントが検査する間隔を調整するためのカスタマイズ・パネルが用意されています。

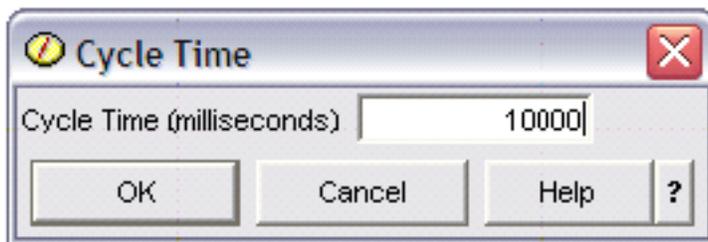


図 1: カスタマイズ・パネルの使用例

**「情報 (About)」パネル**

エージェントには、エージェントの詳細情報を提供するためのパネルが用意されています。

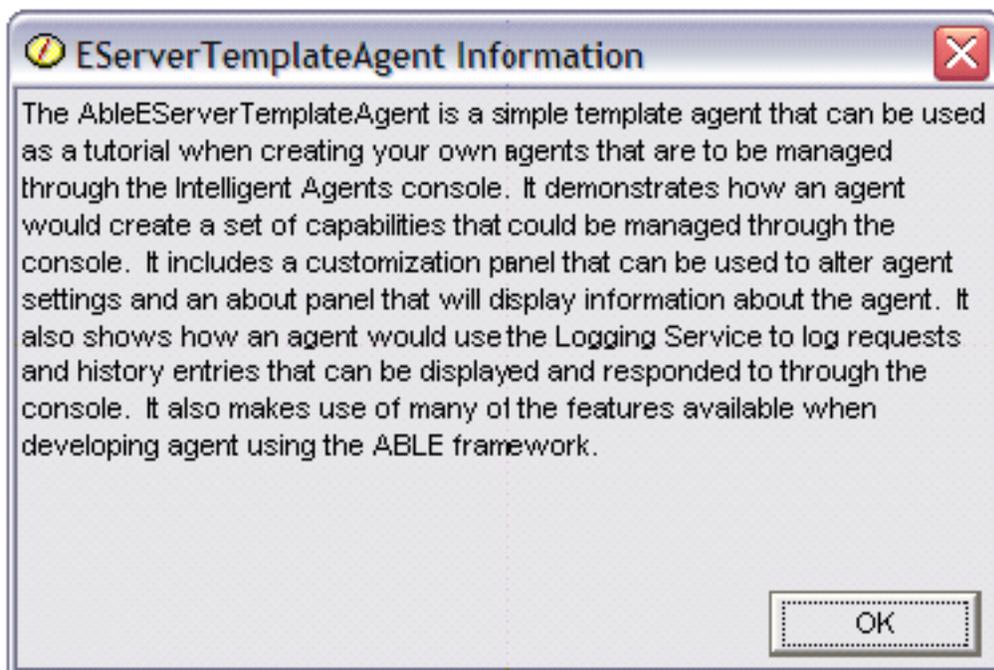


図 2: テンプレート・エージェントの情報パネルの表示

## エージェント環境のセットアップ

iSeries<sup>TM</sup> ナビゲーターのインテリジェント・エージェント・コンソールは、システム上、または分散ネットワーク全体で実行するエージェント・プラットフォームに接続することによって機能します。71 ページの『エージェント・プラットフォーム』はエージェント・プール (JVM) を定義します。その中でエージェント・サービスとエージェントが実行することになります。エージェント・プラットフォームのセットアップを開始する前に、セキュリティー設定を決定する必要があります。セキュア・プラットフォームでは、Kerberos を構成することが必要です。次のトピックでは、エージェント・プラットフォームのセットアップ、およびセキュリティーの構成のためのステップを詳しく説明しています。

### 『エージェント・プラットフォームの構成』

iSeries ナビゲーターでインテリジェント・エージェント・コンソールを使用する前に、まずエージェント・プラットフォームを構成することが必要です。

### 79 ページの『エージェント環境の保護』

セキュア・プラットフォーム、または分散プラットフォーム上の、あるいはそれを介するユーザー、エージェント・プール、およびエージェント・サービスを相互に認証するため、Kerberos ユーザーおよびサービス・プリンシパルを使用することを強くお勧めします。

### 84 ページの『エージェント・プラットフォームの開始』

エージェント・プラットフォームを定義し、オプションでプラットフォームを保護したら、その後、iSeries CL コマンドを使用して、エージェント・サービスと関連するすべての Java<sup>TM</sup> 仮想マシンを開始する必要があります。

**エージェント・プラットフォームの構成:** このトピックでは、エージェント・プラットフォームを簡潔に概説してから、プラットフォーム・プリファレンス・ファイルを変更するための構成ステップを詳細に説明します。

**エージェント・プラットフォームの概要:** インテリジェント・エージェント・コンソールを使用してエージェントを管理するには、まずコンソールの接続先になるエージェント・プラットフォームを定義、保護、および開始することが必要です。エージェント・プラットフォームとは、プラットフォームのサービスまたはエージェントを実行する Java<sup>TM</sup> 仮想マシン、つまりエージェント・プールに過ぎません。プラットフォームの定義には、**ableplatform.preferences** および **able.preferences** ファイルを使用します。

セキュリティをオフにした最も単純な形式の場合、**ableplatform.preferences** は以下の事柄を定義します。

- 各プールのロケーション (システムおよびポート)。
- プラットフォームで実行するサービス。
- プラットフォームで実行を許可されるエージェント。

いったんエージェント・プラットフォームをセットアップすると、プラットフォーム上またはプラットフォームを介して実行するサービスにより、エージェントは固有の名前を受け取ったり、ディレクトリー中の他のエージェントを検索したり、ヒストリーまたは要求をログ記録したり、メッセージをやり取りしたり、エージェントの状態を制御したりすることができるようになります。分散プラットフォームの概念、および使用可能なエージェント・サービスの詳細については、71 ページの『エージェント・プラットフォーム』にある概念の説明部分を参照してください。

**エージェント・プラットフォームの定義:** プラットフォームの構成を開始するには、ファイル **ableplatform.preferences** を変更することにより、エージェント・プール、エージェント・サービス、許可エージェントを定義し、Kerberos セキュリティ・プリンシパルを追加する必要があります。

**ableplatform.preferences** のデフォルト位置は、**QIBM/ProdData/OS400/Able** です。

注: 複数のプラットフォームの構成が可能です。プラットフォームが、同じポートを使用する既存のプラットフォームと同じ位置にないことを必ず確認する必要があります。詳しくは、84 ページの『エージェント・プラットフォームの開始』のトピックを参照してください。

次のコード例は **ableplatform.preferences** から取られたもので、プラットフォーム設定を変更する方法の例を示します。

注: ファイルを開いて内容に変更を加えるにあたっては、小さな誤りやつづりの誤りでエージェントのプラットフォームに障害が起こることと、現時点では間違いをデバッグする簡単な方法がないことを理解しておく必要があります。未使用のプロパティをコメント化しないようにしてください。未使用のプロパティをコメント化すると、プラットフォームに障害が起こることがあります。たとえば、セキュリティをオフにしてプラットフォームを実行することにした場合、ファイルを介してプリンシパル・プロパティをコメント化しないでください。

## 1. エージェント・プールの定義

プラットフォームは、分散 Java 仮想マシンのセットに過ぎません。各 JVM はエージェント・プールと呼ばれ、各 JVM つまりプールは、複数のサービスおよびエージェントをホスティングすることができます (エージェント・プールはサービスをホスティングしなければならないというわけではなく、単にエージェントを実行するために使用することもできます)。プリファレンス・ファイルに IP アドレス (完全修飾されたシステム名) とポートを指定することによって、各 Java 仮想マシン (エージェント・プール) の位置を定義する必要があります。また、各エージェント・プールごとに別名 (任意の固有の名前) を指定します。セキュリティをオンにする場合、サービス・プリンシパルを各エージェント・プールに関連付けることが必要です。Kerberos サービス・プリンシパルについて詳しくは、79 ページの『エージェント環境の保護』のトピックを参照してください。以下は、エージェント・プールの集合を定義する方法の例です。

```

#-----
# Java Virtual Machines
#-----
AgentPool.1.Alias      = Pool1
AgentPool.1.Address   = systemname.ibm.com
AgentPool.1.Port      = 55551
AgentPool.1.Principal = servicePrincipal1

AgentPool.2.Alias      = Pool2
AgentPool.2.Address   = systemname.ibm.com
AgentPool.2.Port      = 55552
AgentPool.2.Principal = servicePrincipal1

AgentPool.3.Alias      = Pool3
AgentPool.3.Address   = systemname.ibm.com
AgentPool.3.Port      = 55553
AgentPool.3.Principal = servicePrincipal2
#-----

```

## 2. エージェント・サービスの定義

プラットフォームで実行するエージェント・サービスを定義し、それらを実行するエージェント・プールの別名を指定します。各エージェント・サービスはファクトリーを指定する必要があります。ファクトリーは、エージェント・サービスを作成する Java クラスです。プラットフォームを直前の状態で再始動するには、持続性サービスを使用します。持続性をオンにするかオフにするかを指定してください。持続性をオンにする場合、Database、Table、および Schema を指定して、持続性サービスがバックアップ・データを保管するための場所を用意しなければなりません。また、PersistenceRetry プロパティに値を指定することもできます。持続性サービスが失敗し、PersistenceRetry プロパティに値 5000 と指定しておいた場合、再試行が 5000 ミリ秒ごとに実行されます。以下のコード例では、ディレクトリー・サービス、ロギング・サービス、および持続性サービスという 3 つの異なるサービスを定義する例を示します。

```
Services=Agent-Directory-Service,Agent-Logging-Service, Persistence-Service
```

```

Agent-Directory-Service.AgentPool      = Pool1
Agent-Directory-Service.Factory        = com.ibm.able.platform.RMIVerifiableDirectoryServiceFactory
Agent-Directory-Service.Persistence   = off
Agent-Directory-Service.PersistenceDatabase = *LOCAL
Agent-Directory-Service.PersistenceTable   = qahadir
Agent-Directory-Service.PersistenceSchema = QUSRSYS
Agent-Directory-Service.PersistenceRetry  = 5000

Agent-Logging-Service.AgentPool      = Pool1
Agent-Logging-Service.Factory        = com.ibm.able.platform.RmiAgentLoggingServiceFactory
Agent-Logging-Service.Persistence   = off
Agent-Logging-Service.PersistenceDatabase = *LOCAL
Agent-Logging-Service.PersistenceTable   = qahalog
Agent-Logging-Service.PersistenceSchema = QUSRSYS
Agent-Logging-Service.PersistenceRetry  = 5000
Agent-Logging-Service.Properties      = history-log-max : 100

```

**注:** ロギング・サービスに history-log-max プロパティを追加して、パフォーマンスを制御するよう指定することができます。 history-log-max=100 を指定すると、各エージェントは過去 100 件の履歴・ログしか保持しなくなります。

```

Persistence-Service.AgentPool      = Pool1
Persistence-Service.Factory        = com.ibm.able.platform.RmiPlatformPersistenceServiceFactory
Persistence-Service.Properties   =
persistence-driver : com.ibm.db2.jdbc.app.DB2Driver,
persistence-protocol : jdbc,
persistence-subProtocol : db2,
blob-type : BLOB,
persistence-dbFlushTime : 1000,
persistence-dbResetAll : off

```

持続性サービスは、エージェント・プラットフォームのためにバックアップとリカバリーを提供します。エージェント・サービスがプラットフォーム上またはプラットフォームを介して実行している場合、持続性を使用するには、いくつかの `Persistence-Service.Properties` を定義する必要があります。

- **persistence-driver**

持続性サービスが使用する JDBC ドライバーを定義します。デフォルトでは、`persistence-driver` はネイティブの DB2<sup>(R)</sup> ドライバーを使用するように設定されています。

- **persistence-protocol および subProtocol**

持続性サービスが使用するデータベース・プロトコルを定義します。デフォルトでは、`protocol` は `jdbc` に、`subProtocol` は `db2` に設定されています。

- **blob-type**

使用中の JDBC ドライバーと関連した BLOB タイプを定義します。DB2 のデフォルトは BLOB に設定されていますが、別のデータベース (Cloudscape など) を使用することにした場合は、BLOB タイプを `blob-type : LONG VARBINARY` と定義できるかもしれません。

- **persistence-dbFlushTime**

持続性サービスがデータベースにデータをフラッシュする頻度を、ミリ秒単位で指定します。

- **persistence-dbResetAll**

このプロパティを `on` に指定すると、プラットフォームの再始動時、前に残しておいたデータが全部データベースから消去されます。

### 3. 許可エージェントを定義する

プラットフォーム、およびプラットフォーム上またはプラットフォームを介して実行されるエージェント・サービスへのアクセスを許可するエージェントをすべて定義する必要があります。以下は、エージェントを定義する方法の例です。例の後に、各エージェント・プロパティの詳細を示します。

```
Agent.1.Alias=Agent1
Agent.1.AutonomyLevel=Medium
Agent.1.ClassName=com.ibm.able.platform.examples.EServerTemplateAgent
Agent.1.ConstructorArgs=String:agentName
Agent.1.EligiblePrincipals=principalAlias1, principalAlias2
Agent.1.EligibleAgentPools=pool1, pool2, pool3
Agent.1.InitArgs=
Agent.1.LastChangedDate=January 11, 2003 11:11am
Agent.1.Type=Tester1
Agent.1.Vendor=IBM1
Agent.1.Version=1.1
```

- **Alias**

エージェントに固有の名前を指定します。エージェント・コンソールでこの名前が使用されることになります。

- **AutonomyLevel**

エージェントの初期自律レベルを指定します。ユーザーは、この設定をコンソールから変更できません。エージェントと関連付ける独立性のレベルを決定し、それに従って自動化レベルを設定します。設定する自動化レベルが高ければ高いほど、エージェントがアクション実行の許可を要求する回数が少なくなります。エージェントを「**高レベル自動化**」に設定すると、ほとんどのアクションは、最初に応答を求めることなく実行されます。エージェントの動作が心配な場合は、設定を「**中レベル自動化**」に変更し、自動化レベルを下げる (エージェントがアクション実行の許可を要求する頻度が増える) と良いかもしれません。

- **ClassName**

実際のエージェント Java クラスを指定します。

- **ConstructorArgs**

エージェントに渡す引数をプロパティ・ファイルに指定できます。

- **EligiblePrincipals**

セキュリティーをオンにする場合、各エージェントに 1 つ以上のユーザー・プリンシパル別名を関連付けることにより、エージェントのインスタンスを開始する権限を持つ人を定義する必要があります。Kerberos サービス・プリンシパルについて詳しくは、『エージェント環境の保護』のトピックを参照してください。

- **EligibleAgentPools**

プラットフォームでエージェントを実行するのに使用する、1 つ以上のエージェント・プールの別名を指定します。

- **InitArgs**

これにより、プリファレンス・ファイルから Init 引数をエージェントに渡すことができます。

#### 4. エージェント・プラットフォームの保護

エージェント・プール、エージェント・サービス、および許可エージェントを定義した後、プラットフォームでセキュリティーを構成する必要があるかもしれません。Kerberos プリンシパル、信用レベル、およびエージェント・プラットフォームを保護するためにそれらを使用および定義する方法について詳しくは、『エージェント環境の保護』を参照してください。

エージェント・プール、エージェント・サービス、および許可エージェントを定義し、オプションでセキュリティーをセットアップしたら、次に 84 ページの『エージェント・プラットフォームの開始』する必要があります。

**エージェント環境の保護:** プラットフォーム・セキュリティーは、オン/オフを切り替えられます。セキュリティーをオフにしたプラットフォーム上、またはそのプラットフォームを介して実行する場合、誰でも他人のエージェント記述の登録を取り消したり、変更したりできるようになります。誰でも、任意のエージェントの機能や状態を変更できるようになります。誰もが、任意の要求を除去したり、応答したりできるようになります。これは、要求が自分自身のものでない場合にもあてはまります。エージェントの使用法を間違えたり、間違ったユーザーがこれを使用したりすると、破壊アクションに至る恐れがあります。エージェントが意図した方法で確実に使用されるように、プラットフォームのインフラストラクチャーにはセキュリティー機能が追加されています。

セキュリティーをオンにした場合、エージェントおよびサービスは、プラットフォーム上、またはプラットフォームを介して実行されるすべてのアクションを認証、および許可できます。エージェントは、自分自身のエージェント記述を登録取り消しまたは変更することしかできません。エージェントはすべての応答済み要求と機能変更を許可することが必要になります。エージェントの状態を変更するためには特定の権限レベルが必要です。エージェントの使用を、特定のユーザーと位置に限定することができます。セキュリティーをオンにすると、発生するすべてのアクションが既知のユーザーにトレースバックされるので、プラットフォーム認証および許可が行えます。

エージェント・プラットフォームを保護する場合、プラットフォームを定義する **able.preferences** ファイルで、Security プロパティを **Security=on** に変更することによって、セキュリティーをオンにできます。

セキュリティーをオンにする前に、以下のステップが実行されていることを確認し、構成する必要があります。

##### 1. 80 ページの『Kerberos を使用するためのプラットフォームの構成』

インテリジェント・エージェント・プラットフォームでは、エージェント・プラットフォーム上、またはエージェント・プラットフォームをまたいだユーザーおよびサービスを認証するために、Kerberos プリンシパルを使用します。プリンシパルの認証は、鍵配布センター (KDC) と呼ばれる中央サーバーを介して行われます。V5R3 では、ネイティブ Kerberos KDC が iSeries<sup>(TM)</sup> 上で提供されます。

## 2. 83 ページの『プラットフォーム・セキュリティの構成』

セキュリティをオンにすると、**ableplatform.preferences** は、自分が定義するプラットフォームのセキュリティのポリシー・ファイルとなります。以下のトピックに、プリンシパル、信用レベル、および許可の構成のためのステップを示します。

**Kerberos を使用するためのプラットフォームの構成:** インテリジェント・エージェント・プラットフォームでは、エージェント・プラットフォーム全体のユーザーおよびサービスを認証するために、Kerberos プリンシパルを使用します。Kerberos プロトコル (Massachusetts Institute of Technology が開発) により、プリンシパル (ユーザーまたはサービス) は自分の身元証明を、保護されていないネットワーク内の他のサービスに対して行うことができます。プリンシパルの認証は、鍵配布センター (KDC) と呼ばれる中央サーバーを介して行われます。KDC は、Kerberos チケットを使ってユーザーを認証します。これらのチケットにより、プリンシパルの身元がネットワーク内の他のサービスに対して証明されます。これらのチケットでプリンシパルの認証が済むと、暗号化されたデータをターゲット・サービスと交換できるようになります。

プラットフォームでは、Kerberos を使ってユーザー・サインオンと初期プラットフォームの開始を認証します。Kerberos を使ってプラットフォームを保護するには、既存の KDC を検出するか、プラットフォーム全体が使用する作業 KDC を作成する必要があります。プラットフォームの一部を実行するすべてのシステムと、このプラットフォームに接続するコンソールを実行するすべての PC が、この KDC を使用できるように構成する必要があります。すべての Kerberos プリンシパルを **ableplatform.preferences** ファイルにリストする必要があります。プラットフォームは、ユーザーとサービスを認証するためにこのファイルを使用します。各プラットフォームの Java<sup>(TM)</sup> 仮想マシン (エージェント・プール) には、サービス・プリンシパルが関連付けられます。コンソールからプラットフォームにログオンする各ユーザーには、ユーザー・プリンシパルが必要です。これらのすべてのプリンシパルを KDC に追加する必要があります。

### 1. 使用可能な Kerberos 鍵配布センター (KDC) を検索または作成する

エージェント・プラットフォームでは、KDC が OS/400<sup>(R)</sup> 上になくても構いません。KDC は、どのプラットフォームで稼働していても機能します。使用できる既存の KDC が見つからない場合、独自の KDC を作成できます。V5R3 の OS/400 では、OS/400 PASE でネイティブの Kerberos サーバーがサポートされています。Kerberos サーバーの構成と管理は、iSeries<sup>(TM)</sup> システムから行えます。OS/400 PASE で Kerberos サーバーを構成するには、以下のタスクを完了する必要があります。

- 文字ベースのインターフェースで、**call QP2TERM** と入力します。このコマンドにより、OS/400 PASE アプリケーションを処理するための対話式シェル環境が開きます。
- コマンド行で、**export PATH=\$PATH:/usr/krb5/sbin** と入力します。このコマンドは、実行可能ファイルを実行するのに必要な Kerberos スクリプトを指定します。
- コマンド行で、**config.krb5 -S -d iseriesa.myco.com -r MYCO.COM** と入力します。このコマンドは、krb5.config ファイルに Kerberos サーバーのドメイン名とレルムの更新を加え、統合ファイル・システムに Kerberos データベースを作成し、OS/400 PASE で Kerberos サーバーを構成します。データベースのマスター・パスワードと、Kerberos サーバーの管理に使用される管理/管理プリンシパルのパスワードを追加するよう、プロンプトが出されます。
- コマンド行で、**/usr/krb5/sbin/start.krb5** と入力し、サーバーを開始します。

iSeries で KDC を構成する方法については、OS/400 PASE での Kerberos サーバーの構成 (Configure a Kerberos server in OS/400 PASE) を参照してください。

### 2. Kerberos を使用するためにエージェント環境のシステムを構成する

Kerberos サーバー (KDC) の作成が済んだら、この Kerberos サーバーを使用するよう、セキュア・プラットフォームへの接続を試みることになる各クライアント PC と、エージェント・プラットフォーム内の各 iSeries システムとを、個別に構成しなければなりません。

#### • クライアント PC を構成する

クライアント PC を構成するには、iSeries ナビゲーターのインテリジェント・エージェント・コン

ソールを実行する JVM のセキュリティー・フォルダーに、**krb5.conf** というテキスト・ファイルを作成することが必要です。場所は以下のとおりです (c: はクライアント・アクセス・ドライバーのインストール先ドライブ)。

C:\Program Files\IBM\Client Access\JRE\Lib\Security

. **krb5.conf** ファイルは、この JRE から開始したすべての JVM に対して、Kerberos を処理するときを使用すべき KDC を通知します。以下の例に、KDC\_REALM.PASE.COM というレルムが system1.ibm.com にある場合に、一般的な **krb5.conf** がどうなるかを示します。

```
[libdefaults]
default_realm = KDC_REALM.PASE.COM
default_tkt_enctypes = des-cbc-crc
default_tgs_enctypes = des-cbc-crc

[realms]
KDC_REALM.PASE.COM = {
    kdc = system1.rchland.ibm.com:88
}

[domain_realm]
.rchland.ibm.com = KDC_REALM.PASE.COM
```

#### • iSeries システムを構成する

iSeries システムが使用する KDC を指定するには、以下のファイルに変更を加えることが必要です。

/QIBM/userdata/OS400/networkauthentication/**krb5.conf**

**krb5.conf** ファイルは、この JRE から開始したすべての JVM に対して、Kerberos を処理するときを使用すべき KDC を通知します。以下の例に、KDC\_REALM.PASE.COM というレルムが system1.ibm.com にある場合に、一般的な **krb5.conf** が iSeries でどうなるかを示します。

```
??(libdefaults??)
    default_realm = KDC_REALM.PASE.COM
??(appdefaults??)
??(realms??)
    KDC_REALM.PASE.COM = {
        kdc = system1.rchland.ibm.com:88
    }
??(domain_realm??)
    system1.rchland.ibm.com = KDC_REALM.PASE.COM
```

作成した KDC を iSeries に使用させる方法の詳細については、ネットワーク認証サービスを構成するを参照してください。

### 3. Kerberos ユーザーおよびサービス・プリンシパルを獲得する

KDC の構成が済んだら、プラットフォームの保護に使用する予定のユーザーおよびサービス・プリンシパルを作成し、これらのプリンシパルを KDC に登録することが必要になります。

#### サービス・プリンシパル:

**ableplatform.preferences** で定義される各エージェント・プール (JVM) には、サービス・プリンシパルを関連付ける必要があります。サービス・プリンシパルは稼働するシステムに特有なもので、そのシステム名を含めて **ServicePrincipalName/systemName@KDCRealm** という形式にすることが必要です。プラットフォーム上の各エージェント・プールが同じサービス・プリンシパルを使用するようにすることもできますし、各プールが独自のサービス・プリンシパルを使用するように指定することもできます。各エージェント・プールの権限レベルが異なる場合は、各権限レベルに合わせて別々のプリンシパルを使用するべきです。

### ユーザー・プリンシパル:

コンソールからセキュア・プラットフォームへ接続することを許可される各ユーザーには、ユーザー・プリンシパルが必要です。ユーザー・プリンシパルは、**ableplatform.preferences** にリストされる各エージェント定義と関連付けることができます。ユーザー・プリンシパルは、コンソールが実行するシステムに関係なく、コンソールからプラットフォームに接続できます。このため、ユーザー・プリンシパルには、プリンシパル名と、プリンシパルが所属する KDC レalmを含めるだけで構いません。したがって、**UserPrincipalName@KDCRealm** となります。

プラットフォームが使用することになる Service および User プリンシパルごとに、プリンシパルを 1 つ KDC に追加する必要があります。ネイティブの KDC を iSeries で使用する場合、次のステップでプリンシパルを KDC に追加できます。

- a. 文字ベースのインターフェースで、**call QP2TERM** と入力します。
- b. コマンド行で、**export PATH=\$PATH:/usr/krb5/sbin** と入力します。このコマンドは、実行可能ファイルを実行するのに必要な Kerberos スクリプトを指定します。
- c. コマンド行に、**kadmin -p admin/admin** と入力して **Enter** を押します。
- d. 管理者のパスワードを使ってサインインします。
- e. コマンド行で、以下のように入力します。
  - iSeries で実行するプールのサービス・プリンシパルを追加する場合:  
**addprinc -pw secret servicePrincipalName/iSeries fully qualified host name@REALM**
  - ユーザー・プリンシパルを追加する場合:  
**addprinc -pw secret jonesm**。これで、ユーザーがコンソールからログインするためのプリンシパルが作成されます。
  - PC で実行するプールのサービス・プリンシパルを追加する場合:  
**addprinc -requires\_preauth -e des-cbc-crc:normal -pw host/pc1.myco.com**

iSeries でネイティブ KDC を使用する場合、KDC にプリンシパルを追加する方法についての詳細を以下のトピックで参照してください。

iSeries で実行するプールのサービス・プリンシパルを追加する場合:

OS/400 プリンシパルを Kerberos サーバーに追加する (Add OS/400 principals to the Kerberos server)

PC で実行するプールの User プリンシパルまたは Service プリンシパルを追加する場合:

Windows 2000 ワークステーションおよびユーザー用のホスト・プリンシパルを作成する (Create Host principals for Windows 2000 workstations and users)

#### 4. サービス・プリンシパルを各 keytab ファイルに追加する

セキュア・プラットフォームを開始するとき、各エージェント・プールは、開始用に定義されたプリンシパルを使用します。また、自分自身を認証するためにもこのプリンシパルを使用します。そのためには、各プール JVM が、使用するプリンシパル用の有効な Kerberos 証明書にアクセスできなければなりません。iSeries の STRAGTSRV コマンドはこれを処理します。ただし、keytab ファイルに使用するプリンシパルの項目がなければなりません。各プラットフォーム・システムで実行する予定のサービス・プリンシパルごとに keytab ファイルにエントリを追加するには、以下のステップを実行します。

iSeries でネイティブ KDC を実行する場合:

- a. 文字ベースのインターフェースで、**STRQSH** と入力します。このコマンドにより、qsh shell インタープリターが開始します。
- b. 以下のコマンドを入力します (ServicePrincipal は追加するサービス・プリンシパルの名前、system@KDCRealm は完全に修飾された iSeries システム名および Kerberos レalm、および

thePassword はサービス・プリンシパルと関連するパスワード)。

```
keytab add ServicePrincipal/system@KDCRealm -p thePassword
```

KDC のセットアップと、ユーザー・プリンシパルおよびサービス・プリンシパルの作成が済んだら、『プラットフォーム・セキュリティの構成』が必要です。

**プラットフォーム・セキュリティの構成:** 始める前に、80 ページの『Kerberos を使用するためのプラットフォームの構成』を済ませておいてください。

セキュリティをオンにすると、**ableplatform.preferences** は、自分が定義するプラットフォームのセキュリティのポリシー・ファイルとなります。以下のステップは、プリンシパル、信用レベル、および許可を構成する方法の例です。

### 1. ユーザーおよびサービス・プリンシパルを定義する

ユーザーおよびサービス・プリンシパルを獲得して KDC に登録したら、これらのプリンシパルを **ableplatform.preferences** に追加する必要があります。セキュリティがオンの場合、ユーザーは有効な Kerberos ユーザー・プリンシパルで定義されないとプラットフォームにアクセスできません。また、すべてのエージェント・サービスおよびエージェント・プールには、有効な Kerberos サービス・プリンシパルを割り当てる必要があります。KDC に登録したユーザーまたはサービス・プリンシパルを追加し、各プリンシパルごとに別名を指定します (別名には、任意の固有の名前を使用できます)。

```
#-----  
# Principals  
#-----  
Principal.1.Alias      = servicePrincipal1  
Principal.1.Principal = name1/systemName@REALM  
  
Principal.2.Alias      = servicePrincipal2  
Principal.2.Principal = name2/systemName@REALM  
  
Principal.3.Alias      = userPrincipal1  
Principal.3.Principal = name1@REALM  
  
Principal.4.Alias      = userPrincipal2  
Principal.4.Principal = name2@REALM
```

### 2. 信用レベルを定義する

ユーザーおよびサービス・プリンシパルを追加したら、各プリンシパルに関連した信用レベルを定義する必要があります。信用レベルは、プラットフォーム上のユーザーまたはサービスの機能を定義するのに役立つために、プリンシパルと関連付けられます。信用レベルをプリンシパルと関連付けることは、プリンシパルをグループ化する方法でもあります。同じ信用レベルを、複数のユーザーおよびサービス・プリンシパルと関連付けることができます。ステップ 1 でサービスおよびユーザー・プリンシパルに割り当てていたプリンシパル別名を、関連付けたい信用レベルに追加し (コンマ区切り)、信用レベル別名として固有の名前を指定します。

```
#-----  
# Trust Levels  
#-----  
TrustLevel.1.Alias      = HighlyTrusted  
TrustLevel.1.Principals = servicePrincipal1,userPrincipal1  
  
TrustLevel.2.Alias      = SomewhatTrusted  
TrustLevel.2.Principals = servicePrincipal2,userPrincipal2
```

### 3. サービス・プリンシパルをエージェント・プールに関連付ける

分散プラットフォームは、複数システムの複数ポートにまたがる場合があります。各エージェント・プールは、一部 (Java<sup>TM</sup> 仮想マシン) またはプラットフォームがどこで実行されるかを定義します。各エージェント・プール項目には、別名、IP アドレス、ポート、およびサービス・プリンシパル別名が入

っています。プリンシパル別名は、このプールが関連付けられるサービス・プリンシパルを指定します。上記で定義した、エージェント・プールと関連付けたいサービス・プリンシパル別名を追加します。

```
#-----  
# Agent Pools (Java Virtual Machines)  
#-----  
AgentPool.1.Alias      = Pool1  
AgentPool.1.IPAddress = systemname.ibm.com  
AgentPool.1.Port       = 55551  
AgentPool.1.Principal = servicePrincipal1  
  
AgentPool.2.Alias      = Pool2  
AgentPool.2.IPAddress = systemname.ibm.com  
AgentPool.2.Port       = 55552  
AgentPool.2.Principal = servicePrincipal1  
  
AgentPool.3.Alias      = Pool3  
AgentPool.3.IPAddress = systemname.ibm.com  
AgentPool.3.Port       = 55553  
AgentPool.3.Principal = servicePrincipal2
```

#### 4. エージェント開始権限を定義する

セキュア・プラットフォームで定義された各エージェントを開始できるユーザーは誰かを定義します。EligiblePrincipal パラメーターに、1 つ以上のユーザー・プリンシパル別名を追加します。

```
#-----  
# Permitted Agents  
#-----  
Agent.1.Alias=Agent1  
Agent.1.AutonomyLevel=Medium  
Agent.1.ClassName=com.ibm.able.platform.examples.EServerTemplateAgent  
Agent.1.ConstructorArgs=String:AgentName1  
Agent.1.EligiblePrincipals=userPrincipal1,userPrincipal2  
Agent.1.EligibleAgentPools=Pool2,Pool3  
Agent.1.InitArgs=  
Agent.1.LastChangedDate=January 11, 2003 11:11am  
Agent.1.Type=Tester1  
Agent.1.Vendor=IBM1  
Agent.1.Version=1.1
```

#### 5. アルゴリズムおよびプロバイダーを定義する

プラットフォームが使用する KeyPairs のアルゴリズムおよびプロバイダーを定義する必要があります。デフォルトでは、プリファレンス・ファイルに次の設定が入っています。

```
#-----  
# Cryptography parameters  
#-----  
CryptographyAlgorithm = DSA  
CryptographyProvider  = IBMJCE
```

必要なセキュリティー・データを **ableplatform.preferences** に追加したら、変更内容を保管します。プラットフォームを正しく構成した後、そのセキュリティーをオンにするのは簡単です。プラットフォームを定義した **able.preferences** を開き、Security プロパティーを **Security=on** にするだけです。非セキュア・プラットフォームを実行するには、『エージェント・プラットフォームの開始』して、セキュリティーの変更を有効にしなければなりません。

**エージェント・プラットフォームの開始:** 75 ページの『エージェント・プラットフォームの構成』と、オプションで 79 ページの『エージェント環境の保護』の構成が済んだら、エージェント・プラットフォームを開始する必要があります。プラットフォームは 1 つ以上の Java<sup>TM</sup> 仮想マシンで構成されているので、プラットフォームを開始するためには、プラットフォームを構成するすべての JVM を開始することが必要です。

以下に、iSeries<sup>(TM)</sup> でエージェント・プラットフォームを開始する方法を説明します。

**iSeries でのエージェント・プラットフォームの開始:** 以下のコマンドは、iSeries 上でエージェント・プラットフォームの開始および停止の処理をします。

**STRAGTSRV** (エージェント・サービスの開始) および **ENDAGTSRV** (エージェント・サービスの終了)

- **STRAGTSRV** (エージェント・サービスの開始)

STRAGTSRV を実行すると、コマンド実行元のシステムで、各プールごとに個別の JVM が開始します。このコマンドを実行すると、QAHASBMTER ジョブが開始します。このジョブは、開始する必要があるすべての JVM (エージェント・プール) を検索します。エージェント・プールが発見されると、プールごとに個別の QAHASBMTEE ジョブが開始します。コマンドが正常に完了すると、QSYSWRK に、各プールごとに個別の SIGW 状態の QAHAPLTFRM ジョブがあるはずですが、STRAGTSRV のキーワードは次のとおりです。

**PREFDIR**

ファイル `able.preferences` および `ableplatform.preferences` の場所を設定します。PREFDIR パラメーターをデフォルト値のままにすると、`/QIBM/ProdData/OS400/able/` ディレクトリーの `ableplatform.preferences` および `able.preferences` ファイルで定義されるプラットフォームが開始または終了します。PREFDIR パラメーターを使ってさまざまなディレクトリーを指すことにより、同じシステムで複数のプラットフォームを開始できます。これを行う場合、2 つのプラットフォームがシステムの同一ポートを使用してオーバーラップしないよう、注意する必要があります。

**HOMEDIR**

ホーム・ディレクトリーの位置を設定します。

**CIASSPATH**

各 JVM が含める必要のある付加的なクラスパスを追加できます。STRAGTSRV は、デフォルトのクラスパスを自動的に以下のように設定します。

```
classpath=
/QIBM/ProdData/Java400/:/qibm/proddata/os400/able:
/qibm/proddata/os400/able/ableplatform.jar:
/qibm/proddata/os400/able/able.jar:
/qibm/proddata/os400/able/ablebeans.jar:
/qibm/proddata/os400/able/jas.jar:
/qibm/proddata/os400/able/Jlog.jar:
/qibm/proddata/os400/Java400/ext/ibmjgssiseriesprovider.jar:
/qibm/proddata/os400/jt400/lib/jt400Native.jar:
/qibm/proddata/os400/Java400/ext/db2_classes.jar:
/qibm/proddata/os400/able/auifw.jar:
```

**SBMJOBUSER**

コマンドの呼び出しに使用している現在のプロファイルとは異なるプロファイルで JVM (エージェント・プール) を実行します。

**PoolIdentifier**

異なるプロファイルで実行する必要があるプール。

ユーザー・プロファイル

エージェント・プールを開始するのに使用するプロファイル (PoolIdentifier)。

- **ENDAGTSRV** (エージェント・サービスの終了)

ファイル `ableplatform.preferences` でエージェント・プールとして指定されている、このシステム上のすべてのプラットフォーム JVM を終了します。このコマンドを実行すると、QAHAPLTEND ジョブが開始します。このジョブは、すべてのエージェント・プールを検索して、終了させます。ENDAGTSRV のキーワードは次のとおりです。

## PREFDIR

ファイル `able.preferences` および `ableplatform.preferences` の場所を設定します。 `PREFDIR` パラメーターをデフォルト値のままにすると、`/QIBM/ProdData/OS400/able/` ディレクトリーの `ableplatform.preferences` および `able.preferences` ファイルで定義されるプラットフォームが開始または終了します。 `PREFDIR` パラメーターを使ってさまざまなディレクトリーを指すことにより、同じシステムで複数のプラットフォームを開始できます。これを行う場合、2 つのプラットフォームがシステムの同一ポートを使用してオーバーラップしないよう、注意する必要があります。

注: エージェント・プラットフォームの開始または終了時に問題が生じる場合、`QAHA_TRACE` システム環境変数を追加して「1」に設定し、起動プログラムのトレースをオンにできます。これにより、`QUSRSYS/QAAHALOG` にログ・ファイルが作成されます。 `QSBR<job number>`、`QSBE<job number>`、および `QEND<job number>` という名前のファイルが、実行した `QAHASBMTER`、`QAHASBMTEE`、および `QAHAPLTEND` ジョブごとに作成されます。

## エージェントの管理

iSeries<sup>TM</sup> のインテリジェント・エージェント・コンソールは、エージェントを扱う仕事をするための強力な管理ツールであり、確実にエージェントが望みどおりの動作をするようにします。 iSeries ナビゲーターでインテリジェント・エージェント・ノードを表示するには、メインメニューから「表示」→「インテリジェント・エージェント」を選択します。

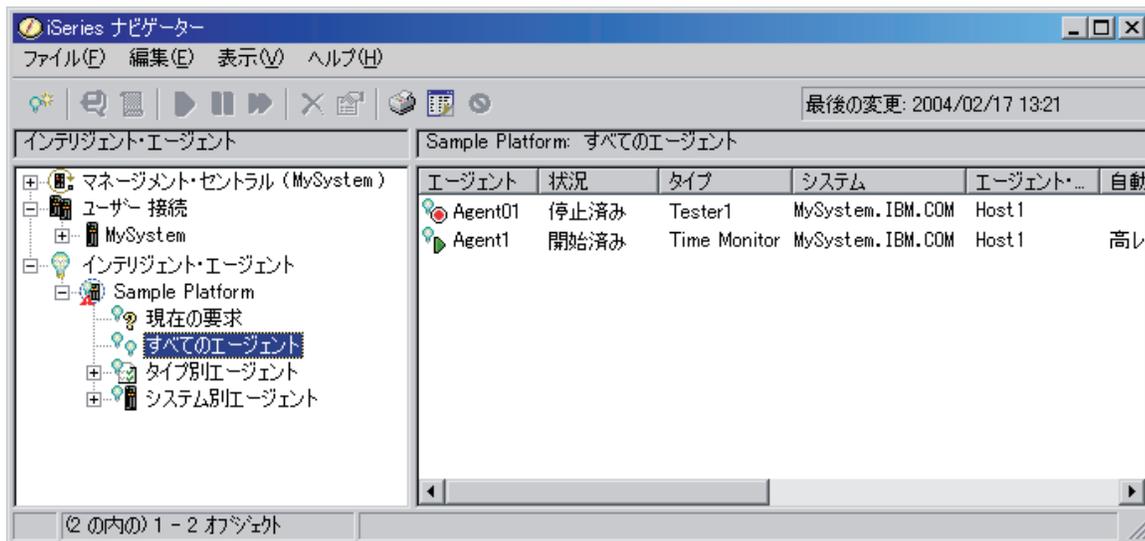


図 1: iSeries ナビゲーターでエージェントを処理する

エージェント環境のセットアップが済んだら、ホスト・システム (複数も可) に接続し、そのシステムで実行するエージェントのインスタンスを作成することによって、エージェント・コンソールを使った作業を開始できます。コンソールを使用して、システム (複数も可) 上で実行するエージェントを開始、停止、中断、削除したり、これに回答したり、その履歴を表示したりすることができます。また、コンソールを使って、エージェントが自動的に実行できるアクションや、許可が必要なアクションの制限をセットアップすることもできます。

### 『エージェントの自動化』

エージェント・コンソールでは、エージェントに自動化のレベルを関連付けることによって、エージェントの動作を制御したり、カスタマイズしたりできます。

### 88 ページの『エージェントの通信』

アクションを実行するにあたって確認または許可を求めるエージェントを追跡して、これに応答することが容易に行えます。

### 89 ページの『エージェントのヒストリー』

エージェント・コンソールは、すべてのエージェント・アクションのヒストリーをログに記録します。

**エージェントの自動化:** インテリジェント・エージェントのコンソールを使うと、エージェントで実行可能な自動化アクションを制御できます。

iSeries<sup>(TM)</sup> ナビゲーターにエージェントの機能を表示し、エージェントの自動化設定を変更するには、次のステップを実行します。

1. 「インテリジェント・エージェント」を展開します。
2. インテリジェント・エージェントのプラットフォームを展開します。
3. 「すべてのエージェント」を選択します。
4. 処理したいエージェントを右マウス・ボタン・クリックし、「プロパティ」を選択します。
5. 「自動化」タブを選択し、現在エージェントで構成されている自動化レベルを表示します。
6. 「機能」をクリックして、エージェントが実行できるアクションのリスト、およびこれらの機能に関連する自動化レベルを表示します。

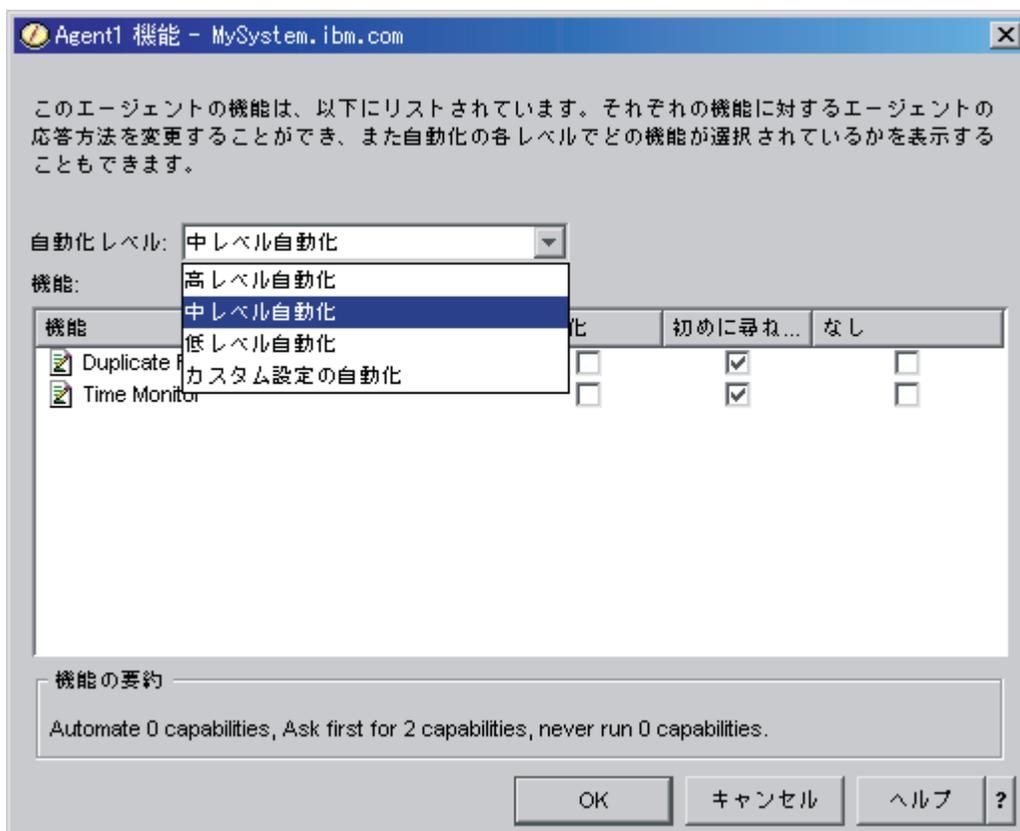


図 1: TimeMonitor エージェントの機能に関連付けられた自動化レベルの表示

どのエージェントにも、自分が実行できるアクションの種類を定義するための機能のセットが揃っています。エージェント・コンソールには、エージェントの対応する自動化レベルに関連した、エージェントの使用可能な機能が表示されます。各自動化レベル設定（「高レベル自動化」、「中レベル自動化」、「低レベル自動化」、および「カスタム設定の自動化」）により、エージェントの使用可能な機能の状態（「自動化」、「初めに尋ねる」、「なし」）が変化します。

たとえば、ログ・ファイルがいっぱいになったら消去する機能がエージェントにある場合、自動化のレベルを「高レベル自動化」から「中レベル自動化」に変更すると、エージェントの機能は「自動化」の状態から「初めに尋ねる」に変化します。すると、エージェントはログ・ファイルを削除する前に許可を求めるようになります。

エージェントの自動化レベルの指定により、エージェントがアクションを自動的に実行するか、アクションの実行前に確認を求めるか、またはアクションを全く実行しないかどうかを決定します。指定可能な自動化の値は次のとおりです。

- 「高レベル自動化」

エージェントはほとんどのアクションを自動的に実行しますが、特定の破壊アクションの実行前には確認を行います。エージェントによっては、「高レベル自動化」に設定されてはいても、一部のアクションを実行する前に、常に外部からの介入を求めるべき場合もあります。

- 「中レベル自動化」

エージェントは一部のアクションを自動的に実行し、その他のアクションについては実行前に確認を行います。エージェントによっては、「中レベル自動化」に設定されてはいても、一部のアクションを実行する前に、常に外部からの介入を求めるべき場合もあります。

- 「低レベル自動化」

エージェントはほとんどのアクションを自動的に実行しません。エージェントは、ほとんどの場合、どのアクションを実行する場合でも、事前に外部からの介入を要求します。

- 「カスタム設定の自動化」

エージェントがアクションを自動実行するか、まず確認するか、実行しないかは、機能の手動構成に従います。

**エージェントの通信:** エージェントの機能に関連した自動化設定を「初めに尋ねる」に設定すると、エージェントはアクションを実行する前にユーザーからの応答を要求します。エージェントによっては、現行の自動化設定に関係なく、常に応答を要求します。エージェントが応答を要求する場合、またはアクションの実行を待機している場合、エージェントの「状況」フィールドに「応答が必要」と表示されます。

iSeries<sup>(TM)</sup> ナビゲーターで内臓のエージェントに応答する方法は次のとおりです。

1. 「インテリジェント・エージェント」を展開します。
2. インテリジェント・エージェントのプラットフォームを展開します。
3. 「すべてのエージェント」を選択します。
4. エージェントを右マウス・ボタン・クリックし、「応答...」を選択します。
5. 処理する応答を選択し、「応答」ボタンをクリックします。
6. 現在応答が必要とされている問題が、エージェントによって表示されます。「応答」フィールドの中の選択可能な応答のリストから応答を選択し、「OK」をクリックします。

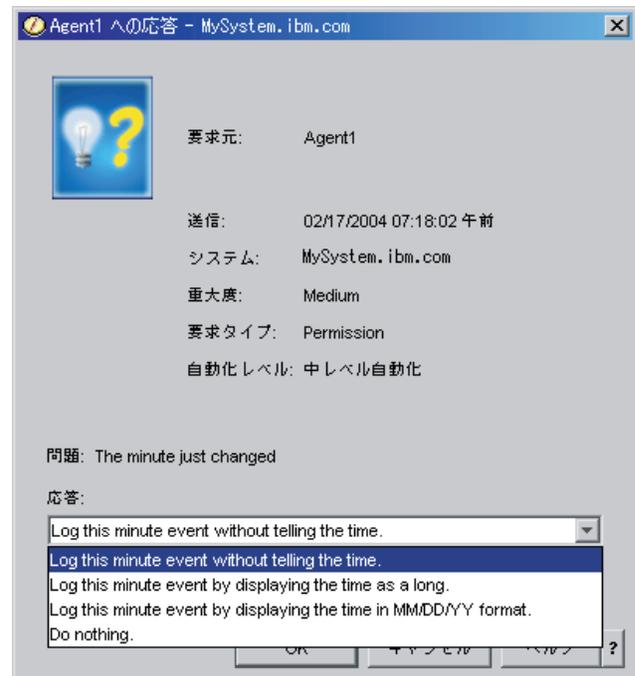


図 1: エージェントの要求に応答する

メインの「インテリジェント・エージェント」メニューから「現在の要求」を選択して、すべての現行の要求のリストを表示することもできます。

**エージェントのヒストリー:** エージェント・コンソールに、エージェントの要求およびアクションのヒストリーを表示できます。ヒストリーには、現行の要求ではなく、応答済みの要求とアクションだけが表示されます。ヒストリー・ログは 1000 項目までに制限されており、1000 を超えると新しい項目が追加されるたびに最も古い項目が消去されます。

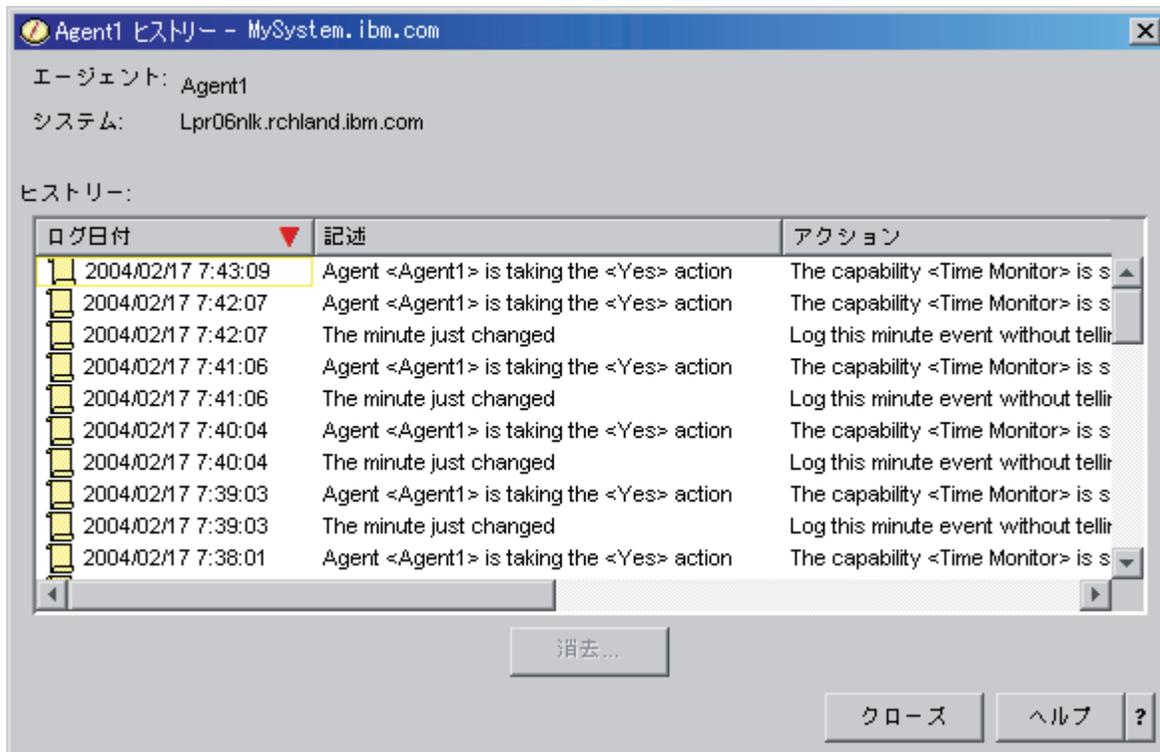


図 1: エージェントの要求とアクションの履歴の表示

iSeries<sup>(TM)</sup> ナビゲーターでエージェントの履歴を表示するには、次のステップを実行します。

1. 「インテリジェント・エージェント」を展開します。
2. インテリジェント・エージェントのプラットフォームを展開します。
3. 「すべてのエージェント」を選択します。
4. 履歴を表示するエージェントを右マウス・ボタン・クリックし、「履歴」を選択します。

## パフォーマンス・データ・ファイル

パフォーマンス・データとは、応答時間およびスループットを理解するために使用できるシステム（システムのネットワーク）の操作に関する一連の情報です。パフォーマンス・データを使用することにより、プログラム、システム属性、および操作に調整を加えることができます。これらの調整を行うことにより応答時間とスループットを向上させることができます。また調整は、システム、操作、またはプログラムへの特定の変更の影響を予測する上で役立ちます。

36 ページの『収集サービス』は、管理収集オブジェクト (\*MGTCOL) にパフォーマンス・データを収集します。パフォーマンス・データの作成 (CRTPFRDTA) コマンドは、その収集オブジェクトからデータを処理し、パフォーマンス・データベース・ファイルに結果を保管します。データベース・ファイルは、以下のカテゴリーに分けられます。

### 時間間隔データを含むパフォーマンス・データ・ファイル

これらのファイルには、それぞれの間隔を収集されるパフォーマンス・データが含まれています。これらのファイルのリストについては、『91 ページの『時間間隔データを含むパフォーマンス・データ・ファイル』』を参照してください。そこには、各ファイルに関する要旨および完全な情報へのリンクも含まれています。これらのファイル内のデータがどこから収集されるかを理解するには、『93 ページの『パフォーマンス・データ・ファイル: 収集サービス・システム・カテゴリーとファイルの

関係』を参照してください。これらのファイルを表示するときに、92 ページの『パフォーマンス・データ・ファイル: ファイルの略語』が役立つことにお気付きになることでしょう。

### 構成データ・ファイル

構成データは、セッションごとに一度収集されます。これらのファイル内のデータがどこから収集されるかを理解するには、『93 ページの『パフォーマンス・データ・ファイル: 収集サービス・システム・カテゴリとファイルの関係』を参照してください。構成データ・ファイル内に QAPMCONF、QAPMHDWR、および QAPMSBSD ファイルを見つけることができます。

### トレース・データ・ファイル

トレース・データは、データのトレースを選択する場合に限り収集されます。トレース・データ・ファイル内に QAPMDMPT を見つけることができます。

バイト数およびバッファ位置などの追加のフィールド情報は、システム上で使用可能なファイル・フィールド記述の表示 (DSPFFD) コマンドを使用すれば、利用可能になります。たとえば、コマンド行で以下のコマンドを入力します。

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

iSeries<sup>(TM)</sup> のパフォーマンスについての詳細は、『1 ページの『パフォーマンス』を参照してください。

## 時間間隔データを含むパフォーマンス・データ・ファイル

パフォーマンス・データ・ファイルに関する完全な情報を表示するには、以下のリストから表示したいファイルを選択します (アルファベット順で示されています)。

ファイル	説明
QAPMAPPN	APPN データ
QAPMASYN	非同期統計 (リンクにつき 1 つ)
QAPMBSC	2 進データ同期統計 (リンクにつき 1 つ)
QAPMBUS	バス・カウンター (バスにつき 1 つ)
QAPMCIOP	通信 IOP データ (IOP につき 1 つ)
QAPMDDI	Distributed Digital Interface (DDI) データ
QAPMDIOP	ストレージ・デバイス IOP データ (IOP につき 1 つ)
QAPMDISK	ディスク装置データ (読取/書込ヘッドにつき 1 つ)
QAPMDOMINO	ドミノ (iSeries 版) (Domino <sup>(TM)</sup> for iSeries <sup>(TM)</sup> ) データ (ドミノ・サーバーにつき 1 つのレコード)
▶▶ QAPMDPS	データ・ポート・サービス ◀◀
QAPMECL	トークンリング・ファイル項目 (リンクにつき 1 つ)
QAPMETH	イーサネット統計 (リンクにつき 1 つ)
QAPMFRLY	フレーム・リレー・データ (リンクにつき 1 つ)
QAPMHDLC	HDLC 統計 (リンクにつき 1 つ)
QAPMHTTPB	IBM <sup>(R)</sup> HTTP server (powered by Apache) の基本データ (サーバーにつき 1 つ)
QAPMHTTPD	IBM HTTP server (powered by Apache) の詳細データ (サーバー構成要素につき 1 つ)
QAPMIDLC	統合サービスのデジタル・ネットワークのデータ・リンク制御ファイル項目
QAPMIOPD	拡張 IOP データ ▶▶ (ネットワーク・サーバーおよび仮想入出力データ) ◀◀
QAPMJOBMI	MI ジョブ・データ (ジョブ、タスク、またはスレッドにつき 1 つのレコード)。この文書を使用するときに、タスク・タイプ・エクステンダーに関する情報が役立ちます。

ファイル	説明
QAPMJOBOS	ジョブ・オペレーティング・システム・データ (ジョブにつき 1 つのレコード)
QAPMJOBS および QAPMJOBL	ジョブ・データ (ジョブ、タスク、またはスレッドにつき 1 つのレコード)
QAPMJOBWT	ジョブ、タスク、およびスレッド待ち条件
QAPMJOBWTD	ファイル QAPMJOBWT にあるカウンター設定の説明。
QAPMJSUM	ジョブ・グループごとのジョブの要約データ (ジョブ・グループにつき 1 つのレコード)
QAPMLAPD	統合サービス・デジタル・ネットワークの LAPD ファイル項目 (リンクにつき 1 つ)
QAPMLIOP	平衡型ワークステーション制御装置データ (物理制御装置につき 1 つ)
QAPMLPAR	論理区画 (論理区画ごとに 1 つのレコード)
QAPMMIOP	多機能 IOP (IOP につき 1 つ)
QAPMPOOL および QAPMPOOLL	主記憶装置データ (システム記憶域プールにつき 1 つ)
QAPMPOOLB	記憶域プール・データ (リンクにつき 1 つ)
QAPMPOOLT	記憶域プールの調整データ (リンクにつき 1 つ)
QAPMPPP	Point-to-Point Protocol データ (リンクにつき 1 つ)
QAPMRESP	ローカル・ワークステーションの応答時間 (ワークステーションにつき 1 つ)
QAPMRWS	リモート・ワークステーションの応答時間
QAPMSAP	TRLAN、イーサネット、DDI、およびフレーム・リレー SAP ファイル項目
QAPMSNA	SNA データ
QAPMSNADS	SNADS データ (SNADS ジョブにつき 1 つ)
QAPMSTND	DDI 端末データ
QAPMSTNE	イーサネット端末ファイル項目
QAPMSTNL	トークンリング端末ファイル項目
QAPMSTNY	フレーム・リレー端末ファイル項目
QAPMSYS および QAPMSYSL	システム・パフォーマンス・データ
QAPMSYSCPU	システム CPU の使用状況データ
QAPMSYSTEM	システム・レベルのパフォーマンス・データ
QAPMTCP	TCP/IP データ
QAPMTCPIFC	個々の TCP/IP インターフェースに関する TCP/IP データ
QAPMUSRTNS	ユーザー定義トランザクション・データ (各ジョブは、トランザクションのタイプごとに 1 つのレコードを持つ)
QAPMX25	X.25 統計 (リンクにつき 1 つ)

## パフォーマンス・データ・ファイル: ファイルの略語

90 ページの『パフォーマンス・データ・ファイル』は、フィールドおよびバイト・データ表内で略語を使用します。これらの略語には、以下のものが含まれます。

略語	説明
1 次ファイル	これらのファイルは、カテゴリと関連付けられ、カテゴリから生成されます。

C	属性列内の文字
▶▶ H	属性列内の 16 進数 ◀◀
PD	属性列内のパック 10 進数
Z	属性列内のゾーン 10 進数
IOP	入出力処理機構。ディスク、ディスプレイ装置、および通信回線など、ホスト・システムとその他の装置との間の活動を制御する処理機構。
DCE	データ回線終端装置。
MAC	中間アクセス制御。通信 IOP 内のエンティティ。
LLC	論理リンク制御。通信 IOP 内のエンティティ。
ビーコン・フレーム	リングが実行不能のときに送信されるフレーム。
タイプ II フレーム	システム・ネットワーク体系 (SNA) によって使用されるコネクション・オリエンテッドのフレーム (情報フレーム)。
I フレーム	情報フレーム。
▶▶ B	属性列内の 4 桁 (2 バイト) の DDS バイナリー・データ。◀◀

## パフォーマンス・データ・ファイル: 収集サービス・システム・カテゴリとファイルの関係

36 ページの『収集サービス』を使用してパフォーマンス・データを収集する場合、そのデータは管理収集 (\*MGTCOL) オブジェクトに保管されます。CRTPFRTDTA コマンドは、管理収集オブジェクトからデータをエクスポートしてから、90 ページの『パフォーマンス・データ・ファイル』にデータを書き込みます。収集サービスが別個に制御して収集するそれぞれのデータ・タイプは、データ・カテゴリと表現します。それぞれのデータ・カテゴリには、1 つまたは複数のパフォーマンス・データ・ファイルに書き込まれるデータを含んでいるかまたは提供します。作成されるデータベース・ファイルまたはメンバーの場合、ファイルまたはメンバーが依存しているカテゴリ (またはカテゴリのグループ) が存在しており、CRTPFRTDTA によって処理されなければなりません。下記の表は、カテゴリとファイルの関係を示しています。以下の 3 つのタイプの関係があります。

関係	説明
1 次ファイル	これらのファイルは、カテゴリと関連付けられ、カテゴリから生成されます。
互換性ファイル	これらのファイルは、パフォーマンス・データベース互換性を提供するための 1 次ファイルを、以前のファイル構造と結合する論理ファイルです。システムが、関係するすべてのファイル (1 次カテゴリ) を生成する場合、互換性ファイルも生成されます。
2 次ファイル	これらのファイルは、カテゴリまたは 1 次ファイル内に含まれているデータから派生した一部のデータと関連付けられており、それらを含んでいます。ただし、これらのファイルは、そのカテゴリによって制御されません。

ユーザーは、以下のことに注意する必要があります。

1. CRTPFRTDTA コマンドは、データベース・ファイルが、選択されたカテゴリ用の 1 次ファイルである場合にのみ、そのファイルを生成します。
2. 1 次ファイルが複数のカテゴリ用にリストされている場合、ファイルを生成するためにそれらの各カテゴリを選択する必要があります。

3. 1つのカテゴリ用の1次ファイルが別のカテゴリ用の2次ファイルとしてリストされている場合、生成されたデータベース・ファイル内の情報を完全なものにするために2番目のカテゴリを選択する必要があります。たとえば、下記の表で示しているように、QAPMECL用の完全なデータベース・ファイルを生成するには、\*CMNBASEと\*CMNSTNの両方を選択する必要があります。

4. 関連したすべての1次ファイルを生成する場合にのみ、システムは互換性ファイルを生成します。

下記の表では、システム・カテゴリとパフォーマンス・データベース・ファイルとの間の関係を示しています。

カテゴリ	1次ファイル	互換性ファイル	2次ファイル
*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	 QAPMIOPD  QAPMLIOP QAPMDIOP QAPMCIOP QAPMMIOP		
*IPCS	QAPMIOPD QAPMTSK		
*CMNBASE	QAPMASYN QAPMBSC QAPMDDI QAPMECL QAPMETH QAPMFRLY QAPMHDLC QAPMIDLC QAPMLAPD QAPMPPP QAPMX25		

*CMNSTN	QAPMSTND QAPMSTNE QAPMSTNL QAPMSTNY なし		QAPMDDI QAPMETH QAPMECL QAPMFRLY QAPMX25
*CMNSAP	QAPMSAP		
*LCLRSP	QAPMRESP		
*APPN	QAPMAPPN		
*SNA	QAPMSNA		
*EACACHE	なし		QAPMDISK (注を参照)
*TCPBASE	QAPMTCP		
*TCPIFC	QAPMTCPIFC		
*DOMINO	QAPMDOMINO		
*HTTP	QAPMHTTPB QAPMHTTPD		
*USRTNS	QAPMUSRTNS		
 *DPS	QAPMDPS		
*LPAR	QAPMLPAR 		
注:			
このカテゴリは、CRTPFRTA には選択可能ではありません。ただし、追加データが *DISK カテゴリによって報告されるようにします。			

## iSeries<sup>TM</sup> ナビゲーターのモニター

iSeries ナビゲーターに組み込まれているモニターは、36 ページの『収集サービス』・データを使用して、特定のシステム・パフォーマンスの要素を追跡します。さらに、CPU 使用率のパーセンテージやジョブの状況など、特定のイベントが生じた場合に、指定されたアクションを実行できます。モニターを使用することによって、複数のシステムやグループのシステム・パフォーマンスをリアルタイムに表示および管理できます。

モニターを使用する場合、まずモニターを開始してから、iSeries ナビゲーターや PC からサーバーに対して他のタスクを実行できます。実際、自分の PC をオフにすることさえ可能です。その間も iSeries ナビゲーターはモニターを継続し、しきい値コマンドやアクションも指定どおりに実行します。停止処置を実行するまでモニターは実行し続けます。さらにモニターを使用して、147 ページの『iSeries<sup>TM</sup> ナビゲーター (ワイヤレス対応)』にアクセスし、パフォーマンスをリモート側で管理することもできます。

iSeries ナビゲーターには、以下のタイプの iSeries ナビゲーターが備えられています。

### システム・モニター

発生時または最高 1 時間まで、パフォーマンス・データを収集して表示します。詳細グラフは、発生時にサーバーで何が起きているかを視覚的に示すのに役立ちます。システム・パフォーマンスの特定の局面を正確に示すために、さまざまなメトリック (パフォーマンス測定) から選択してください。たとえば、サーバー上の平均 CPU 使用率をモニターする場合には、グラフ上の任意の収集ポイント (コレクション・ポイント) をクリックして、CPU 使用率が最も高い 20 個のジョブを表示した詳細図表を表示できます。それから、これらのジョブのいずれかを右クリックして、直接そのジョブを処理できます。

## ジョブ・モニター

ジョブ名、ジョブ・ユーザー、ジョブ・タイプ、サブシステム、またはサーバー・タイプに基づいてジョブまたはジョブのリストをモニターします。ジョブのパフォーマンス、状況、またはエラー・メッセージをモニターするために、さまざまなメトリックから選択してください。直接ジョブを処理するには、「ジョブ・モニター (Job Monitor)」ウィンドウに表示されたリストから、ジョブを右クリックするだけで行えます。

## メッセージ・モニター

アプリケーションが正常に完了するか、または業務上の必要に不可欠な特定のメッセージについてモニターするかどうかを調べます。「メッセージ・モニター」ウィンドウから、メッセージの詳細の表示、メッセージへの応答、メッセージの送信、およびメッセージの削除を行うことができます。

## B2B 活動モニター

「iSeries の接続 (Connect for iSeries)」のようなアプリケーションが構成されている場合は、B2B 活動モニターを使用して B2B トランザクションをモニターすることができます。活動状態のトランザクションのグラフを時間で表示し、さらに、しきい値でトリガーが出される時に自動的にコマンドを実行することができます。特定のトランザクションを検索して表示し、その特定のトランザクションの詳細ステップの棒グラフを表示できます。

## ファイル・モニター

1 つ以上の選択したファイルで、指定されたテキスト・ストリング、指定されたサイズ、またはファイルに対する修正をモニターします。

モニターに関する詳細は、以下のトピックを参照してください。

### 『モニターの概念』

モニターは、リアルタイムにパフォーマンス・データを表示できます。さらに、その間もシステムのモニターを継続し、指定されたしきい値に達したなら選択されたコマンドを実行できます。モニターの働きと、モニターできる内容、および特定のパフォーマンス状態に応答する方法について学ぶことができます。

### 97 ページの『モニターの構成』

iSeries ナビゲーター中でモニターを構成できます。このトピックを使用すると、モニターをセットアップする方法や、使用可能なオプションを最大限に活用できるように構成する方法を学べます。

### 97 ページの『シナリオ: iSeries<sup>TM</sup> ナビゲーターのモニター』

このトピックにはシナリオが記載されており、さまざまなタイプのモニターの一部を使用してシステム・パフォーマンスの特定の側面を参照する方法が説明されています。

## モニターの概念

システム・モニターは、36 ページの『収集サービス』によって生成されて保守される収集オブジェクト中に保管されるデータを表示します。システム・モニターは、データが収集されるつど、最大 1 時間分表示します。1 時間より長い期間のデータを表示するには、103 ページの『グラフ・ヒストリー』を使用する必要があります。モニター・プロパティでデータ収集の頻度を変更できます。この場合、収集サービス中の設定はオーバーライドされます。

モニターを使用して、システム・パフォーマンスの多種多様な要素を追跡したり調べたりすることができます。また多種多様なモニターを同時に実行することもできます。複数のモニターを同時に使用すると、システム・パフォーマンスの監視や管理を行う精巧なツールが得られます。たとえば、新しい対話式アプリケーションを実装する場合に、システム・モニターを使用してジョブのリソース使用率を優先順位付けし、ジョ

ブ・モニターを使用して問題のあるジョブの監視や処理を行い、メッセージ・モニターを使用して指定されたメッセージがいずれかのシステムに表示された場合にアラートを出すことができます。

### しきい値とアクションの設定

新しいモニターを作成する際には、システム・メトリックが指定されたしきい値レベルに達した場合や、イベントが起きた場合に実行させるアクションを指定できます。しきい値レベルに達したりイベントが起きたりすると、メッセージの送信やジョブ待ち行列の保持などの、OS/400<sup>®</sup> コマンドをエンドポイント・システム上で実行するよう選択できます。さらに、イベント・ログを更新したり、PC 上でアラーム音を鳴らすかモニターを立ち上げるかしてアラートを出したりするなどの、複数の事前定義されたアクションをモニターが実行するよう選択することもできます。最後に、2 番目のしきい値を指定してモニターを自動的にリセットできます。このしきい値に達すると、モニターは通常の活動を再開します。

### モニターの構成

システム・モニターは、高度な対話式ツールで、エンドポイント・システムからのリアルタイム・パフォーマンス・データを収集および表示します。新しいモニターを作成する手順は、短時間でできる簡単なもので、「新しいモニター」ウィンドウから始めます。

1. iSeries<sup>™</sup> ナビゲーターで、「マネージメント・セントラル」を展開し、「モニター」を選択して「システム」を右クリックし、次いで「新しいモニター」を選択します。
2. モニター名を指定します。「新しいモニター」-「一般」ページで、モニターの名前を指定します。モニターのリストからモニターを見つけられるように、簡単な説明も加えてください。
3. メトリックを選択します。「新しいモニター」-「メトリック」ページで、メトリックを選択します。任意の数のエンドポイント・システムまたはシステム・グループについて、任意の数のメトリックをモニターすることができます。
4. メトリック情報を表示して変更を加えます。「新しいモニター」-「メトリック」ページで、各メトリックのプロパティを編集します。選択した各メトリックについて、収集間隔、グラフの最大値、および表示時間を編集できます。
5. しきい値コマンドを設定します。「メトリック」ページの「しきい値」タブで、しきい値を有効にし、そのトリガーしきい値またはリセットしきい値に達したときにエンドポイント・システムで実行するコマンドを指定します。
6. しきい値アクションを設定します。「新しいモニター」-「アクション」ページで、メトリックがトリガーしきい値またはリセットしきい値に達したときに実行するアクションを指定します。
7. システムとグループを選択します。「新しいモニター」-「システムとグループ」ページで、モニターを開始したいエンドポイント・システムとシステムのグループを選択します。

モニターを作成した後、モニター名を右クリックして「開始」をクリックすると、モニターを実行してモニター・グラフでの作業を開始することができます。

### シナリオ: iSeries<sup>™</sup> ナビゲーターのモニター

iSeries ナビゲーターに組み込まれているモニターには、システム・パフォーマンスの調査や管理を行う強力なツールの集合が備えられています。iSeries ナビゲーターに備えられているモニターのタイプの概要については、『95 ページの『iSeries<sup>™</sup> ナビゲーターのモニター』』を参照してください。

詳細な使用例とサンプルの構成については、以下のシナリオを参照してください。

#### 98 ページの『シナリオ: システム・モニター』

この例のシステム・モニターは、CPU 使用率が高すぎるために使用可能なリソースが増えるまで優先順位の低いジョブを一時的に保留する場合に、アラートを出します。

### 99 ページの『シナリオ: CPU 使用率のジョブ・モニター』

この例のジョブ・モニターは、指定されたジョブの CPU 使用率を追跡し、CPU 使用率が高すぎる場合はそのジョブの所有者にアラートを出します。



### 101 ページの『シナリオ: Advanced Job Scheduler 通知を使ったジョブ・モニター』

ジョブのしきい値限度を超えたときにオペレーターに E メールを送信する、ジョブ・モニターの例を参照してください。 <<

### 102 ページの『シナリオ: メッセージ・モニター』

この例のメッセージ・モニターは、iSeries サーバー上で生じた、メッセージ・キューに関する照会メッセージを表示します。このモニターは、メッセージを検出すると即時にそのメッセージをオープンして表示します。

## シナリオ: システム・モニター:

### 状態

システム管理者は、ユーザーの要件や業務上の要件に基づく現在の要求を満たせるだけの資源が、確実に iSeries<sup>(TM)</sup> システム上にあるようにする必要があります。ご使用のシステムでは、CPU 使用率が特に重要な関心事です。CPU 使用率が高すぎるために、使用可能なリソースが増えるまで優先順位の低いジョブを一時的に保留する場合に、システムがアラートを出すようにしたいと思っています。

そのためには、CPU 使用率が 80% を超えたらメッセージを送信するように、システム・モニターをセットアップできます。さらに、CPU 使用率が 60% に下がるまですべてのジョブを QBATCH ジョブ待ち行列中に保留し、60% になったらジョブを保留解除して通常の操作を再開することもできます。

### 構成の例

システム・モニターをセットアップするには、追跡したいメトリックと、そのメトリックが指定のレベルに達した場合にモニターが行う処理を定義する必要があります。この目標を達成するようにシステム・モニターを定義するには、以下のステップを完了してください。

1. iSeries ナビゲーターで、「マネージメント・セントラル」>「モニター」を展開し、「システム・モニター (System Monitor)」を右クリックし、「新しいモニター... (New Monitor...)」を選択します。
2. 「一般」ページで、このモニターの名前と説明を入力します。
3. 「メトリック」タブで、以下の値を入力します。
  - a. 「使用可能なメトリック」のリストから「CPU 使用率基本 (平均)」を選択して、「追加」を選択します。「CPU 使用率基本 (平均)」が「モニターするメトリック」の下にリストされるようになり、ウィンドウの下部にこのメトリックの設定が表示されます。
  - b. 「収集間隔」で、このデータを収集する頻度を指定します。この値は、収集サービスの設定をオーバーライドします。この例では、「30 秒 (30 seconds)」を例示します。
  - c. このメトリックに関するモニターのグラフの縦軸の目盛りを変更するには、「最大グラフ値」を変更します。このメトリックに関するモニターのグラフの横軸の目盛りを変更するには、「表示時間」の値を変更します。
  - d. メトリック設定の「しきい値 1」タブをクリックし、以下の値を入力して、CPU 使用率が 80% 以上の場合に照会メッセージを送信するようにします。

- 1) 「しきい値を使用可能にする」を選択します。
  - 2) しきい値トリガーの値として、「>= 80」(80 % 以上が使用中)を指定します。
  - 3) 「期間」に、間隔「1」を指定します。
  - 4) 「OS/400<sup>®</sup> コマンド」に、以下の値を指定します。  

```
SNDMSG MSG('Warning,CPU...') TOUSR(*SYSOPR) MSGTYPE(*INQ)
```
  - 5) しきい値リセットの値として、「< 60」(60 % 未満が使用中)を指定します。この場合、CPU 使用率が 60% 未満に下がるとモニターがリセットされます。
- e. 「しきい値 2」タブをクリックし、以下の値を入力して、5 回の収集間隔の間 CPU 使用率が 80% を超える状態が続いたら、すべてのジョブを QBATCH ジョブ待ち行列中に保留するようにします。

- 1) 「しきい値を使用可能にする」を選択します。
- 2) しきい値トリガーの値として、「>= 80」(80 % 以上が使用中)を指定します。
- 3) 「期間」に、間隔「5」を指定します。
- 4) 「OS/400 コマンド」に、以下の値を指定します。

```
HLDJOBQ JOBQ(QBATCH)
```

- 5) しきい値リセットの値として、「< 60」(60 % 未満が使用中)を指定します。この場合、CPU 使用率が 60% 未満に下がるとモニターがリセットされます。
- 6) 「期間」に、間隔「5」を指定します。
- 7) 「OS/400 コマンド」に、以下の値を指定します。

```
RLSJOBQ JOBQ(QBATCH)
```

このコマンドは、5 回の収集間隔の間 CPU 使用率が 60% 未満の状態が続いたら、QBATCH ジョブ待ち行列を保留解除します。

4. 「アクション」タブをクリックして、「トリガー」と「リセット」の両方の列で、「イベントのログ」を選択します。このアクションを選択すると、しきい値が起動したりリセットされたりする際に、イベント・ログ中に項目が作成されます。
5. 「システムおよびグループ」タブをクリックして、モニターしたいシステムとグループを指定します。
6. 「OK」をクリックして、モニターを保管します。
7. システム・モニターのリストから、新しいモニターを右クリックして、「開始」を選択します。

## 結果

新しいモニターは、CPU 使用率を表示し、指定された収集間隔に従って 30 秒ごとに新しいデータ・ポイントを追加します。CPU 使用率が 80% に達すると、PC がオフになっている場合も含めて、必ずモニターは指定されたしきい値アクションを実行します。

注: このモニターは、CPU 使用率のみを追跡します。しかしながら、同一のモニターに使用可能なメトリックをいくつでも組み込むことができ、個々のメトリックに独自のしきい値とアクションを指定できます。さらに、複数のシステム・モニターを同時に実行することもできます。

## シナリオ: CPU 使用率のジョブ・モニター:

### 状態

現在 iSeries<sup>TM</sup> サーバー上で新しいアプリケーションを実行しており、一部の新しい対話型ジョブが許容量を超えるリソースを使用していることに着目しています。ジョブが使用する CPU キャパシティが多すぎる場合に、常にその問題のジョブの所有者に通知するようにしたいと思っています。

新しいアプリケーション中のジョブを監視し、ジョブが使用する CPU キャパシティが 30% を超えたらメッセージを送信するように、ジョブ・モニターをセットアップできます。

## 構成の例

ジョブ・モニターをセットアップするには、監視対象のジョブ、監視対象のジョブ属性、および指定したジョブ属性が検出された場合にモニターが行う処理を定義する必要があります。この目標を達成するようにジョブ・モニターをセットアップするには、以下のステップを完了してください。

1. iSeries ナビゲーターで、「**マネージメント・セントラル**」>「**モニター**」を展開し、「**ジョブ・モニター (Job monitor)**」を右クリックし、「**新しいモニター... (New Monitor...)**」を選択します。
2. 「**一般**」ページで、以下の値を入力します。
  - a. このモニターの名前と説明を指定します。
  - b. 「**モニターするジョブ (Jobs to monitor)**」タブで、以下の値を入力します。
    - 1) 「**ジョブ名 (Job name)**」で、監視したいジョブの名前 (MKWIDGET など) を指定します。
    - 2) 「**サブシステム**」で、QINTER を指定します。
    - 3) 「**追加**」をクリックします。
3. 「**メトリック**」タブで、以下の情報を入力します。
  - a. 「**使用可能なメトリック**」リストで、「**合計数値 (Summary Numeric Values)**」を展開し、「**CPU 使用率のパーセンテージ (CPU Percent Utilization)**」を選択して、「**追加**」をクリックします。
  - b. メトリック設定の「**しきい値 1**」タブで、以下の値を入力します。
    - 1) 「**トリガーを使用可能にする**」を選択します。
    - 2) しきい値トリガーの値として、「**>= 30**」(30 % 以上が使用中) を指定します。
    - 3) 「**期間**」に、間隔「**1**」を指定します。
    - 4) 「**OS/400<sup>®</sup> トリガー・コマンド (OS/400 trigger command)**」に、以下の値を指定します。

```
SNDMSG MSG('Your job is exceeding 30% CPU capacity') TOUSR(&OWNER)
```
    - 5) 「**リセットを使用可能にする (Enable reset)**」をクリックします。
    - 6) しきい値リセットの値として、「**< 20**」(20 % 未満が使用中) を指定します。
4. 「**収集間隔**」タブをクリックして、「**15 秒 (15 seconds)**」を選択します。この値は、収集サービスの設定をオーバーライドします。
5. 「**アクション**」タブをクリックして、「**トリガー**」と「**リセット**」の両方の列で、「**イベントのログ**」を選択します。
6. 「**サーバーおよびグループ (Servers and groups)**」タブをクリックして、このジョブをモニターする対象にしたいサーバーとグループを選択します。
7. 「**OK**」をクリックして、新しいモニターを保管します。
8. ジョブ・モニターのリストから、新しいモニターを右クリックして、「**開始**」を選択します。

## 結果

新しいモニターは、15 分ごとに QINTER サブシステムをチェックし、ジョブ MKWIDGET の CPU 使用率が 30 % を超えると、このジョブの所有者にメッセージを送信します。このジョブが使用する CPU のキャパシティが 20% 未満の場合は、このモニターはリセットします。

### シナリオ: Advanced Job Scheduler 通知を使ったジョブ・モニター:

#### 状態

現在、iSeries<sup>(TM)</sup> サーバーでアプリケーションを実行しており、CPU 使用率が指定されたしきい値に達したら通知してもらいたいと思っています。

Advanced Job Scheduler がエンドポイント・システムにインストールされているなら、JS を使用した配布の送信 (SNDDSTJS) コマンドを使用して、しきい値を超えたときに誰かに E メールで通知することができます。たとえば、対象の受信者がメッセージを停止して応答しない場合には、次の人へ通知を段階的に拡大するよう指定できます。当番スケジュールを作成し、当番の人だけに通知を送信することができます。また、複数の E メール・アドレスに通知を送信することもできます。

#### ジョブ・モニター構成例

この例では、SNDDSTJS コマンドを使用して、E メール・アドレスのユーザー定義リストである OPERATOR という名前の受信者にメッセージを送信します。また、受信者の代わりに E メール・アドレスを指定したり、その両方を指定したりすることもできます。この目標を達成するようにジョブ・モニターをセットアップするには、以下のステップを完了してください。

1. iSeries ナビゲーターで、「**マネージメント・セントラル**」>「**モニター**」を展開し、「**ジョブ・モニター (Job monitor)**」を右クリックし、「**新しいモニター... (New Monitor...)**」を選択します。
2. 「**一般**」ページで、以下の値を入力します。
  - a. このモニターの名前と説明を指定します。
  - b. 「**モニターするジョブ (Jobs to monitor)**」タブで、以下の値を入力します。
    - 1) 「**ジョブ名 (Job name)**」で、監視したいジョブの名前 (MKWIDGET など) を指定します。
    - 2) 「**サブシステム**」で、QINTER を指定します。
    - 3) 「**追加**」をクリックします。
3. 「**メトリック**」タブで、以下の情報を入力します。
  - a. 「**使用可能なメトリック**」リストで、「**合計数値 (Summary Numeric Values)**」を展開し、「**CPU 使用率のパーセンテージ (CPU Percent Utilization)**」を選択して、「**追加**」をクリックします。
  - b. メトリック設定の「**しきい値 1**」タブで、以下の値を入力します。
    - 1) 「**トリガーを使用可能にする**」を選択します。
    - 2) しきい値トリガーの値として、「**>= 30**」(30 % 以上が使用中) を指定します。
    - 3) 「**期間**」に、間隔「**1**」を指定します。
    - 4) 「**OS/400<sup>(R)</sup> トリガー・コマンド (OS/400 trigger command)**」に、以下の値を指定します。  
SNDDSTJS RCP(OPERATOR) SUBJECT('Job monitor trigger') MSG('Job &JOBNAME is still running!')
    - 5) 「**リセットを使用可能にする (Enable reset)**」をクリックします。
    - 6) しきい値リセットの値として、「**< 20**」(20 % 未満が使用中) を指定します。
4. 「**収集間隔**」タブをクリックして、「**15 秒 (15 seconds)**」を選択します。この値は、収集サービスの設定をオーバーライドします。

5. 「アクション」タブをクリックして、「トリガー」と「リセット」の両方の列で、「イベントのログ」を選択します。
6. 「サーバーおよびグループ (Servers and groups)」タブをクリックして、このジョブをモニターする対象にしたいサーバーとグループを選択します。
7. 「OK」をクリックして、新しいモニターを保管します。
8. ジョブ・モニターのリストから、新しいモニターを右クリックして、「開始」を選択します。

#### メッセージ・モニター構成例

メッセージ・モニターの使用時には、メッセージ・テキストを受信側に送信することができます。以下は、メッセージ・テキストを検索し、SNDDSTJS コマンドを使用してすべての呼び出し時の受信者に E メールを送信する CL プログラムの例です。

```
PGM PARM(&MSGKEY &TOMSGQ &TOLIB)

DCL &MSGKEY *CHAR 4
DCL &TOMSGQ *CHAR 10
DCL &TOLIB *CHAR 10

DCL &MSGTXT *CHAR 132

RCVMMSG MSGQ(&TOLIB/&TOMSGQ) MSGKEY(&MSGKEY)
      RMV(*NO) MSG(&MSGTXT)
      MONMSG CPF0000 EXEC(RETURN)

SNDDSTJS RCP(*ONCALL) SUBJECT('Message queue trigger') MSG(&MSGTXT)
      MONMSG MSGID(CPF0000 IJS0000)

ENDPGM
```

以下は、CL プログラムを呼び出すコマンドです。

```
CALL SNDMAIL PARM('&MSGKEY' '&TOMSG' '&TOLIB')
```

#### 結果

モニターは、15 分ごとに QINTER サブシステムをチェックし、ジョブ MKWIDGET の CPU 使用率が 30 % を超えると、オペレーターに E メールを送信します。このジョブが使用する CPU のキャパシティが 20% 未満の場合は、このモニターはリセットします。

Advanced Job Scheduler の通知機能について詳しくは、通知の処理 (Work with notification) を参照してください。 <<

#### シナリオ: メッセージ・モニター:

##### 状態

貴社で複数の iSeries<sup>™</sup> サーバーを実行しており、個々のシステムのメッセージ・キューをチェックするのに時間がかかります。システム管理者は、システム全体のどこでも照会メッセージが生成されたら、そのことに気付く必要があります。

いずれかの iSeries システムで生じた、メッセージ・キューに関する照会メッセージを表示するように、メッセージ・モニターをセットアップできます。このモニターは、メッセージを検出すると即時にそのメッセージをオープンして表示します。

## 構成の例

メッセージ・モニターをセットアップするには、監視したいメッセージのタイプと、それらのメッセージが生成された場合にモニターが行う処理を定義する必要があります。この目標を達成するようにメッセージ・モニターをセットアップするには、以下のステップを完了してください。

1. iSeries ナビゲーターで、「マネージメント・セントラル」>「モニター」を展開し、「メッセージ・モニター (Message monitor)」を右クリックし、「新しいモニター... (New Monitor...)」を選択します。
2. 「一般」ページで、このモニターの名前と説明を入力します。
3. 「メッセージ」タブで、以下の値を入力します。
  - a. 「モニターするメッセージ・キュー (Message queue to monitor)」で、「QSYSOPR」を指定します。
  - b. 「メッセージ・セット 1 (Message set 1)」タブ上で、「タイプ」で「照会」を選択して、「追加」をクリックします。
  - c. 「このメッセージ数で起動 (Trigger at the following message count)」を選択して、メッセージ数「1」を指定します。
4. 「収集間隔」タブをクリックして、「15 秒 (15 seconds)」を選択します。
5. 「アクション」タブをクリックして、「モニターを開く」を選択します。
6. 「システムおよびグループ」タブをクリックして、照会メッセージをモニターしたいシステムとグループを指定します。
7. 「OK」をクリックして、新しいモニターを保管します。
8. メッセージ・モニターのリストから、新しいモニターを右クリックして、「開始」を選択します。

## 結果

新しいメッセージ・モニターは、モニターされているいずれかの iSeries サーバー上の QSYSOPR に送信された照会メッセージを表示します。

注: このモニターは、QSYSOPR に送信される照会メッセージだけに応答します。しかしながら、1 つのモニターに 2 種類のメッセージの集合を組み込んだり、複数のメッセージ・モニターを同時に実行したりできます。また、指定されたメッセージが受信された時点で、メッセージ・モニターが OS/400<sup>(R)</sup> コマンドを実行することもできます。

## グラフ・ヒストリー

グラフ・ヒストリーには、36 ページの『収集サービス』を使用して数日、数週間、数カ月、または数年にわたって収集されたパフォーマンス・データのグラフィカル・ビューが備えられています。システム・モニターを実行してパフォーマンス・データを表示する必要はありません。収集サービスを使用してデータを収集する限り、「グラフ・ヒストリー」ウィンドウを表示することができます。

### • 104 ページの『グラフ・ヒストリーの概念』

グラフ・ヒストリーで使用できるヒストリー・データの量は、収集サービス中の収集保存期間の値と、105 ページの『IBM Performance Management for eServer iSeries』<sup>(TM)</sup> が使用可能かどうかによって、大きく変わります。パフォーマンス・データのレコードの管理や表示を行うのに使用できるオプションの説明については、このトピックを参照してください。

- 『グラフ・ヒストリーの使用』

グラフ・ヒストリーには、iSeries ナビゲーターからアクセスできます。ステップ形式の指示については、このトピックを参照してください。

システム・パフォーマンスのモニターについて詳しくは、『18 ページの『パフォーマンスの追跡』』のトピックを参照してください。

## グラフ・ヒストリーのご概念

「グラフ・ヒストリー」には、36 ページの『収集サービス』で作成された収集オブジェクトに入っているデータが表示されます。したがって、使用可能なデータのタイプと量は、収集サービスの構成によって異なります。

グラフ化が可能なデータの量は、「収集サービス」プロパティから選択した設定値（特に収集保存期間）によって決まります。iSeries<sup>(TM)</sup> ナビゲーターを使用して、複数のシステムで 108 ページの『PM iSeries の活動化』。PM iSeries をアクティブにすると、グラフ・ヒストリー機能を使用して、数日前、数週間前、または数カ月前に収集されたデータを表示することができます。これはリアルタイムのモニター機能を超えて、要約または詳細データにアクセスすることができます。PM iSeries が使用可能になっていないと、グラフ・データ・フィールドは 1 から 7 日をサポートします。PM iSeries が使用可能になっていると、システム上での管理収集オブジェクトの保存期間を定義します。

- **詳細データ**

管理収集オブジェクトが削除される前に、ファイル・システム内にそれらが保存される時間の長さ。特定の時間間隔を時間数または日数で選択するか、あるいは「永続」を選択することができます。「永続」を選択する場合、管理収集オブジェクトは自動的に削除されません。

- **グラフ・データ**

「グラフ・ヒストリー」ウィンドウに示される詳細およびプロパティ・データのデータが、表示されてから削除されるまでのシステムに留まっている時間の長さです。PM iSeries を開始しない場合、1 から 7 日を指定することができます。PM iSeries を開始する場合、1 から 30 日を指定することができます。デフォルト値は 1 時間です。

- **要約データ**

グラフのデータ収集ポイントが削除される前に、「グラフ・ヒストリー (Graph History)」ウィンドウ内にそれらが表示されるか、またはシステムに保存される時間の長さ。詳細データまたはプロパティ・データは使用できません。PM iSeries を開始して、要約データ・フィールドを使用可能にする必要があります。デフォルトは 1 カ月です。

## グラフ・ヒストリーの使用

グラフ・ヒストリーは、iSeries<sup>(TM)</sup> ナビゲーターに組み込まれています。36 ページの『収集サービス』でモニターしているデータのグラフ・ヒストリーを表示するには、以下のステップを行ってください。

1. 単一システムまたはシステム・グループでの収集サービスの開始の方法の詳細は、iSeries ナビゲーターのオンライン・ヘルプを参照してください。
2. 必要な場合、「収集サービスの開始 - 一般」ページから、「IBM<sup>(R)</sup> Performance Management for eServer<sup>(TM)</sup> iSeries を開始 (Start IBM Performance Management for eServer iSeries)」を選択します。
3. 収集保存期間の残りの値に変更を加えます。
4. 「OK」をクリックします。
5. システム・モニターまたは収集サービス・オブジェクトのどちらかを右マウス・ボタンでクリックして「グラフ・ヒストリー」を選択することによって、グラフ・ヒストリーを表示することができます。
6. グラフィカルに表示するには、「最新表示」をクリックしてください。

グラフ・履歴の立ち上げが完了したなら、グラフ化された一連の収集ポイントを示したウィンドウが表示されます。グラフ線上の収集ポイントは、使用できるデータの 3 つのレベルと対応する 3 つの異なるグラフで表示されます。

- 四角の収集ポイントは、詳細情報とプロパティ情報の両方がデータ内にあることを意味します。
- 三角形の収集ポイントは、詳細情報を含む要約データを表しています。
- 円形の収集ポイントは、詳細情報またはプロパティ情報を含まないデータを表しています。

## IBM Performance Management for eServer iSeries

IBM<sup>(R)</sup> Performance Management for eServer<sup>(TM)</sup> iSeries<sup>(TM)</sup> (PM  iSeries または PM iSeries) は自動化および自己管理されているため、簡単に使用することができます。PM iSeries は、収集サービスを自動的に起動して、所有権の付いていないパフォーマンス・データおよびキャパシティー・データをサーバーから収集してから、そのデータを IBM に送信します。すべてのコレクション・サイトはネットワーク保護されており、転送時間は、ユーザーによって完全に制御されます。IBM にデータを送信すると、すべての傾向データを自分で保管する必要がなくなります。IBM がユーザーのためにデータを保管し、サーバーの拡張およびパフォーマンスを示す一連の 123 ページの『PM iSeries 報告書』およびグラフを提供します。従来のブラウザを使用して、電子的に報告書にアクセスすることができます。そのような報告書を参考にして、かぎとなるパフォーマンス指標の継続的分析を通して、システム資源を計画して管理することができます。

IBM Operational Support Services for PM iSeries には一連の報告書、グラフ、およびプロファイルが組み込まれています。これらは、現行アプリケーションおよびハードウェア・パフォーマンスを最大化する (パフォーマンス傾向分析を使用して) のに役立ちます。またこの製品を使用すると、CPU またはディスクなどのハードウェアに必要なアップグレードのタイミングと業務傾向がどのように関連しあっているかが分かりやすくなります (キャパシティー・プランニングを使用して) 。キャパシティー・プランニング情報はシステム使用率資源およびスループット・データの傾向から得ることができ、ご使用のサーバーについての早期警告システムと見なすことができます。PM iSeries は、システムの「正常性」を知らせる仮想資源であると考えてください。

PM iSeries は、中央処理装置 (CPU) の 1 % 未満しか使用しません。PM/400 は、約 58 MB のディスク・スペースを使用します。これはハードウェア・モデルおよび収集間隔のサイズによって異なります。

### 106 ページの『PM iSeries の概念』

PM iSeries で利用できる機能と長所についてと、実装時の重要な考慮事項について考察してみてください。

### 108 ページの『PM iSeries の構成』

PM iSeries の使用を開始するには、それを活動化し、データの送信と報告書の受信のための伝送方式をセットアップしてから、最後にデータの収集とストレージをカスタマイズする必要があります。

### 119 ページの『PM iSeries の管理』

ネットワークのセットアップが完了したので、PM iSeries でのさまざまなタスクを実行することができます。

### 123 ページの『PM iSeries 報告書』

PM iSeries を使用して収集サービス・データを IBM へ直接送信するように iSeries サーバーを構成することができます。その後 IBM でいくつかの報告書が生成されます。これは、Web 上で表示できますが、直接ユーザーに返送することも可能です。PM iSeries を活動化して自動的にレポートを生成するにすれば、時間と資源の節約になるだけでなく、将来の成長ニーズを見込んで事前に計画をたてることができます。

## PM iSeries の概念

PM iSeries<sup>TM</sup> は、収集サービスを使用して、所有権の付いていないパフォーマンス・データおよびキャパシティー・データをサーバーから収集してから、そのデータを IBM<sup>®</sup> に送信します。そのような情報には、CPU 使用率とディスク・キャパシティー、応答時間、スループット、アプリケーションとユーザーの使用量などがあります。IBM にデータを送信すると、すべての傾向データを自分で保管する必要がなくなります。IBM がユーザーのためにデータを保管し、サーバーの拡張およびパフォーマンスを示す一連の報告書およびグラフを提供します。従来のブラウザを使用して、電子的に報告書にアクセスすることができます。

### 『PM iSeries の利点』

PM iSeries を使用すると、システム資源の管理とキャパシティー・プランニングがかなり簡素化されます。PM iSeries を使用する具体的な方法を考察してください。

### 107 ページの『PM iSeries 製品の操作サポート・サービス』

PM iSeries には、広範囲にわたるオプションが用意されています。以下の情報を使用して、各自のニーズに最適のサービスの組み合わせを決めてください。

### 107 ページの『PM iSeries のデータ収集に関する考慮事項』

PM iSeries では、パフォーマンス・データの収集には 36 ページの『収集サービス』を使用します。PM iSeries と収集サービスがどのように連携して、必要なデータを提供するかを確認してください。

**PM iSeries の利点:** PM iSeries<sup>TM</sup> 機能を使用すると、次のような利点が得られます。

- **不測の事態が起きないようにするのに役立ちます。**  
思いがけない失策を免れます。ユーザーがシステムの拡張およびパフォーマンスの管理を制御しますが、これは、ユーザーがシステムを管理するのであって、システムがユーザーを管理するのではないことを意味します。
- **時間を節約します。**  
パフォーマンス・データの収集および報告を自動的に行うことによって、それらの作業にかかる労力や費用を節約します。これにより、ユーザー資源をシステムおよびアプリケーションの管理に集中できるという利点があります。
- **最大限の効率を得られるように前もって計画することができます。**  
システムを最大効率で実行し続けるための財政的な要件を前もって計画することができます。
- **情報を理解することが容易になります。**  
情報を理解すれば、上司から「なぜアップグレードする必要があるのか」と質問されたときに、容易に返答することができます。
- **将来を予測することができます。**  
実際の傾向情報に基づいてデータ処理の拡張を予測することができます。
- **システム問題を識別することができます。**  
PM iSeries データを使用してパフォーマンス障害を識別することができます。
- **次のアップグレードのサイズを見積もるときの参考になります。**

PM iSeries データを Workload Estimator for iSeries  にアップロードすれば、次のアップグレードのサイズを設定することができます。

PM iSeries を使用する前にしておくことについては、105 ページの『IBM Performance Management for eServer iSeries』で学習してください。

**PM iSeries 製品の操作サポート・サービス:** グラフおよび報告書は電子的または印刷形式のいずれかで受け取ることができます。電子グラフは毎月受け取ることができます。印刷グラフは毎月または 3 カ月ごとに受け取ります。PM iSeries<sup>TM</sup> サービス料金は、パフォーマンス情報を受け取る回数とその形式 (電子的または印刷形式) の選択によって変わります。これらの報告書オプションのいくつかは無料ですが、有料のものもあります。それぞれの国のマーケティングおよびサービスに関する組織は、使用可能なサポートの詳細を提供することができます。無償および有償のオプションの詳細は、PM eServer iSeries の Web サイト  を参照してください。

この機能を使用する前にする必要のあることについては、105 ページの『IBM Performance Management for eServer iSeries』で学習してください。

**PM iSeries のデータ収集に関する考慮事項:** システムの使用率、ワークロード、およびパフォーマンス測定の詳細な傾向を設定するための最も重要な要件は、整合性です。理想的に言って、パフォーマンス・データは 1 日に 24 時間収集される必要があります。PM iSeries<sup>TM</sup> と収集サービスとの関係のため、PM iSeries の使用時に起こり得る事柄を理解している必要があります。

ここで、PM iSeries の使用時に収集の定義を助けるためのいくつかの指針があります。

- **QMPGDATA ライブラリーを選択して、データを保管します。**

PM iSeries が活動状態にある場合、「コレクションを保管する場所」フィールドには、デフォルト値 /QSYS.LIB/QMPGDATA.LIB が使用されます。QMPGDATA が他の値に変更されると、PM iSeries はすぐにコレクションを循環させ、その値を QMPGDATA に戻します。異なるライブラリーにデータを収集したい場合には、PM iSeries がデータを検索する場所を変更します。コマンド行で **GO PM400** と入力し、オプション 3 (カスタマイズの処理) を選択して、ライブラリー名を変更します。

- **収集サービスを使用して、継続的にデータを収集します。**

PM iSeries は、収集サービスを使用して 1 日に 24 時間データを収集することにより、この要件を満たします。PM iSeries は、15 分間隔でパフォーマンス・データを収集します。PM iSeries は、デフォルトでは 15 分間隔を使用しますが、設定されている間隔は変更しません。推奨される間隔は 15 分間隔です。

- **「標準 + プロトコル」プロファイルを選択します。**

標準 + プロトコルは、収集プロファイルのデフォルト値です。収集プロファイルは、どのデータが収集されるかを示します。「標準 + プロトコル」プロファイル内のデータ・カテゴリーは、パフォーマンス・モニターの開始 (STRPFRMON) コマンド上の DATA パラメーターの \*ALL 値に対応しています。この値が他の値に変更されると、PM iSeries はすぐに値を元に戻します。これは、「カスタム」を選択し、すべてのカテゴリーを含めている場合も同じです。変更は直ちに有効になります。コレクションは、(他の理由で必要とされない限り) 循環しません。このアクションは、PM iSeries 報告書に関する十分な情報を収集するために行います。

- **PM iSeries が活動状態にあるときは、収集のパラメーターに一時変更を加えないようにします。**

たとえば、PM iSeries を活動化したときは、「収集時にデータベース・ファイルを作成」フィールドにデフォルト値のチェックが付いています。この値が変更されると、PM iSeries はすぐにデフォルト値を元に戻します。変更は直ちに有効になります。コレクションは、(他の理由で必要とされない限り) 循環しません。

- **収集サービスを終了します。**

iSeries ナビゲーターからいつでも収集サービスを終了することができます。収集サービスを終了する場合、PM iSeries が実行しているときに以下の考慮事項が適用されます。

- PM iSeries スケジューラーは、次の時間の始めに収集サービスを開始します。
- 収集されるデータがほとんどない日は、傾向の計算に含まれません。したがって、収集サービスを頻繁に中断しないでください。

収集サービスを開始しない場合には、122 ページの『PM iSeries の暫時オフ』ことができます。

## PM iSeries の構成

PM iSeries<sup>(TM)</sup> では、収集サービスを通してパフォーマンス・データの収集が自動化されています。どのライブラリーにそのデータを入れるかを指定できますが、基本補助記憶域プール (ASP) に置かれているライブラリーであることが前提になります。そのライブラリーを、独立した補助記憶域プールに移動してはなりません。独立した補助記憶域プールはオフに変更される可能性があり、もしオフに変更されると、PM iSeries の収集プロセスは停止するからです。ライブラリーが存在しないと、PM iSeries が起動時に作成します。

PM iSeries の使用を開始するには、以下のタスクを実行する必要があります。

### 『PM iSeries の活動化』

PM iSeries は OS/400<sup>(R)</sup> の付属製品ですが、収集機能を利用するには活動化する必要があります。

#### 109 ページの『どの PM iSeries 送信方式を使用するか』の決定』

データの送信方法を決定します。マネージメント・セントラルのインベントリー機能を使用してデータを収集してから、Electronic Service Agent (エクストリーム・サポート) を使用してデータを送信することができますが、PM iSeries を使用してデータを収集して SNA プロトコルを介してデータを送信してもかまいません。

#### 116 ページの『PM iSeries のカスタマイズ』

ネットワークのセットアップが完了したなら、ユーザーの必要を満たすために PM iSeries をカスタマイズする必要があります。

**PM iSeries の活動化:** データ収集機能を利用するには、PM iSeries<sup>(TM)</sup> を開始しなければなりません。以下の方法のいずれかを使用して、PM iSeries を開始することができます。

### iSeries ナビゲーターを使用します。

iSeries ナビゲーターを使用して、複数のシステムで PM iSeries をアクティブにします。PM iSeries をアクティブにすると、103 ページの『グラフ・ヒストリー』機能を使用して、数日前、数週間前、または数カ月前に収集されたデータを表示することができます。リアルタイム・モニター機能を超えています。要約データまたは詳細データにアクセスすることができます。PM iSeries が使用可能になっていないと、グラフ・データ・フィールドは 1 から 7 日をサポートします。PM iSeries が使用可能な場合は、データ保存の時間の長さを選択します。

iSeries ナビゲーターから PM iSeries を開始するには、以下のステップを実行します。

1. PM iSeries を開始したいシステムを iSeries ナビゲーターで拡張表示します。
2. 「構成およびサービス」を展開します。
3. 「収集サービス」を右クリックします。
4. 「PM eServer iSeries」を選択します。
5. 「開始」を選択します。
6. PM iSeries を開始したいシステムを選択します。
7. 「OK」をクリックします。

### QSYSOPR メッセージ・キュー内のメッセージ CPAB02A への応答

QSYSWRK サブシステムが開始するとき、このメッセージによって PM iSeries を活動化するかどうかを尋ねられます。

1. 文字ベースのインターフェースから、QSYSOPR 内のメッセージ「Do you want to activate PM iSeries? (I G C)」に G と応答します。 QSYSOPR メッセージ・キューは PM eServer<sup>(TM)</sup> iSeries を活動化するというメッセージを受信します。
2. 連絡先情報を更新します。 **GO PM400** コマンドを発行して、オプション 1 を指定します。

### PM eServer iSeries の構成 (CFGPM400) コマンドの発行

文字ベースのインターフェースから、PM eServer iSeries の構成 (CFGPM400) コマンドを発行することができます。

セットアップ・プロセス内の次のステップ『『どの PM iSeries 送信方式を使用するか』』に進むことができます。

iSeries のパフォーマンスの概要の詳細は、『1 ページの『パフォーマンス』』を参照してください。

**どの PM iSeries 送信方式を使用するか**の決定: V5R1 以降では、PM iSeries<sup>(TM)</sup> 送信プロセスでは、セントラル・システムとエンドポイント・システムをセットアップするためにマネージメント・セントラルを使用して実行するネットワーク構成の利点が生かされています。ただし、引き続き文字ベースのインターフェースを使用して PM iSeries を構成することもできます。使用したい送信方式を選択します。

- 『エクストリーム・サポートでの Service Agent による PM iSeries<sup>(TM)</sup> データの送信 (ユニバーサル・コネクション)』  
この送信方式を選択する場合、マネージメント・セントラルのインベントリ機能によってデータを収集するように PM iSeries を構成する必要があります。サーバーに V4R5 以降のオペレーティング・システムがインストールされている場合 (ユニバーサル・コネクションの修正も適用しなければなりません)、PM iSeries に対してこの構成を実行します。エクストリーム・サポートを使用する場合は、このメソッドを選択します。
- 110 ページの『SNA プロトコルでのデータの送信』  
この送信方式を選択する場合、文字ベースのインターフェースを使用して PM iSeries を構成する必要があります。PM iSeries は SNA を使用してデータを収集し、それを送信します。サーバーに OS/400<sup>(R)</sup> V4R5 またはそれ以前がインストールされている場合、PM iSeries に対してこの構成を実行します。

使用したい送信方式の実装が完了したなら、次に 119 ページの『PM iSeries の管理』を実行するためのその他のタスクを行うことができます。

**エクストリーム・サポートでの Service Agent による PM iSeries<sup>(TM)</sup> データの送信 (ユニバーサル・コネクション):** PM iSeries は、収集サービスを使用して、所有権の付いていないパフォーマンス・データおよびキャパシティー・データをサーバーから収集します。このデータを収集した後、エクストリーム・サポートで Electronic Service Agent<sup>(TM)</sup> を使用して、IBM<sup>(R)</sup> にデータを送信することができます。

これらの機能を利用するには、サーバー上に V5R1 または V5R2 か、あるいはユニバーサル・コネクションの修正を適用された V4R5 がインストールされていなければなりません。以下に、PM iSeries を構成するためのステップを示します。

1. 108 ページの『PM iSeries の活動化』  
データ収集機能を利用するには、PM iSeries を開始しなければなりません。
2. マネージメント・セントラル・ネットワークをセットアップします。  
どのサーバーがセントラル・システムであり、どのサーバーがエンドポイント・システムであるかを定義します。IBM にデータを送信する前に、このネットワーク階層を使用して、エンドポイント・システムから中央設置場所にデータを送信することができます。
3. IBM に接続し、ユニバーサル・コネクションを使用してデータを送信します。  
これは、マネージメント・セントラルが IBM に PM iSeries データを送信するために使用する接続で

す。以前のリリースでは、SNA 上で実行したエレクトロニック支援 (ECS) 接続を使用していました。ユニバーサル・コネクションを使用する際には、TCP/IP 上でデータを送信することができます。

- 『PM iSeries パフォーマンス・データの収集』します。  
マネージメント・セントラル・インベントリー機能を使用して、データを収集します。
- IBM へデータを送信します。  
Electronic Service Agent (マネージメント・セントラル階層内のエクストリーム・サポートの下で使用可能) を使用して、IBM にデータを送信します。Electronic Service Agent はユニバーサル・コネクションを使用します。

また、『SNA プロトコルでのデータの送信』することもできます。

PM iSeries を構成し終わったら、119 ページの『PM iSeries の管理』を実行するためのその他のタスクを行う準備ができました。

**PM iSeries パフォーマンス・データの収集:** 以下のタスクを完了した場合、マネージメント・セントラルを使用して、PM iSeries<sup>(TM)</sup> パフォーマンス・データを収集することができます。

- 108 ページの『PM iSeries の活動化』
- ユニバーサル・コネクションの構成
- マネージメント・セントラル・ネットワークのセットアップ
- Electronic Service Agent がシステム上にインストールされているか、またはシステムからアクセス可能かどうかの確認。

エンドポイント・システムまたはシステム・グループ上に PM iSeries パフォーマンス・データを収集するには、以下のステップを実行します。

- iSeries ナビゲーターで、「マネージメント・セントラル」を展開します。
- 「エンドポイント・システム」または「システム・グループ」を展開します。
- エンドポイント・システムまたはシステム・グループを右クリックして、「インベントリー」を選択します。
- 収集を選択します。
- 収集する 1 つまたは複数のインベントリーを選択します。この場合、「PM iSeries パフォーマンス・データ (PM iSeries performance data)」を選択します。
- 収集の完了時にセントラル・システム上でアクションを実行する場合には、リストからアクションを選択します。
- すぐにデータの収集を開始する場合は「OK」をクリックします。あるいは「スケジュール」をクリックして、データの収集をいつ実行するかを指定します。

サーバーを構成し終わったら、119 ページの『PM iSeries の管理』を実行するためのその他のタスクを行う準備ができました。

**SNA プロトコルでのデータの送信:** 109 ページの『エクストリーム・サポートでの Service Agent による PM iSeries<sup>(TM)</sup> データの送信 (ユニバーサル・コネクション)』を利用しないことにした場合でも、やはり文字ベースのインターフェースを使用してデータを送信することができます。PM iSeries<sup>(TM)</sup> は、サーバーの構成および使用に関する一連の質問を尋ねてきます。「PM eServer iSeries の構成 (Configure PM eServer iSeries)」画面では、サーバーが PM iSeries パフォーマンス・データを送受信する方法に関する質問を尋ねられます。プロセスの最初の部分には、ネットワークのセットアップが含まれています。2 番目の部分は、データの送信方法に関するものです。文字ベースのインターフェースを使用する際には、直接ダイヤル回線を使用してデータを送信することができます。

SNA を使用してデータを送信するには、以下のタスクを行います。

1. 108 ページの『PM iSeries の活動化』します。  
データ収集機能を利用するには、PM iSeries を開始しなければなりません。
2. 使用するネットワーク構成を選択します。

データの送信に使用するネットワーク構成を決定します。118 ページの『PM iSeries 用の直接ダイヤル回線の設定』、既存のインターネット・サービス・プロバイダー (ISP)、または仮想私設ネットワーク (VPN) を使用することによって、IBM<sup>(R)</sup> に接続する方法を選択します。ISP または VPN を使用する場合には、ユニバーサル・コネクションを構成しなければなりません。

直接ダイヤル回線を使用して IBM にデータを報告することにした場合、ネットワークを構成する方法としていくつかの選択項目があります。ご使用のネットワークに適切な構成を選択し、「PM eServer iSeries の構成 (Configure PM eServer iSeries)」画面からその特定の構成について概説しているステップを実行します。

- 『単一サーバー用の PM iSeries ネットワーク』として、IBM に直接そのデータを送信します。
  - 112 ページの『ホスト・サーバー用の PM iSeries ネットワーク』として、ご使用のサーバーが他のサーバー (リモート・サーバー) からパフォーマンス・データを受信してから、IBM にデータを転送します。ホスト・サーバーは、その他のサーバーより前のリリース・レベルであってはなりません。つまり、ホスト・サーバーは、その他のサーバーと同じかまたはそれ以降のリリース・レベルでなければなりません。
  - 113 ページの『リモート・サーバー用の PM iSeries ネットワーク』として、パフォーマンス・データをホスト・サーバーに送信することができます。「PM eServer iSeries の構成 (Configure PM eServer iSeries)」画面で、リモート・サーバーが必要であることを確認し、PM eServer iSeries メニューのオプション 5 (「リモート iSeries システムでの処理 (Work with remote iSeries systems)」) を使用してリモート・サーバーを定義します。
3. 113 ページの『リモート・サーバーでの処理』  
ご使用のネットワークをホスト・サーバー用にセットアップすることにした場合、ホスト・サーバーにそれらのデータを送信するサーバーを識別する必要があります。単一サーバーまたはリモート・サーバーを使用する場合には、このステップを無視することができます。
  4. 116 ページの『PM iSeries のカスタマイズ』  
ネットワークを構成した後、PM iSeries ソフトウェアの操作用のグローバル・パラメーターを設定する必要があります。直接ダイヤル回線で IBM に接続する場合には、PM iSeries データ電話番号を定義する必要があります。

サーバーを構成し終わったら、119 ページの『PM iSeries の管理』を実行するためのその他のタスクを行う準備ができました。

**単一サーバー用の PM iSeries ネットワーク:** 単一サーバーは、IBM<sup>(R)</sup> に直接そのデータを送信します。以下に示すステップは、PM iSeries<sup>(TM)</sup> がデータを収集し、SNA 上でデータを送信する場合にのみ、単一サーバー用に PM iSeries を構成するために実行しなければなりません。サーバーの「PM eServer iSeries の構成 (CFGPM400) (Configure PM/400 (CFGPM400))」画面から:

1. コマンド行に **CFGPM400** と入力します。
2. 「IBM へのパフォーマンス・データの送信 (Send performance data to IBM)」フィールドに \*YES を指定します。
3. 「パフォーマンス・データの受信 (Receive performance data)」フィールドに \*NO を指定します。
4. QMPGDATA のデフォルトのライブラリーを受け入れます。

5. 「IBM へのパフォーマンス・データの送信 (Send performance data to IBM)」フィールドに \*YES を指定した場合、該当する通信オブジェクトが存在するかどうかを示す追加情報が表示されます。そのようなオブジェクトが存在しないと、伝送用の通信オブジェクトが PM iSeries によって作成されます。その他の表示に対しては、適宜応答してください。
6. 「連絡先情報の処理 (Work with Contact Information)」画面に会社の連絡先情報を入力します。

単一サーバーのセットアップは必要ないと判断した場合、別の 110 ページの『SNA プロトコルでのデータの送信』を選択することができます。

サーバーを構成し終わったら、119 ページの『PM iSeries の管理』を実行するためのその他のタスクを行う準備ができました。

**ホスト・サーバー用の PM iSeries ネットワーク:** ホスト・サーバーは、その他のサーバーからパフォーマンス・データを受信してから、IBM<sup>(R)</sup> にデータを転送します。以下に示すステップは、PM iSeries<sup>(TM)</sup> がデータを収集し、SNA 上でデータを送信する場合にのみ、ホスト・サーバー用に PM iSeries を構成するために実行しなければなりません。

1. ホスト・サーバー上の「PM eServer<sup>(TM)</sup> iSeries の構成 (Configure PM eServer iSeries)」画面から
  - コマンド行に **CFGPM400** と入力します。
  - 「IBM へのパフォーマンス・データの送信 (Send performance data to IBM)」フィールドに \*YES を指定します。
  - 「パフォーマンス・データの受信」フィールドに \*YES を指定します。
  - QMPGDATA のデフォルトのライブラリーを受け入れます。
2. ホスト・サーバー上の「リモート iSeries システムでの処理 (Work with Remote iSeries Systems)」画面から、次のようにします。
  - F6 (作成) を押して、どのサーバーがホスト・サーバーにそれらのデータを送信するかを識別します。
  - フィールドを完成させて、Enter を押します。

注: 以下の状態は、PM iSeries がデータを収集し、SNA 上でデータを送信する場合にのみ生じます。ネットワーク・システムを利用している場合、iSeries ナビゲーターにおいてユニバーサル・コネクションとマネージメント・セントラルを使用して、それらのシステム用のデータの収集と伝送を行うことをお勧めします。

PM iSeries は、1 次サーバーから IBM へのデータの送信を、データがリモート・サーバーから受信される翌日に、自動的にスケジュールします。自動スケジュールリングがご使用の作業管理体系に合わない場合には、1 次サーバーからデータの送信を手動でスケジュールすることができます。

ここに、データの送信をスケジュールするときに覚えておく必要のあるヒントがあります。週全体で、1 次サーバーへのデータの送信を均等にスケジュールします。このアクションにより、1 次サーバーのパフォーマンスの影響を最小限にします。たとえば、12 のサーバーから成るネットワークで、4 つのシステムの 3 つのグループを持っているとします。月曜日、水曜日、および金曜日にデータを送信するようにそれぞれのグループをスケジュールすることができます。これにより、1 次サーバーに送信されるデータの量は均等に分散されます。

ホスト・サーバーのセットアップは必要ないと判断した場合、別の 110 ページの『SNA プロトコルでのデータの送信』を選択することができます。

サーバーを構成し終わったら、119ページの『PM iSeries の管理』を実行するためのその他のタスクを行う準備ができました。

**リモート・サーバー用の PM iSeries ネットワーク:** リモート・サーバーは、ホスト・サーバーにそのパフォーマンス・データを送信します。以下に示すステップは、PM iSeries<sup>TM</sup> がデータを収集し、SNA 上でデータを送信する場合にのみ、リモート・サーバー用に PM iSeries を構成するために実行しなければなりません。リモート・サーバー上の「PM eServer iSeries の構成 (Configure PM eServer iSeries)」画面 (CFGPM400) で、次のようなステップを行います。

1. コマンド行に **CFGPM400** と入力します。
2. 「**IBM へのパフォーマンス・データの送信 (Send performance data to IBM)**」フィールドに \*NO を指定します。
3. 「**パフォーマンス・データの受信 (Receive performance data)**」フィールドに \*NO を指定します。
4. QMPGDATA のデフォルトのライブラリーを受け入れます。

注: ネットワーク・システムを利用している場合、iSeries ナビゲーターのインベントリー機能を使用してデータを収集して、ユニバーサル・コネクションを介してそれらのシステム用のデータを伝送することをお勧めします。

リモート・サーバーのセットアップは必要ないと判断した場合、別の 110 ページの『SNA プロトコルでのデータの送信』を選択することができます。

サーバーを構成し終わったら、119ページの『PM iSeries の管理』を実行するためのその他のタスクを行う準備ができました。

**リモート・サーバーでの処理:** サイトによっては、処理に必要なパフォーマンス・データはネットワーク内のホスト・サーバーから IBM<sup>®</sup> に送信されます。ホスト・サーバー・ネットワークを使用すると、ネットワーク内の他のサーバーがこのホスト・サーバーにパフォーマンス・データを送信して IBM に送信します。ホスト・サーバーを使用するようにネットワークをセットアップするには、その他のリモート・サーバーを識別し、それらのデータ送信用のスケジュールを設定しなければなりません。「リモート iSeries<sup>TM</sup> システムでの処理 (Work with Remote iSeries Systems)」画面では、その他のサーバーを定義することができます。

注:

1. リモート・サーバーまたは単一サーバーとしてネットワークをセットアップしている場合、この画面を使用する必要はありません。PM iSeries がデータを収集し、SNA 上でデータを送信する場合にのみこのタスクを実行します。
2. ネットワーク・システムを利用している場合、iSeries ナビゲーターのインベントリー機能を使用してデータを収集して、ユニバーサル・コネクションを介してそれらのシステム用のデータを伝送することをお勧めします。

リモート・サーバーを定義するには、以下のステップを実行します。

1. コマンド行で **GO PM400** と入力します。
2. PM eServer<sup>TM</sup> iSeries メニューで 5 (リモート ISERIES システムの処理) と入力して、Enter を押しします。最初はリモート・サーバーは表示されません。新規リモート・ロケーションを作成する必要があります。
3. F6 (作成) を押して、新規リモート・ロケーションを作成します。
4. 以下の情報に関する値を記録します。ネットワーク属性の表示 (DSPNETA) コマンドを使用して、リモート・システムからこれらの値を表示します。

- ローカル・ネットワーク ID
- デフォルトのローカル・ロケーション

「リモート iSeries システムでの処理 (Work with Remote iSeries Systems)」画面は、リモート・サーバーのリストを表示します。このリストには、サーバーの状況 (活動状態または非活動状態) およびそれぞれのサーバーごとの説明が含まれています。

5. 「PM eServer iSeries リモート・サイトの保守 (PM eServer iSeries Remote Site Maintenance)」画面または「リモート・サイト iSeries の変更 (Change Remote Site iSeries)」画面を使用して、リモート・サイト・サーバーに関する説明を作成または変更します。リモート・ロケーション名は、すべてのリモート・サーバーを通して固有でなければなりません。

PM iSeries は、1 次サーバーから IBM へのデータの送信を、データがリモート・サーバーから受信される翌日に、自動的にスケジュールします。自動スケジュールリングがご使用の作業管理体系に合わない場合には、1 次サーバーからデータの送信を手動でスケジュールすることができます。データの送信を手動でスケジュールするには、『120 ページの『PM iSeries を使ったジョブのスケジュール』』を参照してください。

PM iSeries ソフトウェアは、データを受信するサーバー (ホスト・サーバー) とデータを送信するサーバー (リモート・サーバー) との間で拡張分散ネットワーク機能 (APPN) リンクを定義してあることを前提とします。システム値 QCRTAUT (デフォルトの共通認可の作成) が \*EXCLUDE または \*USE に設定されている場合、制御装置記述を定義する方法については、『116 ページの『PM iSeries 用の装置記述の作成』』を参照する必要があります。ネットワークがこれらの前提事項を満たしていない場合、それぞれのリモート・サーバーへの接続をサポートするための装置のペアの作成については、『『非 APPN ネットワーク内のリモート・サーバーの処理』』を参照してください。

リモート・サーバーを定義し終わったら、特定の回線接続を使用するために 116 ページの『PM iSeries のカスタマイズ』を行う準備ができています。

**非 APPN ネットワーク内のリモート・サーバーの処理:** 1 次サーバーは、その他のサーバーから PM iSeries<sup>(TM)</sup> データを受信してから、IBM<sup>(R)</sup> にデータを送信します。リモート・サーバーは、1 次サーバーに PM iSeries データを送信します。以下の情報は、参照している制御装置が前もって定義されていることを前提としています。

PM iSeries がデータを収集し、SNA 上でデータを送信する場合にのみ、それぞれのリモート・サーバーへの接続をサポートするための装置のペアを作成する必要があります。

1. 装置記述の作成 (APPC) (CRTDEVAPPC) コマンドを使用します。リモート・サーバー上で、CRTDEVAPPC と入力します。F4 を押してパラメーターを入力するように求め、以下の情報を持つ値を定義します。

#### リモート・システム

DEV(D(Q1PLOC)	装置記述の名前を指定します。
RMTLOCNAME(Q1PLOC)	リモート・ロケーションの名前を指定します。
ONLINE(*YES)	システムの始動または再始動時にこの装置がオンラインに変更されるかどうかを指定します。
LCLLOCNAME(Q1PRMxxx)	ローカル・ロケーション名を指定します。Q1PRMxxx は、1 次サーバーの RMTLOCNAME と一致します。xxx は、それぞれのリモート・ロケーションごとに固有です。

CTL(yyyyyy)	接続された制御装置の名前を指定します。 yyyyyy は、1次サーバーに接続する制御装置です。
MODE(Q1PMOD)	モード名を指定します。
APPN(*NO)	装置が APPN 対応であるかどうかを指定します。

2. 1 次サーバーに関する以下の情報を指定します。コマンド行に、CRTDEVAPPC と入力します。 F4 を押してパラメーターを入力するように求め、以下の情報を持つ値を定義します。

### 1 次サーバー

DEV(D)(Q1PRMxxx)	装置記述の名前を指定します。ここで使用されている名前は、リモート・システムの装置記述名と一致します。
RMTLOCNAME(Q1PRMxxx)	リモート・ロケーションの名前を指定します。ここで使用されている名前は、リモート・サーバーの LCLLOCNAME 値と一致します。 xxx は、それぞれのリモート・ロケーションごとに固有です。
ONLINE(*YES)	システムの始動または再始動時にこの装置がオンラインに変更されるかどうかを指定します。
LCLLOCNAME(Q1PLOC)	ローカル・ロケーション名を指定します。この値は、リモート・サーバーの RMTLOCNAME と一致します。
CTL(aaaaaa)	接続された制御装置の名前を指定します。 aaaaaa は、リモート・サーバーに接続する制御装置です。
MODE(Q1PMOD)	モード名を指定します。
APPN(*NO)	装置が APPN 対応であるかどうかを指定します。

3. APPC 装置の定義が完了したなら、装置をオンに変更します (構成の変更 (VRYCFG) コマンド)。リモート・サーバー上で、VRYCFG と入力します。 F4 を押して、パラメーターを入力するように求めます。

### リモート・システムをオンに変更

CFGOBJ(Q1PLOC)	構成オブジェクトを指定します。
CFGTYPE(*DEV)	構成オブジェクトのタイプを指定します。
STATUS(*ON)	状況を指定します。

4. PM e Series<sup>TM</sup> iSeries メニューでオプション 5 を入力して、リモート・サーバーとして Q1PRMxxx を追加します。リモート・サーバーの追加方法については、『113 ページの『リモート・サーバーでの処理』』を参照してください。

これで PM iSeries の構成が完了しました。PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries 用の装置記述の作成:** 以下のステップでは、それぞれのリモート・サーバーで、デフォルトの共通認可の作成 (QCRTAUT) システム値が \*EXCLUDE または \*USE に設定されている必要があります。QUSER が装置記述 Q1PLOC への \*CHANGE 権限を持っていない場合、リモート送信は失敗します。これらのステップでは、装置が自動的に作成または削除されないようにします。

注: このタスクは、PM iSeries<sup>(TM)</sup> がデータを収集し、SNA 上でデータを送信する場合にのみ必要です。

装置を自動的に作成できる場合、装置記述は、QCRTAUT に設定されている値に応じて、PUBLIC \*EXCLUDE または \*USE 権限付きで作成されます。装置を自動的に作成または削除できるかどうかは、制御装置によって制御されます。これらのパラメーターがシステム上で定義される方法を決定するには、以下のコマンドを参照してください。

- 制御装置記述の作成 (APPC) (CRTCTLAPPC) コマンド
- 制御装置記述の変更 (APPC) (CHGCTLAPPC) コマンド
- 制御装置記述の表示 (DSPCTLD) コマンド

APPN を使用するよう構成されていないシステムの場合、装置記述の作成方法については、『114 ページの『非 APPN ネットワーク内のリモート・サーバーの処理』』を参照してください。

以下の情報は、ホスト・サーバーと通信するために使用される制御装置がリモート・サーバー上で前もって定義されていることを前提とします。

リモート・サーバー上で、装置記述 Q1PLOC を再作成します。

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

DLTDEVD     DEVD(Q1PLOC)

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

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

VRFCFG      CFGOBJ(Q1PLOC)
             CFGTYPE(*DEV)
             STATUS(*ON)
```

これで PM iSeries の構成が完了しました。PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries のカスタマイズ:** 「PM eServer<sup>(TM)</sup> iSeries<sup>(TM)</sup> カスタマイズの処理 (Work with PM eServer iSeries Customization)」画面では、以下の機能を提供します。

#### PM iSeries ソフトウェアの操作のグローバル・パラメーターの設定

グローバル・パラメーターにより、以下の項目をカスタマイズすることができます。以下のフィールドの説明については、オンライン・ヘルプを参照してください。

- 優先順位の限界
- 傾向およびシフト・スケジュール
- パフォーマンス・データ・ライブラリー
- 除去仕様

### PM iSeries データ電話番号の定義

米国およびカナダ以外ではデータの受け取り先の IBM<sup>(R)</sup> の所在地の電話番号を PM iSeries に指定しておかなければなりません。ほとんどのロケーションでは、PM iSeries の構成を開始すると PM iSeries はユーザーのロケーションに合った正しいデータ電話番号の選択を試みます。

### 118 ページの『PM iSeries での回線のオンおよびオフの変更』

「PM eServer iSeries 回線制御 (PM eServer iSeries Line Control)」画面では、PM iSeries を使用して、回線をオフに変更し、PM iSeries データを送信してから、回線を接続保留状態に戻すことができます。

グローバル・パラメーターをカスタマイズするには、以下のステップを実行します。

1. コマンド行で **GO PM400** と入力します。
2. 「PM iSeries カスタマイズの処理 (Work with PM iSeries Customization)」を表示するには、PM iSeries メニューで a 3 と入力して、Enter を押します。

収集サービスを使用して PM iSeries データを収集する場合には、107 ページの『PM iSeries のデータ収集に関する考慮事項』に配慮する必要があります。

PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries データ番号の確認:** サーバーが IBM<sup>(R)</sup> への直接ダイヤル接続を使用している場合、PM iSeries<sup>(TM)</sup> 電話番号が正しいことを確認する必要があります。電話番号には、ユーザー回線用の正しい接頭部が含まれていなければなりません。

注: これは、SNA 送信にのみ当てはまります。

エレクトロニック支援の電話番号の形式をチェックするには、以下のステップを実行します。

1.
 

DSPDTAARA DTAARA(QUSRSYS/QESTELE)

と入力して Enter キーを押します。
2. オフセット 0 にある接続番号の接頭部を判別します。たとえば、オフセット 0 が **'T9:1800xxxxxxx'** である場合、接頭部は **T9:** です。
3.
 

DSPDTAARA DTAARA(QUSRSYS/Q1PGTELE)

と入力して Enter キーを押します。
4. オフセット 0 (ゼロ) が、使用するダイヤリング・ストリングです。(他の番号は使用しません。)
5. ECS 回線を使用して PTF をオーダーする場合、オフセット 0 (ゼロ) のフォーマットを、ECS 回線用に使われるフォーマットである CALL QESPHONE に対して比較し、使用するストリングを書き留めて、それをステップ 2 にある値と比較することができます。

電話番号は異なっても、接頭部は同じ（つまり、SST9:1800...、SST:1800... など）でなければなりません。

電話番号を変更する必要がある場合、データ域の変更 (CHGDTAARA) コマンドを次のように使用します。

**CHGDTAARA** と入力します。ただし DTAARA は Q1PGTELE、LIB は QUSRSYS、サブストリングの開始位置は \*ALL、そして New の値は 'SST:18005475497' です。

注: New の値は、各自のダイヤル接頭部の後に 18005475497 (米国およびカナダの場合) が続いている値でなければなりません。

これで PM iSeries 構成が完了しました。次に実行できるタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries 用の直接ダイヤル回線の設定:** ほとんどのロケーションでは、PM iSeries<sup>(TM)</sup> は、ユーザーのロケーションに合った正しいデータ電話番号を選択しようとします。常に、117 ページの『PM iSeries データ番号の確認』をする必要があります。PM iSeries データ電話番号および PM iSeries サポート番号を含む情報を持っていない場合には、IBM<sup>(R)</sup> サポート担当者に連絡してください。IBM サポート担当者が正しい電話番号をお知らせします。

注: この電話番号は、ユニバーサル・コネクションを介してデータを送信する場合には必要ではありません。この電話番号は、直接ダイヤル回線を使用している場合にのみ必要です。

PM iSeries データ電話番号を定義するか、または PM/400 データ電話番号を変更するには、以下のステップを実行します。

1. コマンド行で **GO PM400** と入力します。
2. 「PM eSeries<sup>(TM)</sup> カスタマイズの処理 (Work with PM iSeries Customization)」を表示するには、PM iSeries メニューで a 3 と入力して、Enter を押します。
3. この画面で、電話番号フィールドを示す画面のセクションが表示されるまで前方にスクロールします。
4. 「IBM PM eSeries 電話番号 (IBM PM eSeries iSeries phone number)」フィールドに正しいダイヤル順序を入力します。多くの IBM モデムの場合、ダイヤル音用のコロン (: ) 文字を使用する必要があります。

**PM iSeries での回線のオンおよびオフの変更:** 時折、PM iSeries<sup>(TM)</sup> が使用する回線が接続保留状態になることがあります。この状態では、PM iSeries が回線にアクセスしてデータを送信することができません。「PM eServer<sup>(TM)</sup> iSeries 回線制御 (PM eServer iSeries Line Control)」画面では、PM iSeries を使用して、回線をオフに変更し、データを送信してから、回線を接続保留状態に戻すことができます。この画面を使用すると、PM iSeries 送信タスク (Q1PCMI) を変更して回線状態をチェックしたり、該当する回線をオフに変更することができます。いったん送信が完了すると、同じ回線は接続保留状態に入れられます。

注: このタスクは、PM iSeries がデータを収集し、SNA 上でデータを送信する場合にのみ必要です。

回線をオンおよびオフに変更するには、以下のステップを実行します。

1. コマンド行で **PMLINMON** と入力して、PM iSeries 回線モニター機能を開始します。「PM eServer iSeries 回線制御 (PM eServer iSeries Line Control)」画面が表示されます。
2. 最初の画面上に示されている警告を読んでから、Enter を押します。
3. PM iSeries がオフに変更する必要がある回線、制御装置、および装置の組み合わせを定義します。

4. 機能のマスター制御スイッチとして、プロンプト **Do you want PM iSeries automatic line control active?** を使用します。 **YES** を指定する場合、PM iSeries 機能は活動状態になります。 **NO** を指定する場合、機能は使用不可になります。

**NO** を指定する場合、 **YES** を指定したときに再び回線制御リストを定義する必要はありません。回線だけを指定することによって、回線をオンおよびオフに変更することができます。 3 つの説明すべてを指定することによって、回線、制御装置、および装置をオンおよびオフに変更することができます。

5. 定義した回線、制御装置、および装置を確認します。 **Enter** を押して、選択項目の要約を表示します。
6. **Enter** を押して選択項目を確認するか、**F12** を押して、項目を変更するための前の画面に戻ります。

PM eServer iSeries の構成 (CFGPM400) コマンドを使用して、PM/400 回線制御を設定することもできます。

PM iSeries で実行できる付加的なタスクについては、『『PM iSeries の管理』』を参照してください。

## PM iSeries の管理

PM iSeries<sup>(TM)</sup> を使用するようにネットワークをセットアップした後、以下のタスクを実行することができます。

### 『PM iSeries の非活動化』

PM iSeries を停止する方法について学習します。

### 120 ページの『PM iSeries 連絡先情報の変更』

連絡先情報の元の設定を変更する方法について学習します。

### 120 ページの『PM iSeries を使ったジョブのスケジュール』

PM iSeries を使用してジョブをスケジュールする方法を学習します。

### 121 ページの『PM iSeries 分析での項目の省略』

PM iSeries を使用して分析を実行するときに、ジョブ、ユーザー、および通信回線を省略する方法について学習します。

### 122 ページの『PM iSeries の暫時オフ』

PM iSeries を暫時停止する方法について学習します。

### 122 ページの『PM iSeries の状況の表示』

iSeries ナビゲーターまたは PM eServer<sup>(TM)</sup> iSeries メニューを使用して、PM iSeries 状況を表示する方法について学習します。

### 123 ページの『PM iSeries 報告書の表示』

PM iSeries 報告書の例およびそれらの報告書を解釈する方法に関する説明が示されます。

### 103 ページの『グラフ・ヒストリー』

グラフ・ヒストリーには、指定の期間中に収集されたパフォーマンス・データがグラフィック表示で示されます。このデータを表示する方法が解説されています。

**PM iSeries の非活動化:** PM iSeries<sup>(TM)</sup> の実行を停止するには、以下の方法のいずれかを使用することができます。

**iSeries ナビゲーターを使用する場合**

以下のステップを実行します。

1. iSeries ナビゲーターで、PM iSeries を実行しているシステムを拡張表示します。
2. 「構成およびサービス」を展開します。
3. 「収集サービス」を右クリックします。
4. 「PM eServer iSeries」を選択します。
5. 停止を選択します。
6. PM iSeries を停止したいシステムを選択します。
7. 「OK」をクリックします。

## API の使用

PM eServer iSeries の終了 (Q1PENDPM) API を使用して PM iSeries を非活動化します。

PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries 連絡先情報の変更:** PM iSeries<sup>(TM)</sup> ソフトウェアの構成中に、連絡先の担当者を識別し、所属の組織に関するメール情報が入力されています。後で情報を更新する必要がある場合には、「連絡先情報の処理 (Work with Contact Information)」オプションを使用します。連絡先情報を変更するには、以下のステップを実行します。

1. コマンド行で **GO PM400** と入力します。
2. PM eServer<sup>(TM)</sup> iSeries メニューで 1 と入力して、Enter を押します。「連絡先情報の処理 (Work with Contact Information)」画面が表示されます。
3. 連絡先情報を変更し、適当であれば Enter を押します。

PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries を使ったジョブのスケジュール:** PM iSeries<sup>(TM)</sup> ソフトウェアに不可欠なのは、PM iSeries パフォーマンス・データの収集および分析をサポートするのに必要なジョブを自動的に開始するスケジューラーです。

PM iSeries ソフトウェア活動化プロセスの一環として、Q1PSCH というジョブも開始することになります。次にこのジョブは、以下の表に示されている他のジョブを開始します。

PM iSeries のスケジュール・ジョブにアクセスするには、次のようにします。

1. コマンド行で **GO PM400** と入力します。
2. PM eServer<sup>(TM)</sup> iSeries メニューで 2 と入力して、Enter を押します。「自動スケジュールされたジョブの処理 (Work with Automatically Scheduled Jobs)」画面が表示されます。
3. 各ジョブの状況をアクティブから非アクティブに変更することができます。変更したいジョブの隣に 2 (変更) を入力します。「自動スケジュールされたジョブの変更 (Change Automatically Scheduled Jobs)」画面が示されます。

以下の表は、選択できる PM iSeries ジョブを一覧で示しています。

PM iSeries スケジュール済みジョブ		
ジョブ	スケジュール	機能

Q1PTEST	活動化時	PM iSeries が活動化されてから非活動化状態になることを確認します。
Q1PCM1	毎週	削減されたパフォーマンス・データを IBM <sup>(R)</sup> に送信します。このジョブは、直接ダイヤル回線を使用している場合にのみ活動状態です。
Q1PCM2	毎日	通信をオフラインに変更します。
Q1PPMSUB	1 時間ごと	収集サービスがデータを収集していることを確認します。
Q1PDR	毎日	データ削減を実行し、パフォーマンス・データを除去します。
Q1PPG	毎月	削減されたパフォーマンス・データを除去します。
Q1PCM3	必要時	直接ダイヤル送信が回線をオフに変更できなかった後で、通信をオフラインに変更します。
Q1PCM4	必要時	リモート・サーバーから PM iSeries データにアクセスします。このジョブが開始するのは、PM iSeries メニューでオプション 5 を使用してリモート・システムを追加していた場合のみです。
Q1PPMCHK	4 時間ごと	データ収集が活動状態であることを確認します。
Q1PMONTH	毎月	その月中に送信を追加する必要がある場合に毎月送信できるようにします。デフォルト値は、非活動状態に設定されます。このジョブは、直接ダイヤル回線を使用している場合にのみ使用可能です。

PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries 分析での項目の省略:** PM iSeries<sup>(TM)</sup> ソフトウェア・アプリケーションの要約には、バッチ・ジョブ、ユーザー、および通信回線に関する上位 10 項目の分析が含まれています。ただし、ジョブ、ユーザー、または通信回線によってはそのような分析に該当しないものがあります。たとえば、実行時カテゴリー内の通常より長い実行時間を設定したジョブ (自動開始ジョブなど) を除外することができます。

汎用除外機能を使用することによって上位 10 項目の分析からバッチ・ジョブおよびユーザーのグループを省略することができます。たとえば、MYAPP で始まるすべてのジョブを省略するには、MYAPP\* を指定します。

省略項目を処理するには、以下のステップを実行します。

1. コマンド行で **GO PM400** と入力します。
2. PM eServer<sup>(TM)</sup> iSeries メニューで 4 と入力して、Enter を押します。「上位 10 項目の省略の処理 (Work with Top Ten Omissions)」画面が表示されます。
3. 省略したい項目によって適切なオプション番号を入力します。

- ジョブを処理するには 1 を入力します。
  - ユーザーを処理するには 2 を入力します。
  - 通信回線を処理するには 3 を入力します。
4. 該当するフィールドに 1 を入力すると、特定の 카테고리からユーザーまたはジョブのどちらかを省略します。通信回線の場合には、回線の名前を入力してから、該当するフィールドに 1 を入力します。

PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries の 暫時オフ:** 収集サービスがデータを収集していることを確認するために PM iSeries<sup>(TM)</sup> を停止する必要がある場合、スケジューラー・ジョブを使用して、日付を Q1PPMSUB ジョブを行う今後の日付に変更することができます。

1. コマンド行で **GO PM400** と入力します。
2. 2 (自動的にスケジュール済みジョブを処理する) と入力します。
3. Q1PPMSUB ジョブの隣に 2 (変更) と入力します。
4. 今後の日付および時刻に日付または時刻を変更します。
5. Enter を押します。この変更により、収集サービスがデータを収集していることを確認するために PM iSeries を一瞬停止します。現在収集されているものを終了する必要があります。

注: Q1PPMSUB ジョブに設定された日付および時刻に達するまで、PM iSeries は収集サービスを開始、循環、または変更しません。

スケジューラーを使用して行うことのできるその他の事柄については、『120 ページの『PM iSeries を使ったジョブのスケジュール』』を参照してください。

PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries の 状況の表示:** iSeries<sup>(TM)</sup> ナビゲーターまたはサーバー上の PM eServer<sup>(TM)</sup> iSeries メニューのいずれかを使用して、PM iSeries の状況を表示することができます。「IBM<sup>(R)</sup> Performance Management/400 for eServer iSeries 状況 (IBM Performance Management for eServer iSeries Status)」ダイアログを使用して、1 つ以上のサーバーまたはグループ上の PM/400 の状況全体を表示します。たとえば、PM iSeries が活動状態であるかどうかに関する詳細が表示されます。PM iSeries メニューを使用して、収集サービス状況、PM iSeries スケジューラー状況、パフォーマンス・データのリリース、最後の送信の試み、パフォーマンス・データ・メンバー、およびパフォーマンス・データ・サイズを表示します。

iSeries ナビゲーターから PM iSeries の状況全体を表示するには、以下のステップを実行します。

1. iSeries ナビゲーターのエンドポイント・システムまたはシステム・グループを拡張表示します。
2. 「構成およびサービス」を展開します。
3. 「収集サービス」を右クリックします。
4. 「Performance Management eServer iSeries」を選択します。
5. 状況を選択します。

PM iSeries メニューから PM iSeries の詳細状況を表示するには、以下のステップを実行します。

1. コマンド行で **GO PM400** と入力します。
2. コマンド行から 6 と入力して、Enter を押します。それぞれのフィールドの説明については、オンライン・ヘルプを参照してください。

PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

**PM iSeries 報告書の表示:** PM iSeries<sup>TM</sup> 関連の出力は、月単位または四半期単位の管理報告書およびグラフのセットです。PM iSeries オファリングの報告書には、2 つの 107 ページの『PM iSeries 製品の操作サポート・サービス』があります。

報告書およびグラフの目的は、サーバーの現行パフォーマンスおよび正確な拡張傾向を管理者が明確に理解できるようにすることです。それぞれの報告書およびグラフを詳しく表示し、それらの利点および使用方法を確かめるには、PM iSeries の Web サイト  を参照してください。

PM iSeries で実行できるその他のタスクについては、『119 ページの『PM iSeries の管理』』を参照してください。

## PM iSeries 報告書

サーバーは、通常の実行時に、そのオペレーティング環境についてのさまざまな統計を自動的に記録します。36 ページの『収集サービス』には、それらの統計を整理統合する機能があります。105 ページの『IBM Performance Management for eServer iSeries』はそれらの統計を収集して IBM<sup>®</sup> に送信しますが、これは生成されるすべての PM iSeries<sup>TM</sup> 報告書の基礎になります。Web 上で表示したり印刷するためにこれらの報告書を生成するには、PM iSeries を活動化して、これらの統計を IBM に少なくとも月に一度、できればさらに頻繁に送信する必要があります。

報告書およびグラフの目的は、サーバーの現行パフォーマンスおよび正確な拡張傾向を管理者が明確に理解できるようにすることです。それぞれの報告書およびグラフを詳しく表示し、それらの利点および使用方法について学習するには、PM eServer iSeries Web サイト  をご覧ください。

## Performance Tools

Performance Tools for iSeries<sup>TM</sup> ライセンス・プログラムを使用すると、パフォーマンス・データをさまざまな方法で分析することができます。Performance Tools は、パフォーマンス・データの表示、報告、およびグラフ化を行うためのツールとコマンドを集めたものです。Performance Tools for iSeries を使用することにより、36 ページの『収集サービス』によって収集されたパフォーマンス・データを表示したり、パフォーマンス・トレースの開始 (STRPFTRC) コマンドによって収集されたトレース・データを表示することができます。そしてデータを報告書に要約して、システムのパフォーマンス上の問題を調べることができます。パフォーマンス・データのグラフを作成して、ある時間帯の資源の使用率を調べることもできます。

Performance Tools for iSeries には、基本プロダクトと 2 つのフィーチャー (マネージャーとエージェント) があります。基本に加えていずれかのフィーチャーが必要です。Performance Tools のマネージャーおよびエージェント・フィーチャーについての詳細は、『125 ページの『マネージャーおよびエージェント・フィーチャーの比較』』トピックを参照してください。

### 124 ページの『Performance Tools の概念』

パフォーマンス情報の収集と分析に役立つさまざまなツールを説明しています。具体的にどのツールにどの機能があって、それらがどのように動作するかという詳細情報があります。

### 130 ページの『Performance Tools のインストールと構成』

このトピックにはインストールとセットアップの手順が記載されています。

### 131 ページの『Performance Tools 報告書』

Performance Tools 報告書には、ある時間帯に収集されたデータの情報が示されます。この報告書を使用して、システム資源のパフォーマンスと使用状況についての追加情報を得ることができます。

システム、ジョブ、またはプログラムのパフォーマンスに関するデータを Performance Tools を使用して収

集する方法の詳細は、Performance Tools for iSeries  を参照してください。そこでは、問題の識別と訂正に役立つ、データの分析と印刷の方法についても説明しています。

## Performance Tools の概念

Performance Tools for iSeries<sup>(TM)</sup> ライセンス・プログラムは、サンプル・データとトレース・データという別個の 2 つのタイプのパフォーマンス・データを分析します。36 ページの『収集サービス』はサンプル・データを収集します。サンプル・データは一定の時間間隔で取り込まれる要約データのことです。サンプル・データは、傾向分析やパフォーマンスの分析を行うために収集されます。このデータは、記憶域プールや応答時間といった事柄と関係があります。しかし、収集サービスは、トレース・データの収集をサポートしていません。トレース・データというのは、特定のジョブやトランザクションについての付加的な情報を得るために収集される詳細データです。トレース・データを収集するには、パフォーマンス・トレースの開始 (STRPFTRC) コマンドか Performance Explorer を使用します。

### 『Performance Tools が提供する機能』

Performance Tools には、パフォーマンス・データの収集、分析、および報告のための種々のアプリケーションが組み込まれています。特定のタスクに対してどの機能が使用可能でどれが最適かを識別するのは複雑です。このライセンス・プログラムに組み込まれている機能の説明は、このトピックを参照してください。

### 125 ページの『マネージャーおよびエージェント・フィーチャーの比較』

マネージャー・フィーチャーとエージェント・フィーチャーを使用して、分散環境での Performance Tools の必要な機能を効果的に分割することができます。このトピックでは、これらの 2 つのフィーチャー、それぞれに含まれる機能、およびそれらを最も効果的に使用方法について記述されています。

### 126 ページの『Performance Tools プラグイン』

システム資源の使用率のデータを iSeries ナビゲーターで見ることができます。データの表示、そのデータのグラフ化および報告書への要約を行うことができます。この機能の利用方法については、このトピックを参照してください。



### 127 ページの『CPU 使用率の報告』

仮想プロセッサで消費される合計 CPU が報告される方法を調べます。

### 129 ページの『構成済みキャパシティーのレポート』

構成済みキャパシティーの情報が記録される場所を調べます。

### 130 ページの『5250 オンライン・トランザクション処理 (OLTP)』

5250 オンライン・トランザクション処理の意味、およびこの作業に関連するジョブまたはスレッドを調べます。 <<

**Performance Tools が提供する機能:** Performance Tools は、報告書、対話式コマンド、およびその他の機能などで構成されています。Performance Tools にはたとえば以下のものが含まれます。

## ツール

システム活動の処理  
(WRKSYSACT) コマンド

126 ページの『Performance Tools  
プラグイン』

131 ページの『Performance Tools  
報告書』

グラフィックス機能

134 ページの『Performance  
Explorer』

## 説明

システム活動の処理 (WRKSYSACT) コマンドを使用すると、現在システムで実行されているジョブ、スレッド、およびタスクを対話式で処理することができます。 WRKSYSACT コマンドは、共用処理プールを使用する区画でのタスクごとの CPU 使用量などのシステム・リソースの使用状況を報告します。

iSeries<sup>TM</sup> ナビゲーター内から「パフォーマンス・データの表示 (Display Performance Data)」グラフィカル・ユーザー・インターフェースを使用して、パフォーマンス・データを表示し、データをレポートに要約し、傾向を表示するためのグラフを表示し、システム・パフォーマンスの詳細を分析します。

報告書は、論理的で役立つ形式で、収集サービス・パフォーマンス・データを編成します。報告書については、 Performance Tools for iSeries  に詳しく説明されています。

Performance Tools のグラフィックス機能を使用すると、グラフ様式でパフォーマンス・データを処理することができます。グラフを対話式に表示したり、印刷や作図を行うことができます。また、データを図形データ形式 (GDF) ファイルに保管し、他のユーティリティーで使用することができます。このツールについて

は、 Performance Tools for iSeries  に詳しく説明されています。

Performance Explorer はデータ収集のツールであり、これを用いて、収集サービスによって収集されたサンプル・データや、一般的な傾向分析によっては識別できないパフォーマンスの問題の原因を究明することができます。 Performance Explorer を使用して、プログラム、プロシージャ、モジュール、またはメソッド・レベルで詳細なアプリケーション分析を行ってください。たとえば、個々のプログラムまたはプロシージャ CPU および入出力統計、または個々のオブジェクト入出力の特性に関するトレース・データを収集することができます。この

ツールについては、 Performance Tools for iSeries  に詳しく説明されています。

**マネージャーおよびエージェント・フィーチャーの比較:** Performance Tools は、別々にインストール可能な 2 つのフィーチャーで使用できます。このトピックでは、ユーザーのアプリケーションにとってどちらのフィーチャーが適しているかを判断できるように、その 2 つのフィーチャーの違いを説明します。

### マネージャー・フィーチャー

Performance Tools マネージャー・フィーチャーは、分散環境での中央側システムまたは単一システムでの使用を目的とした全機能を持つパッケージです。トレース・データの分析、データのグラフ化表示、システムの活動のリアルタイム表示、またはシステムの拡大の管理と追跡が必要な場合は、 Performance Tools ライセンス・プログラムのマネージャー・フィーチャーの方が役に立ちます。

### エージェント・フィーチャー

Performance Tools エージェント・フィーチャーはマネージャー機能のサブセットを持ち、より基本的な機能を備えた低価格パッケージです。分散環境では、詳細な分析が必要な場合はデータをマネージャーに送信できるため、エージェント・フィーチャーはネットワーク内の管理対象システム用に適しています。適度なレベルの自己完結性が必要だがエキスパートのスキルを利用できないサイトにとっては、これも有効なツールです。

Performance Tools のエージェント・フィーチャーには、パフォーマンス・データの収集、管理、オンライン表示、データ削減、および分析を単純化する機能があります。 134 ページの『Performance Explorer』の報告機能とその関連コマンドは、 Performance Tools for iSeries<sup>TM</sup> ライセンス・プログラムの基本オプションに組み込まれているため、マネージャー・フィーチャーまたはエージェント・フィーチャーで使用す

ることができます。エージェント・フィーチャーに含まれていない Performance Tools の主要機能は、パフォーマンスとトレースの報告、パフォーマンス・ユーティリティ (ジョブ・トレース、ファイルおよびアクセス・グループの選択)、システム活動のモニター、およびパフォーマンス・グラフです。

**Performance Tools プラグイン:** Performance Tools は、iSeries<sup>™</sup> ナビゲーターのプラグインである、「パフォーマンス・データの表示 (Display Performance Data)」グラフィカル・ユーザー・インターフェース (GUI) からパフォーマンス・データを表示できます。GUI から、パフォーマンス・データを表示し、データを報告書に要約し、傾向を示すグラフを表示し、システム・パフォーマンスの詳細を分析することができます。

### メトリック

iSeries ナビゲーターは選択した時間間隔でのパフォーマンス・メトリックを表示します。「パフォーマンス・データの表示 (Display Performance Data)」GUI の「グラフ」ペインに表示できるパフォーマンス・メトリックには、次のものがあります。

- トランザクション・カウント
- トランザクション応答時間
- 合計 CPU 使用率
- 対話型 CPU 使用率
- バッチ CPU 使用率
- 対話型機能使用率
- 高ディスク使用率
- マシン・プール・ページ不在/秒
- ユーザー・プール・ページ不在/秒
- 例外

「詳細」ペインには、選択した時間間隔での詳細なパフォーマンス・データをさまざまな方法で表示することができます。システム・パフォーマンスを分析するために、ジョブ・データ、サブシステム・データ、プール・データ、またはディスク装置のデータを表示できます。

### 報告書

「パフォーマンス・データの表示 (Display Performance Data)」GUI からは、グラフと詳細データの表示だけでなく、報告書の印刷も行えます。パフォーマンス上の問題の原因になっているシステムの領域をパフォーマンス報告書から調べることができます。さまざまな報告書を実行してシステム資源が使われている場所を調べることができます。Performance Tools の報告書を印刷できるのは、Performance Tools for iSeries (5722-PT1) のオプション 1 (マネージャー・フィーチャー) がセントラル・システムにインストールされている場合のみです。マネージャー・フィーチャーについての詳細は、『125 ページの『マネージャーおよびエージェント・フィーチャーの比較』』トピックを参照してください。

「パフォーマンス・データの表示 (Display Performance Data)」GUI から印刷できる報告書には、次のものがあります。

- システム
- 構成要素
- ジョブ
- プール
- 資源

## iSeries ナビゲーターを介したアクセス

「パフォーマンス・データの表示 (Display Performance Data)」GUI は、iSeries ナビゲーターのプラグインです。すでにプラグインをインストールしてある場合、以下のステップを実行して iSeries ナビゲーターからアクセスできます。

1. iSeries ナビゲーターで「**ユーザー接続**」(またはアクティブ環境)を展開します。
2. 表示したいパフォーマンス・データがあるサーバーを展開します。
3. 「**構成およびサービス**」を展開します。
4. 「**収集サービス**」を右クリックして「**Performance Tools**」を選択し、「**パフォーマンス・データ (Performance Data)**」を選択します。
5. 表示したいパフォーマンス・データ・ファイルを選択します。
6. 「**表示 (Display)**」をクリックします。

iSeries ナビゲーターでの「パフォーマンス・データの表示 (Display Performance Data)」GUI の使用方法についての詳細は、iSeries ナビゲーターのオンライン・ヘルプを参照してください。

**CPU 使用率の報告:** ➤ V5R3 より前のバージョンでは、プロセッサ使用率は使用可能な CPU 時間のパーセンテージとして計算されてきました。収集サービスが、各プロセッサが使用した時間と、経過した間隔の時間を、パフォーマンス・データベース・ファイルに報告していました。このデータのユーザー (Performance Tools レポートおよび表示など) は、消費されたシステム CPU の合計を得るために、各プロセッサで使用した時間を合計する必要がありました。使用可能な CPU 時間は、区画内のプロセッサの数を、データ収集間隔の所要時間で乗算して計算されてきました。最後に、計算された使用可能時間で CPU 時間を除算し、使用率のパーセンテージを得ていました。

以前の方法の問題点は、データのユーザーすべてが仮想プロセッサ全体を想定していたことと、構成されたキャパシティーに変更がないことが前提となっていた、という点です。この方法では、部分的なプロセッサ・キャパシティーおよび動的構成を実行する機能を持つ論理区画を計算することはできなくなりました。これらの問題の影響を最小限に抑えるための一時的な解決策としては、システム・プロセッサの使用率を拡大縮小して整数個のプロセッサの報告のようにすることや、構成が変更されたら収集サービスを循環させることがありました。個々のジョブ CPU 時間は拡大縮小されないで、追加の時間は HVLPTASK で消費されたものと報告することで計上されました。HVLPTASK タスクは実際には CPU を使用していなかったものの、計算の都合上、CPU 時間は HVLPTASK によって消費されたものとして示されました。実際のジョブが実行した作業量を、HVLPTASK に負わされた CPU 時間で拡大縮小し、その結果として、実行された顧客作業量に対する直接の比率としてのシステム CPU 使用率 (パーセント) が、0 から 100 までの範囲で得られました。

V5R3 の収集サービスでは、消費される合計 CPU と、間隔の間に区画で使用可能な合計 CPU が報告されます。共用プロセッサ環境における、HVLPTASK や、CPU を仮想プロセッサ全体に拡大縮小するという概念はありません。収集サービスは、構成が変更された場合も収集を循環しなくなりました。

収集サービスは、構成されている仮想プロセッサ数や、構成されている区画単位数、およびこれらが間隔中にどのように変更されるかにかかわらず、区画で消費される合計プロセッサ時間と、区画内で消費可能だったプロセッサ時間の量とを報告するようになりました。このデータのユーザーは、報告された消費 CPU を、使用可能なキャパシティーで除算して、使用率を計算します。CPU 使用率をこの方法で計算することにより、使用可能な CPU 時間を計算するという間違いを起こしがちなタスクを除去することができます。こうした新しいメトリックで計算される CPU 使用率は、存在する処理単位 (全体または断片) の数、いつ処理単位が変更されるか、または単位が変更される頻度に関係なく正確です。

CPU 使用率の計算にこの変更が加えられた理由がいくつかあります。1 つの理由は、拡大縮小を使用すると、ジョブまたはジョブのグループの使用率が、予期していたよりもずっと小さく見えるということです。

以下の例でこの概念を示します。別の理由は、構成変更によって CPU 報告が無効になるということです。従来、CPU の数は収集開始時に構成された値に基づいており、それを変更するには IPL が必要でした。動的構成が採用されたとき、変更は頻繁に生じないとの前提のもと、収集サービスは収集を循環して構成の変更を処理するようになりました。しかし、変更が頻繁になると、循環も頻繁に発生します。変更が頻繁すぎると、パフォーマンス・データの収集ができません。最後の理由として、各間隔で報告および使用される構成データが正しくても、間隔が開始してから完了するまでの時間に何が起きたかはわからないということがあります。構成の変更が 1 つ以上行われた間隔では、依然として不正確な使用率の計算が行われる恐れがありました。

## 例

区画 A のキャパシティーは 0.3 プロセッサ単位で、1 つの仮想プロセッサを使用するように定義されています。収集間隔の時間は 300 秒です。システムは CPU を 45 秒使用します (対話式ジョブが 15 秒、バッチ・ジョブが 30 秒)。この例では、使用可能な CPU 時間は 90 秒です (300 秒の 30%)。合計 CPU 使用率は 50% になります。

V5R3 より前の場合は、数が拡大縮小され、システム CPU 使用量は 150 秒と報告されます。150 秒を間隔の時間の 300 秒で除算した結果、使用率は 50% になります。対話式の使用率は、15 秒を 300 秒で除算して、5% です。バッチの使用率は、30 秒を 300 秒で除算して、10% です。HVLPTASK は 35% の使用率を担います (150 秒から 45 秒を減算)。つまり、105 秒を 300 秒で除算したものです。これらのパーセンテージを合計すると、50% になります。

V5R3 以降、使用量の 45 秒は拡大縮小されず、そのまま報告されます。報告された消費 CPU 時間を、報告された使用可能なキャパシティーで除算すると、計算された CPU 時間は 50% になります (45 秒を 90 秒で除算)。対話式の使用率のパーセンテージは 17% (15 秒を 90 秒で除算) です。バッチの使用率のパーセンテージは 33% (30 秒を 90 秒で除算) です。

リリース	Total CPU (合計 CPU)	対話式	バッチ	HVLPTASK
OS/400 <sup>(R)</sup> V5R2 以前	50%	5%	10%	35%
OS/400 V5R3 以降	50%	17%	33%	N/A

## 考慮事項

V5R3 では、パフォーマンス・データの変換 (CVTFPRDTA) コマンドは正常に実行されます。しかし、変換されたファイルのデータは、拡大縮小されていないシステム CPU データ (QAPMSYSCPU データベース・ファイル) と一貫するように変更されます。結果は、V5R3 システムでデータが収集された場合と同じになるはずですが、データは以前のリリースのファイルに存在していた値とは異なります。

CPU 使用率を計算するための既存の未変更のツールでは、共用プロセッサ区画や、データ収集中に構成の変更があった区画の正しい結果が表示されません。これには、パフォーマンス・データベースを使用するツールも、QPMLPFRD API を使用するツールも含まれます。

V5R3 管理収集オブジェクト (\*MGTCOL) を前のリリースにコピーして、データベース・ファイルを生成することができます。ただし、以下のことに注意する必要があります。

- 報告される CPU データは拡大縮小されていない状態のもので (共用プロセッサ環境)。つまり、仮想プロセッサ (Performance Tools を含む) を使ったツールが報告する合計システム CPU は正しくありません。
- 構成の変更をまたぐ管理収集オブジェクト (\*MGTCOL) では、変更後の間隔の CPU のパーセンテージの計算が不正確になります。◀

**構成済みキャパシティーのレポート:** ➤ 区画キャパシティー値は、区画の構成プロファイルを使用して、Hardware Management Console (HMC) を介して区画が開始されるときに最初に決定され、そのときに使用可能なキャパシティー資源に依存します。これらの初期値は、区画の活動中に構成変更を使って変更できません。論理区画について詳しくは、共用プロセッサ・プールのトピックを参照してください。

論理区画により、いくつかの区画は特定の状態で、構成済みキャパシティーを超えることができます。このようなときは、これらの区画のプロセッサ使用率メトリックは、構成済みキャパシティーの 100% を超えても構いません。

使用法およびキャパシティー情報は、QAPMSYSTEM データベース・ファイルに記録されます。仮想プロセッサ情報は、QAPMSYSCPU データベース・ファイルに記録されます。以下の値は、この情報を要約したものです。

#### **仮想プロセッサ**

共用プロセッサ・プール内のプロセッサ・キャパシティーを共有している論理区画に割り当てられる、プロセッサの数。この値は、論理区画で並行して活動状態にすることが可能なプロセッサの数を決定します。この値は、SCTACT という欄で、iSeries<sup>(TM)</sup> パフォーマンス・データベース・ファイルに含まれます。

#### **使用可能なプロセッサ・プール・キャパシティー**

共用プロセッサの論理区画が使用するための、共用プロセッサ・プールの合計プロセッサ・キャパシティー。この値は、SYSPLA という欄で、iSeries パフォーマンス・データベース・ファイルに含まれます。

上限なしとして構成された区画が、保証された量を超過することで使用可能な共用プール・キャパシティーと競合する場合、プロセッサ・キャパシティーの分布は、論理区画に割り当てられる上限なしのウェイトによって決定します。

#### **使用される共用プロセッサ・キャパシティー**

活動中のすべての共用プロセッサ論理区画によって使用されている、共用プロセッサ・キャパシティーの合計量。プールを共有するすべての区画が、共用プール内で使用する CPU の合計。この値は、SYSPLU という欄で、iSeries パフォーマンス・データベース・ファイルに含まれます。

#### **区画が保証するキャパシティー**

共用プロセッサ・プールから共用プロセッサ論理区画に構成されるプロセッサのキャパシティー。この値は、SYSCTA という欄で、iSeries パフォーマンス・データベース・ファイルに含まれます。構成される 5250 OLTP キャパシティーは、SYIFTA という欄に記録されます。

#### **区画プロセッサ使用率**

論理区画によるプロセッサ使用率。上限なしのキャパシティーを持つ共用プロセッサの論理区画では、共用プロセッサ・プールに未使用のキャパシティーがある場合、この値が、保証されたキャパシティーを超えることがあります。この値は、SYSPTU という欄で、iSeries パフォーマンス・データベース・ファイルに含まれます。使用される 5250 OLTP キャパシティーは、SYIFUS という欄に記録されます。

区画の最大のプロセッサ・キャパシティーは、構成される仮想プロセッサの数で決定されます。

#### **区画が使用できるキャパシティー**

論理区画が使用できる可能性のあるプロセッサ・キャパシティーの量。この値は、SYSUTA という欄で、iSeries パフォーマンス・データベース・ファイルに含まれます。これは使用されるプロセッサ・キャパシティー (SYSPTU) に共用プロセッサ・プール (SYSPLA) の未使用のキャパシティーを加えたもので、次の制限があります。

- 最小値は構成済み (保証された) キャパシティーである
- 最小値は区画およびプールに割り当てられる仮想プロセッサの数に基づくキャパシティーである



**5250 オンライン・トランザクション処理 (OLTP):**  5250 対話式およびバッチ・タイプ処理の定義は、V5R3 で変更されました。CPU 使用率およびシステム・リソース使用量についての記述では、5250 対話式という用語の代わりに、5250 オンライン・トランザクション処理 (OLTP) という用語が使用されます。対話式という用語は、ユーザー入力および関連する応答という形のアプリケーション・インターフェースを指す、より汎用的な説明であるという事実を認識して、新しい用語が使用されています。その結果、対話式という用語は、次の事柄を指す場合があります。

- 5250 セッション、パススルー・ジョブ、または Telnet ジョブを介した、なじみのある iSeries<sup>(TM)</sup> 対話。
- Domino<sup>(R)</sup> メール、予定表、またはブラウザー・ベースのアプリケーションからの、ワークステーション・ベースの要求。
- 従来から対話式とみなされてきたワークロード (DDM など)。

V5R3 より前のバージョンでは、すべての OS/400<sup>(R)</sup> ジョブは、いくつかのジョブ属性 (DDM、CA4、および INT など) に基づき、10 のカテゴリーのうち 1 つに分類されます。Performance Tools は、これらのカテゴリーを元に、これらのジョブを対話式またはバッチとして報告します。

かつて、いくつかのカテゴリーのプロセッサ使用率の一部は、対話式とバッチの両方に属しているものとして表示されていたので、対話式およびバッチの CPU 使用率を示すためのこのアプローチは意味がありませんでした。たとえば、iSeries Access ジョブは、機能に応じて対話式およびバッチの両方を使用します。V5R3 より以前のバージョンでは、これらのジョブは CA4 カテゴリーに組み込まれ、対話式としてリストされていました。DDM サーバー・ジョブも対話式としてリストされていました。

V5R3 からは、CPU サイクルを担うプロセッサ・キャパシティー機能に応じてワークロードの分散を改善するためのアプローチが、Performance Tools ライセンス・プログラムにインプリメントされています。対話式 CPU レポートとは、CPU が 5250 OLTP プロセッサ・キャパシティーに割り振られたジョブを意味するようになりました。そのため、iSeries Access ジョブは、Performance Tools レポートの適切なセクションにリストされることとなります。さらに、DDM ジョブは、レポートの対話式ワークロード・セクションから、非対話式ワークロード・セクションに移ります。



## Performance Tools のインストールと構成

Performance Tools をインストールするには、システムの保管 (\*SAVSYS) 権限があるユーザー・プロファイルが必要です。システム・オペレーター・プロファイルを使用してこの権限を得ることができます。

Performance Tools は QPFR という名前のライブラリー内で実行する必要があります。ご使用のシステムにこの名前のライブラリーがある場合は、Performance Tools をインストールする前に、オブジェクトの名前変更 (RNMOBJ) コマンドを使用してそれを名前変更します。このステップによって Performance Tools を正しく操作できるようになります。

次のコマンドを使用して Performance Tools をライブラリー QPFR 内に置きます。

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

次に、以下のいずれかを実行します。

- マネージャー・フィーチャーを購入している場合は、次のコマンドを使用します。

```
RSTLICPGM LICPGM(5722PT1) DEV(tape-device-name) OPTION(1)
```

- エージェント・フィーチャーを購入している場合は、次のコマンドを使用します。

```
RSTLICPGM LICPGM(5722PT1) DEV(NAME) OPTION(2)
```

インストールする CD-ROM が数枚ある場合は、次の状態になることがあります。最初の 1 枚をインストールした後、ライセンス・プログラムは復元されたが言語オブジェクトが復元されていないというメッセージが出されることがあります。このような場合は次の CD-ROM を挿入して、以下を入力します。

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

Performance Tools プログラムをインストールする別の方法として、GO LICPGM と入力してメニュー・オプションを使用する方法があります。

Performance Tools はプロセッサ・ベースのプログラムです。使用タイプは「同時」で、このプログラムは使用法制限 \*NOMAX としてインストールされます。

このプログラムについては、Performance Tools for iSeries  に詳しく説明されています。

## Performance Tools 報告書

Performance Tools には、収集されたデータの調査を容易にする手段が用意されており、パフォーマンス問題を分離します。ある時間帯のパフォーマンス・データを収集した後、報告書を 133 ページの『パフォーマンス報告書の印刷』してシステム資源の使われ方と使われている場所を調べることができます。全体の応答時間が遅くなっている原因の特定のアプリケーション・プログラム、ユーザー、または非効率なワークロードが、報告書から分かります。

36 ページの『収集サービス』は、ほとんどの Performance Tools 報告書のデータにトランザクション、ロック、およびトレース報告書の例外を提供します。これらの 3 つの報告書の 22 ページの『アプリケーションのパフォーマンスについての情報の収集』を収集するには、STRPFTRC および ENDPFTRC コマンドを使用する必要があります。

以下は、各報告書の説明とそれぞれの報告書を使用する理由の簡単な概要、および各報告書のサンプルへのリンクです。さらに、各報告書については、Performance Tools for iSeries  に詳しく説明されています。

### Performance Tools 報告書の概要

報告書	説明	表示される内容	情報の使用方法
システム	収集サービスのデータを使用して、システムの稼働状態の概要を示します。この報告書には、ワークロード、資源の使用、記憶域プール使用率、ディスク使用率、および通信に関する要約情報が記載されます。この報告書をしばしば実行して印刷し、システムの使用に関して一般的に把握します。	システム・ワークロード。この報告書にはデータベース権限データが記載されず。	ワークロードの予測

構成要素	<p>収集サービスのデータを使用してシステム報告書の場合と同じシステム・パフォーマンス構成要素についての情報を示しますが、レベルが詳細になります。この報告書は、CPU やディスクなどのシステム資源を大量に消費しているジョブを見つけるのに役立ちます。</p>	<p>資源の使用、通信、システム、およびユーザー・ジョブ。この報告書には、データベース権限データおよび対話型のフィーチャー使用率も含まれます。</p>	<p>ハードウェアの拡張および構成処理の傾向</p>
トランザクション	<p>トレース・データを使用して、パフォーマンス・データ収集時に起きたトランザクションについての詳細情報を示します。</p>	<p>CPU のワークロードおよび使用率、ディスク、主記憶装置、トランザクション・ワークロード、オブジェクト競合</p>	<p>ワークロードの予測、プール構成、アプリケーション設計、ファイル競合、およびプログラムの使用</p>
ロック	<p>トレース・データを使用して、システム操作時のロックおよび占有の競合についての情報を示します。この情報をもとに、不十分なロック要求や内部のマシン占有の競合が原因でジョブの処理が遅れているかどうかを判別することができます。このような状態を待機とも呼びます。このような状態が起きている場合は、ジョブがどのジョブを待機しているかということと、待機の長さを判別することができます。</p>	<p>時間別のファイル、レコード、またはオブジェクト競合; 保留ジョブまたはオブジェクト名; 要求ジョブまたはオブジェクト名</p>	<p>問題分析。オブジェクト競合の低減または除去。</p>
バッチ・トレース・ジョブ	<p>トレース・データを使用して、時間を通してトレースされたさまざまなジョブ・タイプ (例: バッチ・ジョブ) の経過を示します。使用された資源、例外、および状態遷移が報告されます。</p>	<p>ジョブ・クラスのタイム・スライス終了およびトレース・データ</p>	<p>問題分析およびバッチ・ジョブの進行</p>

ジョブ間隔	収集サービスのデータを使用して、すべてのまたは選択した間隔およびジョブに関する情報を示します。これには対話式ジョブと非対話式ジョブの詳細および要約情報が含まれます。報告書が長くなる場合は、含めたい間隔とジョブを選択して出力を制限することもできます。	間隔別のジョブ	ジョブ・データ
プール間隔	収集サービスのデータを使用して、サブシステム活動のセクションとプール活動のセクションを示します。データはサンプル間隔ごとに示されます。報告書が長くなる場合は、含めたい間隔とジョブを選択して出力を制限することもできます。	間隔別のプール	プール・データ
リソース間隔	収集サービスのデータを使用して、すべてのまたは選択した間隔についての資源の情報を示します。報告書が長くなる場合は、含めたい間隔を選択して出力を制限することもできます。	間隔別のリソース	システム・リソースの使用

Performance Explorer と収集サービスは、別々の収集エージェントです。それぞれは、グループ化された収集データのセットを含む独自のデータベース・ファイルのセットを生成します。同時に両方のコレクションを実行することができます。

他のツールでの報告書のリストについては、次を参照してください。

- 140 ページの『Performance Explorer 報告書』
- 123 ページの『PM iSeries 報告書』

**パフォーマンス報告書の印刷:** 収集したパフォーマンス・データを使用して、報告書を印刷することができます。V5R1 より前のリリースでは、オプション 3 (パフォーマンス報告書の印刷) を選択すると、QAPMCONF ファイルにあったパフォーマンス・メンバーのリストが表示されました。このリストには、パフォーマンス・モニター開始 (STRPFRMON) コマンドによって収集されたサンプル・データと追跡データの両方が表示されていました。収集サービスは、追跡データを収集しません。ただし、STRPFRTRC および TRCINT コマンドを使用すれば、追跡データを収集することができます。このデータは、QAPMDMPT ファイルにあります。したがって、V5R1 以降では「パフォーマンス報告書の印刷」画面にサンプル・データ用と追跡データ用の 2 つの画面があります。

**注:** 追跡データとサンプル・データが両方とも現行ライブラリーにある場合は、F20 を使用すると、2 つの「パフォーマンス報告書の印刷」画面間で切り替えをすることができます。

データを収集した後で管理収集 (\*MGTCOL) オブジェクトに保管されているパフォーマンス情報から、パフォーマンス・データ・ファイルのセットを作成する必要があります。パフォーマンス・データ作成 (CRTPFRDTA) コマンドを使用します。データ・ファイルが作成されたら、報告書の印刷を要求できます。

収集サービスを用いて収集したサンプル・データの報告書を印刷するには、以下のコマンドを使用します。

- システム報告書印刷 (PRTSYSRPT)
- 構成要素報告書印刷 (PRTCPTRPT)
- ジョブ間隔報告書印刷 (PRTJOBTRPT)
- プール報告書印刷 (PRTPOLRPT)
- リソース報告書印刷 (PRTRSCRPT)

パフォーマンス追跡開始 (STRPFRTRC) コマンドおよび内部追跡 (TRCINT) コマンドを用いて収集したサンプル・データの報告書を印刷するには、以下のコマンドを使用します。

- トランザクション報告書印刷 (PRTTNSRPT)
- ロック報告書印刷 (PRTLCKRPT)
- ジョブ追跡報告書印刷 (PRTTRCRPT)

注: トランザクション報告書を印刷するには、パフォーマンス追跡終了 (ENDPFRTRC) コマンドを使用して、パフォーマンス追跡データの収集を停止し、オプションでパフォーマンス追跡データをデータベース・ファイルに書き込む必要があります。

## Performance Explorer

Performance Explorer はデータ収集ツールであり、これを使用すると、収集サービスを使用したデータの収集や一般的な傾向分析では特定できないパフォーマンスの問題の原因を特定することに役立ちます。

Performance Explorer を使用する理由として次の 2 つがあります。

- パフォーマンス上の問題を問題の原因になっている、システム資源、アプリケーション、プログラム、プロシージャ、またはメソッドに分離する。
- アプリケーションのパフォーマンスを分析する。

Performance Explorer の収集機能および関連コマンドは、OS/400<sup>(R)</sup> ライセンス・プログラムの一部です。報告機能とその関連コマンドは、Performance Tools for iSeries<sup>(TM)</sup> ライセンス・プログラムの基本オプションの一部であるため、マネージャー・フィーチャーまたはエージェント・フィーチャーで使用することがで

きます。AS/400<sup>(R)</sup> Performance Explorer Tips and Techniques  には、Performance Explorer の機能の補足例と拡張 Performance Explorer のトレース・サポートの例が記載されています。

Performance Explorer は、一般的なパフォーマンスのモニターを行うツールを使用しても特定できないパフォーマンスの問題の原因を見つけるのに役立つツールです。コンピューター環境がサイズと複雑さの両面で拡大すると、当然パフォーマンス分析も同様に複雑になります。Performance Explorer は、複合したパフォーマンスの問題に関するデータを収集することにより、そのような複雑さの拡大に対処しています。

注: Performance Explorer は、他のツールで試行した後で使用されるツールです。このツールは、パフォーマンスの問題に関与する要因を容易に切り分けることができる特定の形式のデータを収集しますが、そのデータを収集するときはシステムのパフォーマンスに著しい影響を与えることがあります。

このツールは、自身のプログラムのパフォーマンスの理解や改良に関心があるアプリケーション開発者を対象に設計されています。これはまた、パフォーマンス管理を十分理解しているユーザーが、複合したパフォーマンスの問題を識別して切り分ける場合にも役立ちます。

Performance Explorer についての詳細は、Performance Explorer に関する次のトピックを参照してください。

### 『Performance Explorer の概念』

Performance Explorer は、指定されたシステムのプロセスまたは資源についての詳細情報を収集します。このトピックでは Performance Explorer の動作と最適な使用方法を説明しています。

#### 141 ページの『Performance Explorer の構成』

詳細なトレース情報を収集するには、トレース対象のアプリケーション・プロセスと最適に連動できるように Performance Explorer を調整する必要があります。

#### 140 ページの『Performance Explorer 報告書』

Performance Explorer セッションでパフォーマンス・データを収集した後、そこに含まれる報告書を実行するか、またはデータベース・ファイルを直接照会してそのデータを表示することができます。

詳しくは、Performance Tools for iSeries  を参照してください。

## Performance Explorer の概念

Performance Explorer は、36 ページの『収集サービス』のように、あとで分析を行うためのデータを収集します。しかし収集するデータのタイプは大きく異なります。収集サービスは、システム資源の消費を最小限にして、一定のスケジュール間隔で広範囲のシステム・データを収集します。一方、Performance Explorer は、トレース・レベルのデータを収集するセッションを開始します。このトレースによって、アプリケーション、ジョブ、またはスレッドが消費する資源についての詳細情報が大量に生成されます。具体的には、Performance Explorer を使用して、システム生成のディスク入出力、プロシージャ呼び出し、

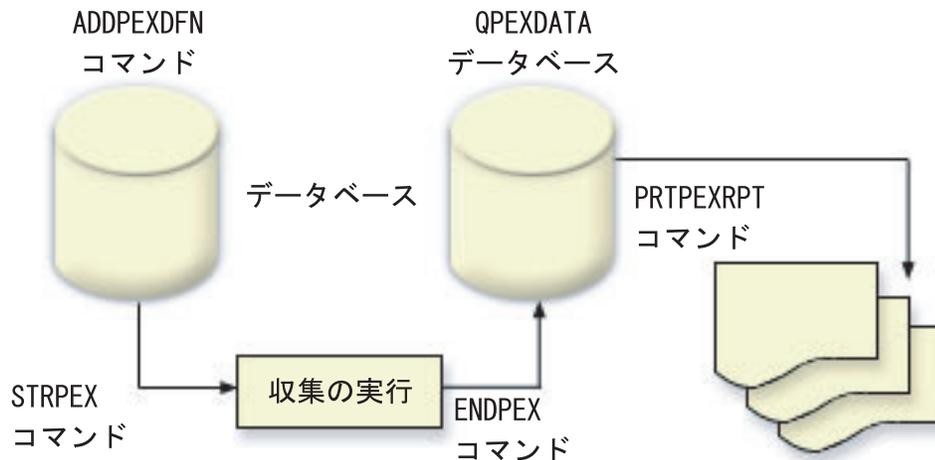
Java<sup>TM</sup> メソッド呼び出し、ページ不在、および他のトレース・イベント  のような領域についての特定の疑問を解くことができます。Performance Explorer でパフォーマンスの問題の切り分けを効率的に行えるのは、これが非常に限定された非常に詳細な情報を収集することができるためです。たとえば、収集サービスを使用すると、ディスク記憶域が急速に消費されているということが分かります。Performance Explorer を使用すると、ディスク・スペースの消費が多すぎるプログラムやオブジェクトとその理由を特定することができます。

注: Performance Explorer のデータと収集サービスのデータを同時に収集することができます。

### Performance Explorer の動作

次に示す図は、Performance Explorer での典型的なパスを示したものです。これらのステップのそれぞれの詳細は、『141 ページの『Performance Explorer の構成』』を参照してください。この図は基本動作サイクルを示しており、次のステップで構成されます。

1. Performance Explorer のデータ収集を定義します。特定のイベントについての比較値を指定することによりフィルターを追加して、収集するデータの量を制限することもできます。
2. Performance Explorer を開始して、定義に基づいたデータの収集を行います。
3. ユーザーのプログラム、コマンド、またはワークロードを実行します。
4. 収集を終了します。収集されたデータは一連のデータベース・ファイルに保管されます。
5. データベース・ファイルから報告書を作成して印刷します。



Performance Explorer についての詳細は、Performance Explorer に関する次のトピックを参照してください。

#### 『Performance Explorer の定義』

Performance Explorer が収集するデータとその収集方法を決定するパラメーターと条件は、Performance Explorer の定義を使用して構成して保管します。このトピックではそれらの定義の使用方法を説明し、簡単な定義のサンプルを示しています。

#### 139 ページの『Performance Explorer データベース・ファイル』

Performance Explorer が収集するデータは、Performance Explorer データベース・ファイルに保管されます。

#### 141 ページの『Performance Explorer の利点』

Performance Explorer には、詳細なパフォーマンス情報の収集と分析に役立つ数々の機能があります。このトピックではそれらのさまざまな機能の概要を説明しています。

**Performance Explorer の定義:** Performance Explorer データを収集するには、収集するデータについて Performance Explorer に通知する必要があります。これは、PEX 定義の追加 (ADDPEXDFN) コマンドを使用して Performance Explorer 定義を作成することにより行えます。定義が完了し、保管された後、作業のサイクルの次のタスクを続行することができます。

新しい定義の作成前に、必要な情報の種類、および必要な詳細事項の量を考慮してください。Performance Explorer は、次のタイプのデータ収集を提供します。

#### 統計タイプ定義

CPU を過剰に使用していたり、数多くのディスク入出力操作を実行しているアプリケーション、および IBM<sup>(R)</sup> プログラムまたはモジュールを識別します。通常、統計タイプは、潜在的なパフォーマンス・ボトルネックをさらに調査すべきプログラムを識別するのに使用します。

- OS/400<sup>(R)</sup> プログラム、プロシージャー、および MI 複合指示の最初のオーダー分析に適しています。
  - 呼び出しの数を指定する
  - インラインおよび累積 CPU 使用量の両方をマイクロ秒単位で指定する
  - 同期および非同期入出力のインラインおよび累積数を指定する

- 実行される呼び出しの数を指定する
- 短い、または長い実行に合うように作動します。
- 収集されるデータのサイズは、すべての実行に対してかなり小さく、一定しています。
- ILE プロシーチャーのランタイム収集オーバーヘッドは、呼び出しの頻度のため問題である可能性があります。ランタイムのレベルが下がっても、収集される統計は正確です。これは Performance Explorer が、データからほとんどの収集オーバーヘッドを除去するためです。
- 組み合わせられた、または個別にされたデータ域を使用します。ADDPEXDFN コマンドの MRGJOB パラメーターは、すべてのプログラム統計が 1 つのデータ域に累積されるか、それとも個別にされるか (例: 各ジョブごとに 1 つのデータ域) を指定します。

統計は、階層方式または水平方式のどちらかで構造化されます。

- 階層構造は、ツリー中の各ノードが、ジョブまたはタスクによって実行されるプログラム・プロシーチャーを表す呼び出しツリー形式に統計を編成します。
- 水平構造は、統計を、プログラムまたはプロシーチャーそれぞれが統計の独自のセットを持つように、単純リストに編成します。

この例は、MYSTATS という名前の Performance Explorer 統計定義の例で、プログラムまたはプロシーチャー・レベルごとに CPU およびディスク資源使用量を示します。

```
ADDPEXDFN DFN(MYSTATS) /* The name of the definition. */
TYPE(*STATS) /* The type of definition */
JOB(*ALL) /*All Jobs */
TASKS(*ALL) /*All tasks */
DTAORG(*FLAT) /* Do not keep track of who calls who */
```

#### プロファイル・タイプ定義

ソース・プログラム・ステートメント番号に基づき、CPU を過度に消費する高水準言語 (HLL) プログラムを識別します。プログラムの開始とプログラムの終了時のサブルーチンとの間で、絶えず分岐しているプログラムも識別できます。プログラムの大きさが十分なものである場合、この往復の繰り返しによって、主記憶域が限られているシステムにおいて、ページ不在率が極端に増えることがあります。

- プログラム・プロファイル (ADDPEXDFN コマンドで TYPE(\*PROFILE) および PRFTYPE(\*PGM) を指定する)
  - 特定のジョブ内の一連のプログラムで、時間がかかっている部分の明細を提供します。
  - プログラム、モジュール、プロシーチャー、ステートメント、または指示によって、データを要約できます。
  - 実行の長さに関係なく、収集のサイズはかなり小さく、一定です。
  - 16 MI プログラムの制限とは、これを 2 番目のオーダー分析ツールとして使用するべきであるということです。
  - サンプル間隔を変更することにより、オーバーヘッドが変更されます。ベンチマークには、2 ミリ秒の間隔が最初の選択として適しています。
  - 指定されるプログラムの数、または指定されるプログラムのサイズのため、画面区画サイズには制限がありません。

この例は、PGMPROF という名前の Performance Explorer プログラム・プロファイル定義の例で、特定のプロシーチャーの使用量を示します。

```

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

```

- ジョブ・プロファイル (ADDPDXDFN コマンドで、TYPE(\*PROFILE) および PRFTYPE(\*JOB) を指定する)
  - 一連のジョブまたはタスクの収集で、時間がかかっている部分の明細を提供します。
  - 収集のサイズは比較的小さいですが、一定ではありません。サイズは、実行が長くなればなるほど大きくなります。
  - システム上のすべてのジョブおよびタスクのプロファイルを作成できるか、または収集されるデータの範囲を数個のジョブまたは対象となるタスクだけに絞ることができます。
  - サンプル間隔を変更することにより、オーバーヘッドが変更されます。ベンチマークには、2 ミリ秒の間隔が最初の選択として適しています。

この例は、ALLJOBPROF という名前の Performance Explorer ジョブ・プロファイル定義の例で、すべてのジョブの使用量を示します。

```

ADDPDXDFN 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) /* 1-millisecond samples will be taken. */

```

### トレース定義

システム上の 1 つ以上のジョブによって生成されるパフォーマンス・アクティビティのヒストリー・トレースを収集します。トレース・タイプは、イベントがいつ発生したか、および発生した順序についての特定の情報を収集します。トレース・タイプは、プログラム、ライセンス内部コード (LIC) タスク、OS/400 ジョブ、およびオブジェクト参照情報についての、詳細な参照情報を収集します。

- 一部の一般的なトレース・イベントは次のとおりです。
  - プログラムおよびプロシージャー呼び出しおよび戻り。
  - 記憶域 (例: たとえば割り振りおよび割り振り解除)。
  - ディスク入出力 (例: 読み取り操作と書き込み操作)。
  - Java<sup>TM</sup> メソッド (例: 入り口と出口)。
  - Java (例: オブジェクト作成とガーベッジ収集)。
  - ジャーナル (例: コミットの開始とコミットの終了)。
  - 同期化 (例: mutex ロックとアンロック、またはセマフォ待機)。
  - 通信 (例: TCP、IP、または UDP)。
- 実行が長いと、より多くのデータを収集します。

この例は、DISKTRACE という名前の Performance Explorer トレース定義で、すべてのディスク・イベントの使用量を示します。

```

ADDPDXDFN 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. \*/

**Performance Explorer データベース・ファイル:** 次の表は、データ収集コマンドを使用したときにシステムによって収集される Performance Explorer のデータ・ファイルを示したものです。単一のファイルの内容を表示するには、ファイル・フィールド記述の表示 (DSPFFD) コマンドを次のように入力します。

DSPFFD FILE(xxxxxxxx)

xxxxxxxx は表示するファイルの名前です。

ファイルに含まれる情報のタイプ	ファイル名
参照情報	QAYPEREF
一般情報	QAYPERUNI
PMC 選択	QAYPEFQCFG
基本構成情報	QAYPECFG
収集されたマシン・インターフェース (MI) 複合命令	QAYPELCLPX
収集されたジョブ	QAYPELJOB
データ収集対象のメトリック	QAYPELMET
収集されたマシン・インターフェース (MI) プログラム、モジュール、またはプロシージャ	QAYPELMI
データ収集対象のライセンス内部コード (LIC) モジュール	QAYPELLIC
データ収集対象のタスク名	QAYPELNAMT
データ収集対象のタスク番号	QAYPELNUMT
マシン・インターフェース (MI) 複合命令のマッピング	QAYPEMICPX
イベント・タイプとサブタイプのマッピング	QAYPEEVENT
ハードウェア・マッピング・データ	QAYPEHWMAP
ライセンス内部コード (LIC) アドレス解決マッピング	QAYPEPROCI
セグメント・アドレス解決マッピング	QAYPESEGI
プロセスおよびタスク解決マッピング	QAYPETASKI
すべてのイベントについての共通トレース・データ	QAYPETIDX
補助記憶域管理イベント・データ	QAYPEASM
基本イベント・データ	QAYPEBASE
ディスク・イベント・データ	QAYPEDASD
ディスク・サーバー・イベント・データ	QAYPEDSRV
ページ不在イベント・データ	QAYPEPGFLT
資源管理プロセス・イベント・データ	QAYPERMPM
資源管理占有ロック・イベント・データ	QAYPERMSL
拡張 36 <sup>(R)</sup> イベント・データ	QAYPES36
セグメント・アドレス範囲 (SAR) データ	QAYPESAR
不明イベント・データ	QAYPEUNKWN
基本統計データ	QAYPESTATS
統計プロファイル要約データ	QAYPEPSUM
ライセンス内部コード (LIC) ブラケット化データ	QAYPELBRKT

ファイルに含まれる情報のタイプ	ファイル名
マシン・インターフェース (MI) ユーザー・イベント・データ	QAYPEMIUSR
マシン・インターフェース (MI) プログラム・ブラケット化データ	QAYPEMBRKT
マシン・インターフェース (MI) ポインターのアドレス	QAYPEMIPTR
ユーザー定義のブラケット化フック・データ	QAYPEUSRDF
ハードウェア・モニター・データ	QAYPEHMON
ハードウェア・モニター合計データ	QAYPEHTOT
リリース、バージョン、修正レベル	QRLVRM
Performance Explorer レベル標識	QRLVL
Performance Explorer Java <sup>(TM)</sup> イベント・データ	QAYPEJVA
Performance Explorer Java クラス情報データ	QAYPEJVICI
Performance Explorer Java メソッド情報データ	QAYPEJVMI
Performance Explorer Java 名前情報データ	QAYPEJVNI
同期イベント・データ	QAYPESYNC
通信イベント・データ	QAYPECMN
ファイル・サービス・イベント・データ	QAYPEFILSV
ヒープ・イベント・データ	QAYPEHEAP
PASE イベント・データ	QAYEPASE
トレース・ジョブ同等イベント・データ	QAYPETBRKT
タスク切り替えイベント・データ	QAYPETSWSW
同期イベント・データ	QAYPESYNC
プログラム・プロファイル・データ	QAYPEPPANE

**Performance Explorer 報告書:** Performance Explorer は、プログラムやジョブの動作とパフォーマンスについての詳細情報を収集して、その情報を 139 ページの『Performance Explorer データベース・ファイル』に保管します。これらのファイルは、SQL を使用して、またはさまざまな報告書のいずれかを実行することにより照会することができます。Performance Explorer で、統計、プロファイル、トレース、および基本の、4 つの異なる報告書を生成することができます。特定の定義を使用してこれらの報告書の 1 つを生成する理由については、『136 ページの『Performance Explorer の定義』』を参照してください。各報告書については、Performance Tools for iSeries  に詳しく説明されています。

Performance Explorer 報告書の作成と印刷は、Print Performance Explorer Report (PRTPEXRPT) コマンドを使用して行います。トレース報告書をカスタマイズする場合は、OUTFILE パラメーターを使用します。Performance Explorer の各タイプのデータの報告書を印刷する場合のコマンドの例を次に示します。

- \*STATS 報告書を、使用された CPU 時間別にソートして印刷する。

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

- プロファイル報告書を、プロシージャ別に要約して印刷する。

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

- トレースを、タスク ID 別に分類して印刷する。

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

Performance Explorer は収集したデータを QAVPETRCI ファイルに保管します。このファイルは QPFR ライブラリーに置かれています。単一レコードの内容を表示するには、次のコマンドを入力します。

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

**Performance Explorer の利点:** Performance Explorer は、iSeries<sup>(TM)</sup> サーバーにおいて詳細なパフォーマンス分析が必要な人に役立ちます。Performance Explorer を使用すると、次のことが可能です。

- ユーザー、ジョブ、ファイル、オブジェクト、スレッド、タスク、プログラム、モジュール、プロシージャー、ステートメント、または命令アドレスのレベルまで下りて、システムにおけるパフォーマンス上の問題の原因を判別できます。
- ユーザーが開発したソフトウェア、およびシステム・ソフトウェアに関するパフォーマンス情報を収集できます。
- システム上の他の操作のパフォーマンスに影響を与えずに、ある特定のジョブの詳細な分析を行えます。
- データの収集元でないシステム上で、そのデータを分析できます。たとえば、ネットワークの管理対象システムでデータを収集した場合、そのデータをセントラル・サイト・システムに送信して分析することができます。

## Performance Explorer の構成

Performance Explorer を構成するには、以下のステップを実行します。

1. iSeries<sup>(TM)</sup> サーバーに、収集するパフォーマンス・データを通知する 136 ページの『Performance Explorer の定義』します。「PEX 定義の追加 (ADDPEXDFN)」画面で、収集タイプ、および定義の名前を指定します。この定義は、ライブラリー QUSRSYS の QAPEXDFN ファイルに、その名前でデータベース・メンバーとして保管されます。指定される名前は、Performance Explorer の開始 (STRPEX) コマンドで使用されます。
2. フィルターを追加 (ADDPEXFTR コマンド) します。Performance Explorer フィルターは、Performance Explorer セッション中に収集されるパフォーマンス・データを識別し、特定のイベントに比較値を指定することによって収集されるデータの量を制限するためのものです。
3. データの収集を開始 (STRPEX コマンド) します。\*PMCO イベントが収集されていない場合、ジョブは複数の Performance Explorer に入っている可能性があります。\*PMCO イベントが収集されている場合、すべての収集の間隔指定が同じであれば (ADDPEXDFN INTERVAL() パラメーター)、ジョブが複数の収集に入っていることが考えられます。
4. 分析したいデータでコマンド、プログラム、またはワークロードを実行します。
5. 142 ページの『Performance Explorer の終了』し、分析を行うためにデータベース・ファイルに保管します。収集を停止するには、Performance Explorer の終了 (ENDPEX) コマンドを使用します。
6. パフォーマンス・データを分析します。Performance Tools ライセンス・プログラムに入っている Print Performance Explorer Report (PRTPEXRPT) コマンドは、データの各タイプ (統計、プロファイル、トレース・プロファイル、またはトレース) ごとに、固有のレポートを提供します。分析用の他のオプションは、データベース・ファイルのセット全体で、独自の Query を作成するためのものです。

すべての Performance Explorer コマンドは、以下のどちらかの方法でアクセスできます。

- コマンド・インターフェース。コマンド行でコマンドを入力します。Print Performance Explorer Report (PRTPEXRPT) コマンド以外のすべてのコマンドは、OS/400<sup>(R)</sup> オペレーティング・システムの一部です。
- Performance Tools のメニュー・オプション。

Performance Explorer の作業の周期を確認するには、135 ページの『Performance Explorer の概念』を参照してください。

## Performance Explorer の終了

Performance Explorer セッションを終了するには、Performance Explorer の終了 (ENDPEX) コマンドを使用します。ENDPEX コマンドは、収集されたデータ上で次の処置を取ります。

- 収集されたデータを、指定されたライブラリー中のファイル QAYPExxx に入れます。  
これには、OPTION(\*END) および DTAOPT(\*LIB) を使用します。すべての QAYPExxx ファイルのデータベース・メンバー名は、DTAMBR パラメーターに名前を指定しない限り、セッション名をデフォルトとして使用します。RPLDTA(\*NO) を指定すると、このセッション名を使用して収集されたデータを消去することができ、RPLDTA(\*YES) を指定すると、収集されたデータを既存のデータに追加することができます。このセッションにかなり慣れていない限り、RPLDTA(\*NO) を使用してください。
- 収集したデータを、単一の IBM<sup>(R)</sup> 定義のファイルに入れます。  
これには、OPTION(\*END) および DTAOPT(\*MGTCOL) を使用します。通常は、IBM サービス技術員の指示の下で、\*MGTCOL だけを使用します。DTAOPT パラメーターで \*MGTCOL 値を指定すると、収集情報が管理収集オブジェクトに保管されます。管理収集オブジェクト・オプションは、データが IBM に送られる場合のみ使用します。Performance Tools が分析できるのはデータベース・ファイルのみです。
- 収集したデータを廃棄します。  
データを保管したい場合は OPTION(\*END)、収集したデータを廃棄したい場合は DTAOPT(\*DLT) を使用します。廃棄するのは、収集されたデータが使用できないと判断した場合です。たとえば、予期していたジョブの 1 つが期待どおりに開始しなかった場合などが考えられます。\*DLT オプションを選択すると、そのセッションに収集されたパフォーマンス・データは保管されません。
- 収集セッションを中断しますが、終了はしません。  
これには OPTION(\*SUSPEND) を使用します。特定のセッション ID に、OPTION(\*RESUME) を指定した STRPEX コマンドを出すことによって、データ収集を後で再開することができます。

注: 活動収集セッション名を忘れた場合、ENDPEX SSNID(\*SELECT) コマンドを使用してください。

## iDoctor for iSeries

iDoctor for iSeries<sup>(TM)</sup> は、Consulting Services、Job Watcher、Java<sup>(TM)</sup> Watcher、PEX Analyzer、および PTDV というコンポーネントで構成されるツールのスイートです。

### Consulting Services

専門のコンサルタントに iDoctor for iSeries スイートの精密なソフトウェア・ツールの 1 つ (PEX Analyzer または Job Watcher) を使ってシステム分析してもらいたい場合は、Consulting Services コンポーネントを使用します。

### Job Watcher

Job Watcher は、ジョブが実行していること、および実行していない理由を非常に詳細に表す、リアルタイムの表およびグラフィカル・データを表示します。Job Watcher は、一定間隔ごとに詳細なジョブ統計を提供する、さまざまな報告書をいくつか作成します。これらの統計により、CPU 使用率、DASD カウンター、待ち、障害、呼び出しスタック情報、対立情報などを判別することができます。

### Java<sup>(TM)</sup> Watcher

Java Watcher は、Java および WebSphere<sup>(R)</sup> の領域で、非常に複雑な問題をデバッグする際の援助として、きわめて貴重な情報を提供します。

## PEX Analyzer

PEX Analyzer は、システムのパフォーマンスを全体的に評価する機能があり、Performance Tools ライセンス・プログラムを使用して取得したデータを利用します。Analyzer は、大量のトレース・データを、パフォーマンス問題を発見するのに役立つグラフにしたり表示したりした報告書に要約し、全体の問題判別時間を削減します。Analyzer は、CPU 使用率、物理ディスク操作、論理ディスク入出力、データ域、およびデータ待ち行列の分析のために、使いやすいグラフィカル・インターフェースを提供します。また、Analyzer はアプリケーションのスローダウンの原因を見つけるのにも役立ちます。

## PTDV

Performance Trace Data Visualizer for iSeries (PTDV) は、Performance Explorer のデータベース・ファイルにある Performance Explorer 収集トレースの処理、分析、および表示のためのツールです。PTDV は iDoctor for iSeries の無料のコンポーネントです。

詳細については、iDoctor for iSeries  Web サイトを参照してください。

## Performance Trace Data Visualizer (PTDV)

Performance Trace Data Visualizer (PTDV) は、iSeries<sup>(TM)</sup> サーバー上で実行されるアプリケーションのパフォーマンス分析に使用できる Java<sup>(TM)</sup> アプリケーションです。PTDV は OS/400<sup>(R)</sup> 基本オペレーティング・システムの Performance Explorer から、分析者がプログラム・フローを見てトレース、ジョブ、スレッド、およびプロシージャごとにまとめられた詳細 (CPU 時間、現在のシステム時刻、サイクル数、命令数など) を入手できるようにします。Java アプリケーション・トレースを視覚化すると、作成されるオブジェクトの数やタイプ、および Java ロックの振る舞いについての情報などの付加的な情報も表示することができます。また、WebSphere<sup>(R)</sup> Application Server が生成する Performance Explorer イベントもサポートします。PTDV では、カラムのソート、データのエクスポートおよびさまざまなレベルでのデータ要約が可能です。

詳細については、Performance Trace Data Visualizer  Web サイトを参照してください。

## Performance Management API

Performance Management API を使用すると、収集サービス、Performance Collector、Performance Explorer および PM iSeries<sup>(TM)</sup> を使用してパフォーマンス・データを収集および管理できます。

Performance Management API には以下のものが含まれます。

- Collection Services API
- Performance Collector API
- Performance Explorer (PEX) API
- IBM<sup>(R)</sup> Performance Management for eServer<sup>(TM)</sup> iSeries (PM iSeries) API

## OS/400 パフォーマンス用の処理コマンド

OS/400<sup>(R)</sup> には、文字ベースのインターフェースからパフォーマンス・データのモニターをリアルタイムで実行するための多数のコマンドがあります。これらのコマンドを使用して、システム・パフォーマンスに関する特定の質問に回答したり、システムの調整に役立てたりすることができます。iSeries<sup>(TM)</sup> ナビゲーターからのリアルタイムのモニターについての詳細は、95 ページの『iSeries<sup>(TM)</sup> ナビゲーターのモニター』を参照してください。

コマンド	機能
活動ジョブの処理 (WRKACTJOB)	システム上で実行する、ジョブの属性および資源の使用効率を見直し、変更します。
ディスク状況の処理 (WRKDSKSTS)	システム・ディスク装置のパフォーマンス情報、および属性を表示します。
システム状況の処理 (WRKSYSSTS)	現行システムの活動の概要を示します。特に、システム上のジョブの数、および記憶域プールの使用状況に関する情報を表示します。
システム活動の処理 (WRKSYSACT)	システム上のジョブおよびタスクを処理します。このコマンドは、Performance Tools ライセンス・プログラム (PT1) の一部です。
オブジェクト・ロックの処理 (WRKOBJLCK)	適用を待機しているロックも含め、指定されたオブジェクトのロックを処理し、表示します。
共用記憶域プールの処理 (WRKSHRPOOL)	マシンおよび基本プールを含めた、共用記憶域プールの使用状況に関する情報を表示し、その属性を変更します。

## 拡張キャッシュ

拡張キャッシュを使用して、現在の iSeries<sup>™</sup> システムのパフォーマンスを向上させます。拡張キャッシュは、ディスクから読み取られる物理入出力要求の数を減らすことによって入出力サブシステムとシステム応答時間の両方を向上させる、先端の大規模読み取りキャッシュ技術です。拡張キャッシュは、データに関する統計情報を生成し、複合的な管理方針を使用してキャッシュに入れるデータを決定します。拡張キャッシュは、多くのタイプのワークロードに対してかなりの効果を持つことが証明されています。

さらに、以下のトピックも参照してください。

- 『拡張キャッシュの概念』  
拡張キャッシュについて紹介しています。このツールの使用を開始する前に考慮すべき、計画、制約事項、および重要な考慮事項に関する情報を扱います。
- 145 ページの『拡張キャッシュの 制約事項および考慮事項』  
拡張キャッシュに必要な構成要素を示すと共に、何が期待できるかについてさらに詳しく説明します。

### 拡張キャッシュの概念

『拡張キャッシュ』を使用してシステム・パフォーマンスを向上させます。拡張キャッシュは、ディスクから読み取られる物理入出力要求の数を減らすことによって入出力サブシステムとシステム応答時間の両方を向上させる、先端の読み取りキャッシュ技術です。拡張キャッシュは、データベースの読み取りアクションに限らず、すべての読み取りアクションのパフォーマンスを向上させます。これには、統合 xSeries<sup>®</sup> サーバーなどの他のシステム構成要素による読み取りアクションも含まれます。また、拡張キャッシュは、装置パリティ保護やミラー保護のある記憶域サブシステムでも効果的に機能します。拡張キャッシュは、多くのタイプのワークロードに対してかなりの効果を持つことが証明されています。

### 拡張キャッシュが働く仕組み

拡張キャッシュは、iSeries<sup>™</sup> 入出力サブシステムに統合されています。拡張キャッシュはディスク・サブシステム制御装置のレベルで操作し、iSeries システム・プロセッサには影響を与えません。記憶域入出力アダプターは、Read Cache Device (solid state disk など) を使用してキャッシュ・メモリーを提供することにより、この拡張キャッシュを管理しています。

拡張キャッシュは、データに関する統計情報を生成し、混ぜ合わされた管理戦略を使用してキャッシュに入れるデータを決定します。キャッシュの管理は、入出力アダプター内で自動的に行われ、予測のアルゴリズム

ムを使用してデータのキャッシュを行うように設計されています。このアルゴリズムは、前もって定められた範囲のデータに対し、ホストがどれほど最近に、そしてどれほど頻繁にアクセスしたかを考慮します。

拡張キャッシュの設計は、iSeries サーバー・システムの特別なデータ管理戦略に基づいています。ディスクが装置パリティ保護されているか、ミラー保護されているか、あるいは無保護であるかに関係なく、そのディスクに保管されているデータには、バンド状に発生する傾向があります。つまり、ディスク記憶域には物理的な連続区域があって、データが活動状態で読み取られている区域、頻繁に書き込みが行われる物理的な連続区域、データが活動状態で読み取られていながらかつ書き込まれている物理的な連続区域、あるいはあまり頻繁にアクセスされない記憶域の物理的な連続区域が存在することを意味します。

この「バンド状の」データは、拡張キャッシュの設計によって説明付けられます。拡張キャッシュの目的は、読み取り/書き込み、および読み取り専用の特徴があるバンドをキャッシュに入れることです。書き込み専用の特徴があるバンドは、記憶域サブシステムの書き込みキャッシュに入れられながらも、大部分は、拡張キャッシュの影響を受けることなく残されます。加えて、拡張キャッシュは、大きなブロックでデータを順次書き込む、または読み取るパフォーマンスに影響を与えないようにも設計されています。この場合は、ディスクの事前取り出し機能が即時応答を保証しています。これは、システム内の他のキャッシュでも同じです。

必要なコンポーネントについては、『『拡張キャッシュの 制約事項および考慮事項』』を参照してください。

**拡張キャッシュの 制約事項および考慮事項:** 144 ページの『拡張キャッシュの概念』の使用を開始する前に、ご使用のコンピューティング環境に関係し得る制約事項や考慮事項を検討するため、いくつかの事前の計画を立てる必要があります。

## 制約事項

拡張キャッシュを使用するためには、システムが以下を備えている必要があります。

- 拡張キャッシュをサポートしている 1 つ以上の記憶域入出力アダプター (V5R2以降が実行されているシステムの場合は CCIN 2780)
- Performance Tools for iSeries<sup>(TM)</sup> ライセンス・プログラム。報告された情報を表示します。

拡張キャッシュは、サポートされる入出力アダプター上で自動的に使用可能になります。このオン/オフを制御するスイッチはありません。入出力アダプターをサブシステムに挿入すると、拡張キャッシュが活動状態になります。拡張キャッシュがデータ・フローをモニターし、読み取りキャッシュ・メモリーに移植するまでには、1 時間ほどかかります。1 時間ほど拡張キャッシュを実行すると、システムのパフォーマンスの向上 (現在のワークロードによる) や入出力スループットの増加が確認できるようになります。

入出力アダプター内の他のディスクに対する装置パリティ保護やミラー保護に関して、拡張キャッシュに制約事項はありません。最後に、拡張キャッシュは、特に iSeries エキスパート・キャッシュを補うために設計されており、これと一緒に使用することも、単独で使用することも可能です。

## 考慮事項

拡張キャッシュを使用すると、ほとんどの環境において、入出力の応答時間をかなり短縮し、システム入出力のスループットを高めることができます。一般のキャッシュと同様、拡張キャッシュの効果は、システムの構成やワークロードに影響を受けます。拡張キャッシュは、記憶域サブシステム・レベルで行われます。そして、特定のサブシステム内にある一連のディスクのデータをキャッシュに入れます。したがって、システム内の活動状態にある記憶域サブシステムやパフォーマンス重視の記憶域サブシステムに対する拡張キャッシュの追加は、ほとんどの場合、論理的なものです。拡張キャッシュは、事前取り出しタイプのキャッシュとは見なされないため、ディスク内の先読み機能を妨げることはありません。

活発に入出力要求を受け取るディスク記憶域の区域が大きいと、新しいデータをキャッシュに入れようとするときに決定する拡張キャッシュの選択肢が広がります。このような適応性を備えているため、拡張キャッシュは、さまざまなタイプやサイズのワークロードに対して効果的です。

**拡張キャッシュの開始:** 144 ページの『拡張キャッシュの概念』を開始し、システムのパフォーマンスを向上させるためには、Read Cache Device を購入してください。Read Cache Device がサブシステムのディスク・スロットに挿入されると、拡張キャッシュが活動状態になります。このオン/オフをユーザーが制御するためのスイッチはありません。拡張キャッシュがデータ・フローをモニターし、Read Cache Device に移植するまでには、1 時間ほどかかります。1 時間ほど拡張キャッシュを実行すると、システムのパフォーマンスの向上 (現在のワークロードによる) や入出力スループットの増加が確認できるようになります。

ご使用の iSeries<sup>(TM)</sup> システムで拡張キャッシュを使用できるかどうかは、145 ページの『拡張キャッシュの 制約事項および考慮事項』を参照してください。

## 拡張キャッシュの入手

拡張キャッシュを使用してシステムのパフォーマンスを向上させることを決定したなら、Read Cache Device (RCD) を購入する必要があります。拡張キャッシュは、RCD を通して自動的に使用可能になります。

拡張キャッシュを実際に使用するためには、以下のシステムが必要です。

- 拡張キャッシュをサポートしている 1 つ以上の記憶域入出力アダプター (CCIN 2748 (V4R4 以降が稼働するシステム用)、CCIN 2778 (V5R1 以降が稼働するシステム用)、CCIN 2757 (最新リリースの V5R2 が稼働するシステム用))
- 拡張キャッシュを活動化する、各記憶域入出力アダプター用の Read Cache Device (RCD) (CCIN 6731 (V4R4 以降が稼働するシステム用))

拡張キャッシュはこの RCD を通して自動的に使用可能になるため、そのオン/オフを制御するスイッチはありません。RCD は、システム割り込みがなくても、並行メンテナンスによって追加される場合があります。RCD は、内部ディスク・スロットに常駐し、他のすべてのディスク・タイプおよびキャパシティーで作業します。拡張キャッシュ内の全データは、必ずディスク上にも存在します。ほとんどあり得ないイベントですが、RCD が失敗しても、データが失われることはありません。

Read Cache Device は、iSeries<sup>(TM)</sup> ハードウェアが置かれている店舗などで購入できます。あるいは、地域の IBM<sup>(R)</sup> 担当員までご連絡ください。

## Workload Estimator for iSeries

Workload Estimator  は、特定のワークロード・タイプのワークロードを見積もることにより、それに基づいて行われるシステムの必要の判別を支援します。PM iSeries<sup>(TM)</sup> は、プロセッサの保証を受けられるユーザーや IBM<sup>(R)</sup> のメンテナンス契約に加入しているユーザーが、追加の課金なしで利用できる統合 OS/400<sup>(R)</sup> 機能です。これを利用すると、システムの成長やパフォーマンスを計画および管理するのに便利な、キャパシティーとパフォーマンスの分析グラフが送られてきます。

Workload Estimator と PM iSeries は、相互に連携するよう拡張されています。Web ベース・アプリケーションを通して、既存のシステムの使用状況、パフォーマンスおよび PM iSeries によって報告される成長に適合した、必要とされる iSeries システムへのアップグレードのサイジングが行えます。また追加のオプションとして、ドミノ (Domino<sup>(TM)</sup>)、Java<sup>(TM)</sup>、および WebSphere<sup>(R)</sup> のような特定のアプリケーションを追加するためのキャパシティーや、複数の AS/400<sup>(R)</sup> または iSeries の従来型の OS/400 ワークロードを 1

つのシステムに統合するためのキャパシティーも、見積もりに含めることができます。このようなキャパシティーを含めておこなら、使用している独自のシステムから得られた既存の使用状況データに基づいて、将来のシステム要件を計画することが可能です。

## iSeries<sup>(TM)</sup> ナビゲーター (ワイヤレス対応)

iSeries ナビゲーター (ワイヤレス対応) では、インターネット対応の電話、ワイヤレス・モデムを備えた個人用デジタル・アシスタント (PDA)、または従来の Web ブラウザーを使用して、システムのパフォーマンスや状況をリモートにモニターできます。ワイヤレス装置では、以下のことを行えます。

- 複数システム間でのコマンドの実行
- システム、ジョブ、およびメッセージ・モニターの開始と表示
- モニターからのジョブやメッセージの処理 (保留、開放、終了、応答、詳細表示)
- 統合 xSeries<sup>(R)</sup> サーバーの管理

iSeries ナビゲーター (ワイヤレス対応) がリモート・モニターを始めるのにどのように役立つかについては、トピック『iSeries ナビゲーター (ワイヤレス対応)』を参照してください。

リモート・モニターに関する完全で最新の情報は、iSeries Navigator for Wireless  ホーム・ページを参照してください。

## PATROL for iSeries (AS/400) - Predict

PATROL for iSeries<sup>(TM)</sup> (AS/400<sup>(R)</sup>) - Predict 製品は、将来の iSeries 要件を推定して、トランザクション・スループットおよびアプリケーション・ワークロードの増加に順応させるのに役立つキャパシティー・プランニング・ツールです。見積もりプロセスは、リソース使用率、パフォーマンス、および iSeries サーバーで測定される 5250 オンライン・トランザクション処理 (対話式) 応答時間情報を提供する収集サービスのデータに基づきます。予測分析は、PC ワークステーションのグラフィカル・インターフェースによって実行されます。

詳細は、BMC 製品の Web サイト  を参照してください。

---

## シナリオ: パフォーマンス

パフォーマンス管理について学ぶ最良の方法の 1 つは、サンプル・ビジネス環境でいくつかのアプリケーションと機能が使用され得るかを示す例を見ることです。次のシナリオと構成例を考慮して、パフォーマンスの管理に関する理解を深めてください。

### 24 ページの『シナリオ: アップグレードまたはマイグレーション後にシステム・パフォーマンスを改善する』

このシナリオは、システムをアップグレードまたは移行したところ、以前よりも実行速度が遅くなったように思える、というものです。このシナリオは、パフォーマンスの問題を識別し、修正するのに役立つでしょう。

### 98 ページの『シナリオ: システム・モニター』

この例のシステム・モニターは、CPU 使用率が高すぎるために使用可能なリソースが増えるまで優先順位の低いジョブを一時的に保留する場合に、アラートを出します。

### 102 ページの『シナリオ: メッセージ・モニター』

この例のメッセージ・モニターは、iSeries<sup>(TM)</sup> サーバー上で生じた、メッセージ・キューに関する照会メッセージを表示します。このモニターは、メッセージを検出すると即時にそのメッセージをオープンして表示します。

### 99 ページの『シナリオ: CPU 使用率のジョブ・モニター』

この例のジョブ・モニターは、指定されたジョブの CPU 使用率を追跡し、CPU 使用率が高すぎる場合はそのジョブの所有者にアラートを出します。



### 101 ページの『シナリオ: Advanced Job Scheduler 通知を使ったジョブ・モニター』

ジョブのしきい値限度を超えたときにオペレーターに E メールを送信する、ジョブ・モニターの例を参照してください。 <<

---

## 関連情報

以下に、パフォーマンスのトピックと関連がある PDF 形式の iSeries<sup>(TM)</sup> 資料 (「ホワイト・ブック」と呼ばれる) および IBM<sup>(R)</sup> Redbooks<sup>(TM)</sup> をリストします。以下の PDF のいずれかを表示または印刷することもできます。

- マニュアル

- Performance Tools for iSeries**



この資料では、システム、ジョブ、またはプログラム・パフォーマンスに関するデータを収集するために必要な情報をプログラマーに提供します。資料には、存在する可能性のある非効率性を識別して訂正するための、パフォーマンス・データの印刷と分析に関するヒントが載せられているほか、マネージャー機能やエージェント機能に関する情報が示されています。

- Web サイト

- **iSeries Performance Capabilities Reference**



この解説書には、パフォーマンス・ベンチマークに役立つサーバー・パフォーマンス、キャパシティ・プランニング、およびサーバー・パフォーマンスの計画に関する高水準の技術情報を提供しています。

- >> **Three-In-One Benchmark**



IBM 開発のベンチマークは、Three-in-One Benchmark と呼ばれ、IT 関連会社が直面する実際の要求を反映しています。この報告書は、iSeries サーバーが現在の中小および中堅規模ビジネスにとって、パフォーマンスを心配せずに必要なアプリケーションを実行するのに役立つ優れた解決策であることを、明確に示します。 <<

- >> **Performance Management for IBM eServer iSeries**



Performance Management により、お客様がコンピューティング環境のパフォーマンスを理解して、これを管理することが可能になります。最新の Performance Management 機能およびツールに関しては、この Web サイトをお読みください。◀

• レッドブック:

- **IBM eserver iSeries Universal Connection for Electronic Support and Services**



この資料は、ユニバーサル・コネクションについて紹介するものです。またこの中では、マシンのソフトウェアやハードウェアのインベントリーを IBM に報告するさまざまなサポート・ツールを使用して、システム・データに基づいた個別のエレクトロニック支援を受けられるようにする方法も説明されています。

- **Lotus<sup>(R)</sup> Domino<sup>(R)</sup> for AS/400<sup>(R)</sup>: Performance, Tuning, and Capacity Planning**



この資料では、パフォーマンス管理のための方法論について説明します。この資料には、パフォーマンス目標の設定、パフォーマンス・データの収集と検討、資源の調整、およびキャパシティー・プランニングが含まれています。パフォーマンスに関する指針およびアプリケーション設計のヒントも提供されます。

- **AS/400 Performance Management**



この資料では、パフォーマンス管理のための方法論について説明します。この資料には、パフォーマンス目標の設定、パフォーマンス・データの収集と検討、資源の調整、およびキャパシティー・プランニングが含まれています。パフォーマンスに関する指針およびアプリケーション設計のヒントも提供されます。

- **AS/400 HTTP Server Performance and Capacity Planning**



インターネットおよび Web ブラウザー・ベースのアプリケーションは、どのように組織が情報を配布し、ビジネス・プロセスを実行し、顧客にサービスを提供し、新規のマーケットに到達するかについて多大の影響を与えています。この資料は、Web ベースのアプリケーションおよび情報システムの設計、開発、および拡張を担当する iSeries プログラマー、ネットワークおよびシステム管理の専門家、およびその他の情報技術者を対象としています。

- **Java<sup>(TM)</sup> and WebSphere<sup>(R)</sup> Performance on IBM eserver iSeries Servers**



この資料では、Java を使用した作業を行うためのヒント、テクニック、および方法論を扱っているほか、iSeries サーバーに関係した特定の視点から WebSphere Application Server のパフォーマンス関連の問題に触れます。

- **▶ Managing OS/400<sup>(R)</sup> with Operations Navigator V5R1, Volume 1: Overview and More**



この分冊では、オペレーション・ナビゲーター V5R1 を概説します。ジョブ、サブシステム、ジョブ・キュー、およびメモリー・プールの管理、システム・パフォーマンス・メトリックのモニター、ジョブおよびメッセージ、および収集サービスなどを説明します。 <<

– >> **Managing OS/400 with Operations Navigator V5R1, Volume 5: Performance Management**



この分冊では、ボリューム 1 で説明されるモニター、グラフ・ヒストリー、および収集サービスの機能に基づいて構成されています。このマニュアルは、アプリケーション環境でこれらの機能を使用する方法を説明します。 <<

– **AS/400 Performance Explorer Tips and Techniques**



この資料は、V3R6 で使用できた Performance Explorer 機能に関する説明と詳細な例を提供します。この中には、特定のアプリケーションの例やレポートが含まれています。

– **DB2<sup>(R)</sup> UDB/WebSphere Performance Tuning Guide**



このマニュアルは、WebSphere Application Server アーキテクチャー、およびその主な構成要素を概説し、鍵となるアプリケーション・チューニング・パラメーター、およびシステム・チューニング・パラメーターを紹介します。

iSeries のパフォーマンスに関する完全な情報を確認していただくため、必ず『1 ページの『パフォーマンス』』のトピックを参照してください。

---

## 付録. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032  
東京都港区六本木3-2-31  
IBM World Trade Asia Corporation  
Licensing

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、IBM 機械コードのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

#### 著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

強行法規で除外を禁止されている場合を除き、IBM、そのプログラム開発者、および供給者は「プログラム」および「プログラム」に対する技術的サポートがある場合にはその技術的サポートについて、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

IBM、そのプログラム開発者、または供給者は、いかなる場合においてもその予見の有無を問わず、以下に対する責任を負いません。

1. データの喪失、または損傷。
2. 特別損害、付随的損害、間接損害、または経済上の結果的損害
3. 逸失した利益、ビジネス上の収益、あるいは節約すべかりし費用

国または地域によっては、法律の強行規定により、上記の責任の制限が適用されない場合があります。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

(C) (お客様の会社名) (年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。(C) Copyright IBM Corp. \_年を入れる\_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

---

## 商標

以下は、IBM Corporation の商標です。

AIX

AIX 5L

Domino

e(ロゴ)server

eServer

Operating System/400

OS/400

IBM

iSeries

pSeries

xSeries

Java<sup>™</sup> およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Linux<sup>™</sup> は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

---

## 資料に関するご使用条件

お客様がダウンロードされる情報につきましては、以下の条件にお客様が同意されることを条件にその使用が認められます。

**個人使用:** この情報は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、この情報またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

**商業的使用:** この情報は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこの情報の二次的著作物を作成したり、お客様の企業外で情報またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

情報の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。IBM は、この情報の内容についていかなる保証もしません。この情報は、特定物として現存するままの状態を提供され、商品性の保証、不侵害の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

これらの資料の著作権はすべて、IBM Corporation に帰属しています。

お客様が、このサイトから情報をダウンロードまたは印刷することにより、これらの条件に同意されたものとさせていただきます。

---

## コードに関する特記事項

IBM<sup>(R)</sup> は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作使用权を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。

強行法規で除外を禁止されている場合を除き、IBM、そのプログラム開発者、および供給者は「プログラム」および「プログラム」に対する技術的サポートがある場合にはその技術的サポートについて、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

IBM、そのプログラム開発者、または供給者は、いかなる場合においてもその予見の有無を問わず、以下に対する責任を負いません。

1. データの喪失、または損傷。
2. 特別損害、付随的損害、間接損害、または経済上の結果的損害
3. 逸失した利益、ビジネス上の収益、あるいは節約すべかりし費用

国または地域によっては、法律の強行規定により、上記の責任の制限が適用されない場合があります。





Printed in Japan