



IBM i  
システム管理  
パフォーマンス

7.1







**IBM i**  
**システム管理**  
**パフォーマンス**

*7.1*

**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、213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 i 7.1 (製品番号 5770-SS1) に適用されます。また、改訂版で断りが無い限り、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。

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： IBM i  
Systems management  
Performance  
7.1

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2010.4

© Copyright IBM Corporation 2002, 2010.

---

## 目次

パフォーマンス	1	データの表示と分析	78
IBM i 7.1 の新機能	1	シナリオ: パフォーマンス	201
パフォーマンスの PDF ファイル	3	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	201
システム・パフォーマンスの管理	4	シナリオ: システム・モニター	202
パフォーマンス管理戦略の選択	4	シナリオ: メッセージ・モニター	204
システムの拡張時および拡張方法の決定	6	シナリオ: CPU 使用率のジョブ・モニター	205
システム変更前後のパフォーマンス・メトリックの比較	7	シナリオ: Advanced Job Scheduler 通知を使ったジョブ・モニター	206
パフォーマンスの追跡	8	パフォーマンスの関連情報	208
パフォーマンスの問題の調査	9		
パフォーマンス・データの表示	21	<b>付録. 特記事項</b>	<b>213</b>
パフォーマンスの調整	22	プログラミング・インターフェース情報	214
e-business のパフォーマンス	27	商標	215
パフォーマンス管理用のアプリケーション	31	使用条件	215
パフォーマンス・データ・コレクター	34		



---

## パフォーマンス

システムのパフォーマンスのモニターおよび管理は、業務の変わりつつある要求と確実に歩調を合わせるために重要です。

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となります。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。

システムのパフォーマンスの管理は、システムのハードウェアとソフトウェアの完全な理解を必要とする複雑な作業となる場合があります。IBM® i は、パフォーマンス管理の領域における業界のリーダーです。他に類を見ないパフォーマンス・メトリックや常時オンの収集サービス、パフォーマンス・データのグラフィカル・ビューなど、他のシステムにはない多くの特色を備えています。システム・パフォーマンスに影響を与えるさまざまなプロセスをすべて理解することは難しい場合があり、またパフォーマンス上の問題を解決するためには、ツールの大規模スイートを効果的に使用することが求められます。IBM i が提供する機能は、このような作業をユーザーが容易に行えるようにすることを目的としています。

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。

注: 以下のコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

### 関連概念

#### 実行管理機能

実行管理機能は、システム操作とシステム上の日常的ワークロードを制御するために必要なコマンドと内部機能をサポートします。

---

## IBM i 7.1 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。

### 収集サービス

- データ収集カテゴリーが追加されました。
  - 取り外し可能ストレージ - このカテゴリーは、IBM i 区画に接続された磁気テープ装置のデータ収集をサポートします。パフォーマンス・データの作成 (CRTPFDRDTA) コマンドは、このデータを新しいデータベース・ファイル QAPMTAPE にエクスポートします。
  - 外部ストレージ - このカテゴリーは、IBM i 区画に接続された特定の外部ストレージ・サブシステムに関連した非標準データの収集をサポートします。こうしたデータのフォーマットはサブシステムに依存するので変わる可能性があるため、CRTPFDRDTA はデータを新しいデータベース・ファイル QAPMXSTGD に文字ストリングとしてエクスポートします。このデータは iDoctor 内で表示できません。
  - システム内部 - このカテゴリーは、内部システム・データの収集をサポートします。提供された場合、このデータは IBM 内部で使用されます。

- 収集と報告への短持続期間のスレッドやタスクの影響が少なくなりました。これらのタスクやスレッドのデータは、個別ではなく累積で報告されます。追加情報については、65 ページの『短持続期間のスレッドおよびタスク』を参照してください。
- ディスク入出力操作をより詳しく分析できるように、ディスク応答時間グループ (入出力応答時間の平均ではなく分散で示す) が拡張されました。
  - グループ定義の数が増えました。
  - 時間の単位がミリ秒に代わってマイクロ秒になります。
  - 読み取り時間と書き込み時間が個別のグループに分かれました。
  - 応答時間グループの新しいデータが、新しいファイル QAPMDISKRB で提供されます。

注: 7.1 より前に定義された応答時間グループは、互換性のためにこれまでと同様 QAPMDISK ファイルで提供されます。

- 保管または復元操作を実行したジョブでは、追加データが収集されます。このデータは、新しいファイル QAPMJOBRSR で提供されます。
- 既存のいくつかのファイルが変更され、フィールドが追加されました。詳しくは、『収集サービス・データ・ファイル』のトピックを参照してください。

## IBM Systems Director Navigator for i Performance インターフェース

Performance Data Investigator で、以下のようにいくつかの機能拡張が行われました。これらの機能は、「データの調査」タスクを選択することによってアクセスできます。

- **全般ヘルス標識:** 区画の全般的な正常性を示す新しいコンテンツ・パッケージが組み込まれており、これにユーザー定義のヘルスしきい値を構成できます。
- **区画間 CPU メトリックの表示:** 収集サービスは、オペレーティング・システムに関係なく、同一の単一物理サーバー上のすべての論理区画を対象に、高水準の区画間プロセッサ・パフォーマンス・メトリックを収集できます。この機能は、POWER6<sup>®</sup> 以上のシステム (ファームウェア・レベルが少なくとも xx340\_061) で使用できます。このデータを使用できる場合は、「物理システム」の下にあるさまざまなパースペクティブを通して表示できます。
- **エクスポート:** イメージ (図表のみ) ファイル、コンマ区切りファイル、またはタブ区切りファイルにデータをエクスポートできるようになりました。
- **次期アップグレードの見積もり:** 「データの調査」を使用して、現在のパフォーマンス特性に基づいた今後のシステムの見積もりで使用される Workload Estimator に、現行セッションからデータを送信できるようになりました。
- **対話式的パースペクティブ作成:** 新しい図表や表を「データの調査」内から作成できます。ビューの追加、SQL ステートメントの変更、データ・シリーズ情報の変更はすべて、ページ表示のままで行えます。
- **メトリック・ファインダー:** 「データの調査」では多数のメトリックを使用できるので、調べたいメトリックがどのパースペクティブに含まれているかを知るのには、ささいな作業では済まない場合もあります。メトリック名を検索してパースペクティブを選択できるようになりました。この機能は、「データの調査」が起動していれば「検索」ボタンから使用できます。
- **Performance Explorer コンテンツ・パッケージ:** Performance Explorer データの分析を開始するための簡単な Performance Explorer コンテンツ・パッケージが組み込まれています。
- **新しいメトリック:** 「データの調査」にさまざまなパースペクティブが新しく追加されました。また、既存のパースペクティブにさまざまなメトリックが新しく追加されました。新しいメトリックとして、SAN、仮想入出力、エネルギー制御、仮想メモリー、通信データをはじめ、多数あります。



- **その他の機能拡張:** パフォーマンス・タスクの機能を全体として向上させるために、各種変更が行われました。「データの調査」の範囲で例を挙げると、永続的オプション、各パースペクティブの上部に新しく表示されるようになった収集コンテキスト、より速いナビゲーションとより完全なヒストリー・データを可能にする新しいメニュー・バーなどがあります。

注: これらの新機能は、PTF の SI35659、SI35663、および SI36093 を適用した IBM i 6.1 でも使用できます。

## Performance Management Agent

PM Agent が SNA を使用してパフォーマンス・データを IBM に送信する機能がなくなりました。パフォーマンス管理データの送信には、Electronic Service Agent™ のみが使用されます。GO PM400 のオプション 2 (「自動スケジュール済みジョブの処理」) に表示されていた 3 つの項目がなくなりました。これらのジョブは、パフォーマンス・データの送信に SNA を使用していました。

GO PMAGT のサポートが追加されており、これは GO PM400 と同じ機能を持っています。



PMAGT/PM400 メニューに、次の 2 つのオプションが新しく追加されました。

- Electronic Service Agent が IBM に接続されているかどうかを検査する機能。
- パフォーマンス・データの送信を開始する機能。

PMLINMON と CFGPMLIN の 2 つのコマンドがシステムから削除されました。

## 新規情報または変更情報の見分け方

技術的な改訂箇所には、以下のマークを使います。

-  イメージにより、新規または変更された情報の開始点を示します。
-  イメージにより、新規または変更された情報の終了点を示します。

このリリースでの新機能または変更点に関する他の情報を調べるには、プログラム資料説明書を参照してください。

---

## パフォーマンスの PDF ファイル

この情報の PDF ファイルを表示または印刷できます。


- | • 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、パフォーマンスを選択します。この PDF にはパフォーマンスの参照情報は含まれていません。
- | • パフォーマンスの参照情報の PDF 版を表示またはダウンロードするには、パフォーマンスの参照情報を
- | 選択します。

## 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. ご使用のブラウザで PDF リンクを右クリックする。
2. PDF をローカルに保存するオプションをクリックする。
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® 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 Reader がご使用のシステムにインストールされている必要があります。このアプリケーションは、Adobe Web サイト (<http://get.adobe.com/reader/>)  から無償でダウンロードできます。

### 関連資料

208 ページの『パフォーマンスの関連情報』

パフォーマンスのトピックと関連のある製品マニュアル および IBM Redbooks® (PDF 形式)、Web サイト、および information center トピックをここにリストします。PDF は、どれでも表示または印刷することができます。

---

## システム・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。

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。

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。

### 関連情報

 [IBM i でのパフォーマンス管理 \(英語\)](#)

ホワイト・ペーパー、記事、ツール関連最新資料をはじめ、IBM i オペレーティング・システムに関するさまざまな参照情報については、このパフォーマンス管理 Web サイト (英語) をご覧ください。

## パフォーマンス管理戦略の選択

よいパフォーマンス管理戦略を開発することは、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戦略は、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。

異なる業務ニーズには、異なるパフォーマンス管理戦略が必要です。基本的なパフォーマンス管理戦略を決定し、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、管理戦略もそれに応じて異なります。

## 中小規模ビジネス

中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。 Performance Management for IBM i (PM for IBM i) を使用して、パフォーマンス・データを IBM に直接送信できます。パフォーマンス・データはそこでコンパイルされてユーザーのための報告書が生成されます。これは時間の節約になるだけでなく、ユーザーの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。

中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。

- IBM Systems Director Navigator for i Performance インターフェース: パフォーマンス・データの表示および管理を行います。
- 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。
- Performance Management for IBM i: 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。
- 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。
- IBM i ナビゲーター・モニター: 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や状態に対する応答を自動化します。

## 中規模ビジネス

ほとんどの場合、中規模ビジネスは中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 for IBM i を使用することによって得られるものもあります。

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。

- IBM Systems Director Navigator for i Performance インターフェース: パフォーマンス・データの表示および管理を行います。
- 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。
- Performance Management for IBM i: 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。
- 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。
- IBM i ナビゲーター・モニター: 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や状態に対する応答を自動化します。

## 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。

- IBM Systems Director Navigator for i Performance インターフェース: パフォーマンス・データの表示および管理を行います。
- 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。
- Performance Management for IBM i: 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。
- 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。
- IBM i Job Watcher: 特定のジョブまたはスレッド・リソースに関する詳細な情報を収集します。

- IBM i Disk Watcher: ディスク・パフォーマンス・データに関する詳細な情報を収集します。
- 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。

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

66 ページの『IBM i Job Watcher』

IBM i Job Watcher は、システム上の一部またはすべてのジョブ、スレッド、およびタスクについてジョブ・データの収集を行います。コール・スタック、SQL ステートメント、待機されているオブジェクト、Java™ JVM 統計、待機統計などを提供し、これらはジョブ関連のパフォーマンス上の問題の診断に使用されます。

67 ページの『IBM i Disk Watcher』

IBM i Disk Watcher は、ディスク関連のパフォーマンス上の問題を診断するためのディスク・パフォーマンス・データの収集を行います。

116 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。

119 ページの『IBM Performance Management for Power Systems - IBM i のサポート』

IBM i オファリングをサポートする IBM Performance Management for Power Systems™ (PM for Power Systems) 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・リソース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。

68 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・リソース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。

## 関連資料

134 ページの『Performance Tools』

Performance Tools ライセンス・プログラムには、オペレーティング・システムで使用可能な基本パフォーマンス・ツールの機能を補足または拡張する、数多くの補足機能があります。

103 ページの『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らないとされることでしょう。

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。

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 では、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、以下のようないくつかのツールを提供しています。

- IBM Performance Management for System i5<sup>®</sup>
- IBM Systems Workload Estimator

#### 関連概念

119 ページの『IBM Performance Management for Power Systems - IBM i のサポート』

IBM i オファリングをサポートする IBM Performance Management for Power Systems (PM for Power Systems) 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・リソース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。

133 ページの『IBM Systems Workload Estimator』

IBM Systems Workload Estimator は、System i<sup>®</sup>、System p<sup>®</sup>、および System x<sup>®</sup> 用の Web ベースのサイズ変更ツールです。このツールを使用して、新規システムのサイズ変更、アップグレードから既存のシステムへのサイズ変更、または複数のシステムの統合のサイズ変更を実行できます。

#### 関連資料

4 ページの『パフォーマンス管理戦略の選択』

よいパフォーマンス管理戦略を開発することは、システムのパフォーマンスを管理するのに役立ちます。

## システム変更前後のパフォーマンス・メトリックの比較

システム変更前およびシステム変更後のパフォーマンス・メトリックの比較は、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。

新しい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、一連のシステム・パフォーマンス・メトリックを設定する必要があります。正確なシステム・パフォーマンス・メトリックを維持することは、トラブルシューティングに必要な不可欠な情報です。少なくとも、システム・パフォーマンス・メトリックは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。

## 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

## 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。

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。

システムには、パフォーマンスの傾向を追跡し、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の使用率レベルと傾向。

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。

- IBM Systems Director Navigator Performance インターフェース
- 収集サービス
- IBM Performance Management for System i5

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

116 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。

119 ページの『IBM Performance Management for Power Systems - IBM i のサポート』

IBM i オフラインをサポートする IBM Performance Management for Power Systems (PM for Power Systems) 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・リソース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。

## 関連資料

27 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・リソース、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サーバーの通信リソース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。

## 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。

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。

IBM i5/OS® Job Watcher および IBM i5/OS Disk Watcher もサンプル・データを収集します。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 およびパフォーマンス追跡開始 (STRPFTRC) 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、System i ナビゲーター・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、希望するジョブに対して IBM i5/OS Job Watcher 収集を開始し、サーバー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。

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・パフォーマンスと関連するデータを収集および検討するために、i5/OS 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。

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

多くのさまざまなパフォーマンス上の問題がシステムの共通域にしばしば影響します。共通域の問題（たとえば、バックアップおよびリカバリ）を調査し、解決する方法を学習します。

システム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。

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>• IBM Systems Director Navigator for i の Performance Data Investigator</li> <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</li> <li>• System i ナビゲーター の実行管理機能。</li> <li>• System i ナビゲーター モニター内の CPU 使用状況メトリック</li> </ul>



領域	説明	使用可能なツール
主記憶装置	ページ不在や待ち状態から不適合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>• IBM Systems Director Navigator for i の Performance Data Investigator</li> <li>• System i ナビゲーター 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>• System i 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>• IBM Systems Director Navigator for i の Performance Data Investigator</li> <li>• ディスク状況の処理 (WRKDSKSTS) コマンド</li> <li>• System i ナビゲーター 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>• IBM Systems Director Navigator for i の Performance Data Investigator</li> <li>• Performance Tools の構成要素報告書</li> <li>• System i 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>
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>• IBM Systems Director Navigator for i の Performance Data Investigator</li> <li>• Performance Tools の構成要素報告書</li> <li>• System i 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>

領域	説明	使用可能なツール
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>• IBM Systems Director Navigator for i の Performance Data Investigator</li> <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>• System i ナビゲーターの実行管理機能の下にある予期されるジョブの詳細表示。</li> <li>• システム活動の処理 (WRKSYSACT) コマンド</li> <li>• パフォーマンス・データの表示 (DSPPFRDTA) コマンド。</li> </ul>
バックアップおよび回復	バックアップおよびリカバリーに影響を与える領域を調査して、保管および復元操作をします。	<ul style="list-style-type: none"> <li>• System i Performance Capabilities Reference (『Save/Restore Performance』の章)。</li> </ul>

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

## 実行管理機能

iSeries ナビゲーターの実行管理機能について詳しくは、『実行管理機能』のトピックを参照してください。

## 関連資料

106 ページの『モニターのメトリック』


効果的なシステム・パフォーマンスのモニターを行うためには、どのような視点からシステム・パフォーマンスをモニターするかを決めなければなりません。マネージメント・セントラルには、メトリックと呼ばれるさまざまなパフォーマンスの測定法がありますが、このメトリックを使用するとシステム・パフォーマンスのさまざまな局面を正確に示すことができます。

バックアップおよびリカバリーに関するよくある質問

一般的なバックアップおよびリカバリーに関する質問については、『バックアップおよびリカバリーに関するよくある質問』のトピックを参照してください。

## 関連情報

パフォーマンス用の CL コマンド

 「IBM i でのパフォーマンス管理」 Web サイト (英語) - 「Performance Capabilities Reference」 PDF  
「IBM i でのパフォーマンス管理」 Web サイト (英語) から、適切な「Performance Capabilities Reference」 PDF を選択してください。バックアップとリカバリー関連のパフォーマンスに関する情報については、『Save/Restore Performance』の章を参照してください。

## 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システム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システム、または一連のシステム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。

データにアクセスし、収集できる 4 つのコレクターがあります。

- 収集サービス
- Job Watcher
- Disk Watcher
- Performance Explorer

### 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

66 ページの『IBM i Job Watcher』

IBM i Job Watcher は、システム上の一部またはすべてのジョブ、スレッド、およびタスクについてジョブ・データの収集を行います。コール・スタック、SQL ステートメント、待機されているオブジェクト、Java JVM 統計、待機統計などを提供し、これらはジョブ関連のパフォーマンス上の問題の診断に使用されます。

67 ページの『IBM i Disk Watcher』

IBM i Disk Watcher は、ディスク関連のパフォーマンス上の問題を診断するためのディスク・パフォーマンス・データの収集を行います。

68 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・リソース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。

## システム・リソース使用状況についての情報の収集

いくつかのツールは、中央演算処理装置 (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。

システム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。

- IBM Systems Director Navigator Performance インターフェース
- System i ナビゲーター モニター
- パフォーマンス用の CL コマンド
- IBM Performance Management for System i5

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

119 ページの『IBM Performance Management for Power Systems - IBM i のサポート』

IBM i オファリングをサポートする IBM Performance Management for Power Systems (PM for Power Systems) 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・リソース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。

## 関連資料

103 ページの『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。

## 関連情報

パフォーマンス用の CL コマンド

## 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

アプリケーションは、さまざまな理由で実行が遅いことがあります。IBM i 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・プログラムを使用すると、より多くの情報を得る助けとなります。

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 や Job Watcher などの特定のパフォーマンス・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、ジョブ・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、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てください。

パフォーマンス・トレースの開始 (STRPFRTTC) コマンドは、マルチプログラミングおよびトランザクション・データを収集します。このコマンドを実行後、トレース・ダンプ (DMPTRC) コマンド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。

## 関連概念

66 ページの『IBM i Job Watcher』

IBM i Job Watcher は、システム上の一部またはすべてのジョブ、スレッド、およびタスクについてジョブ・データの収集を行います。コール・スタック、SQL ステートメント、待機されているオブジェクト、Java JVM 統計、待機統計などを提供し、これらはジョブ関連のパフォーマンス上の問題の診断に使用されます。

68 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・リソース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。

## 関連資料

103 ページの『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。

134 ページの『Performance Tools』

Performance Tools ライセンス・プログラムには、オペレーティング・システムで使用可能な基本パフォーマンス・ツールの機能を補足または拡張する、数多くの補足機能があります。

パフォーマンス・トレースの開始 (STRPFRTTC) コマンド

マルチプログラミング・レベル (MPL) およびトランザクション・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 (STRPFRTTC) コマンド』を参照してください。

28 ページの『i5/OS における Java パフォーマンス』

i5/OS には、システム上の 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リソースがいくつかあります。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。

## トレース・データのダンプ:

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い方法ではありません。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きたいものです。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、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。収集サービスを使用して、トレース情報の収集と同時に、サンプル・ベースの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、CRTPFRTTA 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。

## 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

## 関連資料

トレースのダンプ (DMPTRC) コマンド

内部トレース表内の情報をデータベース・ファイルに書き込むには、『トレースのダンプ (DMPTRC) コマンド』を参照してください。

## メモリーのダンプ:

主記憶域情報のダンプ (DMPMEMINF) コマンドは、主記憶域のページに関する情報をファイルにダンプします。

記憶域データをダンプするには、次のコマンドを実行します。

```
DMPMEMINF OUTFILE(MYLIBRARY/DMPMEMFILE)
```

ダンプを表示するコマンドは、以下の SQL に類似しています。

```
SELECT count(*),POOL, OBJNAME, LIBNAME FROM mylibrary/dmpmemfile
group BY POOL, OBJNAME, LIBNAME
order by 1 desc
```

## 関連資料

主記憶域情報のダンプ (DMPMEMINF) コマンド

主記憶域のページに関する情報をファイルにダンプするには、『主記憶域情報のダンプ (DMPMEMINF) コマンド』を参照してください。

## IBM i 待機アカウンティングの基礎

待機アカウンティングは IBM i オペレーティング・システムに組み込まれている特許取得済みテクノロジーであり、スレッドまたはタスクが何もしていないように見えるときにそれが何をしているかをユーザーが分かるようにします。

スレッドまたはタスクは、実行中でなければ待機しています。待機アカウンティング (IBM i だけにしかない概念) は、パフォーマンスの詳細分析のための非常に強力な機能です。ここでは、待機の概念とスレッドが待機する理由、およびパフォーマンス上の問題をトラブルシューティングする場合や単にアプリケーションのパフォーマンスを向上させる場合に待機アカウンティングを使用する方法について焦点を当てていきます。

ジョブは、作業を行うための基本的なメカニズムです。どのジョブも少なくとも 1 つのスレッドを持ち、複数のスレッドを持つ場合もあります。1 つ 1 つのスレッドがライセンス内部コード (LIC) タスクによって表されますが、IBM i のスレッド・レベルの構造がなくてもタスクは存在します。LIC タスクは通常、IBM i Performance Tools またはサービス・ツール以外では外部から不可視です。待機アカウンティングの概念はスレッドとタスクの両方に適用されるので、1 つの実行可能作業部分を指す場合にスレッドとタスクという用語を使用します。

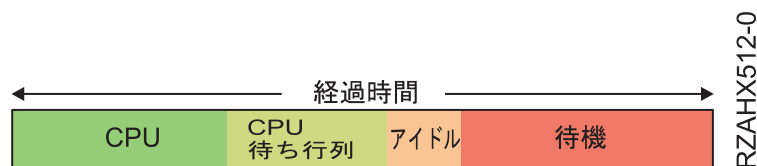
スレッドまたはタスクが取りうる基本状態として、次の 2 つがあります。

- プロセッサで実行中。「実行中」状態です。
- プロセッサで実行待機中。

主に次の 3 つの待ち状態があります。

1. 実行準備済みでプロセッサ待ち。特殊な待ち状態であり、一般に CPU キューイング と呼ばれます。これは、スレッドまたはタスクが待ち行列に入れられていて、CPU での実行を待機していることを意味します。CPU キューイングが起こるのには、いくつかの異なる理由があります。例えば、区画が過負荷状態であり、区画の受け入れ可能限度を超える作業がある場合、作業は待ち行列に入れられて CPU を待機することになります。これは、ランプ・メーターが設置された高速道路に例えることができます。高速道路が混んでいるときは、ランプ・メーターの信号が赤になるので、車は止まって本線に入れるようになるまで待たなければなりません。論理区画と同時マルチスレッド化も、CPU キューイングにつながる場合があります。
2. アイドル待機。アイドル待機は、予期される正常な待ち状態です。アイドル待機になるのは、スレッドが外部入力を待っているときです。この入力、ユーザー、ネットワーク、または別のアプリケーションからもたらされる場合があります。その入力を受け取るまで、行わなければならない作業はありません。
3. ブロック待機。ブロック待機は、共有リソースへのアクセスを同期化するための直列化メカニズムの結果です。ブロック待機は、予期される正常な待ち状態である場合があります。例として、表の行の更新、ディスク入出力操作、または通信入出力操作のための直列化アクセスが挙げられます。一方、ブロック待機が正常でない場合もあります。待ち状態を待機アカウンティングを使用して分析できるのは、このような予期しないブロック・ポイントのときです。

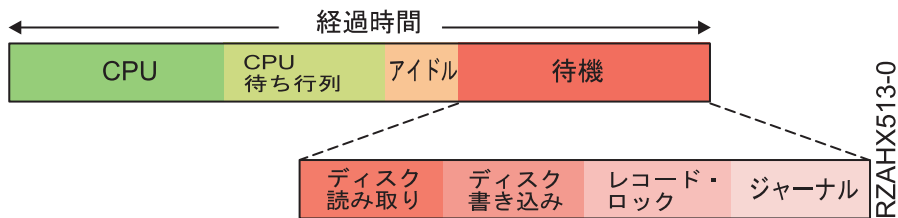
スレッドまたはタスクの存続時間を、実行または待機に費やした時間に分解してグラフィカルに考えることができます。このグラフィカル記述を「実行/待機時間こん跡」といいます。このこん跡は、大まかに次のようになります。



従来、アプリケーションのパフォーマンスを向上させるための焦点は、アプリケーションに CPU をできるだけ効率的に使用させることに当てられていました。待機アカウンティング機能のある IBM i では、待機に費やした時間を調べて、その待機時間の要因を把握できます。削減または除去できる待機要素がある場合は、全体パフォーマンスも向上させることができます。

IBM i オペレーティング・システムにおけるほぼすべての待ち状態は、識別されており列挙済みです。換言すれば、固有待機ポイントそれぞれに数値が割り当てられています。これが可能なのは、IBM がライセンス内部コードとオペレーティング・システムの両方を完全に管理しているからです。IBM i 6.1 のリリースの時点で、固有待ち状態は 268 あります。スレッドおよびタスクごとに 250 を超える固有待ち状態を追跡すると、ストレージを過剰に消費することになるので、グループ化アプローチが採用されています。固有待ち状態はそれぞれ、32 あるグループ（「バケット」）のうちの 1 つに割り当てられています。スレッドまたはタスクが待ち状態に入ったときと待ち状態から抜けたとき、タスク・ディスパッチャーが待ち状態を該当グループにマッピングします。

待機アカウンティングを使用して実行/待機時間こん跡を調べることにより、スレッドまたはタスクの待機時間について、その構成要素を識別できるようになりました。以下に例を示します。



スレッドの待機時間の要因がディスク・データの読み取りと書き込み、直列化アクセスのためのレコードのロッキング、およびデータのジャーナリングだった場合、待機をこのように分解してとらえることができます。関係している待機のタイプが分かると、確認すべきことがいくつか出てきます。この例の場合、その中には以下のような事柄が含まれます。

- ディスク読み取りでページ不在になっていないか。そうであれば、プール・サイズは適切か。
- ディスクの読み取りと書き込みの要因になっているのはどのプログラムか。削減または除去できる不必要な入出力はないか。あるいは、入出力を非同期で行えないか。
- レコード・ロッキング・ストラテジーは最適か。あるいは、レコードを不必要にロックしていないか。
- ジャーナル処理されているのはどのファイルか。すべてのジャーナルが必要か、また最適に構成されているか。

IBM i 6.1 のリリースの時点で、定義済みの 32 の待機グループ（「バケット」）は、以下のとおりです。待機グループの定義はリリースによって異なり、今後変更される可能性があります。

1. CPU 上のディスパッチ時間
2. CPU キューイング
3. 予約済み
4. その他の待機
5. ディスク・ページ不在
6. 障害なしディスク読み取り
7. ディスク・スペース使用の競合
8. ディスク操作開始の競合
9. ディスク書き込み
10. ディスク (その他)
11. ジャーナリング
12. セマフォの競合
13. mutex の競合
14. マシン・レベルのゲートの直列化 - IBM サポートをコール
15. 占有の競合 - IBM サポートをコール
16. データベース・レコード・ロックの競合
17. オブジェクト・ロックの競合
18. 不適格な待機
19. 主記憶域プールの競合 - IBM サポートをコール
20. クラシック Java ユーザー (ロックを含む)
21. クラシック Java JVM
22. クラシック Java (その他)
23. ソケット受け入れ (アイドル)



- | 24. ソケット送信
- | 25. ソケット受信
- | 26. ソケット (その他)
- | 27. IFS
- | 28. PASE
- | 29. データ待ち行列受け取り
- | 30. アイドル/作業待ち
- | 31. 同期トークンの競合
- | 32. 異常な競合 - IBM サポートをコール

| これらの待機グループの多くは、アプリケーションの待機分析をするうえで表面的なものにすぎない場合も  
| あります。これらの状況でアプリケーションが何を行っていてなぜ待機しているのかを理解すると、不必要  
| な待機を削減したり除去したりする助けになることがあります。

| グループ 16 (データベース・レコード・ロックの競合) を例にとると、実際にはこのグループの内容とし  
| て、いくつかの異なる待機が列挙されます。以下の待機があります。

- | • 読み取り
- | • 更新
- | • 脆弱
- | • 転送
- | • 検査
- | • 競合出口

### | **ホルダーとウェイター**

| IBM i は、スレッドまたはタスクが待機しているリソースを追跡するだけでなく、リソースが割り振られ  
| たスレッドまたはタスクも追跡します。これは非常に強力な機能です。「ホルダー」は、直列化リソースを  
| 使用しているスレッドまたはタスクです。「ウェイター」は、その直列化リソースへのアクセスを要求して  
| いるスレッドまたはタスクです。

### | **コール・スタック**

| IBM i は、スレッドまたはタスクごとにコール・スタックも管理します。これは、待機アカウンティング  
| 情報から独立しています。コール・スタックは呼び出されたプログラムを示しており、待ち状態を把握す  
| る、つまり結果としてリソースを保持していたかまたはリソースの利用を要求していたロジック部分を知る  
| うえで、非常に役立つことがあります。ホルダー、ウェイター、コール・スタックの組み合わせは、待ち状  
| 態を分析するための非常に強力な機能になります。

### | **データの収集と分析**

| IBM i には、待機アカウンティング情報を収集するためのパフォーマンス・データ収集メカニズムとし  
| て、収集サービスと Job Watcher の 2 つがあります。Job Watcher は、ホルダーとウェイターの情報お  
| よびコール・スタックも収集します。パフォーマンス・データが収集されると、データをグラフィカルに分  
| 析できます。iDoctor 製品には、パフォーマンス・データをグラフィカル表示するための Windows® クラ  
| イアント機能があります。また、IBM i 6.1 では、IBM Systems Director Navigator for i Web コンソール  
| に、Web ブラウザー・インターフェースでパフォーマンス・データをグラフィカル表示するための「デー  
| タの調査」機能があります。

## 関連概念

- | 78 ページの『データの調査』
- | 「データの調査」タスクを選択すると、強力なツールである Performance Data Investigator が起動します。
- | このツールを使用すると、パフォーマンス収集に保管されたデータを、図表または表の形式で表示および分析できます。
- | 66 ページの『IBM i Job Watcher』
- | IBM i Job Watcher は、システム上の一部またはすべてのジョブ、スレッド、およびタスクについてジョブ・データの収集を行います。コール・スタック、SQL ステートメント、待機されているオブジェクト、Java JVM 統計、待機統計などを提供し、これらはジョブ関連のパフォーマンス上の問題の診断に使用されます。

## 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。

### 状態

最近システム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。

### 詳細

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。i5/OS および Performance Tools ライセンス・プログラム (5770-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 リソースが 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 System i 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 リソースが消費されている場合は、欠落ファイル呼び出ししているか、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、システムおよびコンポーネント報告書を使用して、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・リソースを割り当てる。

- 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、パフォーマンスを自動的に調整することができます。
3. ジョブ要約報告書を実行して、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。

## 関連概念

25 ページの『パフォーマンスの自動的な調整』

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。

## 関連資料

STRPFRTRC コマンド

マルチプログラミング・レベル (MPL) およびトランザクション・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 (STRPFRTRC) コマンド』を参照してください。

## 関連情報

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・リソース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。

## 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。

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、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。

- IBM Systems Director Navigator Performance インターフェース
- パフォーマンス用の CL コマンド
- Performance Tools プラグイン
- System i ナビゲーター・モニター

## 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。

- IBM Systems Director Navigator Performance インターフェース
- IBM Performance Management for System i5
- Performance Tools プラグイン
- System i ナビゲーター・グラフ・ヒストリー

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

119 ページの『IBM Performance Management for Power Systems - IBM i のサポート』

IBM i オファリングをサポートする IBM Performance Management for Power Systems (PM for Power Systems) 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・リソース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。

116 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。

## 関連タスク

136 ページの『Performance Tools プラグイン』

システム・リソースの使用率のデータを System i 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。

## 関連資料

103 ページの『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。

## 関連情報

パフォーマンス用の CL コマンド

## パフォーマンスの調整

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。

パフォーマンス調整の主な目的は、システム・リソースを最も効率的に使用することです。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。

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・リソース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。

## 関連資料

### スレッド類縁性システム値

2 次スレッドが初期スレッドと同じプロセッサとメモリのグループの類縁性を持つかどうかを指定するには、『スレッド類縁性システム値』を参照してください。

### プロセッサ・マルチタスキング・システム値

プロセッサ・マルチタスキングをオン、オフ、またはシステムにより決定のいずれかに指定するには、『プロセッサ・マルチタスキング・システム値』を参照してください。

## 基本システム調整の実行

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。

### このタスクについて

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

マシン・プールを、10 フォールト/秒以下に調整する。

- 初期ユーザー・プール・サイズを決定する

すべてのユーザー・プールに対するフォールトの合計が、プロセッサ時間にプロセッサの使用率 (%) を乗じた値より小さくなるように、ユーザー・プールを調整する。たとえば、4 つのプロセッサが 50 % の使用率 ( $4 * 50 = 200$ ) で実行しているシステムでは、フォールトを 200 フォールト/秒より小さくなるように設定する。

#### 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。

#### 手順

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 (OPNQRYF)、アプリケーション・エラー、サインオフ活動などがあります。
2. WRKSYSSTS、WRKDSKSTS、WRKACTJOB、および WRKSYSACT CL コマンドを使用して、パフォーマンス・データを表示する。
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。

## タスクの結果

### 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必要があります。

## 関連資料

9 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。

## パフォーマンスの自動的な調整

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。

システム・リソース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

QPFRAJ 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、System i ナビゲーターで、「構成およびサービス」→「システム値 (System Values)」→「パフォーマンス (Performance)」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、QPFRAJ 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、System i ナビゲーターで、「構成およびサービス」→「システム値 (System Values)」→「パフォーマンス (Performance)」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、QPFRAJ 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、System i ナビゲーターで、「構成およびサービス」→「システム値 (System Values)」→「パフォーマンス (Performance)」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、QPFRAJ 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・プール活動レベル (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。

### 関連情報

メモリー・プール

メモリー・プールについては、『メモリー・プール』のトピック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

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。

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。一部のモデルは、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® 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有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、どの程度パフォーマ




ンスの向上が期待できるかのガイドラインについては、 *System i 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。

#### 関連資料

##### QPRCMLTTSK システム値

プロセッサ・マルチタスキングをオン、オフ、またはシステムにより決定のいずれかに指定します。

#### 関連情報


 「IBM i でのパフォーマンス管理」 Web サイト (英語) - 「Performance Capabilities Reference」 PDF  
QPRCMLTTSK システム値を使用することにより期待できるパフォーマンス向上に関するガイドラインについては、「IBM i でのパフォーマンス管理」 Web サイト (英語) から、適切な「Performance Capabilities Reference」 PDF を選択してください。

## e-business のパフォーマンス

e-business 環境のパフォーマンスの管理には、システム管理者にとっての新しい問題がいくつか伴います。

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。

#### 関連情報

 Domino for iSeries sizing and performance tuning  
Domino for iSeries のパフォーマンスの情報については、IBM Redbooks 資料の「Domino for iSeries のサイジングとパフォーマンス・チューニング」を参照してください。

## クライアント・パフォーマンス

多くの場合、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。

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ツーエンド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。

- 十分なメモリ容量にします。複雑なフォームとグラフィックス、およびリソース集中アプレット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、ネットワーク設計、ハードウェア・リソース、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サーバーの通信リソース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。

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、ご使用のサーバーの通信リソースのモニターと調整に使用できるリソース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。

### 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

8 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。

### 関連資料

103 ページの『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。

### 関連情報

 「IBM i でのパフォーマンス管理」 Web サイト (英語) - 「Performance Capabilities Reference」 PDF  
「IBM i でのパフォーマンス管理」 Web サイト (英語) から、適切な「Performance Capabilities Reference」 PDF を選択してください。「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。特に通信リソースの計画と管理について、第 5 章『Communications Performance』を参照してください。

SystemiNetwork.com

この Web サイトには、ネットワーク計画とリソースを最適化するための多数の情報リソースがあります。特に、『Cultivate your AS/400 Networks』と『8 tools for better network performance』を参照してください。

## i5/OS における Java パフォーマンス

i5/OS には、システム上の 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リソースがいくつかあります。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。

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、i5/OS 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。

i5/OS 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、以下の情報リソースを利用してください。

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

Tuning Java program performance


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。

## 関連資料

14 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』

アプリケーションは、さまざまな理由で実行が遅いことがあります。IBM i 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・プログラムを使用すると、より多くの情報を得る助けとなります。

## 関連情報

 「IBM i でのパフォーマンス管理」Web サイト (英語) - 「Performance Capabilities Reference」PDF  
「IBM i でのパフォーマンス管理」Web サイト (英語) から、適切な「Performance Capabilities Reference」PDF を選択してください。「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 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 Performance on IBM eServer iSeries Servers

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 IBM Redbooks 資料をお読みください。

WebSphere J2EE Application Development for the IBM eServer iSeries Server

この IBM Redbooks 資料には、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。

## IBM HTTP Server パフォーマンス

IBM HTTP サーバーは多くの場合 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。

IBM HTTP Server for i5/OS 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。またいくつかの機能によって 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。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。IBM HTTP Server for i5/OS はさらに、Web Performance Monitor および Web Performance Advisor を提供します。

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、以下の情報リソースを参照してください。

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

## 関連情報

収集サービス・データ・ファイル: QAPMHTTPB

収集サービス・データ・ファイル: QAPMHTTPD

IBM HTTP Server for i5/OS

HTTP Server on i5/OS 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。このトピックにはさらに、Web Performance Monitor、Web Performance Advisor、および Fast Response Caching Accelerator (FRCA) の説明も含まれています。

☞ 「IBM i でのパフォーマンス管理」 Web サイト (英語) - 「Performance Capabilities Reference」 PDF  
「IBM i でのパフォーマンス管理」 Web サイト (英語) から、適切な「Performance Capabilities Reference」 PDF を選択してください。「Performance Capabilities Reference」には、システム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、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。

☞ IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eServer iSeries servers  
HTTP Server (Powered by Apache) for i5/OS の詳細な説明については、この IBM Redbooks 資料を参照してください。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 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 IBM Redbooks 資料を参照してください。この資料には、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® Application Server は、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。

WebSphere 環境で、システムのパフォーマンスを管理する場合、管理者にとって、いくつかの課題があります。Web ベースのトランザクションによって消費されるリソースは多くなることもあり、しかもそこで消費されるリソース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システム・リソースの調整方法については、以下のトピックと情報リソースを参照してください。

## 関連情報

収集サービス・データ・ファイル: QAPMWASAPP

収集サービス・データ・ファイル: QAPMWASCFG

収集サービス・データ・ファイル: QAPMWASEJB

パフォーマンス・データ・ファイル: QAPMWASRSC

収集サービス・データ・ファイル: QAPMWASSVR

📄 [WebSphere Application Server Performance の Web サイト](#)

この Web サイトには、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リソースがあります。この情報リソースは、サーブレット、Java Server Pages (JSPs)、および Enterprise JavaBeans (EJBs) を使用する環境では特に有用です。

📄 [DB2 UDB/WebSphere Performance Tuning Guide](#)

この IBM Redbooks 資料には、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 on IBM eServer iSeries Servers](#)

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 IBM Redbooks 資料をお読みください。

📄 [WebSphere V3 Performance Tuning Guide](#)

この IBM Redbooks 資料には、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。

📄 [「IBM i でのパフォーマンス管理」 Web サイト \(英語\) - 「Performance Capabilities Reference」 PDF](#)

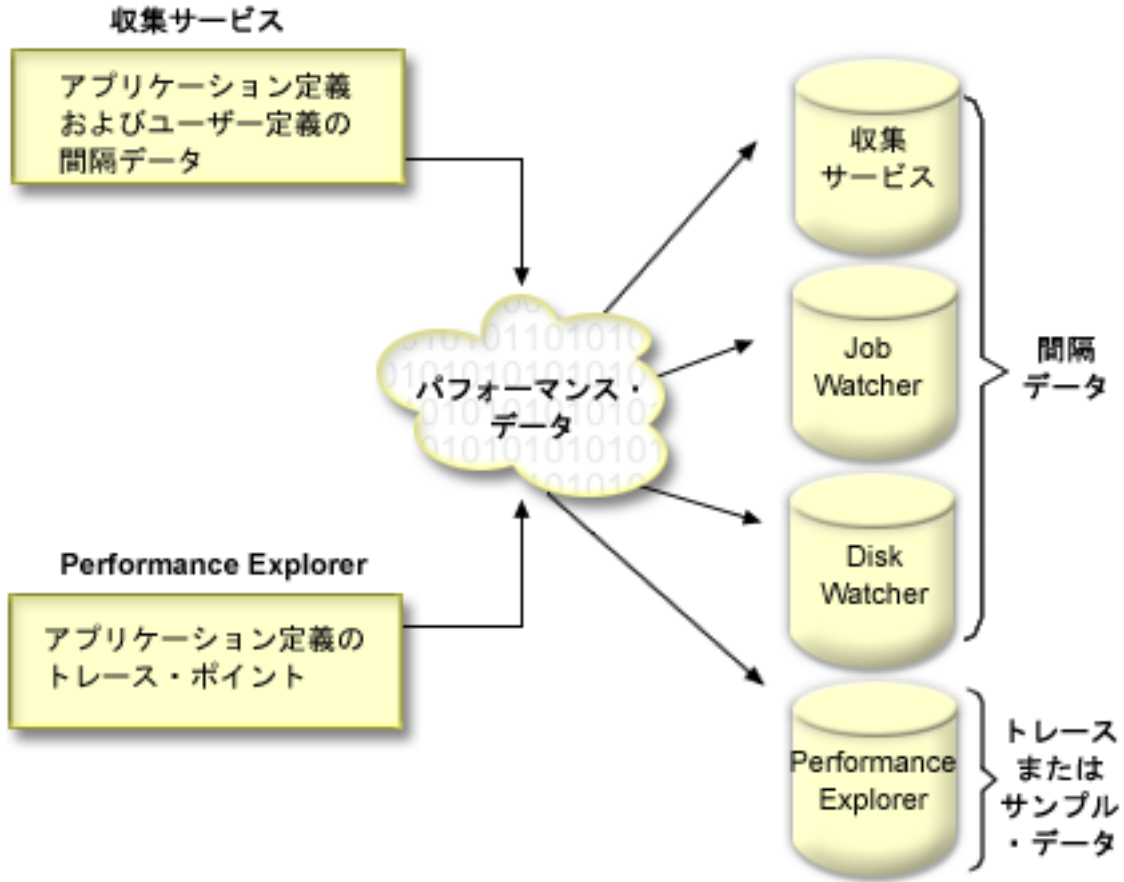
「IBM i でのパフォーマンス管理」 Web サイト (英語) から、適切な「Performance Capabilities Reference」 PDF を選択してください。「Performance Capabilities Reference」には、ご使用のサーバーを最適なパフォーマンスに構成したり調整するときに役立つ詳細な情報、報告書、および例が記載されています。特に、WebSphere Application Server 特有のパフォーマンス上のヒントについて、第 6 章『Web Server and Web Commerce』を参照してください。

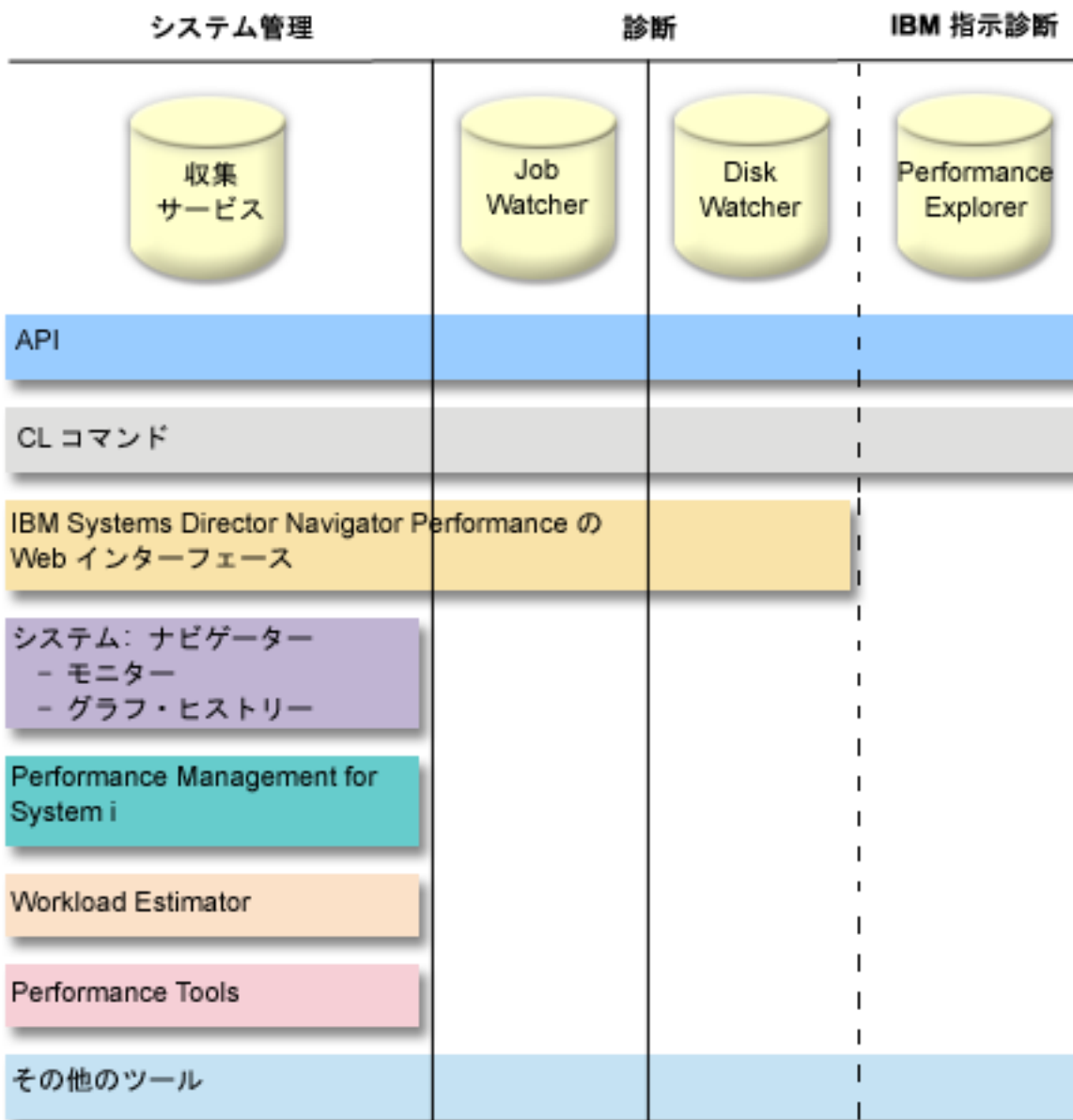
---

## 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。データを収集するアプリケーションもあれば、収集されたデータの表示や分析、モニター、管理に使用されるアプリケーションもあります。

以下の図は、主なパフォーマンス・アプリケーションを表しています。雲の形は、収集可能なシステム内に存在するすべてのデータを表しています。データにアクセスし、収集できる 4 つのコレクターがあります。最終的に、コレクターにより収集されたデータは、一連のデータベース・ファイル内に保管されます。





各コレクターには固有の特性があります。

#### 収集サービス

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。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。収集サービス・データは管理収集オブジェクト内に保管され、変換されてデータベース・ファイル内に置かれます。収集される間隔データは、アプリケーション定義またはユーザー定義のいずれかの間隔データで指定されます。

#### IBM i Job Watcher

IBM i Job Watcher は、システム上の一部またはすべてのジョブ、スレッド、タスクについてジョブ・データの収集を行います。コール・スタック、SQL ステートメント、待機されているオブジェクト、Java JVM 統計、待機統計などを提供し、これらはジョブ関連のパフォーマンス上の問題の診断に使用されます。

## IBM i Disk Watcher

IBM i Disk Watcher は、ディスク関連のパフォーマンス上の問題を診断するためのディスク・パフォーマンス・データの収集を行います。

## Performance Explorer

Performance Explorer は、問題を診断するためのプログラムおよびアプリケーション・レベルでの詳細データの収集を行います。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。Domino<sup>®</sup>、NetServer、または WebSphere サーバーなどがあるアプリケーション定義の Performance Explorer トレース・ポイントにより、収集されるデータが指定されます。これは IBM の指示に従って使用するようになっています。Performance Explorer データは管理収集オブジェクト内に保管され、変換されてデータベース・ファイル内に置かれます。

どのデータベース・ファイルに含まれているパフォーマンス・データでも、API または CL コマンドでアクセスできます。データベース・ファイルのいくつかに含まれているパフォーマンス・データは、さまざまなツール (『データの表示と分析』でさらに説明) の 1 つ以上を使用して調査と分析を行えます。

## パフォーマンス・データ・コレクター

IBM i には、データに対してアクセスと収集が可能なコレクターが 4 つあります。各コレクターには固有の特性があります。

### 収集サービス

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

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。
- 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。



## 関連概念

### iSeries ナビゲーター

パフォーマンス・データの収集および管理に System i ナビゲーターを使用する方法については、『System i ナビゲーター』のトピックを参照してください。

#### 41 ページの『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。

#### 69 ページの『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リソース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。

## 関連タスク

### 120 ページの『PM Agent の活動化』

PM Agent はオペレーティング・システムの付属製品であり、収集機能を利用するには活動化する必要があります。

## 関連資料

### パフォーマンス収集開始 (STRPFCOL) コマンド

データ収集の開始方法については、『パフォーマンス収集開始 (STRPFCOL) コマンド』を参照してください。

### Performance Management APIs

パフォーマンス・データの収集および管理に Performance Management API を使用する方法については、『Performance Management API』のトピックを参照してください。

## 関連情報

### パフォーマンス・データ・ファイル

パフォーマンス・データが含まれているファイルの情報については、『パフォーマンス・データ・ファイル』のトピックを参照してください。

## 収集サービスの動作方法:

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。

単一収集オブジェクト内にパフォーマンス・データを保管すると、パフォーマンス・データの収集時にシステムのオーバーヘッドを低く抑えることができます。収集中にデータベース・ファイルを作成するを選択した場合、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します。この低い収集オーバーヘッドにより、詳細なパフォーマンス・データを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。

次の図は、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。例えば、CL コマンド、API、System i ナビゲーター インターフェース、および IBM Systems Director Navigator Performance 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

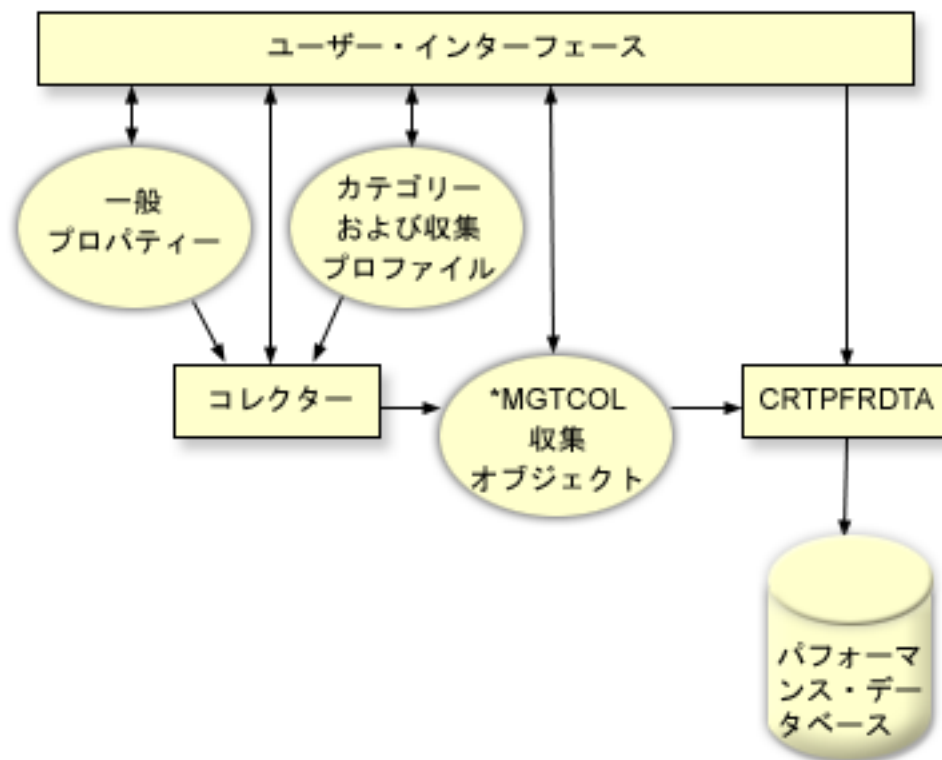
収集オブジェクト \*MGTCOL は、大量のパフォーマンス・データを保持するための効果的なストレージ・メディアとしての機能を果たします。

## 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、およびトレース・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。



収集サービスのデータからのデータベース・ファイルの作成:

この情報は、収集サービス・データからデータベース・ファイルを手動または自動で作成するために使用します。

## このタスクについて

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、後からデータベース・ファイルを作成することもできます。

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、System i 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 Agent を活動化することができます。これは収集時にデータベース・ファイルを作成します。

作成したデータベース・ファイルは、パフォーマンス報告を生成するために、IBM Systems Director Navigator for i5/OS Performance インターフェースや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・ファイルの生成とパフォーマンス報告の生成を行います。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。

## 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・カテゴリー、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、全カテゴリーのセット (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ー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ーで、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。

### 関連タスク

98 ページの『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には、以下のステップを実行します。

### 関連資料

パフォーマンス・データ作成 (CRTPFRDTA) コマンド

パフォーマンス・ファイルの作成の情報については、『パフォーマンス・データ作成 (CRTPFRDTA) コマンド』を参照してください。

### データ収集のカスタマイズ:

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。

### このタスクについて

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」プロファイルは、一般に Performance Tools によって必要とされるカテゴリーに対応します。ただし通信データは例外です。「標準 + プロトコル」プロファイルは、一般に Performance Tools によって必要とされるすべてのカテゴリーに対応し、これには通信データも含まれます。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・カテゴリー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ーごとに指定できます。多くのカテゴリーでは、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、以下のカテゴリーを除くすべてのカテゴリーはこの特定時間を使用します。

- **カスタム**・プロファイルにセットアップできる明示的な時間間隔。
- 少なくとも 5 分ごとに収集する必要がある、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ーなどの、間隔制限があるカテゴリー。

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。このサイクル間隔は、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。

### 関連タスク

98 ページの『収集サービスの構成』

収集サービスを構成するには、以下のようになります。

### 収集サービス収集プロファイル:

収集サービス収集プロファイルの説明です。収集プロファイルでは、収集対象を定義します。

- **最小** - 最も一般的に使用されるセットのパフォーマンス・データ。これは推奨される最小データ収集であり、以下のカテゴリが含まれます。
  - システム・バス: このカテゴリには、各システム・バスの操作に関するデータが入ります。
  - メモリー・プール: このカテゴリには、メモリー・プール構成データおよびメモリー・プール操作データが入ります。
  - ハードウェア構成: このカテゴリには、システムのハードウェア・リソース情報が入ります。このカテゴリには、ハードウェア・リソースの表示 (DSPHDWRSC) コマンドで獲得されるものと同じデータが入ります。複数のインスタンスが検出される場合、このデータの最初のインスタンスだけがデータベースに報告されます。
  - システム CPU: このカテゴリには、各プロセッサのシステム CPU 使用量に関するデータが入ります。
  - システム・レベル・データ: このカテゴリには、システム全体にわたって使用される一般システム・データが入ります。
  - ジョブ (MI タスクおよびスレッド): このカテゴリには、システム内のアクティブなあらゆるタスク、ジョブ、およびスレッドに関する情報が入ります。収集対象のデータは、マシン・インターフェース (MI) により提供されます。
  - ジョブ (オペレーティング・システム): このカテゴリには、システム内のすべてのアクティブ・ジョブに関する情報が入ります。収集対象のデータは、オペレーティング・システムにより提供されます。
  - ディスク・ストレージ: このカテゴリには、システム・ストレージ・ユニット・データが入ります。これには、基本ストレージ・ユニット情報およびディスク・ドライブの操作データが含まれます。
  - 入出力プロセッサ: このカテゴリには、システムの入出力プロセッサ (IOP) に関するデータが入ります。これには、アダプター・リソースによる IOP バス使用および IOP 使用率に関するデータが含まれます。
- **標準** - Performance Tools for IBM i に含まれるツールが通常必要とするデータ・カテゴリです (通信プロトコル・データを除く)。標準プロファイルのデータ・カテゴリには、最小プロファイル内のすべてのカテゴリに加え、以下が含まれます。
  - メモリー・プール調整: このカテゴリには、各システム・メモリー・プールのプール調整構成データが入ります。
  - サブシステム: このカテゴリには、活動サブシステムおよびサブシステム・プールに関するデータが入ります。複数のインスタンスが検出される場合、このデータの最初のインスタンスだけがデータベースに報告されます。
  - SNADS: このカテゴリには、システム内のアクティブ SNADS ジョブに固有のトランザクション境界情報が入ります。
  - ローカル応答時間: このカテゴリには、5254 制御装置に接続されているワークステーションの応答時間情報が入ります。応答時間データはワークステーションごとに報告され、一連の応答時間バケットに保管されます。
  - APPN: このカテゴリには、システムの APPN サポートに関するデータが入ります。記録されたデータには、一般情報と、トランザクション・タイプおよび作業活動に応じて分類されたデータの両方が含まれます。
  - SNA: このカテゴリには、システムの SNA サポートに関するデータが含まれます。データは各活動 T2 タスクごとに報告され、制御装置、タスク、およびセッション情報で構成されます。
  - TCP/IP (基本): このカテゴリには、TCP/IP のシステム全体のパフォーマンス情報が入ります。

- ユーザー定義トランザクション・データ: このカテゴリには、IBM 定義のトランザクションではなく、アプリケーション定義のトランザクションのデータが含まれます。ユーザーは固有のユーザー定義トランザクションを作成することができます。
- IBM Domino for i: このカテゴリは、IBM Domino for i ライセンス・プログラムがシステムにインストールされている場合に、このプロファイルに組み込まれます。
- IBM HTTP Server for i (powered by Apache): このカテゴリは、IBM HTTP Server for i ライセンス・プログラムがシステムにインストールされている場合に、このプロファイルに組み込まれます。
- | - 外部ストレージ: このカテゴリには、IBM i 区画に外部接続されたディスク装置の非標準化データが入ります。
- | - システム内部データ: このカテゴリには、システムの内部データが入ります。
- | - 取り外し可能ストレージ: このカテゴリには、システムに接続された取り外し可能記憶装置に関するデータ (すなわち磁気テープ装置のデータ) が入ります。
- **標準 + プロトコル** - Performance Tools for i に含まれるツールが通常必要とするデータ・カテゴリで (通信プロトコル・データを含む)。「標準 + プロトコル」プロファイル内のデータ・カテゴリには、標準プロファイル内のすべてのカテゴリに加え、以下が含まれます。
  - ネットワーク・サーバー: このカテゴリには、ネットワーク・サーバーに関する情報が入ります。統合 xSeries® サーバーの場合は、CPU 使用状況のデータが報告されます。ホスト区画 (物理リソースを提供する区画) 上の仮想入出力アダプターの場合は、それがゲスト区画に代わって提供している仮想装置サポートのために、この区画内で実行された入出力活動に関するデータが提供されます。
  - 通信 (基本): このカテゴリには、使用可能な (オンに構成されている) 各通信回線の基本プロトコル情報が入ります。
  - 通信 (端末): このカテゴリには、特定の通信回線の端末情報が入ります。データは、使用可能な (オンに構成されている) 各端末ごとに報告されます。このデータをサポートするプロトコルは、トークンリング、イーサネット、DDI、フレーム・リレー、および X.25 です。
  - 通信 (SAP): このカテゴリには、特定の通信回線のサービス・アクセス・ポイント (SAP) 情報が入ります。データは、使用可能な (オンに構成されている) 回線内に構成されている各 SAP ごとに報告されます。このデータをサポートするプロトコルは、トークンリング、イーサネット、DDI、およびフレーム・リレーです。
  - データ・ポート・サービス: このカテゴリには、データ・ポート・サービスから入手されたパフォーマンス・データが入ります。データ・ポート・サービスは、ソース・システムと、iSeries® クラスター環境内の任意数の指定済みターゲット・システムの 1 つとの間で、大容量データの転送をサポートするライセンス内部コードです。データ・ポート・サービスは、独立したリモート補助記憶域プール (ASP) のミラーリングなど、ライセンス内部コード・クライアントによって使用されます。
  - | - 論理区画: このカテゴリには、収集サービスが稼働している区画に IBM Director Server (5761-DR1) ライセンス・プログラムがインストールされている場合、適格な区画から収集されるパフォーマンス・データが入ります。データを他の区画から収集するには、IBM Director Agent (5761-DA1) ライセンス・プログラムがそれらの他の区画にインストールされていなければならない、かつサーバーがそれらの他の区画に対して許可されていなければなりません。
  - TCP/IP インターフェース: このカテゴリには、アクティブな TCP/IP インターフェースそれぞれの情報が入ります。
- **拡張キャパシティー・プランニング** - 拡張キャパシティー・プランニング・プロファイルのデータ・カテゴリには、「標準+プロトコル」プロファイルのすべてのカテゴリに加え、「PEX データ - プロセッサ効率」データ・カテゴリがあります。「PEX データ - プロセッサ効率」データ・カテゴリには、Performance Explorer (PEX) データの命令当たりの循環数が入ります。データは、キャパシティー

ー・プランニング能力を向上させるため、または他の目的のために収集できます。このカテゴリを使用する時には、以下の特別な考慮事項が適用されます。

- Performance Explorer 定義 QPMIPEXPEI が作成されます。Performance Explorer 定義がすでに存在している場合には、それが削除されて、再作成されます。
- このカテゴリでは、Performance Explorer (PEX) 収集 (セッション ID: QPMINTPEXD) を開始するために、収集サービスが必要です。この収集は、他の Performance Explorer の収集と競合する場合があります。
- 収集したデータの妥当性に影響することがあるため、QPMINTPEXD セッションは手動で終了または開始しないでください。
- このカテゴリの収集を停止すると、セッション QPMINTPEXD の Performance Explorer 収集も終了します。

IBM は、このプロファイルの使用を推奨しなくなりました。

- ・ **カスタム** - このオプションを使用すると、収集されるデータのカテゴリだけでなく、間隔の指定のカスタマイズも可能です。データのカテゴリによって異なる収集間隔にすることができます。

#### 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ないで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います。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。

## 関連概念

日付と時間システム値: 時刻

QTIME システム値の情報については、『日付と時間システム値: 時刻』のトピックを参照してください。

日付と時間システム値: 協定世界時 (UTC) からのオフセット

QUTCOFFSET システム値の情報については、『日付と時間システム値: 協定世界時 (UTC) からのオフセット』のトピック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のインプリメント:

収集サービスのユーザー定義カテゴリ機能によって、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。

## このタスクについて

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。

## 手順

1. 収集サービスで新しいカテゴリ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。
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. 新しいカテゴリを登録して、データ収集プログラムを指定します。
  - 登録: QypsRegCollectorDataCategory
  - 登録取り消し: QypsDeregCollectorDataCategory

カテゴリを登録すると、それは収集サービスによって使用可能な収集カテゴリのリストに含まれます。
4. カテゴリを収集サービス・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。
  - 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)
  - 収集オブジェクト属性の検索: QpmRtvMgtcolAttr
  - 収集オブジェクトのオープン: QpmOpenMgtcol
  - 収集オブジェクトのクローズ: QpmCloseMgtcol
  - 収集オブジェクト・リポジトリのオープン: QpmOpenMgtcolRepo
  - 収集オブジェクト・リポジトリのクローズ: QpmCloseMgtcolRepo
  - 収集オブジェクト・データの読み取り: QpmReadMgtcolData



## タスクの結果

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 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の詳細は、javadocs を .zip ファイル形式でダウンロードしてください。

## 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・キューに登録および登録取り消しするには、以下の API 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

## 関連資料

### QpmCloseMgtcol API

管理収集オブジェクトのクローズ (QpmCloseMgtcol) API は、管理収集オブジェクトのオープン (QpmOpenMgtcol) API により以前にオープンされた管理収集オブジェクトをクローズします。

### QpmCloseMgtcolRepo API

管理収集オブジェクト・リポジトリのクローズ (QpmCloseMgtcolRepo) API は、管理収集オブジェクト・リポジトリのオープン (QpmOpenMgtcolRepo) API により以前にオープンされた管理収集オブジェクトのリポジトリをクローズします。

### QpmOpenMgtcol API

管理収集オブジェクトのオープン (QpmOpenMgtcol) API は、処理のために指定の管理収集オブジェクトをオープンし、ハンドルをそのオープンした管理収集オブジェクトに戻します。

### QpmOpenMgtcolRepo API

管理収集オブジェクト・リポジトリのオープン (QpmOpenMgtcolRepo) API は、処理のために指定の管理収集オブジェクトのリポジトリをオープンします。

### QpmReadMgtcolData API

管理収集オブジェクト・データの読み取り (QpmReadMgtcolData) API は、管理収集オブジェクトのリポジトリ内の特定のレコードについての情報を戻します。

### QpmRtvActiveMgtcolName API

活動状態の管理収集オブジェクト名の検索 (QpmRtvActiveMgtcolName) API は、活動状態の管理収集オブジェクトのオブジェクト名とライブラリー名を戻します。

### QpmRtvMgtcolAttrs API

管理収集オブジェクト属性の検索 (QpmRtvMgtcolAttrs) API は、管理収集オブジェクトの属性および管理収集オブジェクトのリポジトリについての情報を戻します。

### QypsAddCollectorNotification API

コレクター通知の追加 (QypsAddCollectorNotification) API は、収集イベントについて指定のデータ待ち行列に通知を提供するコレクターに登録します。

### QypsDeregCollectorDataCategory API

コレクター・データ・カテゴリーの登録取り消し (QypsDeregCollectorDataCategory) API は、マネージメント・セントラルの収集サービス機能からユーザー定義のデータ・カテゴリーを除去します。

### QypsRmvCollectorNotification API

コレクター通知の除去 (QypsRmvCollectorNotification) API は、特定のデータ待ち行列および収集イベントのコレクターから通知登録を除去します。

### QypsRegCollectorDataCategory API

コレクター・データ・カテゴリーの登録 (QypsRegCollectorDataCategory) API は、マネージメント・セントラルの収集サービス機能の 1 つ以上のコレクター定義に、ユーザー定義のデータ・カテゴリーを追加します。

## 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 パフォーマンス・データの作成 (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。

注: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

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 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リソースを初期化します。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 more data 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・リソースは、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のインプリメント:

ここでのサンプル・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。

### 例: データ収集プログラム:

このプログラム例はいくつかのテスト・データを収集してデータ・バッファに保管し、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。

注: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

### 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() */

```

## 関連概念

44 ページの『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。

**例: データ収集プログラムを登録するためのプログラム:**

このサンプル・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: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

## C++ サンプル・コード

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

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

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

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

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

        memset( &catAttr, 0x00, sizeof(catAttr) );
        catAttr.structureSize = sizeof( Qyps_USER_CAT_ATTR );
        catAttr.minCollectionInterval = 0;
        catAttr.maxCollectionInterval = 0;
        catAttr.defaultCollectionInterval = 30; /* Collect at 30 second interval */
        memset( catAttr.qualifiedMsgId, ' ', sizeof(catAttr.qualifiedMsgId) );
        memcpy( catAttr.categoryDesc,
               "12345678901234567890123456789012345678901234567890", sizeof(catAttr.categoryDesc) );

        QypsRegCollectorDataCategory( collectorName,
                                     categoryName,
```

```

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

```

### 例: 収集オブジェクトを照会するためのプログラム:

このサンプル・プログラムは、QIBM/ProdData/OS400/CollectionServices/lib ディレクトリー・パスの ColSrv.jar ファイルに入って出荷されている Java 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: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

### 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パフォーマンス・データを収集する場合、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。

## このタスクについて

システムから収集オブジェクトを削除できます。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。

- 1 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す。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。収集サービスは、構成済み収集ライブラリー内に存在する、有効期限が切れた管理収集オブジェクトだけを削除します。

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。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。

## タスクの結果

### 関連タスク

95 ページの『収集の管理』

パフォーマンス収集の管理には、IBM Systems Director Navigator for i を使用します。

### 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。

- トランザクションの開始 (QYPESTRT, qypeStartTransaction) API
- トランザクションの終了 (QYPEENDT, qypeEndTransaction) API
- トランザクションのログ (QYPELOGT, qypeLogTransaction) API (Performance Explorer のみによる使用)
- トレース・ポイントの追加 (QYPEADDT, qypeAddTracePoint) API (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。パフォーマンス・データの作成 (CRTPFRDTA) コマンドは、このデータをユーザー定義トランザクション・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。

## 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、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。

### 関連概念

68 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・リソース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。

### 関連資料

QYPESTRT, qypeStartTransaction API

QYPEENDT, qypeEndTransaction API

QYPELOGT, qypeLogTransaction API

QYPEADDT, qypeAddTracePoint API

パフォーマンス・データ作成 (CRTPFRDTA) コマンド

### 関連情報

パフォーマンス・データ・ファイル: QAPMUSRTNS

### 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:

この 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: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

```
//*****  
// 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;
            (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 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:

この 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: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

```
import com.ibm.iseries.collectionservices.PerformanceDataReporter;
```

```

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

public class TestTXApi
{
    static TestTXApiThread[] thread;

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

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

    static
    {
        int i;

        // initialize transaction type strings and byte arrays

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

```



```

TxTypeString[17] = "Transaction type 17";
TxTypeString[18] = "Transaction type 18";
TxTypeString[19] = "Transaction type 19";

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

}/* static */

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

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

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

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

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

    } catch (Exception e)
    {
        System.out.println( "Oops! Cannot process parameters!" );
        return;
    }
    else
    {
        System.out.println( "Incorrect Usage." );
    }
}

```

```

        System.out.println( "The correct usage is:" );
        System.out.println( "java TestTXApi numberOfTXPerThread numberOfThreads
log|nolog enable|disable secsToDelay");
        System.out.println( "%tlog will make the program cut 1 log transaction per start / end pair");
        System.out.println( "%tdisable will disable performance collection to minimize overhead");
        System.out.print( "%nExample: %"java TestTXApi 10000 100 log enable 3%" will call " );
        System.out.println( "cause 10000 transactions for each of 100 threads");
        System.out.println( "with 3 seconds between start and end of transaction");
        System.out.println( "Plus it will place additional log call and will enable reporting." );
        return;
    }

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

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

    TestTXApi t = new TestTXApi( );

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

    long startTime = System.currentTimeMillis();

    t.runTests( numberOfThreads );

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

    long endTime = System.currentTimeMillis();

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

}/* main() */

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

}/* prepareTests() */

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

```

```

}/* runTests() */

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

    private PerformanceDataReporter    pReporter;

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

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

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

        }/* constructor */

    public void run()
    {
        int i;

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

        }/* run() */

    }/* class TestTXApiThread */
}/* class TestTXApi */

```

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。このデータは、収集サービスのデータベース・ファイル QAPMJOBWT および QAPMJOBWTD に報告されます。

**注:** QAPMJOBWTD ファイルに Query を出すためには、ジョブの CCSID は、システムにインストール済みの 1 次言語の CCSID に設定する必要があります (65535 バイナリー・データにはありません)。

ジョブの待機統計を表示する別のツールとして、IBM i5/OS Job Watcher があります。IBM i5/OS Job Watcher は、選択した一連のジョブ、スレッド、およびライセンス内部コード (LIC) プログラム・タスクについてのリアルタイム情報を戻します。指定した時間間隔で、IBM i5/OS Job Watcher は、1 つのスレッド (ジョブ単位) からすべてのスレッド (ジョブ単位) まで、どこからでもサンプルを採取します。IBM i5/OS Job Watcher は、ジョブ、タスク、およびスレッドの詳細な待機統計を含めて、多様なパフォーマンス・データを収集します。

待ち状態データを累積する、32 の待ちバケットがあります。これらの静的待ちバケットは、収集サービスおよび IBM i5/OS Job Watcher により使用され、待ち状態データの継続的ビューを提供します。収集サービスでは、これらのバケットからのデータは、ファイル QAPMJOBWT および QAPMJOBWTG で報告されます。Job Watcher では、これらのバケットからのデータは QAPYJWTD および QAPYJWSTS で報告されます。

## 関連概念

66 ページの『IBM i Job Watcher』

IBM i Job Watcher は、システム上の一部またはすべてのジョブ、スレッド、およびタスクについてジョブ・データの収集を行います。コール・スタック、SQL ステートメント、待機されているオブジェクト、Java JVM 統計、待機統計などを提供し、これらはジョブ関連のパフォーマンス上の問題の診断に使用されます。

16 ページの『IBM i 待機アカウンティングの基礎』

待機アカウンティングは IBM i オペレーティング・システムに組み込まれている特許取得済みテクノロジーであり、スレッドまたはタスクが何もしていないように見えるときにそれが何をしているかをユーザーが分かるようにします。

## 関連情報

パフォーマンス・データ・ファイル: QAPMJOBWT

パフォーマンス・データ・ファイル: QAPMJOBWTD

パフォーマンス・データ・ファイル: QAPMJOBWTG

A jobs life

実行管理機能ジョブ属性 (Work management job attributes)

## 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られると想定します。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期できます。

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。

#### すべての区画の CPU 使用率の収集と表示:

複数区画を使用する場合、区画で IBM i、AIX®、または Linux® が稼働しているかどうかにかかわらず、全区画にわたる処理能力の全体使用率を把握することが重要になることがあります。IBM i は、このようなデータを収集して表示するための手段を提供します。

#### データの収集

物理システムの CPU 使用率を収集するには、以下の構成要件を満たす必要があります。

- IBM i 6.1 を搭載し、ファームウェア・レベルが xx340\_061 以降の POWER6 ハードウェア。
- パフォーマンス・データ収集対象区画のパフォーマンス・データの収集が使用可能に設定済み。このデータは 1 つの区画に対して収集するだけで済みますが、その区画は IBM i 区画でなければなりません。収集される CPU 使用率情報には、IBM i が稼働している区画だけでなく、AIX および Linux が稼働している区画で行われた作業も反映されます。一方、AIX および Linux は、このデータの収集をサポートしていません。

このパフォーマンス・データの収集を使用可能にするためには、HMC または Integrated Virtualization Manager (IVM) の構成パラメーターを設定する必要があります。HMC の場合、プロセッサ構成タブに「Allow performance information collection (パフォーマンス情報収集を許可)」チェック・ボックスがあります。このデータを収集する IBM i 区画のこのチェック・ボックスを選択します。IVM を使用する場合は、lssyscfg コマンドを使用して、all\_perf\_collection (区画が共用プロセッサ・プール使用率を検索することを許可) パラメーターを指定します。パラメーターの有効値は、0 (権限を許可しない (デフォルト)) および 1 (権限を許可する) です。

パフォーマンス・データ収集のサポートが使用可能になると、収集サービスがこの追加情報を収集するようになります。収集間隔ごとに、収集サービスがハイパーバイザーから区画構成と使用率の情報を収集します。データは、収集サービスのデータベース・ファイル QAPMLPARH に保管されます。収集サービスのデータベース・ファイル QAPMSYSPRC に、物理プロセッサ使用率を取得することもできます。

#### データの表示

IBM Systems Director Navigator for i の Performance Data Investigator ツールを使用して、収集されたデータを Web 上でグラフィカル表示できます。これらの図表は、収集サービスのコンテンツ・パッケージの中の「物理システム」(Physical System) フォルダーにあります。このデータを使用する図表の例として、以下のものがあります。

- 論理区画の概要
- 論理区画別の提供されたプロセッサ時間
- 論理区画別の上限なしの使用プロセッサ時間
- 物理共用プロセッサ・プール使用率

- 物理プロセッサ別の物理プロセッサ使用率
- 論理区画別の専用プロセッサ使用率
- プロセッサ状況別の物理プロセッサ使用率の概要
- プロセッサ状況別の物理プロセッサ使用率の詳細

#### 関連タスク

- 78 ページの『Performance Data Investigator』
- Performance Data Investigator は、パフォーマンス・データ上の Web ベースの GUI に、インタラクティブな図表と表を提供します。これを使用することにより、コレクター (収集サービス、IBM i Job Watcher、IBM i Disk Watcher、Performance Explorer) それぞれのパフォーマンス・データの表示と分析が可能になります。

#### 関連情報

- 収集サービス・データ・ファイル: QAPMLPARH
- 収集サービス・データ・ファイル: QAPMSYSPRC

#### ARM パフォーマンス・データの収集:

Application Response Measurement (ARM) パフォーマンス・データを収集するのに、収集サービスを使用できます。

ARM API は、ARM トランザクション用のパフォーマンス・データを収集します。(ARM API は、Open Group により開発された API で、アプリケーション・トランザクションの進行をアプリケーションが報告できるようにします。) これらのトランザクションは、QAPMARMTRT および QAPMUSRTNS データベース・ファイルにおいて報告されます。

- ARM API についてさらに詳しく知るためには、[www.theopengroup.org/arm](http://www.theopengroup.org/arm) にある Open Group の Web サイトにアクセスしてください。

#### 短存続期間のスレッドおよびタスク:

- 収集サービスは、サンプル間隔中にプロセッサ時間を使用したすべてのジョブ、タスク、2 次スレッドについて、パフォーマンス・データを収集します。このデータは、QAPMJOBMI ファイル内のレコードを通して報告されます。

- ごくわずかの作業をしてすぐに終了する 2 次スレッドやタスクが作成される場合もあります。これらの存続期間は通常、1 秒未満です。これが頻繁に継続して起これば、問題になることがあります。収集オブジェクトのサイズと QAPMJOBMI ファイルのメンバーが著しく増える可能性があるためです。また、データを収集してファイルを生成するための CPU 使用率が上がるだけでなく、そのデータを取り込むさまざまなツールによって使用されるリソースが増えることとなります。

- ある特定の短存続期間のスレッドやタスクの範囲で消費されるリソースは大したことはないとはいえ、これらがまとまると、合計システム使用率の重要な因子になることがあります。そのため、これらをただ無視するわけにはいきません。

- IBM i 7.1 から、存続期間が特定のしきい値よりも短いタスクと 2 次スレッドのデータを、収集サービスが累算するようになりました。短存続期間の 2 次スレッドは、同種のスレッドを持つジョブのなかでジョブ別に累算されます。短存続期間のタスクは、プロセッサ・ノード別に累算されます。この累積データは、他のタスクや 2 次スレッドが報告されるのと同じように、サンプリング時に報告されます。

| QAPMJOBMI ファイルに、「短持続期間項目カウント」フィールドが新しく追加されました。このフィールドには、短持続期間のスレッドまたはタスクのデータを含んだレコードの値 (ゼロより大) が入ります。この値は、示されたジョブまたはノード分として間隔内で累算されたエンティティの数です。  
| QAPMCONF ファイルは、収集時に使用された短持続期間しきい値を報告します。QAPMCONF の項の GKEY = "F1" の説明を参照してください。

| デフォルトでは、タスクとスレッドの両方に使用されるしきい値は 1000ミリ秒です (1000 ミリ秒より短い持続期間で終了するスレッドおよびタスクは個別に報告されません)。この処理を無効にする必要がある場合、または異なるしきい値が必要な場合は、以下のように環境変数を使用して行うことができます。

| 変数 QPM\_TASK\_SL\_THRESHOLD を作成することによって、短持続期間タスクの処理を制御できます。

| 変数 QPM\_THREAD\_SL\_THRESHOLD を作成することによって、短持続期間 2 次スレッドの処理を制御できます。

| 環境変数に関連付けられた値が、適用するしきい値 (ミリ秒単位) になります。値を 0 または NULL にすると、すべてのスレッドまたはタスクが報告されるようになります。これらの環境変数は、収集サービスのコレクター・ジョブ QYSPFRCOL が認識できるように、システム・レベルの変数として作成する必要があります。値は収集の開始時に 1 度だけ取得されます。そのため、新しい値が使用されるようにするには、変更を行った後でアクティブ収集を循環させる必要があります。

| 次に示すのは、環境変数を作成してデフォルト値として 1000 ミリ秒を設定する例です。

```
| ADDENVVAR ENVVAR(QPM_TASK_SL_THRESHOLD) VALUE(1000) LEVEL(*SYS)  
| ADDENVVAR ENVVAR(QPM_THREAD_SL_THRESHOLD) VALUE(1000) LEVEL(*SYS)
```

## IBM i Job Watcher

IBM i Job Watcher は、システム上の一部またはすべてのジョブ、スレッド、およびタスクについてジョブ・データの収集を行います。コール・スタック、SQL ステートメント、待機されているオブジェクト、Java JVM 統計、待機統計などを提供し、これらはジョブ関連のパフォーマンス上の問題の診断に使用されます。

IBM i Job Watcher は、リフレッシュごとに終了間隔の差分情報が計算されるという点において、CL コマンドの WRKACTJOB および WRKSYSACT とサンプリング機能が似ています。監視されるジョブ、スレッド、またはタスクから収集されるデータは、侵入的ではない方法でそれが実行されます。

IBM i Job Watcher は、IBM Systems Director Navigator for i Performance インターフェースまたは CL コマンドにより構成および管理できます。



## 関連概念

16 ページの『IBM i 待機アカウンティングの基礎』  
待機アカウンティングは IBM i オペレーティング・システムに組み込まれている特許取得済みテクノロジーであり、スレッドまたはタスクが何もしていないように見えるときにそれが何をしているかをユーザーが分かるようにします。

## 関連タスク

101 ページの『IBM i Job Watcher の管理』  
IBM i Job Watcher の管理には、IBM Systems Director Navigator for i を使用します。

## 関連資料

Job Watcher 定義の追加 (ADDJWDFN)

Job Watcher 収集中に収集するパフォーマンス・データの指定については、『Job Watcher 定義の追加 (ADDJWDFN) コマンド』を参照してください。

Job Watcher の終了 (ENDJW)

Job Watcher 収集の終了については、『Job Watcher の終了 (ENDJW) コマンド』を参照してください。

Job Watcher 定義の除去 (RMVJWDFN)

システムからの Job Watcher 定義の除去については、『Job Watcher 定義の除去 (RMVJWDFN) コマンド』を参照してください。

Job Watcher の開始 (STRJW)

Job Watcher 収集の開始については、『Job Watcher の開始 (STRJW) コマンド』を参照してください。

## IBM i Disk Watcher

IBM i Disk Watcher は、ディスク関連のパフォーマンス上の問題を診断するためのディスク・パフォーマンス・データの収集を行います。

IBM i Disk Watcher により、ディスク装置に対する入出力操作に関するデータを、アクセスされているオブジェクト、ファイル、プロセス、スレッド、およびタスクを判別するうえで頻繁に必要とされるランタイム・データと共に入手できます。このツールは、WRKDSKSTS、WRKSYSSTS、および WRKSYSACT などのツールの提供内容を上回る仕方です。Disk Watcher は、ディスク入出力データを関連タスクおよびオブジェクト名と共に収集するために、短期間および長期間のトレースを使用するメカニズムを提供します。

このツールは、次のように使用できる可能性があります。

- マルチパス・ディスク装置上の入出力操作のパフォーマンスの評価
- 入出力キューイングのパフォーマンスの評価
- データを装置間で再展開することによりどのようにパフォーマンスが向上するかの判別
- デバイス、IOA、またはバスの最適な配置の判別

IBM i Disk Watcher は、IBM Systems Director Navigator for i Performance インターフェースまたは CL コマンドにより構成および管理できます。

## 関連タスク

99 ページの『IBM i Disk Watcher の管理』

IBM i Disk Watcher の管理には、IBM Systems Director Navigator for i を使用します。

## 関連資料

Disk Watcher 定義の追加 (ADDDWDFN)

システムからの Disk Watcher 定義の追加については、『Disk Watcher 定義の追加 (ADDDWDFN) コマンド』を参照してください。

Disk Watcher の終了 (ENDDW)

Disk Watcher 収集の終了については、『Disk Watcher の終了 (ENDDW) コマンド』を参照してください。

Disk Watcher 定義の除去 (RMVDWDFN)

システムからの Disk Watcher 定義の除去については、『Disk Watcher 定義の除去 (RMVDWDFN) コマンド』を参照してください。

Disk Watcher の開始 (STRDW)

Disk Watcher 収集の開始については、『Disk Watcher の開始 (STRDW) コマンド』を参照してください。

## 関連情報

Disk Watcher データ・ファイル

## 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・リソース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。

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・リソース、アプリケーション、プログラム、プロシージャー、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。

資料 AS/400® Performance Explorer Tips and Techniques は、Performance Explorer 機能の追加の例、および Performance Explorer 拡張トレース・サポートの例を提供します。

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に使用するツールです。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。

## 関連概念

53 ページの『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。

## 関連情報



AS/400 Performance Explorer Tips and Techniques



Performance Tools (PDF)

## 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リソース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。

Performance Explorer は、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、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 は、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・リソース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リソース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 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。

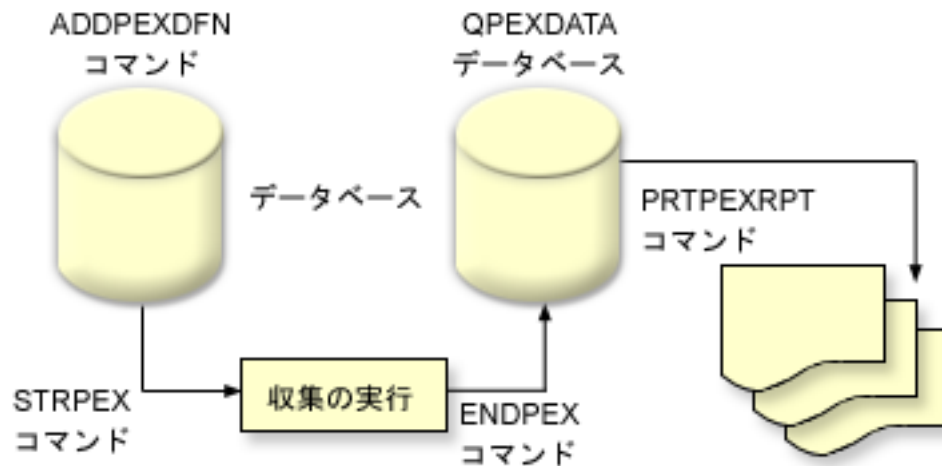
Performance Explorer が実行しているとき、それは収集に必要なファイルだけを作成します。

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、『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。

#### 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

#### 関連タスク

76 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。

#### **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。

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 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。

- i5/OS 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 プロシージ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の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・プロシージ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ジ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ジャー・レベルごとに CPU およびディスク・リソース使用量を示します。

```
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 使用率を過度に消費する高水準言語 (HLL) プログラム、モジュール、プロシージャー、およびステートメントを識別します。

- 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、プロシージャー、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、特定のプロシージャの使用量を示します。

```
ADDPEXDFN 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 (ADDPEXDFN 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。

```
ADDPEXDFN DFN(ALLJOBPROF) /* The name of the definition. */
TYPE(*PROFILE) /* The type of definition */
PRFTYPE(*JOB) /* A job profile type will be monitored. */
JOB(*ALL) /*All Jobs */
TASKS(*ALL) /*All tasks */
INTERVAL(1) /* 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、i5/OS 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 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。

```

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

```

この例は、HEAPEVENTS という名前の Performance Explorer トレース定義です。

```

ADDPEXDFN DFN(HEAPEVENTS) /* The name of the definition. */
TYPE(*TRACE) /* The type of definition */
JOB(*ALL) /*All Jobs */
TASK(*ALL) /*All tasks */
MAXSTG (100000) /*Maximum storage. Set to 100000 because the default of
10000 KB is often too small for the large number of heap events that can be
generated when tracing all jobs and all tasks.*/
TRCTYPE(*HEAP) /* Selects all heap events from the STGEVT
(storage events) parameter. */

```

### 関連概念

75 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。

### 関連タスク

76 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。

### 関連資料

Performance Explorer 定義の追加 (ADDPEXDFN) コマンド

### **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。

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 (PEX) 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
トレース・リソース・アフィニティー	QAYPEAFN
補助記憶域管理イベント・データ	QAYPEASM
補助記憶域プール (ASP) 情報データ	QAYPEASPI
基本イベント・データ	QAYPEBASE
基本構成情報	QAYPECFGFI
通信イベント・データ	QAYPECMN
ディスク・イベント・データ	QAYPEDASD

ファイルに含まれる情報のタイプ	ファイル名
ディスク・サーバー・イベント・データ	QAYPEDSRV
イベント・タイプとサブタイプのマッピング	QAYPEEVENT
ファイル・サービス・イベント・データ	QAYPEFILSV
構成済みフィルター情報	QAYPEFTRI
パフォーマンス測定カウンター (PMC) の選択	QAYPEFQCFG
ヒープ・イベント・データ	QAYPEHEAP
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
Performance Explorer Java イベント・データ	QAYPEJVA
Performance Explorer Java クラス情報データ	QAYPEJVCI
Performance Explorer Java メソッド情報データ	QAYPEJVM
Performance Explorer Java 名前情報データ	QAYPEJVNI
ライセンス内部コード (LIC) ブラケット化データ	QAYPELBRKT
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ー	QAYPELMI
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
構成済みタスク	QAYPELTASK
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) 複合命令のマッピング	QAYPEMICPX
マシン・インターフェース (MI) ポインターのアドレス	QAYPEMIPTR
マシン・インターフェース (MI) ユーザー・イベント・データ	QAYPEMIUSR
ポータブル・アプリケーション・ソリューション環境 (PASE) イベント・データ	QAYPEPASE
ページ不在イベント・データ	QAYPEPGFLT
プログラム・プロファイル・データ	QAYPEPPANE
ライセンス内部コード (LIC) アドレス解決マッピング	QAYPEPROCI
リソース管理プロセス・イベント・データ	QAYPERMPM
リソース管理占有ロック・イベント・データ	QAYPERMSL
参照情報	QAYPEREF
各種解決データ	QAYPERINF
データベース・レベル標識	QAYPERLS
一般情報	QAYPERUNI
セグメント・アドレス範囲 (SAR) データ	QAYPESAR
セグメント・アドレス解決マッピング	QAYPESEGI



ファイルに含まれる情報のタイプ	ファイル名
基本統計データ	QAYPESTATS
同期イベント・データ	QAYPESYNC
プロセスおよびタスク解決マッピング	QAYPETASKI
トレース・ジョブ同等イベント・データ	QAYPETBRKT
すべてのイベントについての共通トレース・データ	QAYPETIDX
トレース索引データ (時間とタスクごと)	QAYPETIDXL
トレース索引データ (時間ごと)	QAYPETID2L
タスク切り替えイベント・データ	QAYPETSWSW
ユーザー定義のブラケット化フック・データ	QAYPEUSRDF

### Performance Explorer データベース・ファイルのマイグレーション

Performance Explorer (PEX) データベース・ファイルは、新規イベントや新規データがファイルに追加されるために、リリースからリリースで変更されます。i5/OSの新規リリースにマイグレーションする際に、システムが非互換の PEX データベース・ファイルを発見すると、これらのファイルを QPEXDvrmxx ライブラリーへ移動します。ここで、vrm=バージョン。システムは、ファイルが移動されることを示す、状況メッセージを表示します。ファイルが移動した後、システムは移動が成功したか、または失敗したかを示す完了メッセージを表示します。移動が失敗した場合、システムは 非互換リポジトリ (Incompatible repository) メッセージを表示します。

#### 関連概念

『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。

#### 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。

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 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、『Performance Explorer の定義』を参照してください。それぞれの報告書は、資料 Performance Tools で詳細に説明されています。

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せん。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)

## 関連概念

70 ページの『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。

## 関連資料

73 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。

Print Performance Explorer Report (PRTPEXRPT) コマンド

## 関連情報



Performance Tools (PDF)

## 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。

## このタスクについて

Performance Explorer を構成するには、以下のステップを実行します。

### 手順

1. システムに、収集するパフォーマンス・データを通知するセッション定義を作成します。「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. (オプション) フィルターを追加します (PEX フィルターの追加 (ADDPEXFTR) コマンド)。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・データを識別します。特定のイベントに対して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. データ収集を開始します (Performance Explorer の開始 (STRPEX) コマンド)。\*PMCO イベントが収集されない場合、ジョブが複数の Performance Explorer 収集に入る可能性があります。\*PMCO イベントが収集される場合、すべての収集の間隔指定 (ADDPEXDFN INTERVAL() パラメーター) が同じであれば、ジョブが複数の収集に入る可能性があります。STRPEX コマンドで、定義およびオプション・フィルターを指定できます。
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. データの収集を停止し、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。

- パフォーマンス・データを分析します。Performance Explorer 報告書の印刷 (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。

以下は、その他の分析用オプションです。

- データベース・ファイルのセットに対して、独自の Query を作成する。
  - iDoctor for IBM i - PEX Analyzer を使用する。iDoctor for IBM i は、ソフトウェア・パフォーマンス分析ツールと関連サービスのセットです。詳細情報を収集してそのデータの自動グラフィカル分析を行うので、ご使用のシステムの正常性を幅広く評価できます。
7. 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。

## タスクの結果

Performance Explorer コマンドはすべて、以下のいずれ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。すべてのコマンドが IBM i 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。

## 関連概念

69 ページの『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リソース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。

70 ページの『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。

## 関連資料

PEX フィルターの追加 (ADDPEXFTR) コマンド

Performance Explorer の開始 (STRPEX) コマンド

Print Performance Explorer Report (PRTPEXRPT) コマンド

## *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。

Performance Explorer の終了 (ENDPEX) コマンドは、収集されたデータに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。既存のデータを新規データで上書きしないようにするには RPLDTA(\*NO) を指定し、既存のデータを新規データで上書きするには RPLDTA(\*YES) を指定できます。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 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。

## データの表示と分析

パフォーマンス・データ・コレクターによって収集されたデータは、データベース・ファイルに保管されます。IBM i には、このデータの管理、表示、分析に役立つさまざまなツールが用意されています。

### 関連情報



IBM i でのエンドツーエンド・パフォーマンス管理 (英語)

## IBM Systems Director Navigator for i Performance インターフェース

- | IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

### 関連情報

IBM Systems Director Navigator for i

- | **Performance Data Investigator:**

- | Performance Data Investigator は、パフォーマンス・データ上の Web ベースの GUI に、インタラクティブな図表と表を提供します。これを使用することにより、コレクター (収集サービス、IBM i Job Watcher、IBM i Disk Watcher、Performance Explorer) それぞれのパフォーマンス・データの表示と分析が可能になります。

- | **このタスクについて**

- | 以下のステップを実行することによりアクセスできます。

### 手順

- | 1. IBM Systems Director Navigator for i を起動します。
  - | a. Web ブラウザーから、次の URL にアクセスします。hostA は、ご使用の IBM i 区画名です。  
http://hostA:2001
  - | b. ご使用のシステム ID とパスワードでログインします。
- | 2. 左側パネルから「パフォーマンス」を選択します。
- | 3. 「データの調査」を選択します。

- | **データの調査:**

「データの調査」タスクを選択すると、強力なツールである Performance Data Investigator が起動します。このツールを使用すると、パフォーマンス収集に保管されたデータを、図表または表の形式で表示および分析できます。

「データの調査」メインページから、分析するパースペクティブと収集を選択することになります。各コレクター (収集サービス、Job Watcher、Disk Watcher、Performance Explorer) には、そのデータ収集タイプのパースペクティブが事前定義されたコンテンツ・パッケージ が関連付けられています。IBM 提供のヘルス標識用コンテンツ・パッケージもあります。「ヘルス標識」パッケージには区画の全般的な正常性を示すパースペクティブが含まれており、これにユーザー定義のしきい値を構成できます。ユーザーが作成してリストに保管したカスタムのコンテンツ・パッケージやパースペクティブを使用することも可能です。

次の例は、「データの調査」ページに表示されるコンテンツ・パッケージのリストの階層フォーマットを示しています。

データの調査

**指示**  
下で収集を選択し、「表示」をクリックして、パフォーマンス・データを表示します。  
[メッセージを閉じる](#)

**パースペクティブ** **選択**

- Performance Explorer
- Disk Watcher
- Job Watcher
- ヘルス標識
- 収集サービス
  - CPU 使用率および待機の概要**
  - スレッドまたはタスク別 CPU 使用率
  - リソース使用率の概要
  - ジョブ統計の概要
  - 待機
  - CPU
  - ディスク
  - 物理ディスク入出力
  - 同期ディスク入出力
  - ページ不在
  - 論理データベース入出力
  - 仮想入出力
  - 通信
  - 5250 ディスプレイ・トランザクション
  - 物理システム
  - Java
  - 収集サービス・データベース・ファイル

**名前**  
CPU 使用率および待機の概要

**記述**  
この図表は、選択された収集について、それに貢献しているすべてのジョブおよびタスクの一定期間における CPU 使用率および注目すべき各種の待機カテゴリーを示します。この図表を使用して、さらに詳しく調べるための時間フレームを選択します。

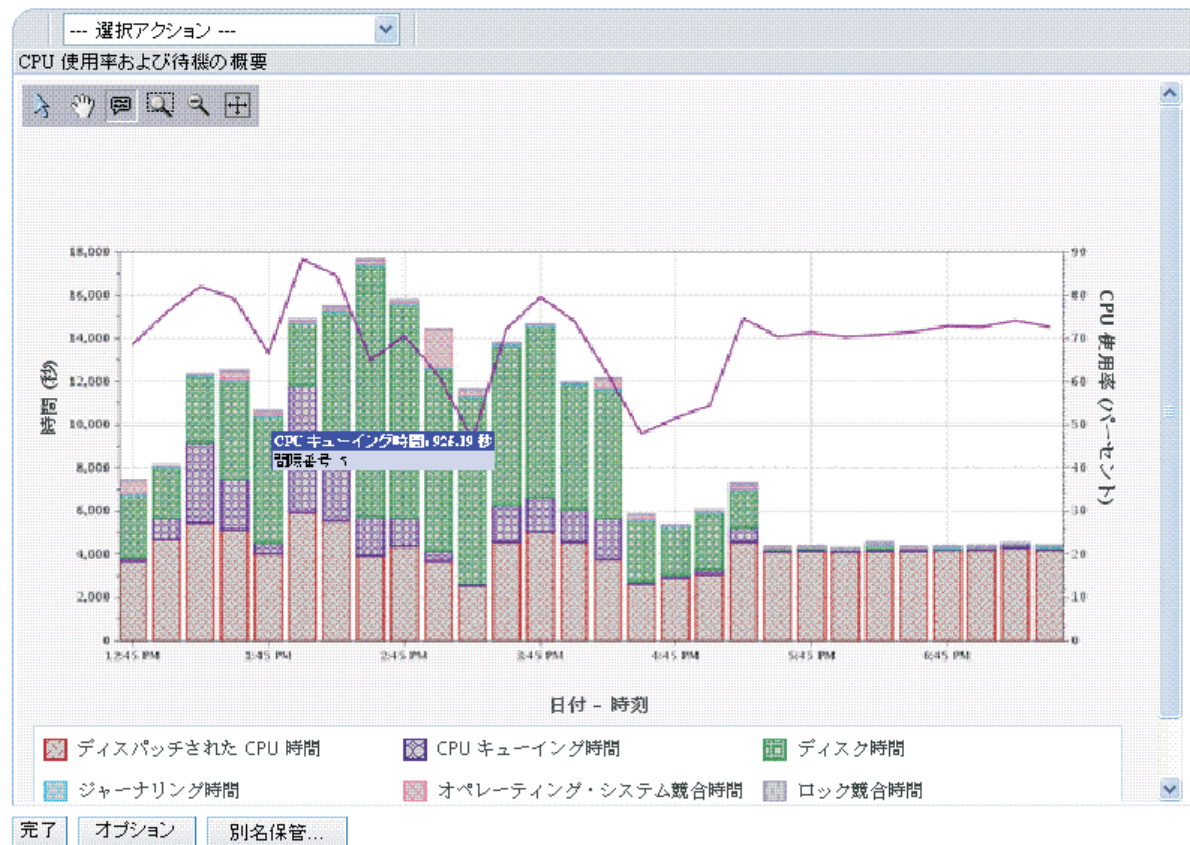
**収集**

収集ライブラリー: LIBRARY1    収集名: 最新

[表示](#)   [検索](#)   [オプション](#)   [閉じる](#)

- 各コンテンツ・パッケージの下には、データに対して異なる解釈 (レンダリング) をするためのパースペクティブのリストがあります。パースペクティブは、分析するデータのレンダリングに使用できる図表または表から成る 1 ページを定義したものです。
- 「データの調査」ページを使用して分析対象のパースペクティブと収集を選択するには、以下のステップを実行します。
1. 作業に使用するコンテンツ・パッケージを展開します。そのためには、横にある正方形をクリックするか、コンテンツ・パッケージの名前を直接選択します。
  2. パースペクティブは、コンテンツ・パッケージの中に階層的に保管されます。サブフォルダーにナビゲートするには、フォルダー名の横にある正方形をクリックします。
  3. 表示するパースペクティブが見つかったら、クリックして選択します。ページの反対側に、パースペクティブの要旨が表示されます。ページの下部には、収集を簡単に選択できる 2 つのオプション・リストが表示されます。
  4. 「**収集ライブラリー**」リストを使用して、作業に使用するライブラリーを選択します。「**収集名**」リストが更新され、選択したライブラリーに含まれる収集が表示されます。このリストには、選択したパースペクティブに対して有効な収集のみが含まれます。
  5. 「**収集名**」リストを使用して、作業に使用する収集を選択します。
  6. 「**表示**」をクリックすると、選択したパースペクティブにレンダリングされた収集データが表示されます。

収集	時間	システム
名前 (複数可): 52345UVXYZ	開始: 2010/01/21 0:00:08	名前: SYSTEM1
ライブラリー: LIBRARY1	終了: 進行中	リリース: V7R1M0
タイプ: 収集サービスのファイル・ベース収集		



IBM i 6.1 以降で作成された収集、または i 6.1 フォーマットに変換された収集のみが、Performance Data Investigator での分析に使用できます。変換には IBM Systems Director Navigator for i の Performance Convert Collection (パフォーマンス変換収集) タスクを使用するか、パフォーマンス収集の変換 (CVTFPRCOL) コマンドを使用します。

必要とする機能のレベルによっては、以下の一部またはすべてをインストールすることが必要な場合があります。

- Performance Tools for i (5761PT1) オプション 1 - マネージャー・フィーチャー - Performance Explorer コンテンツ・パッケージおよび IBM i Disk Watcher 機能に必要。
- Performance Tools for i (5761PT1) オプション 3 - IBM i Job Watcher 機能に必要。

## 関連概念

16 ページの『IBM i 待機アカウンティングの基礎』

待機アカウンティングは IBM i オペレーティング・システムに組み込まれている特許取得済みテクノロジーであり、スレッドまたはタスクが何もしていないように見えるときにそれが何をしているかをユーザーが分かるようにします。

64 ページの『すべての区画の CPU 使用率の収集と表示』

複数区画を使用する場合、区画で IBM i、AIX、または Linux が稼働しているかどうかにかかわらず、全区画にわたる処理能力の全体使用率を把握することが重要になることがあります。IBM i は、このようなデータを収集して表示するための手段を提供します。

## 関連タスク

95 ページの『収集の表示』

IBM Systems Director Navigator for i を使用して収集を表示するには、以下のいずれかの一連のステップを実行します。

## パースペクティブでの作業:

パースペクティブとは、指定した収集データのレンダリングのことです (図表または表のどちらかの形式)。収集データが表示されると、データを対話式に扱ったり分析したりするためのさまざまな機能を使用できるようになります。

## このタスクについて

- 最も強力な機能の 1 つに、収集したデータをドリルダウンできることがあります。現行パースペクティブ・ビューのアクション・リスト・メニューから 2 番目のパースペクティブを選択することにより、データの別のビューにドリルダウンできます。すべての図表はデータについて、それぞれ異なる洞察を提供できますが、Performance Data Investigator ツールの強みは、それぞれのパースペクティブが後続のパースペクティブに影響を与えることができるという点にあります。ドリルダウンでアクセスするパースペクティブごとに、カスタム・ビュー・コンテキストを絞り込むことができます。図表のポイントまたは表の行を選択することにより、その後のビューにおける有効範囲を絞り込むために使用するデータ・セットを示します。例えば、「CPU 使用率および待機の概要」パースペクティブを調べているときに、パフォーマンス上の問題の可能性があると気付いたとします。特定の日時範囲にズームインすることによって、対象の有効範囲を絞り込むことができます。次に、アクション・リスト・メニューから「スレッドまたはタスク別 CPU 使用率」などの別のパースペクティブを選択することで、追加の関連情報を取得できます。データ・ポイントを選択してから別のパースペクティブにドリルダウンするというプロセスは、反復可能です。パースペクティブ上部の「履歴」メニューは、表示したパースペクティブに戻る手段としてだけでなく、どのパースペクティブを表示したか管理するときにも使用できます。

- データの表示方法を変更できることから、変更したパースペクティブを今後参照できるように「保管」アクションが用意されています。

- 「オプション」をクリックすることにより、Performance Data Investigator の内部で使用される永続的なユーザー設定を指定できます。これらのオプションの 1 つである「設計モードを使用可能にする」を選択すると、独自のコンテンツ・パッケージやパースペクティブを作成できるようになります。

- その他にも、Performance Data Investigator を柔軟で使いやすいものにするために、データをエクスポートできたり、パースペクティブの生成に使用される SQL を変更できたりなど、図表と表に関する対話式のさまざまな機能とアクションが用意されています。

## パースペクティブの保管:



「パースペクティブの保管 (Save Perspective)」機能を使用することによって、変更したパースペクティブを後で使用できるように保管できます。

#### このタスクについて

収集に含まれるデータをドリルダウンとコンテキスト変更を行って調べていくと、現行コンテキストのパースペクティブが変更されていきます。このような変更されたパースペクティブを保管すると、後でそのパースペクティブに戻り、作成済みのパースペクティブの特性に合わせて、(適切なタイプの) 収集データを素早くレンダリングできるようになります。

カスタマイズされた表または図表から保管アクションを実行するには、下部にある「別名保管」ボタンをクリックするか、または上部の「パースペクティブ」メニューにある「別名保管」アクションを使用します。「パースペクティブの保管 (Save perspective)」ページで、この新規パースペクティブを識別するための名前と説明を指定できます。保管が正常に完了すると、再びパースペクティブ・ページが追加メッセージと共に表示され、保管が正常に行われたことと、今後その同じパースペクティブに直接戻ることができる URL が示されます。

以下は、この機能の重要なポイントです。

- 保管の結果として戻される URL は、同一システム上の IBM Systems Director Navigator for i を使用している他のユーザーと共有できます。

- 保管されたパースペクティブは、Performance Data Investigator の「データの調査」ページから利用できます。保管されたパースペクティブは、「データの調査」メイン・ページの「カスタム・パースペクティブ - ユーザー名」コンテンツ・パッケージの下に表示されます。

- コンテンツ・パッケージは、「QIBM/UserData/OS400/iSeriesNavigator/config/PML/CCP」の IFS 内の「CCP\_USERNAME.PML」という名前のファイルに保管されます。システム上の保管されたパースペクティブを保護する場合は、コンテンツ・パッケージのバックアップを取る必要があります。

- ライブラリーと名前のドロップダウン・ボックスから収集を選択するだけで、保管されたパースペクティブを使用して異なるパフォーマンス収集を分析できます。新規収集には、正しくレンダリングされるようにコンテキスト情報を適用する必要があります。空の図表が表示される場合は、「コンテキストの変更」アクションを使用して、パースペクティブのコンテキストを検査および変更してください。

#### 関連タスク

89 ページの『コンテキストの変更』

「コンテキストの変更」アクションを使用することにより、図表ビューまたは表ビューの現行コンテキスト情報の検査と変更を行えます。

#### オプション:

「オプション」ページから、Performance Data Investigator 固有の永続的なユーザー設定を構成できます。

#### このタスクについて

このページで設定できるオプションは、以下のとおりです。

- パターンの使用 - パターンを図表の中の適用可能部分で使用するかどうかを指定します。デフォルトで選択されています。

- 図表の表示 - 可能であれば表ではなく常に図表が表示されるようにデフォルトを設定します。デフォルトで選択されています。

- 設計モードを使用可能にする - 新規コンテンツ・パッケージの設計と開発が可能になる拡張機能を有効にします。デフォルトでは選択されていません。

- ヘルプの表示 - さまざまな操作のヘルプ・メッセージを有効にします。デフォルトで選択されています。
- 表サイズの設定 - 表に表示される行と列の数を指定できます。

デフォルトのライブラリーを指定することによって、収集を選択するときに使用されるライブラリーを特定できます。以下のいずれかに設定できます。

- 収集サービスによって構成されたライブラリー
- 最後にアクセスしたライブラリー
- 指定したライブラリー

#### 設計モードの使用可能化:

設計モードを使用可能にするための設定は、「オプション」ページにあります。設計モードを使用可能にすると、新規コンテンツ・パッケージのカスタム開発が可能になります。

#### このタスクについて

設計モードを選択することによって、追加の拡張機能が使用可能になります。これには、以下の機能が含まれます。

- 新規のフォルダーやパースペクティブを作成する。
- フォルダーやパースペクティブを編集する。
- パースペクティブのビューを追加、編集、削除する。
- カスタムのフォルダーやパースペクティブを削除する。
- 図表ビューに使用されるデータ・シリーズを追加、編集、削除する。
- 図表ビューに使用されるしきい値を追加、編集、削除する。
- 「拡張編集」オプションを使用して、パースペクティブの生成に使用される XML を編集する。
- カスタム・パースペクティブを今後編集できないように「ロック」する。
- 「上に移動」または「下に移動」を使用して、フォルダー内のパースペクティブの配列を並べ替える。
- パースペクティブ・リストをリフレッシュする。

#### 関連タスク

91 ページの『カスタム・パッケージの作成』

設計モードが使用可能になっていると、独自のカスタム・コンテンツ・パッケージを作成できます。これらのパッケージに、ユーザーのニーズに合わせたパースペクティブを含めることができます。

#### パースペクティブの最新表示:

「パースペクティブの最新表示」は、システム上のすべてのコンテンツ・パッケージを再ロードする場合に使用します。

#### このタスクについて

Performance Data Investigator の「データの調査」ページの下部に、「パースペクティブの最新表示」ボタンがあります。このボタンをクリックすると、システム上のすべてのコンテンツ・パッケージが強制的に再ロードされ、その結果としてビューが更新されます。この機能は、システム上に新規コンテンツ・パッケージを手動で作成する場合に役立ちます。


「パースペクティブの最新表示」は、設計モードが有効になっている場合のみ使用できます。


## 1 図表での機能:


1 図表ビューでは、データをさらに詳しく調べたり図表ビューのコンテキストを変更したりするためのさまざまな機能を使用できます。


### 1 このタスクについて


1 図表のインタラクティブな機能が、図表をパフォーマンス・データを分析するための強力なツールにしています。グラフ化されたパフォーマンス・データを視覚化できるので、活動の山と谷が明瞭になります。より詳細な項目と特定の時間間隔にドリルダウンできる機能は、非常に便利です。以下の図表アイコンは、分析を補助するためのインタラクターです。


1 •  矢印アイコンをクリックすると、図表上のポイントまたはバーを選択できるようになります。データ・ポイントを選択すると、関心の対象となるコンテキストに基づいて今後の分析を絞り込むのを、Performance Data Investigator が支援できるようになります。図表上のポイントを選択し、次にアクション・リスト・メニューからアクションを選択することにより、データをさらに分析するためにドリルダウンできます。これはデフォルトのインタラクターです。

1 •  手のひらアイコンをクリックすると、図表イメージの任意の部分をクリックしてドラッグすることによって図表をパンできるようになります。このインタラクターは、図表がその一部分にズームインされている場合に、倍率を調整せずにさまざまな部分に移動したいときに便利です。

1 •  吹き出しアイコンをクリックすると、ツールチップ情報が有効になったり無効になったりします。ツールチップは、図表の任意のデータ・ポイントにカーソルを移動して停止させると表示されます。ツールチップによっていくつかの関連情報が表示されるように定義できます。デフォルトでは無効になっています。

1 •  虫眼鏡 (後ろに点線の長方形がある) のように見えるアイコンをクリックすると、ズームイン・アクション・インタラクターが有効になります。ズームインは、ポイントをクリックしてウィンドウをドラッグすることによって行えます。そのウィンドウの範囲が、図表の中のさらに詳しく調べたい部分を意味します。マウス・ボタンを放すと、パースペクティブ図表が再レンダリングされ、選択した領域のみが表示されます。

1 •  負符号 (「-」) が中にある虫眼鏡アイコンをクリックするたびに現行ズーム・レベルからインクリメント方式でズームアウトされていき、最後はパースペクティブの全体図表ビューに戻ります。

1 •  矢印が四方を指している正方形アイコンをクリックすると、図表のズームが最大レベルにリセットされます。その結果、パースペクティブの全体図表ビューが表示されます。

1 同時に活動状態にできる図表インタラクターは 1 つだけです。

1 さらに、図表アクション・リスト・メニューに「表として表示」アクションが追加されました。このアクションを選択すると、現行ビューが変更されて表としてレンダリングされます。

## 1 表での機能:

1 表ビューでは、データをさらに詳しく調べるためのさまざまな機能を使用できます。

## このタスクについて

表のすべての行を選択してそれをデータの関心部分とする機能が、表に備わっています。また、特定の情報を対象とした表のフィルタリング、ソート、または検索を簡単に行えます。元の図表に切り替えると、表に対して行った配列やフィルタリングの変更がすべて反映されます。

表固有のさまざまなアクションを使用できます。以下のアクションは、表アイコン機能をクリックするか、またはアクション・リスト・メニューから特定のアクションを選択することによって使用できます。

- すべて選択 - 「選択」列のすべての行のすべてのチェック・ボックスを選択します。選択された行が、アクション・リスト・メニュー上のアクションに基づく Performance Data Investigator による操作の対象になります。

- すべて選択解除 - 「選択」列のすべてのチェック・マークをクリアします。

- 行フィルターの表示/非表示切り替え (Filter Row Show or Hide Toggle) - 行フィルターはデフォルトで非表示です。行フィルターを表示してデータをフィルターに掛けられるようにするには、このアイコンを選択します。特定のパラメーター (列値) を使用して、表示されるデータを絞り込めるようになります。

- フィルターをすべてクリア (Clear All Filters) - どのカスタム・フィルターも除去します。

- ソート編集 (Edit Sort) - 最大 3 つの列の値に基づいて、表ビューに表示される列をソートできます。表ビューに対してソート編集を行う場合は、このアイコンをクリックします。ソート編集照会が表示されます。ソート位置にする列見出しを最大 3 つ (1 次、2 次、3 次) まで選択します。次に、ソートごとに昇順または降順を選択します。その優先度でソートされたデータが表示されます。

- ソートをすべてクリア (Clear All Sorts) - 有効になっているカスタム・ソートをすべて除去します。

以下のアクションは、表のアクション・リスト・メニューでのみ使用できます。

- 図表として表示 - このアクションを選択すると、現行ビューが変更されて図表としてレンダリングされます。

- このアクションは、「オプション」ページの「図表の表示」が選択されていない場合は使用できません。

- データ・シリーズが定義されていない表を表示しているときに「図表として表示」アクションを選択しようとする場合は、データ・シリーズを定義してから続行する必要があります。

- 列 - 表ビューの列を追加または除去します。列見出しを上下に移動することにより、列の順序を再配置することもできます。

- 検索ツールバーの表示 (Show find toolbar) - 表ビュー内で検索を行えるようになります。

- デフォルトの復元 - 表をデフォルトのソートとフィルタリングの状態に復元します。

以下のアクションは、表に表示されるアイコンをクリックすることによって使用できます。

- 選択 - 行の横にあるチェック・ボックスをクリックすることによって、表の行を選択できます。行を選択すると、関心の対象となるコンテキストに基づいて今後の分析を絞り込むのを、Performance Data Investigator が支援できるようになります。アクション・リスト・メニューから選択したアクションは、選択した行に対して実行されます。

- ソート (Sorting) - 列見出しにあるソート標識 (曲折アクセント記号「^」) をクリックすることによって、表を昇順または降順にソートできます。「ソート編集 (Edit Sort)」または「ソートをすべてクリア (Clear All Sorts)」アイコンを使用して、現行ソート基準を操作することもできます。

- フィルタリング - 「行フィルターの表示 (Show Filter Row)」表アイコンをクリックすることにより、表の列見出しの下に行フィルターが表示されます。特定のパラメーター (列値) を使用して、表示されるデ

データを絞り込めるようになります。行フィルターはデフォルトで非表示です。レンダリングされるデータを列の条件に基づいてフィルターに掛けるには、その列の「フィルター」リンクをクリックします。

#### 表と図表でのアクション:

表ビューまたは図表ビューから、さまざまな便利なアクションを実行できます。

#### エクスポート:

ビューをエクスポートする機能を使用すると、図表または表を後の参照用にファイル・ロケーションにエクスポートできます。イメージ (図表のみ) ファイル、コンマ区切りファイル、またはタブ区切りファイルにデータをエクスポートできます。

#### このタスクについて

図表ビューまたは表ビューから「エクスポート」アクションを選択すると、「エクスポート」ページが表示されます。このページから、保管するパースペクティブ・ビューに使用される以下のフィールドを検査および変更できます。

- 表題 - 保管されるファイルの最上部で使用されるタイトルです。

- フォーマット - パースペクティブ・ビューの保管フォーマットを選択します。

表の場合の選択肢は、コンマ区切り (\*.csv) またはタブ区切り (\*.txt) ファイル・フォーマットです。

図表の場合は、イメージ (\*.png)、コンマ区切り (\*.csv)、またはタブ区切り (\*.txt) フォーマットの中から選択できます。

- データ範囲 - エクスポートされるビューに含まれるデータの範囲を変更できます。以下の中から選択できます。

- すべてのデータ - 現行ビューで使用できるデータのすべてをエクスポートします。

- 表示されるデータ - ビューの現行表示データのみエクスポートします。

- ユーザー定義範囲 - このオプションを選択すると、先頭レコード番号と最終レコード番号を指定できます。レコード番号は、そのデータ・シリーズのすべてのデータのうちの 1 つのデータ要素の索引を表します。これにより、指定した範囲がエクスポートされます。

「エクスポート」ページで「OK」を選択すると、図表ビューまたは表ビューをダウンロードするための別のブラウザ・ウィンドウが開く場合があります。ブラウザのメッセージ・バーに応答してファイルをダウンロードできるようにすることが必要な場合もあります。ファイル・ダウンロードのウィンドウで、データを開くかご使用のクライアントに保管するかを選択できます。

#### SQL の変更:

「SQL の変更」アクションを使用すると、現行コンテキストのパフォーマンス収集からデータを検索するために使用される SQL ステートメントの検査と変更を行えます。

#### このタスクについて

パースペクティブ・ビューは、定義された SQL ステートメントを通じ、選択された収集データを使用してレンダリングされます。「SQL の変更」機能は、データの検索に使用される照会を変更できるようにします。SQL ステートメントを変更する場合は、その前に SQL 開発を経験するようお勧めします。この機能を使用すると、ビューに表示される複雑なメトリックを Performance Data Investigator がどのデータベース・フィールドを使用して計算しているかを理解できます。

選択されたパフォーマンス収集をターゲットにするために必要な特定のデータベース・メンバーを、照会は SQL 別名を使用して表すことを知っておいてください。別名は QTEMP で作成され、ライブラリー名、ファイル名、メンバー名を連結して単一の別名にすることによって命名されます。これらの照会を Performance Data Investigator 外で実行する場合は、独自の対話式セッションで別名を再作成する必要があります。

SQL ステートメントを変更し終えてパースペクティブが再び表示されたら、「別名保管」アクションを使用して、更新したパースペクティブを後で使用できるようにカスタム・コンテンツ・パッケージに保管できます。

「リセット」ボタンも「SQL の変更」パネルの特徴です。このボタンをクリックすると、SQL ステートメントはパネルがロードされた時点の内容にリセットされます。「収集選択の許可」チェック・ボックスもあります。チェック・マークが付いていると (デフォルトの動作)、現在選択されている収集だけでなく、どの収集に対しても照会を実行できます。

#### 関連タスク

92 ページの『パースペクティブの作成』

設計モードを使用可能にすると、「データの調査」メインページと「カスタム・パースペクティブの保管」パネルに、「新規パースペクティブ」ボタンが表示されます。ロックされていない選択したコンテンツ・パッケージまたはパースペクティブ・グループに、新しいカスタム・パースペクティブを作成できます。

89 ページの『ビューの作成および編集』

設計モードが使用可能になっていると、表または図表が表示されているときに「ビューの編集」アクションを使用できます。ロックされていないパースペクティブに対する新規ビューの作成、編集、および追加も可能です。

#### 次期アップグレードの見積もり:

現在のパフォーマンス統計に基づいた今後のシステムの見積もりで使用される Workload Estimator に現行セッションからデータを送信するには、「次期アップグレードの見積もり」アクションを使用します。

#### このタスクについて

「次期アップグレードの見積もり」アクションは、収集サービスのファイル収集を調査しているときに、ビュー・パースペクティブ内で使用できます。選択すると、IBM Systems Workload Estimator (WLE) に誘導されます。WLE に送信されるデータは、現行コンテキストに基づいて現行収集から取り出されます。

「次期アップグレードの見積もり」アクションを選択すると、以下の情報が表示されます。

- フィルター - WLE に送信されるデータをサブセット化するために「日付/時刻」によるフィルタリングが使用されているかどうかを示します。
- システム名、オペレーティング・システム、バージョン、ワークロード ID などの一般情報。
- 「使用率」や「対話式使用率」などの CPU メトリック。
- ディスク・ストレージ容量。

「OK」ボタンをクリックするとデータが Workload Estimator (WLE) Web アプリケーションに送信され、新しいウィンドウに IBM Systems Workload Estimator Web アプリケーションが開きます。

増大傾向履歴データや季節ピーク・データに基づいてアップグレードを見積もる場合は、PM for Power Systems を使用することをお勧めします。PM for Power Systems を使用すると、複数の区画からの見積もりも容易になります。

## | 関連概念

- | 119 ページの『IBM Performance Management for Power Systems - IBM i のサポート』
- | IBM i オファリングをサポートする IBM Performance Management for Power Systems (PM for Power Systems) 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・リソース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。

## | コンテキストの変更:

- | 「コンテキストの変更」アクションを使用することにより、図表ビューまたは表ビューの現行コンテキスト情報の検査と変更を行えます。

## | このタスクについて

- | このパネルに表示される情報は、現行の表または図表に影響を与えている前のパースペクティブからマイニングされたコンテキストを表しています。値を変更すると、現行の表または図表、および同じデータをマイニングする今後のパースペクティブに影響を及ぼします。ただし、以前のパースペクティブには影響しません。現行パースペクティブを終了すると、「コンテキストの変更」を使用して行った変更は失われ、値はマイニングされた前の値にリセットされます。分析とドリルダウンを続行する場合は通常、コンテキストを直接変更するよりも、現行パースペクティブを閉じて新しいデータ・ポイントを選択した方がよいでしょう。

## | 関連タスク

- | 82 ページの『パースペクティブの保管』
- | 「パースペクティブの保管 (Save Perspective)」機能を使用することによって、変更したパースペクティブを後で使用できるように保管できます。

## | ビューの作成および編集:

- | 設計モードが使用可能になっていると、表または図表が表示されているときに「ビューの編集」アクションを使用できます。ロックされていないパースペクティブに対する新規ビューの作成、編集、および追加も可能です。

## | このタスクについて

- | ビューは単一の図表または表です。パースペクティブは、1 つ以上 (最大 10 まで) のビューで構成されます。「データの調査」メイン・ページかまたは「カスタム・パースペクティブの保管」パネルで、ロックされていないパースペクティブを選択してから「編集」ボタンを選択することにより、新規ビューを作成できます。「パースペクティブの編集」ページの「ビュー」セクションにある「追加」ボタンをクリックします。「ビューの追加」パネルが開きます。このパネルから、以下を指定できます。

### | ビュー

- | • 名前
- | • ビューのタイプ (表または図表のどちらか)

### | データ・セット

- | データ・セットはビューを生成するために使用され、SQL 照会によって定義されます。「SQL の変更」をクリックすると、別のパネルが開きます。このパネルで、ビュー用のデータを生成する SQL ステートメントを変更できます。

### | ドリルダウン

「ドリルダウン」セクションには、ビューのドリルダウン・オプションとして追加 (または除去) できるパースペクティブのリストが表示されます。ドリルダウン・パースペクティブはビューのアクション・メニューに表示され、同じコンテキストから他のパースペクティブを調べるために使用できます。

#### 図表のプロパティー

- 軸の転置 - ビューが図表である場合、ドメインと範囲の軸を変えるには、「軸の転置」チェック・ボックスを選択します (ドメインが、下部ではなく左側に表示されるようになります)。
- データ・シリーズ - 図表のレンダリングが期待どおりにならないこともあります。これは、表の中の使用可能なデータが図表に含まれていない場合に起こる可能性があります。あるいは、図表シリーズが定義されていないことも考えられます。以下の操作を行えます。
  - データ・シリーズの追加 - 現行の図表ビューに追加する新しいデータ・シリーズを作成できます。
  - データ・シリーズの編集 - 図表ビューの既存のデータ・シリーズを表示、変更できます。データ・シリーズの属性を変更できます。変更が図表ビューに反映されるのは、「ビューの編集」ページから図表ビューに戻った時点です。
  - 削除 - 図表ビューからデータ・シリーズを削除します。
  - 上に移動 - 選択したデータ・シリーズをビューのリストの上方へ移動します。
  - 下に移動 - 選択したデータ・シリーズをビューのリストの下方へ移動します。
- しきい値 - ビューで使用されるしきい値を定義する場合に、このオプションを使用します。ビューのしきい値の追加、編集、または削除を、対応するボタンを使用して行えます。

#### 関連タスク

92 ページの『パースペクティブの作成』

設計モードを使用可能にすると、「データの調査」メインページと「カスタム・パースペクティブの保管」パネルに、「新規パースペクティブ」ボタンが表示されます。ロックされていない選択したコンテンツ・パッケージまたはパースペクティブ・グループに、新しいカスタム・パースペクティブを作成できます。

87 ページの『SQL の変更』

「SQL の変更」アクションを使用すると、現行コンテキストのパフォーマンス収集からデータを検索するために使用される SQL ステートメントの検査と変更を行えます。

93 ページの『データ・シリーズ』

「データ・シリーズ」パネルでは、図表ビューに使用されるデータ・シリーズを表示、変更したり、図表に使用される新しいデータ・シリーズを定義したりできます。

94 ページの『しきい値』

しきい値は図表をざっと見るための手段であり、表示されている値がガイドラインの範囲に収まっているかどうかや、処置をとる必要があるかどうかを示す視覚標識として表示されます。

#### パースペクティブの検索: メトリック・ファインダー:

特定のメトリックを含んだパースペクティブの検索を支援するメトリック・ファインダー機能を表示するには、「データの調査」ページの「検索」ボタンを使用します。

#### このタスクについて

「データの調査」ページのメトリック・ファインダーのフォーマットは、ユーザーが特定のメトリック (つまり特定の情報タイプ) への関心に基づいたパースペクティブを素早く表示できるように設計されています。定義済みのサポートされるメトリックの広範囲のリストからメトリックを選択することにより、その領域の情報に基づいたビューを提供するパースペクティブのリストを表示できます。



「データの調査」ページのメトリック・ファインダーのフォーマットには以下のセクションがあります。これらは「データの調査」ページの階層フォーマットとは異なります。

- **メトリック** - このメトリック検索リストから、特定の情報タイプが含まれるパースペクティブのリストを表示できます。
  - **フィルター** - 使用できるメトリックは多数あります。「メトリック」セクションのフィルター・フィールドを使用すると、フィルターに入力したテキストとメトリックの名前を突き合わせることによって、「メトリック」リストに表示されるメトリックのリストを制限できます。このフィルターは、ある特定のメトリックをその正確な名前を知らずに検索する場合に役立ちます。
  - **フィルター・テキストの入力後**、「フィルターの適用」をクリックしてメトリック名リストを更新する必要があります。
  - **メトリック・リストがフィルターに掛けられると**、フィルターと (部分的にまたは全体が) 一致するメトリックのみがドロップダウンに含まれるようになります。
  - **フィルター処理なしで表示されるメトリックのリストは**、パースペクティブの階層フォーマットから使用できるすべてのパースペクティブにおけるメトリックそのものです。言い換えれば、該当するメトリックを選択することによって、すべての最上位パースペクティブに到達できることとなります (一致するパースペクティブがパースペクティブの階層フォーマットで検出された場合)。これには、ユーザーが定義したカスタム・パッケージのパースペクティブが含まれます。ドリルダウンでのみアクセスできるパースペクティブは、パースペクティブのリストに含まれません。
- **パースペクティブ** - パースペクティブ・リストが表として示されます。この表には、上記のメトリック・セクションで選択したメトリックに対応するパースペクティブが 1 つ以上含まれます。ここで、収集データをレンダリングするために使用するパースペクティブを選択できます。
  - 各パースペクティブのロケーションは、図表項目に次のフォーマットで示されます。

コンテンツ・パッケージ → フォルダー [-> サブフォルダー] → パースペクティブ名

**注:** パースペクティブにメトリックが含まれていても、データ収集の図表ビューにそのメトリックがデフォルトで組み込まれることを意味しているわけではありません。場合によっては、パースペクティブが表示されたら「表として表示」アクションを強制的に実行して、メトリックを見つける必要があります。

### カスタム・パッケージの作成:

設計モードが使用可能になっていると、独自のカスタム・コンテンツ・パッケージを作成できます。これらのパッケージに、ユーザーのニーズに合わせたパースペクティブを含めることができます。

### このタスクについて

カスタムの図表および表を作成できるインタラクティブな機能は、IBM 提供のコンテンツ・パッケージとは異なる角度からパフォーマンス・データを見る必要のある上級者にとって、大きな助けになります。コンテンツ・パッケージには通常、何らかの共通の目的を持つ個別に定義されたパースペクティブのセットが含まれます。新規のコンテンツ・パッケージ・フォルダーを作成するには、「データの調査」メイン・パネル上部、または「カスタム・パースペクティブの保管」パネルのパースペクティブ・リスト上部の、「新規パッケージ ...」アイコンをクリックします。新しいパッケージの名前と説明を指定できる別のパネルが起動します。

パッケージが作成されると、「編集」ボタンを使用してパッケージ情報を編集できるようになります。編集モード中は、パッケージのデフォルトのパースペクティブを選択することもできます (パッケージ内にパースペクティブが定義されている場合)。また、パッケージ (およびその子孫すべて) をロックして編集できな

いようにする場合には、「ロック」チェック・ボックスを選択できます。パッケージを作成した後、ロックしないでおいた場合は、パッケージ情報を「編集」ボタンを使用して編集したり、「削除」ボタンを使用して削除したりできます。

作成したコンテンツ・パッケージはすべて、パースペクティブ階層メイン・リストに表示されます。パッケージを作成したら、そのパッケージに新しいパースペクティブ・グループやパースペクティブを追加できます。

#### 関連タスク

84 ページの『設計モードの使用可能化』

設計モードを使用可能にするための設定は、「オプション」ページにあります。設計モードを使用可能にすると、新規コンテンツ・パッケージのカスタム開発が可能になります。

#### フォルダーの作成:

設計モードを使用可能にすると、「データの調査」メインページと「カスタム・パースペクティブの保管」パネルに、「新規フォルダー」ボタンが表示されます。これにより、作成したどのパースペクティブも論理グループに編成できるようになります。

#### このタスクについて

パースペクティブ・グループを作成するには、ロックされていないコンテンツ・パッケージまたはパースペクティブ・グループのどちらかをツリーで選択しておく必要があります。「新規フォルダー」ボタンをクリックすると、新しいフォルダーの名前と説明を入力できるパネルが起動します。「新規フォルダー」アクションを実行するときにコンテンツ・パッケージが選択されていた場合は、選択したそのコンテンツ・パッケージの下にパースペクティブ・グループが作成されます。パースペクティブ・グループを選択しておいて「新規フォルダー」アクションを実行すると、選択したそのパースペクティブ・グループの下に新しいパースペクティブ・グループがネストされます。

パースペクティブ・グループが作成されると、「編集」ボタンを使用してフォルダー情報を編集できるようになります。編集モード中は、パースペクティブ・グループのデフォルトのパースペクティブを選択することもできます (パッケージ内にパースペクティブが定義されている場合)。また、フォルダー (およびその子孫すべて) をロックして編集できないようにする場合には、「ロック」チェック・ボックスを選択できます。フォルダーを作成した後、ロックしないでおいた場合は、パースペクティブ・グループ情報を「編集」ボタンを使用して編集したり、「削除」ボタンを使用して削除したりできます。

作成したパースペクティブ・グループはすべて、関連付けられたコンテンツ・パッケージの下のパースペクティブ階層メイン・リストに表示されます。パースペクティブ・グループを作成したら、そのパースペクティブ・グループに新しいパースペクティブ・グループやパースペクティブを追加できます。「上に移動」または「下に移動」ボタンを使用することにより、コンテンツ・パッケージ内のパースペクティブ・グループの配列を並べ替えることもできます。

#### パースペクティブの作成:

設計モードを使用可能にすると、「データの調査」メインページと「カスタム・パースペクティブの保管」パネルに、「新規パースペクティブ」ボタンが表示されます。ロックされていない選択したコンテンツ・パッケージまたはパースペクティブ・グループに、新しいカスタム・パースペクティブを作成できます。

#### このタスクについて

パースペクティブはデータがレンダリングされた 1 つのパネルであり、通常は 1 つ以上の図表または表の形式をとります。パースペクティブには 1 つ以上 (最大 10 まで) のビューが含まれます。ビューは単一

の図表または表です。パースペクティブを作成するには、ロックされていないコンテンツ・パッケージまたはパースペクティブ・グループのどちらかをパースペクティブ・ツリーで選択しておく必要があります。その後で「新規パースペクティブ」ボタンを選択します。パースペクティブの名前と説明を指定できる別のパネルが起動します。パースペクティブをロックして編集できないようにする場合には、「ロック」チェック・ボックスを選択できます。このパネルで、パースペクティブにビューを追加することも指定できます。

パースペクティブを作成してロックしないでおいた場合は、「編集」ボタンを使用してパースペクティブ情報を編集できます。編集モード中は、名前または説明の変更、あるいはロック状況の変更（アンロックからロックへ）が可能です。パースペクティブに関連付けられたビューを追加、編集、または削除することもできます。ロックされていないパースペクティブを、「削除」ボタンを使用して削除することもできます。

Performance Data Investigator は、パースペクティブがどのように定義されているか（例えばデータのマイニングとレンダリングの方法）を、XML ファイルを使用して保管します。このファイルは「パフォーマンス・マークアップ言語」（PML）と呼ばれます。「データの調査」メイン・ページと「カスタム・パースペクティブの保管」パネルにある「拡張編集」ボタンをクリックすることにより、ロックされていないパースペクティブの PML を直接表示したり編集したりできます。

作成したパースペクティブはすべて、関連付けられたコンテンツ・パッケージまたはパースペクティブ・グループの下のパースペクティブ階層メイン・リストに表示されます。

#### 関連タスク

87 ページの『SQL の変更』

「SQL の変更」アクションを使用すると、現行コンテキストのパフォーマンス収集からデータを検索するために使用される SQL ステートメントの検査と変更を行えます。

89 ページの『ビューの作成および編集』

設計モードが使用可能になっていると、表または図表が表示されているときに「ビューの編集」アクションを使用できます。ロックされていないパースペクティブに対する新規ビューの作成、編集、および追加も可能です。

#### データ・シリーズ:

「データ・シリーズ」パネルでは、図表ビューに使用されるデータ・シリーズを表示、変更したり、図表に使用される新しいデータ・シリーズを定義したりできます。

#### このタスクについて

「データ・シリーズ」にある属性は、以下のとおりです。図表の属性が既に指定されている場合、一部の属性はロックされていることがあります。

- ドメイン - この図表用の独立軸に使用されるフィールドを指定します。現行の図表のための別のデータ・シリーズが存在する場合は、既存のドメイン値に合わせるために、このフィールドはロックされます。
- 範囲 - この図表に使用される範囲値を指定できます。使用できるすべての範囲が「使用可能」リストに表示されます。「追加」ボタンをクリックすると、「使用可能」リストで選択した範囲がデータ・シリーズに追加されます。「除去」ボタンをクリックすると、「選択済み」テーブルで選択した範囲がデータ・シリーズから除去されます。「選択済み」テーブルのドロップダウン・メニューを使用して、各範囲の色、背景、パターンを指定できます。
- タイプ - 図表タイプ（バリエーションのある線または棒）を選択します。

- 明細 - 選択した明細フィールドの互いに異なる値ごとに、その明細値に対応する範囲値のすべてを表す固有のデータ・シリーズが生成されます。このメカニズムによって、システム上の固有のジョブごとに個別のデータ・シリーズが時間の経過とともに作成されていきます。例えば、次のデータ・セットがあるとします。

表 1. データ・セットの例

間隔番号	ジョブ CPU	ジョブ名
1	10	Job A
1	15	Job B
2	20	Job A
2	30	Job B
3	30	Job A
3	45	Job B

「間隔番号」のドメインと「ジョブ CPU」の範囲を指定した場合、データ・ポイントとして (1, 10)、(1, 15)、(2, 20)、(2,30)、(3, 30)、(3, 45) を持つ線グラフになります。

「ジョブ名」の明細ディメンションを指定した場合は、シリーズ 1 = (1, 10)、(2, 20)、(3, 30) とシリーズ 2 = (1, 15)、(2, 30)、(3, 45) がデータ・ポイントである 2 つの線が生成されます。

これは分かりにくい機能ですが、強力です。少し試してみれば完全に理解できるでしょう。

- ツールチップのフィールド - このリストからフィールドを選択することにより、データ・シリーズ内の各ポイントのツールチップに、現行ドメインに対して選択したフィールドの値が含まれるようになります。

データ・シリーズ機能は、図表ビューにのみ使用できます。表ビューから図表ビューに切り替える際に、データ・シリーズを追加することが必要な場合があります。データ・シリーズが定義されていない表を表示しているときに「図表として表示」アクションを選択しようとする場合は、データ・シリーズを定義してから続行する必要があります。

## 関連タスク

89 ページの『ビューの作成および編集』

設計モードが使用可能になっていると、表または図表が表示されているときに「ビューの編集」アクションを使用できます。ロックされていないパースペクティブに対する新規ビューの作成、編集、および追加も可能です。

## しきい値:

しきい値は図表をざっと見るための手段であり、表示されている値がガイドラインの範囲に収まっているかどうかや、処置をとる必要があるかどうかを示す視覚標識として表示されます。

## このタスクについて

しきい値は 1 つの境界を表しており、これを越えたデータは新しい状態になったと見なされます。しきい値には永続性があり、ユーザー・プロファイルに従って保管されます。しきい値に関する以下の項目を指定できます。

- 名前 - このしきい値の表示名。
- フィールド - このしきい値が定義されるフィールドです。
- カラー - このしきい値を図表に描画するとき使用される色を指定します。

- | • 現行値 - ユーザーが現在指定しているしきい値を表します。この値は簡単に変更でき、アプリケーション・セッションを越えて存続します。今後「名前」と「フィールド」を同じにしたしきい値を定義した場合、そのしきい値にも同じ値が使用されます。現行値を「デフォルト値」にリセットするには、「デフォルト値にリセット」ボタンをクリックします。
- | • デフォルト値 - コンテンツ・パッケージの作成時に指定された値を表します。ユーザーが「現行値」を指定して意図的にオーバーライドしない限り、この値が使用されます。このしきい値に関してデフォルト値から新しい値に移行するには、「現行値に更新」をクリックします。

#### | 関連タスク

- | 89 ページの『ビューの作成および編集』
- | 設計モードが使用可能になっていると、表または図表が表示されているときに「ビューの編集」アクションを使用できます。ロックされていないパースペクティブに対する新規ビューの作成、編集、および追加も可能です。

#### 収集の管理:

- | パフォーマンス収集の管理には、IBM Systems Director Navigator for i を使用します。

#### 収集の表示:

- | IBM Systems Director Navigator for i を使用して収集を表示するには、以下のいずれかの一連のステップを実行します。

#### 手順

1. IBM Systems Director Navigator for i ウィンドウから、「パフォーマンス」を選択します。
2. 「データの調査」を選択します。Performance Data Investigator ツールが起動します。
3. 収集ライブラリーを選択します。
4. 収集名を選択します。
5. 固形ボックスをクリックして、コンテンツ・パッケージを展開します。
6. ツリー内でノードを展開したままにして、使用したいパースペクティブに進みます。
7. パースペクティブを選択します。
8. 「OK」をクリックします。

#### 次のタスク

- | 収集サービス、Disk Watcher、または Job Watcher ファイル・ベースの収集は、以下のステップを実行することによっても表示できます。
- | 1. IBM Systems Director Navigator for i ウィンドウから、「パフォーマンス」を選択します。
- | 2. 「収集」を選択します。ご使用のシステム上のデータ収集のリストが表示されます。
- | 3. 表示する収集の横にある矢印をクリックし、メニューから「データの調査」を選択します。Performance Data Investigator ツールが起動します。選択した収集データが、コンテンツ・パッケージで定義されたデフォルトのパースペクティブを使用してレンダリングされます。

## 関連概念

78 ページの『データの調査』

「データの調査」タスクを選択すると、強力なツールである Performance Data Investigator が起動します。このツールを使用すると、パフォーマンス収集に保管されたデータを、図表または表の形式で表示および分析できます。

### 収集のコピー:

収集をコピーするには、以下のステップを実行します。

#### 手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**収集**」をクリックします。
3. コピーしたい収集を選択します。
4. 「**Select Actions (アクションの選択)**」メニューから、「**コピー**」を選択します。

### 収集の削除:

収集を削除するには、以下のステップを実行します。

#### 手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**収集**」をクリックします。
3. 削除したい収集を選択します。
4. 「**Select Actions (アクションの選択)**」メニューから、「**削除**」を選択します。

### 収集の保管:

収集を保管するには、以下のステップを実行します。

#### 手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**収集**」をクリックします。
3. 保管したい収集を選択します。
4. 「**Select Actions (アクションの選択)**」メニューから、「**保管**」を選択します。

### 収集のリストア:

収集をリストアするには、以下のステップを実行します。

#### 手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**収集**」を展開します。

4. 「**Restore Collection (収集のリストア)**」をクリックします。

#### **収集の変換:**

前のリリースで収集された収集内容を変換するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**収集**」をクリックします。
3. 変換したい収集を選択します。
4. 「**Select Actions (アクションの選択)**」メニューから、「**変換**」を選択します。

#### **収集プロパティの表示:**

収集プロパティを表示するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**収集**」をクリックします。
3. プロパティを表示したい収集を選択します。
4. 「**Select Actions (アクションの選択)**」メニューから、「**プロパティ**」を選択します。

#### **収集サービスの管理:**

収集サービスの管理には、IBM Systems Director Navigator for i を使用します。

#### **収集サービスの開始:**

収集サービスを開始するには、以下のようになります。

#### **このタスクについて**

収集サービスを開始するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**収集サービス**」を展開します。
5. 「**Start Collection Services (収集サービスの開始)**」をクリックします。

#### **収集サービスの停止:**

収集サービスを停止するには、以下のようになります。

## このタスクについて

収集サービスを停止するには、以下のステップを実行します。

### 手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**収集サービス**」を展開します。
5. 「**Stop Collection Services (収集サービスの停止)**」をクリックします。

### 収集サービスの構成:

収集サービスを構成するには、以下のようになります。

## このタスクについて

収集サービスを構成するには、以下のステップを実行します。

### 手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**収集サービス**」を展開します。
5. 「**Configure Collection Services (収集サービスの構成)**」をクリックします。収集の循環時に要約データを生成するようしておくこと、分析ツールによるパフォーマンス・データベース・データの処理速度を上げることができます。

### 収集サービスの循環:

収集サービスを循環させるには、以下のようになります。

## このタスクについて

収集サービスを循環させるには、以下のステップを実行します。

### 手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**収集サービス**」を展開します。
5. 「**Cycle Collection Services (収集サービスの循環)**」をクリックします。

### 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には、以下のステップを実行します。



## 手順

1. 「IBM Systems Director Navigator for i5/OS」 ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**収集**」をクリックします。
3. パフォーマンス・データの作成対象としたい収集に基づいて、収集サービス \*MGTCOL オブジェクトを選択します。
4. 「**Select Actions (アクションの選択)**」メニューから、「**データベース・ファイルの作成**」を選択します。

## 関連タスク

36 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』

この情報は、収集サービス・データからデータベース・ファイルを手動または自動で作成するために使用します。

### 収集サービス状況の表示:

収集サービス状況を表示するには、以下のようにします。

#### このタスクについて

収集サービス状況を表示するには、以下のステップを実行します。

## 手順

1. 「IBM Systems Director Navigator for i5/OS」 ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**収集サービス**」を展開します。
5. 「**Collection Services Status (収集サービス状況)**」をクリックします。

### IBM i Disk Watcher の管理:

IBM i Disk Watcher の管理には、IBM Systems Director Navigator for i を使用します。

## 関連概念

67 ページの『IBM i Disk Watcher』

IBM i Disk Watcher は、ディスク関連のパフォーマンス上の問題を診断するためのディスク・パフォーマンス・データの収集を行います。

### Disk Watcher の開始:

Disk Watcher を開始するには、以下のようにします。

#### このタスクについて

Disk Watcher を開始するには、以下のステップを実行します。

## 手順

1. 「IBM Systems Director Navigator for i5/OS」 ウィンドウから「**Performance (パフォーマンス)**」を選択します。

2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Disk Watcher**」を展開します。
5. 「**Start Disk Watcher (Disk Watcher の開始)**」をクリックします。

#### **Disk Watcher の停止:**

Disk Watcher を停止するには、以下のようにします。

#### **このタスクについて**

Disk Watcher を停止するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Disk Watcher**」を展開します。
5. 「**Stop Disk Watcher (Disk Watcher の停止)**」をクリックします。

#### **Disk Watcher 定義の追加:**

Disk Watcher 定義を追加するには、以下のようにします。

#### **このタスクについて**

Disk Watcher 定義を追加するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Disk Watcher**」を展開します。
5. 「**Add Disk Watcher Definition (Disk Watcher 定義の追加)**」をクリックします。

#### **Disk Watcher 定義の削除:**

Disk Watcher 定義を削除するには、以下のようにします。

#### **このタスクについて**

Disk Watcher 定義を削除するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。

3. 「**Collectors (コレクター)**」を展開します。
4. 「**Disk Watcher**」を展開します。
5. 「**Disk Watcher Definitions (Disk Watcher 定義)**」をクリックします。
6. 削除したい Disk Watcher 定義を選択します。
7. 「**Select Actions (アクションの選択)**」メニューから、「**削除**」を選択します。

#### **Disk Watcher 定義のプロパティの表示:**

Disk Watcher 定義のプロパティを表示するには、以下のようにします。

#### **このタスクについて**

Disk Watcher 定義のプロパティを表示するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Disk Watcher**」を展開します。
5. 「**Disk Watcher Definitions (Disk Watcher 定義)**」をクリックします。
6. プロパティを表示したい Disk Watcher 定義を選択します。
7. 「**Select Actions (アクションの選択)**」メニューから、「**プロパティ**」を選択します。

#### **IBM i Job Watcher の管理:**

IBM i Job Watcher の管理には、IBM Systems Director Navigator for i を使用します。

#### **関連概念**

66 ページの『IBM i Job Watcher』

IBM i Job Watcher は、システム上の一部またはすべてのジョブ、スレッド、およびタスクについてジョブ・データの収集を行います。コール・スタック、SQL ステートメント、待機されているオブジェクト、Java JVM 統計、待機統計などを提供し、これらはジョブ関連のパフォーマンス上の問題の診断に使用されます。

#### **Job Watcher の開始:**

Job Watcher を開始するには、以下のようにします。

#### **このタスクについて**

Job Watcher を開始するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Job Watcher**」を展開します。

5. 「**Start Job Watcher (Job Watcher の開始)**」をクリックします。

#### **Job Watcher の停止:**

Job Watcher を停止するには、以下のようにします。

このタスクについて

Job Watcher を停止するには、以下のステップを実行します。

手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Job Watcher**」を展開します。
5. 「**Stop Job Watcher (Job Watcher の停止)**」をクリックします。

#### **Job Watcher 定義の追加:**

Job Watcher 定義を追加するには、以下のようにします。

このタスクについて

Job Watcher 定義を追加するには、以下のステップを実行します。

手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Job Watcher**」を展開します。
5. 「**Add Job Watcher Definition (Job Watcher 定義の追加)**」をクリックします。

#### **Job Watcher 定義の削除:**

Job Watcher 定義を削除するには、以下のようにします。

このタスクについて

Job Watcher 定義を削除するには、以下のステップを実行します。

手順

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Job Watcher**」を展開します。
5. 「**Job Watcher Definitions (Job Watcher 定義)**」をクリックします。

6. 削除したい Job Watcher 定義を選択します。
7. 「**Select Actions (アクションの選択)**」メニューから、「**削除**」を選択します。

#### **Job Watcher 定義のプロパティの表示:**

Job Watcher 定義のプロパティを表示するには、以下のようにします。

#### **このタスクについて**

Job Watcher 定義のプロパティを表示するには、以下のステップを実行します。

#### **手順**

1. 「IBM Systems Director Navigator for i5/OS」ウィンドウから「**Performance (パフォーマンス)**」を選択します。
2. 「**Show All Performance Tasks (すべてのパフォーマンス・タスクの表示)**」をクリックします。
3. 「**Collectors (コレクター)**」を展開します。
4. 「**Job Watcher**」を展開します。
5. 「**Job Watcher Definitions (Job Watcher 定義)**」をクリックします。
6. プロパティを表示したい Job Watcher 定義を選択します。
7. 「**Select Actions (アクションの選択)**」メニューから、「**プロパティ**」を選択します。

## **System i ナビゲーター**

System i ナビゲーターは、ご使用のシステムを Windows デスクトップから管理するためのグラフィカル・ユーザー・インターフェースです。System i ナビゲーターが備えている機能を使用すると、ご使用のシステムのパフォーマンス管理に役立ちます。

#### **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、および企業間 (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。

System i 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。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、System i 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 をオフにすることもさえ可能です。System i ナビゲーター はその間もモニターを継続し、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、System i 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。

System i ナビゲーターには、以下のタイプのモニ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。

### 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

### 関連資料

27 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・リソース、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サーバーの通信リソース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。

### 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。モニターの仕事と、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調べることができます。

システム・モニターは、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、グラフ・ヒストリーを使用する必要があります。モニター・プロパティの中でデータ収集の頻度を変更できます。モニター・プロパティの設定は、収集サービスの設定をオーバーライドします。

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、i5/OS 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開始す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。

## 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

116 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。

## モニターの構成:

System i ナビゲーター中でモニターを構成できます。この情報を使用すると、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。

## このタスクについて

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。

## 手順

1. System i 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、「システム」を右クリックし、「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。

- 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。
- 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。
- 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。

## タスクの結果

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。

### モニターのメトリック:

効果的なシステム・パフォーマンスのモニターを行うためには、どのような視点からシステム・パフォーマンスをモニターするかを決めなければなりません。マネージメント・セントラルには、メトリックと呼ばれるさまざまなパフォーマンスの測定法がありますが、このメトリックを使用するとシステム・パフォーマンスのさまざまな局面を正確に示すことができます。

「新しいモニター」ウィンドウの「メトリック」ページでは、モニターに関する一般的な情報 (メトリックを含む) を表示および変更できます。このページを表示するには、「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。フィールドに必要な情報を入力し、「メトリック (Metrics)」タブをクリックします。

モニターを構成する際に、モニターに組み込むリストから任意のメトリック、メトリックのグループ、またはすべてのメトリックを選択して使用することができます。モニターに使用できるメトリックのタイプには、次のものがあります。

表 2.

メトリック・グループ	メトリックの説明
CPU 使用率	<p>処理装置時間のうち、システム上のジョブによって消費される時間のパーセンテージ。以下の中から、モニターで使用する CPU 使用率メトリックのタイプを選択します。</p> <ul style="list-style-type: none"> <li>• CPU 使用率 (平均)</li> <li>• CPU 使用率 (対話型ジョブ)</li> <li>• CPU 使用率 (対話型のフィーチャー)</li> <li>• CPU 使用率 (データベース能力)</li> <li>• CPU 使用率 (2 次ワークロード)</li> <li>• CPU 使用率基本 (平均)</li> </ul> <p>これらのメトリックに関する詳細と使用法については、System i ナビゲーターの「新しいモニター」ウィンドウまたは「モニター・プロパティ」ウィンドウで、「一般」タブからオンライン・ヘルプを参照してください。</p>
対話式応答時間 (平均および最大)	そのシステムでの対話型ジョブの応答時間。
トランザクション率 (平均)	システム上のジョブによって完了したトランザクションの 1 秒当たりの数。



表 2. (続き)

メトリック・グループ	メトリックの説明
トランザクション率 (対話式)	以下のタイプのジョブによってシステム上で完了されたトランザクションの 1 秒当たりの数。 <ul style="list-style-type: none"> <li>対話式</li> <li>複数要求端末 (MRT)</li> <li>システム/36 環境対話式</li> <li>パススルー</li> </ul>
バッチ論理データベース I/O	システム上のバッチ・ジョブによって現在実行されている論理データベース入出力 (I/O) 操作の平均数。
ディスク・アーム使用率 (平均および最大)	データ収集中にシステム上で使用されているディスク・アーム・キャパシティーのパーセンテージ。
ディスク・ストレージ (平均および最大)	データ収集中にシステム上でいっぱいになっているディスク・アーム記憶域のパーセンテージ。
ディスク IOP 使用率 (平均および最大)	データ収集中のシステム上のディスク入出力処理装置 (IOP) が使用されている程度。
通信 IOP 使用率 (最大および平均)	データ収集中のシステム上の通信入出力処理装置 (IOP) が使用されている程度。
通信回線使用率 (平均および最大)	すべてのシステム通信回線から実際に送信および受信したデータの量。
LAN 使用率 (最大および平均)	すべてのローカル・エリア・ネットワーク (LAN) 通信回線から実際に送信および受信したデータの量。
マシン・プール不在	システム上のマシン・プールで発生する 1 秒当たりのページ不在の数。
ユーザー・プール不在 (最大および平均)	システムのすべてのユーザー・プールの中で発生する 1 秒当たりのページ不在の数。

より詳細なヘルプを表示させるには、「新しいモニター」-「メトリック」ウィンドウで、「ヘルプ」ボタンをクリックします。 マネージメント・セントラル・メトリックの使用に慣れたら、ご使用のコンピューティング環境に必要な情報に合わせてメトリックを選択できます。 必要な情報的に絞ってメトリックを選択した後、モニター用に選択した各メトリックごとにメトリックの詳細情報を表示および変更することができます。

#### 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:

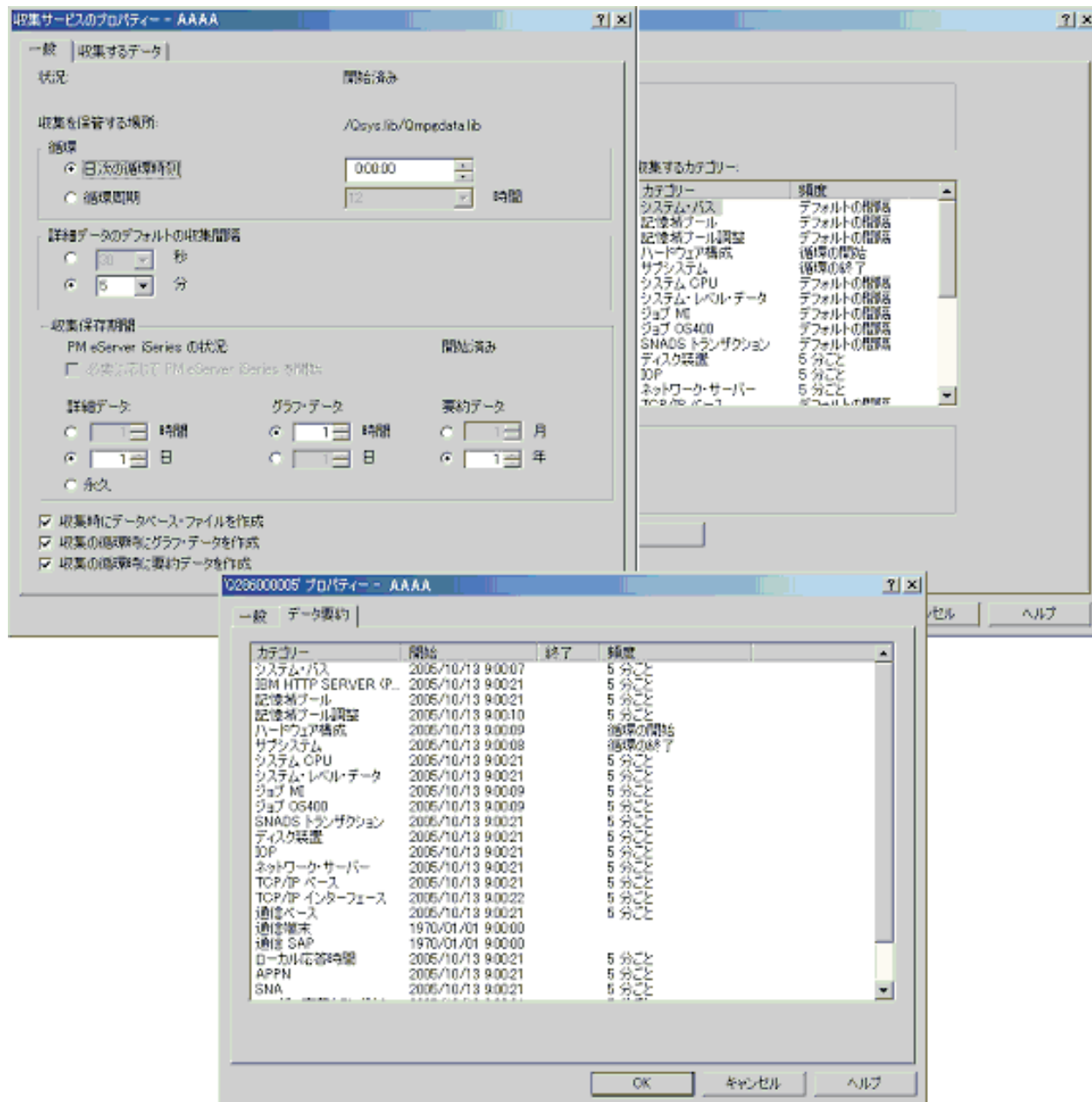
収集サービスは、スタンドアロン・アプリケーションとして、またパフォーマンス・データを収集するために他のアプリケーションが使用するユーティリティーとして、パフォーマンス分析用の貴重なツールです。

時々、システムで生じるアクティビティー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収集サービス・データ収集はただその 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

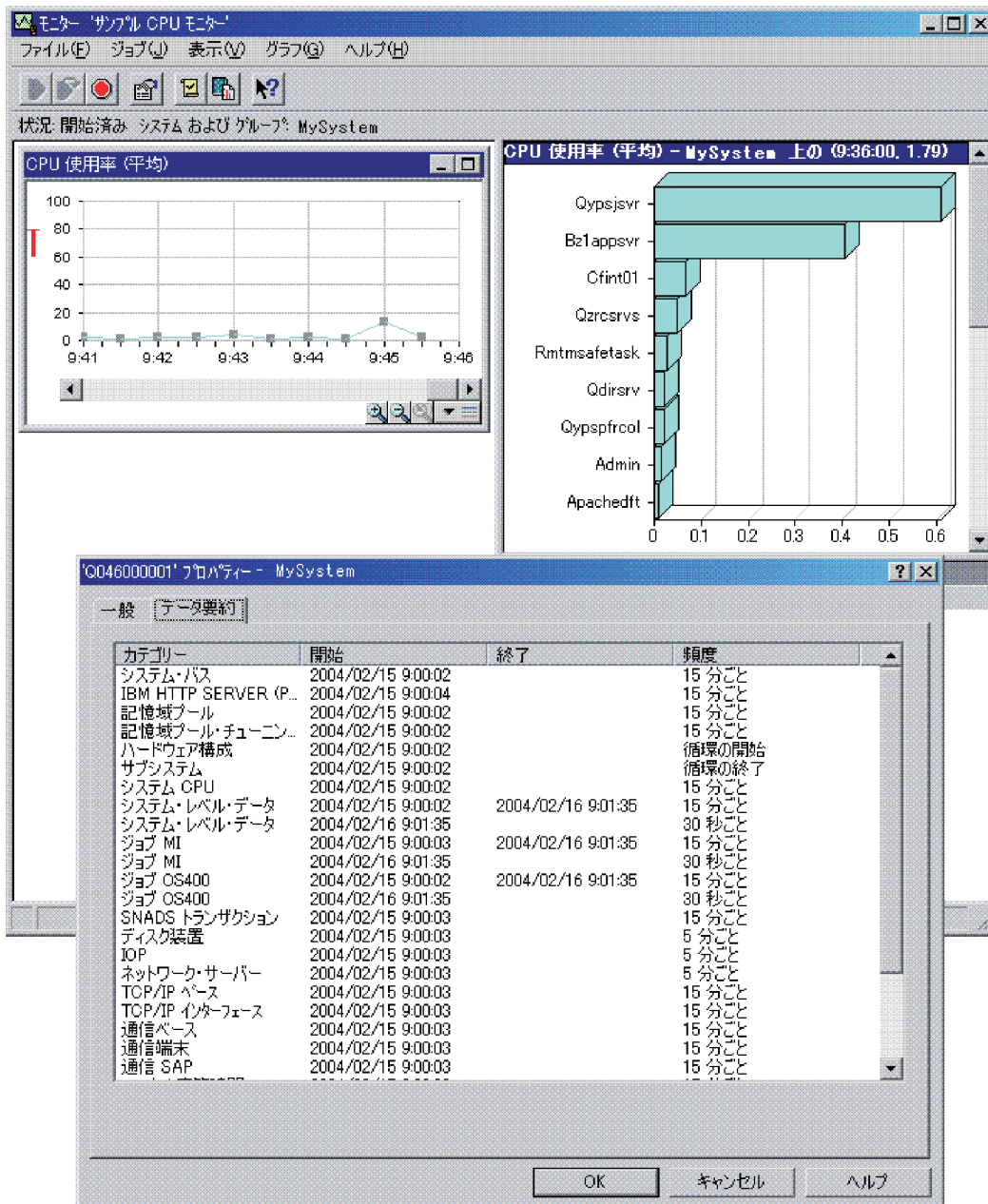
このシナリオでは、システム上で活動状態のシステム・モニターまたはジョブ・モニターはありません。収集サービスのプロパティ・ページおよび \*MGTCOL オブジェクトのプロパティ・ビューを表示すると、以下のようなものが表示されます。



## 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 データのカテゴリの収集間隔が 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。



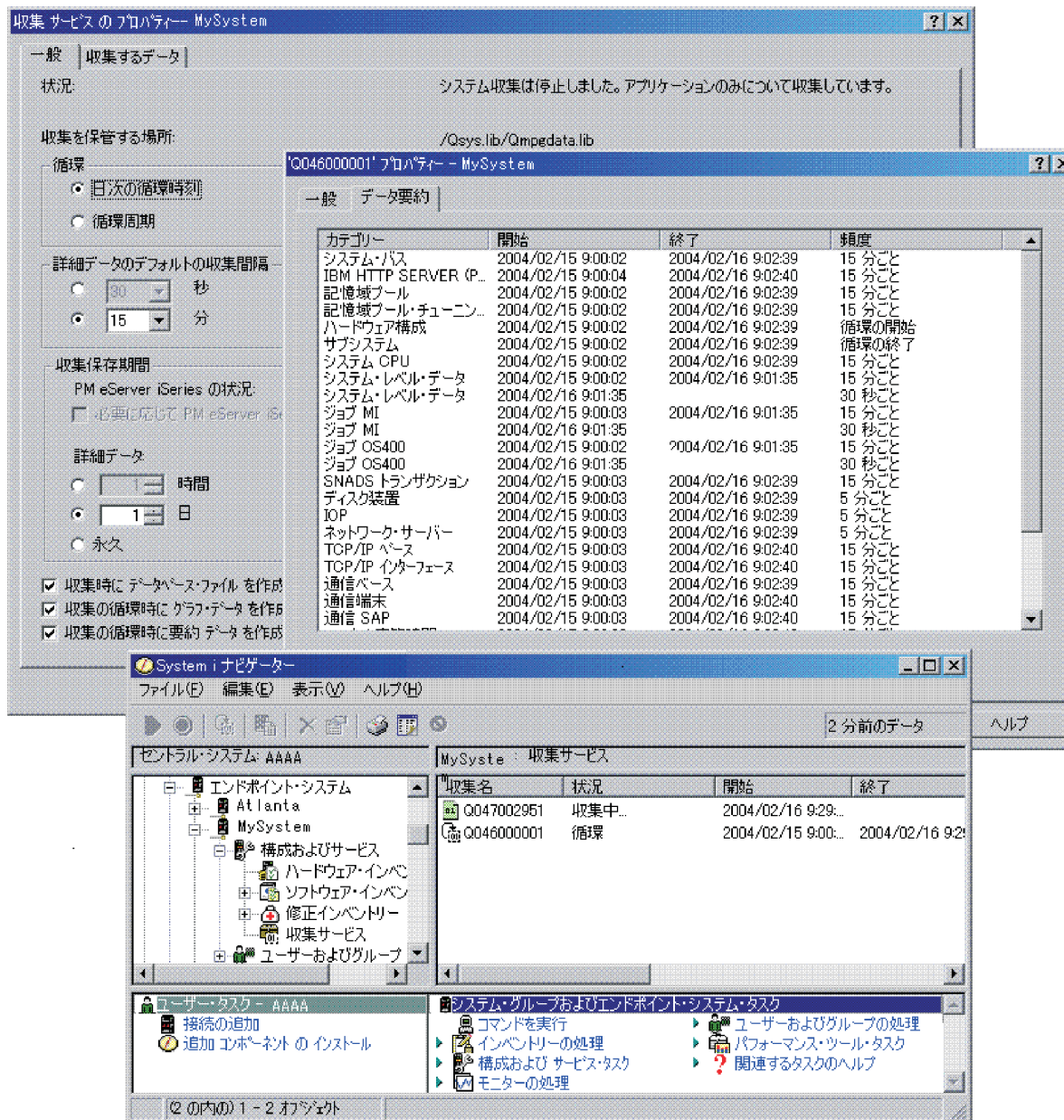
### 収集サービスは停止し、システム・モニターは開始されたままの場合

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で、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。システム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 オブジェクトがシステムについて収集中... 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。



### シナリオ: System i ナビゲーター モニター:

この情報を使用して、さまざまなタイプのモニターのいくつか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を確認します。

System i 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。System i 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、『System i ナビゲーター 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。

### 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。

### 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リソースが、確実に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. System i 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) 「i5/OS コマンド (i5/OS command)」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) 「i5/OS コマンド (i5/OS command)」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) 「i5/OS コマンド (i5/OS command)」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:

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。

## 状態

現在システム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 キャパシティーが多すぎる場合に、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 キャパシティーが 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. System i 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) 「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) 「i5/OS トリガー・コマンド (i5/OS 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:

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。

## 状態

現在、システム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。

注: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

1. System i 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) 「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) 「i5/OS トリガー・コマンド (i5/OS 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。

注: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

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

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

DCL &MSGTXT *CHAR 132

RCVMSG 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。

## 関連概念

### 通知の処理

Advanced Job Scheduler の通知機能の使用方法については、『通知の処理』のトピックを参照してください。

## 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、システム上で生じた、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。

## 状態

貴社に複数のサーバーがあり、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. System i 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システム上の 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 i5/OS 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。

グラフ・ヒストリーには、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。

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

## 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

8 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。

### グラフ・ヒストリーの概念:

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を説明しています。

「グラフ・ヒストリー」には、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。System i ナビゲーターを使用して、複数システムにまたがる Performance Management Agent (PM Agent) をアクティブにします。PM Agent 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 Agent 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 Agent 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。

- **明細データ (QMPGDATA.LIB または QPFRDATA.LIB 内の属性タイプ \*PFR)**

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 (QMGTC2.LIB 内の属性タイプ \*PFRDTL)**

「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 Agent 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 Agent 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。

- **グラフ・ヒストリー・データ (QMGTC2.LIB 内の属性タイプ \*PFRHST)**

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 Agent 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。要約データは、1 時間の間隔で要約され、2 次および 3 次レベルの詳細はサポートされません。

- **Graph history status**

「グラフ・ヒストリー」ウィンドウは、グラフの履歴状況を表示します。また、脱落したグラフ履歴データを再作成することもできます。

## 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

## 関連タスク

120 ページの『PM Agent の活動化』

PM Agent はオペレーティング・システムの付属製品であり、収集機能を利用するには活動化する必要があります。

## グラフ・ヒストリーの表示:

このトピックには、System i ナビゲーターから、グラフ・ヒストリーを表示するための段階的な説明が含まれています。

## このタスクについて

グラフ・ヒストリーは、System i ナビゲーター に含まれています。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。

## 手順

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、System i 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. 「収集サービスの開始 - 一般」ページから、「**Start IBM Performance Management for Power Systems (IBM Performance Management for Power Systems の開始)**」を選択します (必要な場合)。
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。

## タスクの結果

**ヒント:** グラフ・ヒストリー・データが欠落している場合、再作成することができます。グラフ・ヒストリー・データを再作成するには、System i ナビゲーター 内のオブジェクトを右クリックして、「**グラフ・ヒストリー・データの作成 (Create Graph History Data)**」を選択します。

グラフ・ヒストリー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。

## 次のタスク

以下の場合には、システムはアクティブ収集オブジェクト (\*PFR 属性) から \*PFRDTL および \*PFRHST 収集オブジェクトに、データを追加します。

- 収集オブジェクトのプロパティが、循環時にはグラフ・データおよび要約データを追加するように設定されている場合は、収集は循環されます。
- 既に循環されたオブジェクトが選択され、データを要約するメニュー・オプションが選択された場合。
- システム・モニターが実行中である場合は、データは、\*PFRDTL オブジェクトにのみ追加されます (システム・モニターが実行中のため)。

#### 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

## IBM Performance Management for Power Systems - IBM i のサポート

- | IBM i オファリングをサポートする IBM Performance Management for Power Systems (PM for Power Systems) 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・リソース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 for Power Systems オファリングには、Performance Management Agent (PM Agent) が組み込まれています。PM Agent は、オペレーティング・システムの 1 機能です。非専有の収集サービス・データの自動収集を行い、データを削減したうえで IBM に送信します。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。

このオファリングを IBM Systems Workload Estimator とともに使用すると、お客様のビジネス・トレンドが、必要なハードウェア・アップグレード (たとえば、中央演算処理装置 (CPU) またはディスク) のタイミングとどのように関係しているかを、より深く理解できるようになります。IBM Systems Workload Estimator は、PM Agent に複数のシステムまたは区画のデータを IBM Systems Workload Estimator に送信させることで、システム統合のサイズを変更したり、論理区画でのシステムのアップグレードを評価したりできます。

#### 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

#### 関連タスク

88 ページの『次期アップグレードの見積もり』

現在のパフォーマンス統計に基づいた今後のシステムの見積もりで使用される Workload Estimator に現行セッションからデータを送信するには、「次期アップグレードの見積もり」アクションを使用します。

#### 関連情報

 「PM for Power Systems」Web サイト (英語)

PM for IBM i の詳細については、「PM for Power Systems」Web サイト (英語) を参照してください。

#### PM Agent の概念:

PM Agent 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。

PM Agent 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 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。

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 of 正確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 Agent と収集サービスとの間の関係のため、PM Agent 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 Agent 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。

- **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 Agent 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 Agent は、15 分間隔でパフォーマンス・データを収集します。PM Agent 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。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 Agent 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 Agent 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。**

例えば、System i ナビゲーター から PM Agent を活動化すると、「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としてチェック・マークが付けられます。これを変更すると、PM Agent は正時にデフォルト値に戻ります。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。

## 関連資料

38 ページの『収集サービス収集プロファイル』

収集サービス収集プロファイルの説明です。収集プロファイルでは、収集対象を定義します。

## PM Agent の構成:

PM Agent の使用を開始するには、それを活動化し、伝送方式をセットアップしてから、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。

PM Agent 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 Agent 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 Agent が起動時に作成します。

PM Agent 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 Agent の活動化:

PM Agent はオペレーティング・システムの付属製品であり、収集機能を利用するには活動化する必要があります。

### このタスクについて

データ収集機能を利用するには、PM Agent 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 Agent を開始することができます。

#### System i ナビゲーターを使用します。

System i ナビゲーターを使用して、複数のシステムで PM Agent をアクティブにします。PM Agent をアクティブにすると、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。PM Agent を使用可能になっていない場合は、最大で過去 7 日間のデータを表示できます。PM Agent 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。

System i ナビゲーターから PM Agent を開始するには、次のようにします。

#### 手順

1. PM Agent を開始したいシステムを System i 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 Agent」を選択します。
5. 「開始」を選択します。
6. PM Agent 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。

#### タスクの結果

##### 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 Agent 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「PM Agent を活動化しますか? (I G C)(Do you want to activate PM Agent? (I G C))」に G と応答します。QSYSOPR メッセージ・キューは、PM Agent 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 Agent の構成 (CFGPMAGT) コマンドの発行

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

- 1 セットアップ・プロセスの次のステップに進むことができます。次のステップの目的は、IBM にデータを
- 1 送信するために使用される Service Agent 伝送をセットアップすることです。

## 関連概念

116 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。

『データの PM Agent 伝送のセットアップ』

パフォーマンス・データの収集と送信には、Electronic Service Agent のインベントリー機能を使用します。

## 関連タスク

129 ページの『PM Agent の非活動化』

PM Agent を停止する方法について学習します。

### データの PM Agent 伝送のセットアップ:

- | パフォーマンス・データの収集と送信には、Electronic Service Agent のインベントリー機能を使用します。
- | 伝送方式を実装し終わったら、PM Agent を管理するための他のタスクを行うことができます。

## 関連概念

マネージメント・セントラル

## 関連資料

129 ページの『PM Agent の管理』

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

**Electronic Service Agent による PM Agent データの送信:** パフォーマンス・データを収集するためには、PM Agent が活動状態でなければなりません。PM Agent 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、Electronic Service Agent を使用して IBM にデータを送信します。

## 関連概念

ユニバーサル・コネクション

## 関連タスク

エクストリーム・サポートでの Electronic Service Agent

120 ページの『PM Agent の活動化』

PM Agent はオペレーティング・システムの付属製品であり、収集機能を利用するには活動化する必要があります。

## 関連資料

129 ページの『PM Agent の管理』

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

### 単一サーバー用の PM Agent ネットワークの構成:

単一サーバーは、IBM に直接そのデータを送信します。

- | このタスクについて
- | 以下は、PM Agent を単一サーバー用に構成する場合に実行する必要があるステップです。



## 手順

1. コマンド行に **CFGPMAGT** と入力します。
2. 「**IBM へのパフォーマンス・データの送信に関する接続オプションの選択**」フィールドに、「**1 = サービス・エージェントを使用してデータを送信する**」を指定します。
3. 「**パフォーマンス・データの受信**」フィールドに **0 = いいえ (No)** を指定します。
4. 「**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、すべての必須フィールドが入力済みであることを確認します。そうでなければ、構成は完了しません。

## 関連資料

129 ページの『PM Agent の管理』

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

### ホスト・サーバー用の PM Agent ネットワークの構成:

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 にデータを転送します。

### このタスクについて

以下は、PM Agent をホスト・サーバー用に構成する場合に実行する必要があるステップです。

## 手順

1. ホスト・サーバーのコマンド行に **CFGPMAGT** と入力します。「PM エージェントの構成」画面から、以下のようにします。
  - 「**IBM へのパフォーマンス・データの送信に関する接続オプションの選択**」フィールドに、「**1 = サービス・エージェントを使用してデータを送信する**」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 **1 = Yes** を指定します。
2. ホスト・サーバーのコマンド行に **GO PMAGT** と入力します。PM Agent メニュー画面から、以下のようにします。
  - コマンド行で **5** (リモート IBM i システムの処理) を入力し、**Enter** を押します。
  - **F6** (作成) を押して、どのサーバーが自身のデータをホスト・サーバーに送信するかを指定します。
  - フィールドを完成させて、**Enter** を押します。

## タスクの結果

PM Agent 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。ホスト・サーバーへのデータの送信を週全体で均等にスケジュールします。このアクションで、ホスト・サーバーへのパフォーマンスの影響が最小限に抑えられます。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、ホスト・サーバーに送信されるデータの量が均等に分散されます。

サーバーを構成し終わったら、PM Agent 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。

## 関連資料

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

### リモート・サーバー用の PM Agent ネットワークの構成:

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。

#### このタスクについて

以下は、PM Agent をリモート・サーバー用に構成する場合に実行する必要があるステップです。

#### 手順

1. コマンド行に **CFGPMAGT** と入力します。
2. 「IBM へのパフォーマンス・データの送信に関する接続オプションの選択」フィールドに、「**2 = リモート IBM i で SNA を使用する**」を指定します。
3. 「パフォーマンス・データの受信」フィールドに **0 = いいえ (No)** を指定します。

#### タスクの結果

注: システムがネットワークになっている場合は、System i 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、ユニバーサル・コネクションを介してそれらのシステムのデータを伝送することをお勧めします。

サーバーを構成し終わったら、PM Agent 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。

## 関連資料

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

### リモート・サーバーでの処理:

サイトによっては、ホスト・サーバーと 1 つ以上のリモート・サーバーでネットワークが構成されることになります。この場合は、PM データを処理のために IBM に送信するのは、ホスト・サーバーになります。

#### このタスクについて

1. 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 System i システムの処理 (Work with Remote System i systems)」画面 (**GO PMAGT** と入力してから **5** を入力) で、これらの他のサーバーを定義できます。
1. 注: ネットワーク・システムを利用している場合、System i 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。

## 手順

1. コマンド行で **GO PMAGT** と入力します。
2. **PM Agent** メニューで **5 (リモート IBM i システムの処理)** を入力し、Enter を押します。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・システムでの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 IBM システムの処理 (Work with Remote IBM systems)」画面に、リモート・サーバーのリストが表示されます。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 Agent** リモート・サイトの保守 (PM Agent Remote Site Maintenance)」画面または「リモート・サイト IBM i の変更 (Change Remote Site IBM i)」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。

## タスクの結果

PM Agent 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、『リモート・サーバー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 ネットワークに関する考慮事項』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 PM Agent のカスタマイズを行う準備ができています。

### 関連タスク

130 ページの『PM Agent を使用したジョブのスケジューリング』  
PM Agent 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。

127 ページの『PM Agent 用の装置記述の作成』  
PM Agent 用の装置記述を作成できます。

『APPC ネットワーク内のリモート・サーバーの処理』  
ホスト・サーバーは、他のサーバーから PM Agent データを受信してから、IBM にデータを送信します。リモート・サーバーは、ホスト・サーバーに PM Agent データを送信します。

128 ページの『PM Agent 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 Agent をカスタマイズする必要があります。

### APPC ネットワーク内のリモート・サーバーの処理:

1. ホスト・サーバーは、他のサーバーから PM Agent データを受信してから、IBM にデータを送信します。
1. リモート・サーバーは、ホスト・サーバーに PM Agent データを送信します。

## このタスクについて

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 Agent がデータを収集する場合のみ、各リモート・サーバーへの接続をサポートするために装置のペアを作成する必要があります。

### 手順

- 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。

表3. リモート・システム

DEVD(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. ホスト・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。

表4. 1 次サーバー

DEVD(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。

表5. 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。

- PM Agent 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、『リモート・サーバーでの処理』を参照してください。

## タスクの結果

これで PM Agent の構成が完了しました。PM Agent で実行できる他のタスクについては、『PM Agent の管理』を参照してください。

## 関連タスク

124 ページの『リモート・サーバーでの処理』

サイトによっては、ホスト・サーバーと 1 つ以上のリモート・サーバーでネットワークが構成されることになります。この場合は、PM データを処理のために IBM に送信するのは、ホスト・サーバーになります。

## 関連資料

129 ページの『PM Agent の管理』

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

## PM Agent 用の装置記述の作成:

PM Agent 用の装置記述を作成できます。

## このタスクについて

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 Agent がデータを収集する場合のみ必要です。

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。

APPN を使用するように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、『非 APPN 環境内でのリモート・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。

```
VRVCFG  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)D  
USER(QUSER)  
AUT(\*CHANGE)

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

## 関連タスク

125 ページの『APPC ネットワーク内のリモート・サーバーの処理』  
ホスト・サーバーは、他のサーバーから PM Agent データを受信してから、IBM にデータを送信します。  
リモート・サーバーは、ホスト・サーバーに PM Agent データを送信します。

## 関連資料

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

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

## PM Agent 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 Agent をカスタマイズする必要があります。

## このタスクについて

「PM Agent カスタマイズの処理 (Work with PM Agent Customization)」画面では、以下の機能を提供します。

## PM Agent ソフトウェアの操作用のグローバル・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

## PM Agent データ電話番号の定義

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。

## 手順

1. コマンド行で **GO PMAGT** と入力します。
2. 「PM Agent カスタマイズの処理 (Work with PM Agent Customization)」画面を表示するには、PM Agent メニューで 3 と入力して、Enter を押します。

## タスクの結果

PM Agent で実行できるその他のタスクについては、『PM Agent の管理』を参照してください。

## 関連資料

### 『PM Agent の管理』

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

### PM Agent の管理:

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

PM Agent 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。

## 関連資料

PM Agent の終了 (Q1PENDPM) API

### PM パフォーマンス・データ伝送の開始:

PM Agent メニューを使用して、パフォーマンス・データの Service Agent 伝送を開始できます。

### このタスクについて

PM パフォーマンス・データ伝送を開始するには、以下のステップを実行します。

### 手順

1. コマンド行で **GO PM400** と入力します。
2. コマンド行から **9** と入力して、Enter を押します。ジョブ・ログを調べて、戻されたメッセージを確認します。

### PM Agent の非活動化:

PM Agent を停止する方法について学習します。

### このタスクについて

PM Agent 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。

### System i ナビゲーター の使用

以下のステップを実行します。

### 手順

1. System i ナビゲーターで、PM Agent 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 Agent」を選択します。
5. 「停止」を選択します。
6. PM Agent 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。

## タスクの結果

### API の使用

PM Agent の終了 (Q1PENDPM) API を使用して PM Agent を非活動化します。

### 関連タスク

120 ページの『PM Agent の活動化』

PM Agent はオペレーティング・システムの付属製品であり、収集機能を利用するには活動化する必要があります。

### **PM Agent 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。

### このタスクについて

PM Agent 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。

### 手順

1. コマンド行で **GO PM400** と入力します。
2. PM Agent 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 Agent を使用したジョブのスケジューリング:**

PM Agent を使用してジョブをスケジューリングする方法を学習します。

### このタスクについて

PM Agent ソフトウェアに不可欠なのは、PM Agent 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 Agent 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 Agent 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。

### 手順

1. コマンド行で **GO PM400** と入力します。
2. PM Agent 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、Enter を押します。「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。



## タスクの結果

以下の表は、選択できる PM Agent ジョブを一覧で示しています。

PM Agent のスケジュールされたジョブ		
ジョブ	スケジュール	機能
Q1PTEST	活動化時	PM Agent 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。
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。
Q1PCM4	必要時	リモート・サーバーから PM Agent にアクセスします。このジョブが開始するのは、PM Agent 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。

### IBM Performance Management for IBM i 分析からの項目の省略:

IBM Performance Management for IBM i (PM for IBM i) 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説明します。

#### このタスクについて

PM for IBM i 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。例えば、ランタイム・カテゴリの中の実行時間が通常より長いジョブ(自動開始ジョブなど)は、除外することもできます。

汎用除外機能を使用することによって分析からバッチ・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。

#### 手順

1. コマンド行で **GO PM400** と入力します。
2. PM Agent メニューで **4** と入力して、Enter を押します。「省略項目の処理 (Work with 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. 特定の 카테고리からユーザーまたはジョブのどちらかを省略する場合は、該当するフィールドに **1** を入力します。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 Agent の暫時停止:**

PM Agent を暫時停止する方法について学習します。

#### **このタスクについて**

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 Agent 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。

#### **手順**

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 Agent 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。

#### **タスクの結果**

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 Agent は収集サービスを開始、循環、または変更しません。

#### **関連タスク**

130 ページの『PM Agent を使用したジョブのスケジューリング』  
PM Agent 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。

#### **| Service Agent 接続の検査:**

| PM Agent メニューを使用して、サーバーから IBM への Service Agent 接続の状況を検査できます。

#### **| このタスクについて**

| Service Agent 接続を検査するには、以下のステップを実行します。

#### **| 手順**

- | 1. コマンド行で **GO PM400** と入力します。
- | 2. コマンド行から **8** と入力して、Enter を押します。ジョブ・ログを調べて、戻されたメッセージを確認します。

#### **| PM Agent 状況の表示:**

PM Agent 状況を表示するために、System i ナビゲーターまたは PM Agent メニューの使用方法を学習します。

## このタスクについて

System i ナビゲーターまたは PM Agent メニューのいずれかを使用して、PM Agent の状況を表示することができます。1 つ以上のサーバーまたはグループ上の PM Agent の全体状況を表示するには、「PM Agent 状況」ダイアログを使用します。たとえば、PM Agent 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 Agent メニューを使用して、収集サービス状況、PM Agent 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。

System i ナビゲーターから PM Agent 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。

### 手順

1. System i 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 Agent」を選択します。
5. 状況を選択します。

### タスクの結果

PM Agent メニューから PM Agent 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。

### **IBM Performance Management for IBM i 報告書の表示:**

IBM Performance Management for IBM i 報告書の例と、それらの報告書の解釈の仕方についての説明を参照してください。

IBM Performance Management for IBM i オファリングの出力は、管理報告書およびグラフのセットです。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。報告書の表示方法、および報告書の利点と利用法のいくつかについては、IBM Performance Management for IBM i Web サイトにアクセスしてください。

### 関連概念

116 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。

### 関連情報

 「PM for Power Systems」 Web サイト (英語)

## IBM Systems Workload Estimator

IBM Systems Workload Estimator は、System i、System p、および System x 用の Web ベースのサイズ変更ツールです。このツールを使用して、新規システムのサイズ変更、アップグレードから既存のシステムへのサイズ変更、または複数のシステムの統合のサイズ変更を実行できます。

Workload Estimator により、現行ワークロードを最も適切に反映する測定入力が可能になります。この 1 つの方式は、Performance Management for System i からのデータを使用することです。 Workload

Estimator はさらに、新たに生じるアプリケーション要件を反映するための多様な組み込みワークロードも提供しています。仮想化を使用して、さらに堅固なソリューションを作成できます。 Workload Estimator は、総合的なクライアントのパフォーマンス要件を満たす、プロセッサ、メモリー、およびディスクの現状の推奨事項と発展に伴う推奨事項を提供します。

## 関連情報

 [IBM Systems Workload Estimator](#)

Workload Estimator のオンライン・バージョンを実行するには、IBM Systems Workload Estimator の Web サイトを参照してください。

## Performance Tools

Performance Tools ライセンス・プログラムには、オペレーティング・システムで使用可能な基本パフォーマンス・ツールの機能を補足または拡張する、数多くの補足機能があります。

### 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

135 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する方法について記述されています。

### 関連情報

 [Performance Tools \(PDF\)](#)

パフォーマンス用の CL コマンド

### 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。

### 関連概念

34 ページの『収集サービス』

収集サービスは、システム管理データの収集の手段を備えます。これはシステム・データの基本コレクターです。

### 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。このトピックでは、このライセンス・プログラムに組み込まれている機能を説明します。

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	説明
パフォーマンス・データの表示	System i 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。

ツール 報告書	<b>説明</b> 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、およびトレース・データを編成します。報告書は、資料 Performance Tools で詳細に説明されています。
グラフィックス機能	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。このツールは、資料 Performance Tools で詳細に説明されています。
IBM i Job Watcher	IBM Systems Director Navigator for i Performance インターフェースの Job Watcher 機能は、Performance Tools に組み込まれています。
IBM i Disk Watcher	IBM Systems Director Navigator for i Performance インターフェースの Disk Watcher 機能は、Performance Tools に組み込まれています。

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

## 関連タスク

136 ページの『Performance Tools プラグイン』

システム・リソースの使用率のデータを System i 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。

## 関連資料

システム活動の処理 (WRKSYSACT) コマンド

## 関連情報

Performance Tools 報告書



Performance Tools (PDF)

## 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する方法について記述されています。

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。マネージャー・フィーチャーには、IBM Systems Director Navigator Performance インターフェースの IBM i5/OS Disk Watcher 機能も含まれています。

## 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。Performance explorer の報告機能とその関連コマンドは、Performance Tools for i5/OS 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・ユーティリティー (ジョブ・トレースおよびファイル選択ユーティリティー)、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。

## 関連概念

78 ページの『IBM Systems Director Navigator for i Performance インターフェース』

IBM Systems Director Navigator for i Performance インターフェースを使用すると、パフォーマンス・データの表示、収集、管理を、さまざまなパフォーマンス情報やツールが揃った 1 つの環境で集中的に行えます。

## Performance Tools プラグイン:

システム・リソースの使用率のデータを System i 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。

## このタスクについて

Performance Tools は、System i 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

System i 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・リソース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 i5/OS (5770-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。

「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
- リソース

## System i ナビゲーター を使ったアクセス

「パフォーマンス・データの表示 (Display Performance Data)」GUI は、System i 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 System i ナビゲーター からアクセスできます。

### 手順

1. System i 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。

### タスクの結果

System i 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、System i ナビゲーター のオンライン・ヘルプを参照してください。

## 関連概念

135 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。

## 関連タスク

142 ページの『Performance Tools プラグインの System i ナビゲーターへのインストール』  
Performance Tools プラグインを System i ナビゲーターにインストールして、システム・リソース使用状況データを表示することができます。

## 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。

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いました。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いました。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し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	合計 CPU	対話式	バッチ	HVLPTASK
OS/400® V5R2 以前	50%	5%	10%	35%
OS/400 V5R3 以降	50%	17%	33%	N/A

## 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。

区画キャパシティー値は、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・リソースに依存します。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す。

論理区画 (LPAR) 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 というフィールド (または列) で、QAPMSYSCPU パフォーマンス・データベース・ファイルに含められます。

### 使用可能な共用プロセッサ・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、QAPMSYSTEM パフォーマンス・データベース・ファイルに含められます。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。プールを共用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、QAPMSYSTEM パフォーマンス・データベース・ファイルに含められます。

### 区画の保証されたキャパシティー

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、QAPMSYSTEM パフォーマンス・データベース・ファイルに含められます。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。

### 区画のプロセッサ使用率

論理区画により使用された、合計 CPU 時間。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、QAPMSYSTEM パフォーマンス・データベース・ファイルに含められます。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。

### 区画の使用可能なキャパシティー

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、QAPMSYSTEM パフォーマンス・データベース・ファイルに含められます。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

### 関連情報

パフォーマンス・データ・ファイル: QAPMSYSTEM

パフォーマンス・データ・ファイル: QAPMSYSCPU

### 5250 オンライン・トランザクション処理 (OLTP):

このトピックでは、5250 オンライン・トランザクション処理およびこの作業に関連するジョブまたはスレッドを説明します。

オンライン・トランザクション処理 (OLTP) とは、ユーザーからサブミットされた要求を受け取り後直ちに処理する、対話式アプリケーション・タイプです。以下は、OLTP 処理の例です。

- 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、システム対話。
- Domino 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。

System i 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 カテゴリに組み込まれ、対話式としてリストされていました。分散データ管理 (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 では、Performance Tools ライセンス・プログラムが更新され、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をより良く分散するようになりました。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るジョブを意味します。System i 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・セクションに移動しました。

### 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。

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(xxxxPT1) 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(xxxxPT1) 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(xxxxPT1) DEV(NAME) OPTION(2)
```

- マネージャー・フィーチャーまたはエージェント・フィーチャーのいずれかをインストールすることに加え、IBM i5/OS Job Watcher を購入した場合は、次のコマンドを使用します。

```
RSTLICPGM LICPGM(xxxxPT1) DEV(tape-device-name) OPTION(3)
```

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(xxxxPT1) 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。

このプログラムは、資料 Performance Tools で詳細に説明されています。

## 関連情報



Performance Tools (PDF)

### **Performance Tools プラグインの System i ナビゲーターへのインストール:**

Performance Tools プラグインを System i ナビゲーターにインストールして、システム・リソース使用状況データを表示することができます。

#### このタスクについて

Performance Tools プラグインをインストールする前に、まず Performance Tools (5770-PT1) ライセンス・プログラムがインストールされていることを確認する必要があります。 Performance Tools プラグインをインストールするには、次のようにします。

#### 手順

1. System i ナビゲーターで、「**ユーザー接続**」を右クリックして、「**オプションのインストール**」 → 「**プラグインのインストール**」を選択します。
2. 「プラグインのインストール」パネルで、プラグインのインストール元のシステムを選択し (5770-PT1 製品がインストールされているシステムでなければなりません)、「**OK**」をクリックします。
3. i5/OS ユーザー・プロファイル名とパスワードを「パスワード」パネルに入力し、「**OK**」をクリックします。(Windows パスワードを求めるプロンプトが出されますが、i5/OS ユーザー・プロファイル・パスワードにする必要があります。)

**注:** 一部の Windows オペレーティング・システムでは、Windows と i5/OS ユーザー・プロファイル・パスワードとを一致させることが求められる場合があります。

4. 選択したシステム上のプラグインのスキャンが次に実行されます。「プラグイン選択」パネルが表示されたら、「Performance Tools」チェック・ボックスにチェックを付け、「**次へ**」をクリックします。
5. プラグインのインストール後に最初に System i ナビゲーターが使用されると、System i ナビゲーターのスキャン・パネルが表示されます。「**今すぐにスキャン**」をクリックします。「**今すぐにスキャン**」ボタンをクリックしない場合は、直前にインストールしたプラグインは無効になり、System i ナビゲーターには表示されません。

#### **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・リソース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。

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、報告書を印刷してシステム・リソース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。

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 つの報告書のトレース情報を収集するには、パフォーマンス・トレースの開始 (STRPFRTRC) および パフォーマンス・トレースの終了 (ENDPFRTRC) コマンドを使用する必要があります。

#### **Performance Tools 報告書の概要:**

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です。

表 6. 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
システム報告書	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、リソースの使用、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全般的に把握します。	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ません。	ワークロードの予測
構成要素報告書	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・リソース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。	リソースの使用、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。	ハードウェアの拡張および構成処理の傾向
トランザクション報告書	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。	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	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
ロック報告書	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。	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	問題分析。オブジェクト競合の低減または除去。

表 6. Performance Tools 報告書の概要 (続き)

報告書	説明	表示される内容	情報の使用方法
バッチ・ジョブ追跡報告書	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。使用されたリソース、例外、および状態遷移が報告されます。	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	問題分析およびバッチ・ジョブの進行
ジョブ間隔報告書	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
プール間隔報告書	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
リソース間隔報告書	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リソースの情報を示します。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。

### システム報告書 - 作業負荷:

システム報告書の作業負荷セクションは、システムの対話式作業負荷および対話式以外の作業負荷を示します。

システム報告書の作業負荷セクションの「パート 1」は、システムの対話式作業負荷を示しています。作業負荷セクションの「パート 2」は、システムの非対話式作業負荷を示します。

システム報告書 6/26/04 16:06  
作業負荷 ページ 0

メンバー . . . . : PNT6PERF 型/製造番号 . . : 825/10-5M0FM 主記憶域 . . . . . : 8192.0 MB 開 始 . . . . : 04/07/04 19:11  
ライブラリー . . : CARR098R01 システム名 . . . . : CARREGT バージョン/リリース: 5/ 4.0 停 止 . . . . : 04/07/04 20:15  
区画 ID . . . . . : 000 機能コード . . . . : 7415-2472-7415 対話式しきい値 . . : 100.00 %  
仮想プロセッサ: 32 処理装置 . . . . . : 32.0  
QPFRADJ . . . . . : 0 QDYNPTYSCD . . . . : 1 QDYNPTYADJ . . . . . : 1

対話式作業負荷 ジョブ タイプ	トランザクション 数	平均 応答	論理 DB 入出力カウント	----- 印刷装置 ----- 行数	ページ数	通信 入出力カウント	MRT 最大時間
対話式	3,242	.65	16,734	12,910	339	0	0
DDM サーバー	0	.00	864,667	443	23	1,596,096	0
パススルー	6,645	.68	343,262	1,119,009	27,769	240	0
合計	9,887		1,224,663	1,132,362	28,131	1,596,336	
平均		.67					

非対話式作業負荷 ジョブ タイプ	トランザクション 数	論理 DB 入出力カウント	----- 印刷装置 ----- 行数	ページ数	通信 入出力カウント	論理入出力 当たりの CPU	論理 入出力/秒
バッチ	18,151	1,030,253,068	18,656,603	544,032	1,531,738	.0001	95,526.4
スプール	70	1,066	14,933	369	0	.0285	.0
自動開始	56	426,047	1,692,060	41,502	178,288	.0008	39.5
COLLECTION	1	2,910	0	0	0	.0171	.2
SQL	192	3,252,232	3,519	88	0	.0003	301.5
MGMTCENTRAL	2	12,229	0	0	0	.0046	1.1
合計	18,903	1,033,969,357	20,367,115	585,991	1,713,007		
平均						.0003	95,871.0

- 平均 CPU 使用率 . . . . . : 61.0  
 CPU 1 使用率 . . . . . : 55.4  
 CPU 2 使用率 . . . . . : 57.9  
 CPU 3 使用率 . . . . . : 61.5  
 CPU 4 使用率 . . . . . : 62.2  
 CPU 5 使用率 . . . . . : 62.0  
 CPU 6 使用率 . . . . . : 60.1  
 CPU 7 使用率 . . . . . : 61.7  
 CPU 8 使用率 . . . . . : 63.1  
 CPU 9 使用率 . . . . . : 55.4  
 CPU 10 使用率 . . . . . : 56.0  
 CPU 11 使用率 . . . . . : 59.9  
 CPU 12 使用率 . . . . . : 60.6  
 CPU 13 使用率 . . . . . : 60.9  
 CPU 14 使用率 . . . . . : 62.5  
 CPU 15 使用率 . . . . . : 63.7  
 CPU 16 使用率 . . . . . : 64.1  
 CPU 17 使用率 . . . . . : 54.7  
 CPU 18 使用率 . . . . . : 57.3  
 CPU 19 使用率 . . . . . : 59.8  
 CPU 20 使用率 . . . . . : 60.6  
 CPU 21 使用率 . . . . . : 61.6  
 CPU 22 使用率 . . . . . : 62.9  
 CPU 23 使用率 . . . . . : 63.9  
 CPU 24 使用率 . . . . . : 64.7  
 CPU 25 使用率 . . . . . : 57.0  
 CPU 26 使用率 . . . . . : 55.2  
 CPU 27 使用率 . . . . . : 66.2  
 CPU 28 使用率 . . . . . : 61.1  
 CPU 29 使用率 . . . . . : 62.4  
 CPU 30 使用率 . . . . . : 63.2  
 CPU 31 使用率 . . . . . : 66.2  
 CPU 32 使用率 . . . . . : 66.4

合計 CPU 使用率 (対話型機能) . . . . . : .0  
 合計 CPU 使用率 (データベース機能) . . . . : 51.6

**Sample Component Report - ジョブ作業負荷活動:**

構成要素報告書のジョブ作業負荷活動セクションは、各ジョブごとのトランザクション合計数、時間当たりトランザクション数、平均応答時間、ディスク操作の回数、通信操作の回数、PAG 不在の回数、算術オーバーフローの回数、および永続書き込みの回数を示します。

報告書ヘッダーに表示される値は、収集開始時に QAPMCONF ファイルから獲得した構成メトリックを反映します。この値は、論理区画構成での動的変更のために、収集期間内に各間隔ごとに変更することがあります。

構成要素報告書  
ジョブ作業負荷活動

10/02/03 17:12:15  
ページ 6

Perf data from 14:00 to 16:00 at 1 min

メンバー . . . : Q275140000 型式/製造番号 . . : 890/10-3907F 主記憶域 . . . . . : 56.4 GB 開始 . . . . . : 10/02/03 14:00:00  
ライブラリー . . : PTLIBV5R3 システム名 . . . . : ABSYSTEM バージョン/リリース : 5/ 3.0 停止 . . . . . : 10/02/03 16:00:00  
区画 ID . . . . : 003 機能コード . . . . : 7427-2498-7427 対話式しきい値 . . : .00 %  
仮想プロセッサ: 4 処理装置 . . . . . : 4.0

ジョブ名	ユーザー名 / スレッド	ジョブ番号	T Y P	CPU UTIL	DB 能 UTIL	トランザクション / 時間	トランザクション / 時間	応答	--- ディスク入出力 ---			通信入出力	ページフォールト	算術 OVRFLW	永続書出
									同期	非同期	論理				
ADMIN	QTMHHTP	955725	B 02 25	.02	.0	0	0	.000	14771	615	0	0	0	0	2787
ADMIN	QTMHHTP	955727	B 02 25	.00	.0	0	0	.000	24	0	0	0	0	0	2
ADMIN	QTMHHTP	955728	B 02 25	.00	.0	0	0	.000	0	0	165	0	0	0	0
ADMIN	QTMHHTP	956347	B 02 25	.14	.0	0	0	.000	959	343	1349	0	0	0	736
AMQALMPX	QMOM	955751	B 02 35	.00	.0	0	0	.000	0	0	0	0	0	0	0
AMQPCSEA	QMOM	955757	B 02 35	.00	.0	0	0	.000	0	0	0	0	0	0	0
AMQRMPA	QMOM	955773	B 02 35	.01	.0	0	0	.000	14	0	2	0	0	0	0
AMQRRMFA	QMOM	955752	B 02 35	.00	.0	0	0	.000	1	0	0	0	0	0	0
AMQZMAA	QMOM	955753	B 02 35	.00	.0	0	0	.000	0	0	0	0	0	0	0
AMQZLAA0	QMOM	955755	B 02 20	.02	.0	0	0	.000	7	0	0	0	0	0	0
AMQZLAA0	QMOM	955774	B 02 20	.00	.0	0	0	.000	2	0	0	0	0	0	0
AMQZXMA0	QMOM	955749	B 02 20	.00	.0	0	0	.000	1	0	0	0	0	0	0
CFINT01		L 01 00	.26	.0	0	0	0	.000	0	0	0	0	0	0	0
CFINT02		L 01 00	.06	.0	0	0	0	.000	0	0	0	0	0	0	0
CFINT03		L 01 00	.08	.0	0	0	0	.000	0	0	0	0	0	0	0
CFINT04		L 01 00	.08	.0	0	0	0	.000	0	0	0	0	0	0	0
CFINT05		L 01 00	.00	.0	0	0	0	.000	0	0	0	0	0	0	0
CFINT06		L 01 00	.00	.0	0	0	0	.000	0	0	0	0	0	0	0
COLDQT		L 01 02	.00	.0	0	0	0	.000	0	0	0	0	0	0	0
CPUTEST	WLCPU	953645	B 02 51	.00	.0	0	0	.000	0	0	0	0	0	0	0
CPUTEST	WLCPU	953647	B 02 51	.00	.0	0	0	.000	0	0	0	0	0	0	0
CPUTEST	WLCPU	953648	B 02 51	.00	.0	0	0	.000	0	0	0	0	0	0	0
CPUTEST	WLCPU	953649	B 02 51	.00	.0	0	0	.000	0	0	0	0	0	0	0
CPUTEST	WLCPU	953650	B 02 51	.00	.0	0	0	.000	0	0	0	0	0	0	0

- ジョブ名 -- ジョブ名
- ユーザー名/スレッド -- ユーザー名または2次スレッドID
- ジョブ番号 -- ジョブ番号
- TYP -- ジョブ・タイプ
- PL -- ジョブが実行されたプール
- PTY -- ジョブの優先順位
- CPU UTIL -- ジョブによって使用された使用可能 CPU 時間のパーセント。これはすべてのプロセッサの平均値です。
- DB CPB UTIL -- ジョブによってデータベース処理を実行するために使用されたデータベース機能のパーセント
- TNS -- ジョブのトランザクション合計数
- TNS/ 時間 -- 1 時間当たりのトランザクション数
- RSP -- 平均対話式トランザクション応答時間 (秒数)
- 同期ディスク入出力 -- 同期ディスク操作の回数 (読み取りおよび書き出し)
- 非同期ディスク入出力 -- 非同期ディスク操作の回数 (読み取りおよび書き出し)
- 論理ディスク入出力 -- 論理ディスク操作回数 (GET, PUT, UPD, その他)
- 通信入出力 -- 通信操作回数 (GET, PUT)
- ページ不在 -- 処理アクセス・グループを含むページ不在の数
- 算術オーバーフロー -- 算術オーバーフロー例外の数
- 永続書き出し -- 永続的書き出しの数

カラム	TOTAL	平均
CPU 使用率		98.740 *
DB 能力使用率		82.3
トランザクション	2,099	
トランザクション/時間	1,043	
応答		1.610
同期ディスク入出力	304,001	
非同期ディスク入出力	1,906,898	
論理ディスク入出力	6,257,174	
通信入出力	0	
ページ不在	0	
算術オーバーフロー	3	
永続書き出し	1,980,564	

\* ----- 選択した間隔での合計経過時間に基づいた平均

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。



## このタスクについて

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。

## 関連情報

パフォーマンス用の CL コマンド

### パフォーマンス報告書の欄:

各報告書には、情報の欄がありますここではその情報を説明します。

>8.0 (構成要素) 応答時間が 8 秒を超えた回数

#### % Write Cache Overruns

(構成要素) 収集間隔中の書き込みキャッシュ・オーバーランのパーセント。

#### ----- (pgmname(プログラム名))

(トランザクション) トランザクション合計の記録。たとえば、----- QUYLIST。この報告書の行は、ジョブが活動状態から待機状態へのトランザクションを持つ度に示されます。合計は、トランザクションの RSP\* (応答時間)、CPU 秒数、および入出力カウントについて作成されます。

#### A-I WAIT /TNS (A-I 待機/TNS)

(トランザクション) トランザクション当たりの活動状態から不適格状態への待ちの平均時間 (秒数)。この値が高い場合は、多くの対話式ジョブのタイム・スライス値の設定が低すぎるのが原因と考えられます。タイム・スライス値を大きくすることを考慮してください。

#### ABORTS RECD (打ち切り受信数)

(リソース間隔) 受信したフレームに HDLC 打ち切り標識が含まれていたフレームの数。これは、フレームが完了する前に遠隔装置がフレームを終了したことを示します。

**ACT JOBS (活動ジョブ数)**

(ジョブ間隔) 選択されたジョブ (報告書のセクションに応じて対話式または非対話式) で間隔中に活動状態だったジョブの数

**ACT LEVEL (活動レベル)**

(構成要素) 開始プールの活動レベル

**ACT LVL (活動レベル)**

(システム、プール間隔) 活動レベル。プール間隔報告書のプール活動セクションの場合、間隔でのプールの活動レベル。システム報告書の記憶域プール利用率セクションの場合、最初のサンプル間隔時の活動レベル。

**ACT-INEL (活動 - 不適合)**

(システム、構成要素) 活動状態から不適合状態へのジョブの状態の移行の 1 分当りの平均回数

**ACT-WAIT (活動 - 待機)**

(システム、構成要素) このプールに割り当てられた処理による活動状態から待機状態への 1 分当りの移行回数

**ACTIVE (活動状態)**

(ジョブ・トレース) ジョブが処理中であった時間。

**ACTIVE DEVICES (活動装置)**

(システム) 回線上の活動装置の平均数

**ACTIVE DISPLAY STATIONS (LOCAL OR REMOTE) (活動表示装置数 (ローカルまたは遠隔))**

(システム) 測定期間においてトランザクションを入力するローカルまたは遠隔表示装置の数

**ACTIVE JOBS (活動ジョブ数)**

(トランザクション) 間隔において活動状態だった対話式ジョブの数

**ACTIVE JOBS PER INTERVAL (間隔当り活動ジョブ数)**

(システム) サンプル間隔に活動状態であったこのタイプのジョブの平均数

**ACTIVE K/T /TNS (活動 K/T /TNS)**

(トランザクション) 活動ワークステーション (AWS の推定値のところで説明) の平均考慮時間および入力時間 (または 1 つのトランザクションの終了と次のトランザクションの開始との間の遅延時間) の秒数。活動 K/T /TNS 遅延時間は、600 秒を超える遅延時間がすべて 600 秒に丸められるという点が入力/思考 /TNS 遅延時間と異なります。この手法は、偶発的ユーザー (断続的に作業を行うか、長期間ワークステーションを離れるユーザー) が活動ワークステーション数の推定値に及ぼす影響を軽減するために使用されます。

**ACTIVE WRK STN (活動ワークステーション)**

(リソース間隔) 活動状態のワークステーションの数

**ACTIVE/RSP (活動/応答)**

(トランザクション) トランザクション処理中に、活動レベルを保持していたジョブが (待機または活動状態で) 費やす時間

**ACTIVITY LEVEL (活動レベル)**

(システム) 対話式ジョブ活動が実行されていたすべての対話式プールの活動レベルの合計

**ACTIVITY LEVEL TIME (活動レベル時間)**

(トランザクション) 活動、短時間待機 での待機、および占有/競合 (占有競合) での待機に費やされたトランザクション時間の内訳。短時間待機 および占有競合 時間は、活動レベル時間 に含ま

れます。活動レベル・スロットはこれらの時間においても活動しているためです。占有競合時間は活動レベル時間に含まれ、待機時間のようにトランザクション/応答時間を得るために加算されることはありません。

**ARITH OVRFLW (算術オーバーフロー)**

(構成要素、ジョブ間隔) 間隔中に選択された対話式ジョブで起こった算術オーバーフロー例外の数

**ASP ID (補助記憶域プール識別コード)**

(システム、リソース間隔) 補助記憶域プール識別コード

**ASP RSC NAME (補助記憶域プール・リソース名)**

(システム、リソース) 収集時にディスク装置が割り振られた先の ASP リソース名を識別します。

**ASYNCR (非同期)**

(システム、構成要素、トランザクション、ジョブ間隔) 間隔中に選択された対話式ジョブによって開始された非同期ディスク入出力操作の回数。入出力操作を開始したジョブは、入出力操作が完了するのを待たずに、処理を継続することができます。入出力操作は、背景のシステムのテストによって完了します。

**ASYNCR DIO /TNS (非同期 DIO /TNS)**

(トランザクション) 非同期 DB 読み取り、DB 書き込み、NDB 読み取り、および NDB 書き込み要求の平均回数の合計 (ジョブに関するトランザクションごとの非同期入出力要求の平均回数)

**ASYNCR DISK I/O (非同期ディスク入出力)**

(システム、構成要素、トランザクション) トランザクションごとの非同期ディスク入出力操作の回数

**ASYNCR DISK I/O PER SECOND (非同期ディスク入出力/秒)**

(構成要素) 1 秒当りの非同期ディスク入出力操作の平均回数

**ASYNCR DISK I/O REQUESTS (非同期ディスク入出力要求)**

(トランザクション) 優先順位、ジョブ・タイプ、およびプールの所定の組み合わせに関する非同期ディスク入出力要求の合計回数

**ASYNCR I/O /SEC (非同期入出力/秒)**

(ジョブ間隔) 間隔中にジョブによって開始された非同期ディスク入出力操作の 1 秒当りの平均回数。非同期ディスク入出力カウントを経過時間で割って算出されます。

**ASYNCR I/O PER SECOND (非同期入出力/秒)**

(ジョブ間隔) 間隔中に選択された非対話式ジョブによって開始された 1 秒当りの非同期ディスク入出力操作の平均回数

**ASYNCR MAX (非同期最大)**

(トランザクション) 平均 DIO/トランザクションの下にリストされる、単一のトランザクションについてそのジョブによって発生した非同期 DBR、NDBR、および WRT 入出力要求の最大数。ジョブが対話式または自動開始のジョブ・タイプでなければ、ジョブに対する合計ディスク入出力がここにリストされます。

**ASYNCR SUM (非同期合計)**

(トランザクション) 平均 DIO/トランザクションの下にリストされ、非同期 DBR、NDBR、および WRT 要求の平均の合計 (ジョブのトランザクションごとの非同期入出力要求の平均数)

**ASYNCRONOUS DBR (非同期 DBR)**

(システム、ジョブ間隔、プール間隔) 間隔中のジョブのトランザクションごとのディスクに対する非同期データベース読み取り操作の平均回数。非同期データベースの読み取りカウントを処理されたトランザクション数で割って算出されます。このフィールドは、システム中のジョブがトランザ

クシオンを処理しなかった場合は、印刷されません。システム報告書のリソース使用率セクションでは、1 秒当りの非同期データベース読み取り操作の回数です。

注: 非同期入出力操作はシステムの非同期入出力タスクによって実行されます。

#### **ASYNCHRONOUS DBW (非同期 DBW)**

(システム、ジョブ間隔) 間隔中の選択されたジョブのトランザクションごとのディスクに対する非同期データベース書き込み操作の平均回数。非同期データベースの書き込みカウントを処理されたトランザクション数で割って算出されます。このフィールドは、システム中のジョブがトランザクションを処理しなかった場合は、印刷されません。システム報告書のリソース使用率セクションでは、1 秒当りの非同期データベース読み取り操作の回数です。

注: 非同期入出力操作はシステムの非同期入出力タスクによって実行されます。

#### **ASYNCHRONOUS DISK I/O PER TRANSACTION (トランザクションごとの非同期ディスク入出力)**

(システム) 対話式トランザクションごとの非同期物理ディスク入出力操作の平均回数

#### **ASYNCHRONOUS NDBR (非同期 NDBR)**

(システム、ジョブ間隔、プール間隔) 間隔中のシステム内ジョブのトランザクションごとの非同期非データベース読み取り操作の平均回数。非同期非データベース読み取りカウントを処理されたトランザクション数で割って算出されます。このフィールドは、システム中のジョブがトランザクションを処理しなかった場合は、印刷されません。システム報告書のリソース使用率セクションでは、1 秒当りの非同期非データベース読み取り操作の回数です。

注: 非同期入出力操作はシステムの非同期入出力タスクによって実行されます。

#### **ASYNCHRONOUS NDBW (非同期 NDBW)**

(システム、ジョブ間隔、プール間隔) 間隔中のシステム内ジョブのトランザクションごとの非同期非データベース書き込み操作の平均回数。非同期非データベース書き込みカウントを処理されたトランザクション数で割って算出されます。このフィールドは、システム中のジョブがトランザクションを処理しなかった場合は、印刷されません。システム報告書のリソース使用率セクションでは、1 秒当りの非同期非データベース書き込み操作の回数です。

注: 非同期入出力操作はシステムの非同期入出力タスクによって実行されます。

#### **AVAIL LOCAL STORAGE (K) (使用可能ローカル記憶域 (K))**

(リソース間隔) IOP の空きローカル記憶域のキロバイト数

#### **AVAILABLE STORAGE (使用可能記憶域)**

(構成要素) 使用可能なローカル記憶域 (バイト数)。IOP で使用可能な主記憶装置の平均バイト数。空きローカル記憶域は、小さい部分に分割されているため、結合されていないと考えられます。

#### **AVERAGE (平均)**

(トランザクション) すべてのトランザクションについて欄に記載されている項目の平均値

#### **AVERAGE (平均)**

(ジョブ・トレース) フィールドの平均値。SEQUENCE 欄内の AVERAGE 行の項目は、収集した STRTNS および ENDTNS の対の数を示します。対話式ジョブの場合、この数はトレースがオン (デフォルトの STRTNS および ENDTNS 値が使用されていた場合) であった期間中に発生したトランザクションの数を示します。

#### **AVERAGE DISK ACTIVITY PER HOUR (平均ディスク活動/時)**

(構成要素) ディスク・アームのシーク距離を参照

**AVERAGE DIO/TRANSACTION (平均 DIO/トランザクション)**

(トランザクション) 物理ディスク入出力カウントに関する 7 つの情報欄。物理入出力は、これらの報告書の他の部分に示されている論理入出力と対比されます。論理入出力はプログラム・レベルで送られる要求で、結果として補助記憶域 (DASD) へのアクセスを起こさせることになります。物理入出力は、実際に補助記憶域へのアクセスが起こる要求を指します。

- SYNCHRONOUS DBR (同期 DBR)
- SYNCHRONOUS NDBR (同期 NDBR)
- SYNCHRONOUS WRT (同期書き込み)
- SYNCHRONOUS SUM (同期合計)
- SYNCHRONOUS MAX (同期最大)
- ASYNC SUM (非同期合計)
- ASYNC MAX (非同期最大)

**AVERAGE K PER I/O (平均 K/入出力)**

(リソース間隔) 各ディスク読み取りまたは書き込み操作で転送された平均キロバイト数

**AVERAGE PHYS I/O /SEC (平均物理入出力/秒)**

(リソース間隔) システムのすべてのディスクに対して行われた物理ディスク読み取りおよび書き込み操作の 1 秒当りの平均数

**AVERAGE READS/SEC (平均読み取り/秒)**

(リソース間隔) システムのすべてのディスクに対して行われた物理ディスク読み取り操作の 1 秒当りの平均数

**AVERAGE RESPONSE (平均応答)**

(システム) 対話式トランザクションの平均応答時間 (秒数)。合計/平均対話式応答時間には、DDM サーバーのジョブのトランザクションは含まれていません。

**AVERAGE RESPONSE TIME (平均応答時間)**

(システム) 入出力操作ごとのディスクの平均応答時間

**AVERAGE RESPONSE TIME (SECONDS) (平均応答時間 (秒))**

(システム) 平均対話式応答時間

**AVERAGE SERVICE TIME (平均サービス時間)**

(システム) 入出力操作当りの平均ディスク・サービス時間。競合がない場合に要求にかかる時間です。

**AVERAGE WAIT TIME (平均待機時間)**

(システム) 入出力操作当りの平均ディスク待ち時間。通常は、競合のため。

**AVERAGE WRITE/SEC (平均書き込み/秒)**

(リソース間隔) システムのすべてのディスクに対して行われた物理ディスク書き込み操作の 1 秒当りの平均数

**AVG CPU /TNS (平均 CPU /TNS)**

(トランザクション) 所定のカテゴリーに区分されたトランザクションごとの処理装置の平均秒数

**AVG K/T /TNS (平均 K/T /TNS)**

(トランザクション) 対話式ジョブの平均の考慮時間と入力時間、またはトランザクション境界間の遅延時間 (秒数)

**AVG LENGTH (平均長さ)**

(ロック) ロックまたは占有が保持された平均ミリ秒数

**AVG RSP (SEC) (平均応答 (秒数))**

(トランザクション) 平均トランザクション応答時間 (秒数)

**AVG RSP /TNS (平均応答/TNS)**

(トランザクション) 所定のカテゴリに区分されたトランザクションのトランザクションごとの平均応答時間 (秒数)

**AVG RSP TIME (平均応答時間)**

(構成要素) 平均トランザクション応答時間

**AVG SEC LOCKS (ロックの平均秒数)**

(トランザクション) 対話式または非対話式待機側に帰されるロックの平均長さ (秒数)

**AVG SEC SEIZES (占有の平均秒数)**

(トランザクション) 対話式または非対話式待機側に帰される占有の平均長さ (秒数)

**AVG TIME PER SERVICE (サービス当り平均時間)**

(リソース間隔) 所定の要求を処理するためにディスク・アームが使用する時間の量

**AVG UTIL (平均使用率)**

(システム、リソース間隔) リソース報告書のディスク使用率要約において、ディスクが使用中だった使用可能時間の平均パーセント。これは、システムのすべてのディスクの複合平均です。システム報告書の通信要約では、測定時間間隔中に使用された回線容量の平均パーセント。

**BATCH ASYNCHRONOUS I/O PER SECOND (バッチ非同期入出力/秒)**

(システム) バッチ処理における非同期物理ディスク入出力操作の 1 秒当りの平均回数

**BATCH CPU SECONDS PER I/O (バッチ CPU 秒/入出力)**

(システム) すべてのバッチ・ジョブが使用したシステム処理装置秒数を、1 つのバッチ・ジョブの入出力 1 回当りの値に平均した秒数

**BATCH CPU UTILIZATION (バッチ CPU 使用率)**

(構成要素) システムがバッチとみなしたジョブが使用した使用可能な処理装置時間のパーセント。

注: 複数処理装置システムの場合、これはすべての処理装置を通算した平均使用率です。

**BATCH IMPACT FACTOR (バッチ影響係数)**

(システム) モデル化の目的によるバッチ作業負荷調整

**BATCH PERMANENT WRITES PER SECOND (バッチ永続書き込み/秒)**

(システム) バッチ処理における永続書き込み操作の 1 秒当り平均回数

**BATCH SYNCHRONOUS I/O PER SECOND (バッチ同期入出力/秒)**

(システム) バッチ処理における同期物理ディスク入出力操作の 1 秒当りの平均回数

**BCPU / SYNCHRONOUS DIO (BCPU/同期 DIO)**

(トランザクション) 同期ディスク入出力操作ごとのバッチ処理装置の平均秒数

**BIN (トランザクション) 2 進オーバーフロー例外の数****BINARY OVERFLOW (2 進オーバーフロー)**

(構成要素) 1 秒当りの 2 進オーバーフローの回数

**BMPL - CUR AND INL (BMPL - CUR および INL)**

(トランザクション) 現在活動レベルにあるジョブの数 (現行のマルチプログラミング・レベルの開始)、およびジョブが待機状態でなくなった (トランザクションの開始) ときにジョブが実行していた記憶域プールの不適格待ち行列にあるジョブの数 (不適格のマルチプログラミング・レベルの開始)

注: マルチプログラミング・レベル (MPL) は活動レベルと互換的に使用されます。

**BUNDLE WAIT COUNT (バンドル待機数)**

(構成要素) ジャーナル・バンドルがディスクに書き込まれるのを待機するタスクおよびジョブの合計数。

**BUNDLE WAIT PCT (バンドル待機パーセント)**

(構成要素) ジャーナル・バンドルがディスクに書き込まれるのを待機する際にかかる時間のパーセンテージ (間隔経過時間と比較した)。

**BUNDLE WRITES SYSTEM (バンドル書き込みシステム)**

(構成要素) 内部システム・ジャーナルに対するバンドル書き込みの数。バンドル書き込みは、システムによって共に保管されるジャーナル項目のグループです。

**BUNDLE WRITES USER (バンドル書き込みユーザー)**

(構成要素) ユーザー作成ジャーナルに対するバンドル書き込みの数。バンドル書き込みは、システムによって共に保管されるジャーナル項目のグループです。

**BYTES PER SECOND RECEIVED (受信バイト/秒)**

(システム) 1 秒当りの平均受信バイト数

**BYTES PER SECOND TRANSMITTED (送信バイト/秒)**

(システム) 1 秒当りの平均送信バイト数

**BYTES RECD PER SEC (秒当たり受信バイト数)**

(リソース間隔) 1 秒当りの平均受信バイト数

**BYTES TRNSMITD PER SEC (秒当たり送信バイト数)**

(リソース間隔) 1 秒当りの平均送信バイト数

**CATEGORY (カテゴリー)**

(トランザクション) 分類されたトランザクションのグループ。対話式トランザクション・カテゴリーによる分析では、トランザクションは処理装置モデルによって分類されます。トランザクションの区切りに使用される境界値は *AVG CPU /TNS* (平均 CPU/TNS) の欄に示されます。対話式応答時間による分析では、トランザクションはその応答時間によって分類されます。対話式入力/考慮時間による分析では、トランザクションは入力/考慮時間によって分類されます。

**CACHE HIT STATISTICS (キャッシュ・ヒット統計)**

(構成要素) 次の項目を含むキャッシュ使用状況に関する統計データ

- アームごとの装置キャッシュ読み取りヒットのパーセント
- アームごとの制御装置キャッシュ読み取りヒットのパーセント
- 書き込みキャッシュの効率を表すパーセント

**DEVICE READ (装置読み取り)**

装置読み取りは、装置キャッシュ読み取りヒット (DSDCRH) 数を装置読み取り操作 (DSDROP) 数で割った値をパーセントで示します。

**CONTROLLER READ (制御装置読み取り)**

制御装置読み取りは、制御装置キャッシュ読み取りヒット (DSCCRH) 数を読み取りコマンド (DSRDS) 数で割った値をパーセントで示します。

**WRITE EFFICIENCY (書き込み効率)**

書き込み効率は、書き込みコマンド (DSWRTS) 数と装置書き込み操作 (DSDWOP) 数の差を書き込みコマンド (DSWRTS) 数で割った値をパーセントで示します。

**EACS READ (EACS 読み取り)**

拡張キャッシュ・シミュレーターによる読み取りヒットのパーセント

**EACS RESP (EACS 応答)**

拡張キャッシュ・シミュレーターによる応答時間短縮のパーセント

**上限あり**

(システム) 各間隔の終了時に区画が上限ありであったか上限なしであったかを示します。この列は、パフォーマンス・データを収集する i5/OS 区画についてのみ印刷されます。

**チャンネル**

(リソース間隔) IDLC 回線によって使用される B チャンネル (特殊条件)

**CMN (通信)**

(ジョブ間隔) 間隔中に選択された対話式ジョブによって実行された通信入出力操作の回数

**CMN I/O (通信入出力)**

(構成要素) 通信操作 (GET、PUT) の数

**CMN I/O PER SECOND (通信入出力/秒)**

(ジョブ間隔) 間隔中に選択された非対話式ジョブによって実行された通信入出力操作の 1 秒当りの平均回数

**COLLISION DETECT (衝突検出)**

(リソース間隔) 端末装置 (TE) が、同じバスを使用しようと試みた別の TE によって壊された送信フレームを検出した回数

**COMMIT OPS (コミット操作)**

(構成要素) 実行されるコミット操作。アプリケーションおよびシステム提供の参照保全コミットを含みます。

**COMMUNICATIONS I/O COUNT (通信入出力カウント)**

(システム) 通信入出力操作の回数

**COMMUNICATIONS I/O GET (通信入出力 GET)**

(システム) トランザクションごとの通信 GET 操作の回数

**COMMUNICATIONS I/O PUT (通信入出力 PUT)**

(システム) トランザクションごとの通信 PUT 操作の回数

**COMMUNICATIONS LINES (通信回線)**

(システム、構成要素、ジョブ間隔、プール間隔) 報告書選択基準では、含める (SLTLINE パラメーター) または除外する (OMTLNE パラメーター) ために選択された通信回線のリスト。これらは、ユーザーが指定する通信回線名です。

**CONTROL UNITS (制御装置)**

(システム、構成要素、ジョブ間隔、プール間隔) 含める (SLTCTL パラメーター) または除外する (OMTCTL パラメーター) ことにより選択された制御装置のリスト。これらは、ユーザーが指定する制御装置名です。

**COUNT (カウント)**

(トランザクション、ロック) の欄の項目の発生数。たとえば、ロック報告書では、発生したロックまたは占有の数

**CPU (トランザクション) 所定の優先順位でジョブが使用した処理装置合計秒数****CPU (ジョブ・トレース) このトレース項目に使用された CPU の概算。これは使用された時間と実行した CPU モデルに基づいて計算される値です。**



**CPU /TNS**

(トランザクション、ジョブ間隔) トランザクションごとの処理装置の使用可能時間の長さ (秒数)

**CPU MODEL (CPU モデル)**

(システム) 処理装置のモデル番号

**CPU PER I/O ASYNC (非同期入出力当り CPU)**

(システム) 非同期入出力当りの CPU 使用

**CPU PER I/O SYNC (同期入出力当り CPU)**

(システム) 同期入出力当りの CPU 使用

**CPU PER LOGICAL I/O (論理入出力当り CPU)**

(システム) 各論理ディスク入出力操作で使用された処理装置時間

**CPU QM**

(トランザクション) 単純な処理装置のキューイング乗数

**CPU SEC (CPU 秒数)**

(トランザクション) この状態でジョブが使用した処理装置時間

**CPU SEC /SYNC DIO (CPU 秒数/同期 DIO)**

(トランザクション) ジョブの各タイプごとの CPU 秒数を同期ディスク入出力要求の数で割った率

**CPU SEC AVG AND MAX (CPU 秒数 - 平均および最大)**

(トランザクション) ジョブのトランザクションごとの平均処理装置時間、およびジョブでトランザクションに使用した最大処理装置時間。ジョブが対話式または自動開始のジョブ・タイプでない場合、ジョブに対する合計の処理装置時間だけが MAX の欄の見出しの下にリストされます。

**CPU SEC PER TNS (CPU 秒数/TNS)**

(トランザクション) トランザクションごとの処理装置時間

**CPU SECONDS (CPU 秒数)**

(システム、トランザクション、構成要素) トランザクションごとに使用された処理装置平均秒数。システム要約データでは、追跡期間中にジョブによって使用された使用可能な処理装置合計秒数です。優先順位 - ジョブ・タイプ - プール統計では、優先順位、ジョブ・タイプ、およびプールの所定の組み合わせでジョブが使用した処理装置合計秒数です。バッチ・ジョブ分析では、ジョブが使用した処理装置の使用可能時間の長さ (秒数) です。並行バッチ・ジョブ統計では、ジョブ・セット中のジョブが使用した処理装置の使用可能時間の長さ (秒数) です。

**CPU SECONDS (CPU 秒数)**

(ジョブ・トレース) トランザクションに使用された処理装置時間の概算値。

**CPU SECONDS PER TRANSACTION (CPU 秒/トランザクション)**

(システム) トランザクションごとの処理装置の平均秒数

**CPU UTIL (CPU 使用率)**

(システム、構成要素、トランザクション、ジョブ間隔、プール間隔、バッチ・ジョブ・トレース) 使用された使用可能な処理装置時間のパーセント。複数処理装置システムの場合、これは合計使用率を処理装置数で割った値です。

**CPU UTIL PER TRANSACTION (CPU 使用率/トランザクション)**

(構成要素) CPU 使用率をジョブのトランザクション総数で割った値です。

**CPU UTILIZATION (BATCH) (CPU 使用率 (バッチ))**

バッチ・ジョブによって使用される使用可能な CPU 時間のパーセント。これは、すべてのプロセッサの平均です。

### **CPU UTILIZATION (INTERACTIVE) (CPU 使用率 (対話式))**

対話式ジョブによって使用される使用可能な CPU 時間のパーセント。これは、すべてのプロセッサの平均です。

### **CPU UTILIZATION (TOTAL) (CPU 使用率 (合計))**

対話式ジョブおよびバッチ・ジョブによって使用される使用可能な CPU 時間のパーセント。これは、すべてのプロセッサの平均です。

注: 上限のない (uncapped) 区画では、CPU 使用率の合計は 100 パーセントを超えることがあります。

### **CPU/ASYNC I/O (CPU/非同期入出力)**

(ジョブ間隔) 各非同期ディスク入出力操作に要した処理装置時間の平均ミリ秒数。これは、ジョブが使用した処理装置時間 (ミリ秒数) を非同期ディスク入出力カウントで割って算出されます。

### **CPU/SYNC I/O (CPU/同期入出力)**

(ジョブ間隔) 各同期ディスク入出力操作に要した処理装置時間の平均ミリ秒数。これは、ジョブが使用した処理装置時間 (ミリ秒数) を同期ディスク入出力カウントで割って算出されます。

### **CPU/TNS**

(トランザクション) 間隔中のジョブのトランザクションごとの平均処理秒数。これは、使用された処理装置時間の合計を、処理されたトランザクションの数で割って算出されます。

### **CPU/TNS (SEC) (CPU/TNS (秒数))**

(トランザクション) トランザクションごとの処理装置の秒数

### **CTL (構成要素) 制御装置識別コード**

### **CUM CPU UTIL (累積 CPU 使用率)**

(トランザクション) 所定のカテゴリより小さいか等しいトランザクションごとの平均応答時間をもつトランザクションが使用した、処理装置の使用可能時間の累積パーセント。たとえば、合計追跡期間の全ジョブ優先順位別 CPU (システム要約データ) では、所定の優先順位より高いか等しい優先順位でジョブが使用した処理装置時間です。

### **CUM PCT TNS (累積 PCT TNS)**

(トランザクション) トランザクションごとの累積 CPU パーセント。システム要約データでは、所定のカテゴリより小さいか等しいトランザクションごとの平均応答時間をもつすべてのトランザクションの累積 CPU パーセントです。対話式プログラム・トランザクション統計では、リストされたプログラムによるすべてのトランザクションの累積 CPU パーセントです。ジョブ統計セクションでは、リストされたジョブによる合計トランザクションの累計 CPU パーセントです。対話式プログラム統計セクションでは、リストされたプログラムによるすべてのトランザクションの累積 CPU パーセントです。

### **CUM UTIL (累計使用率)**

(システム) 累積 CPU 使用率 (現在高)

注: これは個々のジョブから取られ、作業負荷ページの処理装置使用率の合計とわずかに異なる場合があります。

### **CUR INL MPL (現行不適合 MPL)**

(トランザクション) 記憶域プール内で活動レベルを待機している (不適合) ジョブの数

### **CUR MPL (現行 MPL)**

(トランザクション) 記憶域プールで活動レベルを保持しているジョブの数

## 現行ユーザー

(ジョブ) 各間隔の最後でジョブが実行していたユーザーの名前。

## DASD OPS/SEC (DASD 操作回数/秒)

(構成要素) 1 秒当りのディスク操作回数

## DASD OPS PER SEC READS (DASD 読み取り操作回数/秒)

(リソース) 1 秒当りの読み取り回数

## DASD OPS PER SEC WRITES (DASD 書き込み操作回数)

(リソース) 1 秒当りの書き込み回数

## DATAGRAMS RECEIVED (受信データグラム)

(構成要素) インターフェースから受信された入力データグラムの合計数。この数には、誤って受信されたものも含まれています。

**DB** (ジョブ・トレース) 項目に対し発生した、物理データベースの読み取り回数。

## DB CPB UTIL (DB CPB 利用率)

(構成要素) データベース処理の実行に使用されるデータベース機能のパーセント

## DB FAULT (DB 不在)

(システム、構成要素) データベースのページ不在の 1 秒当りの平均回数

## DB PAGES (DB ページ)

(システム、構成要素) 読み取られたデータベース・ページの 1 秒当りの平均数

## DB READ (DB 読み取り)

(トランザクション) 物理入出力カウント欄にリストされている場合は、ジョブがその状態にあった間のデータベース読み取り要求の数。同期ディスク入出力要求 /TNSの欄にリストされている場合は、トランザクションごとの同期データベース読み取り要求の平均数。

## DB READS ((DB 読み取り)

(ジョブ・トレース) 物理データベースに発生した読み取り回数。

## DB WRITE (DB 書き込み)

(トランザクション) 同期ディスク入出力要求 /TNSの欄にリストされている場合は、トランザクションごとの同期データベース書き込み要求の平均回数

## DB WRT (DB 書き込み)

(トランザクション) 物理入出力カウント欄にリストされている場合は、ジョブがその状態にあった間のデータベース書き込み要求の回数。同期ディスク入出力カウント欄にリストされている場合は、トランザクションごとの同期データベース書き込み要求の回数。

## DDM I/O (DDM 入出力)

(構成要素、ジョブ間隔) 分散データ管理 (DDM) サーバー・ジョブの論理データベース入出力操作の数

## DDM SVR WAIT /TNS (DDM SVR 待機 /TNS)

(トランザクション) トランザクションごとに、ターゲット・システムがデータの要求に対して応答するのを待つために、ソースの分散データ管理 (DDM) サーバー・ジョブが費やした平均時間 (秒数)。この値には、回線時間およびターゲット・システムがデータの要求に応答するのに費やした時間が含まれます。

**DEC** (トランザクション) 10 進オーバーフロー例外の数

**DECIMAL DATA (10 進データ)**

(構成要素) 1 秒当りのデータ例外カウント。データ例外は、無効なデータが演算命令によって検出された場合に起こります。10 進命令での無効な符号または数字コード、または乗算命令での左端のゼロの不足などがその例です。

**DECOMMIT OPS (コミット解除操作)**

(構成要素) 実行されるコミット解除操作。アプリケーションおよびシステム提供の参照保全コミット解除を含みます。

**DECIMAL OVERFLOW (10 進オーバーフロー)**

(構成要素) 1 秒当りの 10 進オーバーフローの数

説明 (構成要素) 例外タイプの詳細な説明

**DETECTED ACCESS TRANSMISSION ERROR (DTSE) In (検出済みアクセス伝送エラー (DTSE) イン)**

(リソース間隔) ネットワーク終端装置 1 (NT1) 終点が、端末装置 (TE) に対して、回線伝送終端装置 (LT) から NT1 終点への ISDN U インターフェースを通るデータのエラーを通知した回数。NT1 終点は、保守チャネル S1 を介して TE へエラーを報告します。

**DETECTED ACCESS TRANSMISSION ERROR (DTSE) OUT (検出済みアクセス伝送エラー (DTSE) アウト)**

(リソース間隔) ネットワーク終端装置 1 (NT1) 終点が、端末装置 (TE) に対して、NT1 終点から LT への ISDN U インターフェースを通るデータのエラーを通知した回数。NT1 終点は、保守チャネル S1 を介して TE へエラーを報告します。

**DEVICE (装置)**

(構成要素) 装置識別コード

**DIO/SEC ASYNC (DIO/秒 - 非同期)**

(システム) 1 秒当り非同期入出力操作の回数

**DIO/SEC SYNC (DIO/秒 - 同期)**

(システム) 1 秒当り同期入出力操作の回数

**DISK ARM SEEK DISTANCE (ディスク・アームのシーク距離)**

(構成要素) 1 時間当りの平均シーク距離の分布

- 0 シーク距離がゼロの回数
- 1/12 シーク数がディスクの 0 から 1/12 の回数
- 1/6 シーク数がディスクの 1/12 から 1/6 の回数
- 1/3 シーク数がディスクの 1/6 から 1/3 の回数
- 2/3 シーク数がディスクの 1/3 から 2/3 の回数
- >2/3 シーク数がディスクの 2/3 を超えた回数

**DISK ARMS (ディスク・アーム)**

(システム) この IOP のディスク・アームの数

**Disk Capacity (ディスクの容量)**

(構成要素) 使用済みまたは使用可能な平均ディスク・スペース量

**MB** ディスクで使用可能な 100 万バイト単位のスペース

**PERCENT (パーセント)**

ディスクで使用可能なスペースのパーセント

**DISK CONTROLLERS (ディスク制御装置)**

(システム) この IOP のディスク装置制御装置の数

**DISK FEATURE (ディスク機構)**

(システム) ディスクのタイプ (9332、9335 など)

**DISK I/O ASYNC (ディスク入出力 - 非同期)**

(システム、構成要素) 非同期ディスク入出力操作の合計回数

**DISK I/O LOGICAL (ディスク入出力論理)**

(構成要素) GET や PUT のような論理ディスク操作の回数

**DISK I/O PER SECOND (ディスク入出力/秒)**

(システム) 1 秒当りの平均物理ディスク入出力操作回数

**DISK I/O READS /SEC (ディスク入出力読み取り/秒)**

(リソース間隔) ディスク IOP による 1 秒当りのディスク読み取り操作の平均回数

**DISK I/O REQUESTS (ディスク入出力要求)**

(トランザクション) 追跡期間中にジョブによって出された同期および非同期ディスク入出力要求の合計回数

**DISK I/O SYNC (ディスク入出力同期)**

(システム、構成要素) 同期ディスク入出力操作の合計回数

**DISK I/O WRITES /SEC (ディスク入出力書き込み/秒)**

(リソース間隔) ディスク IOP による 1 秒当りのディスク書き込み操作の平均回数

**DISK IOP (ディスク IOP)**

(システム) ディスク IOP 制御装置の数

**DISK MIRRORING (ディスク・ミラーリング)**

(システム) ディスク・ミラーリングが活動状態であるかどうかを示します。

**DISK SPACE USED (使用ディスク・スペース)**

(リソース間隔) システム全体で使用された合計ディスク・スペース (ギガバイト数)

**DISK TRANSFER SIZE (KB) (ディスク転送サイズ (KB))**

(システム) ディスク操作ごとに転送された平均キロバイト数

**DISK UTILIZATION (ディスク使用率)**

(システム) ディスク・アームが入出力操作を行っていた時間間隔の比率

**DISK CPU UTIL (ディスク CPU 使用率)**

(システム、リソース間隔) ディスク装置によって使用された CPU 使用率

**DTGM REQ TRANSM DSCRD (データグラム要求送信廃棄)**

(構成要素) 次の理由によって廃棄される IP データグラムのパーセント

- データグラムを宛先に送信するための経路が検出されなかった
- バッファ・スペースの不足

**DTGM REQ FOR TRANSM TOT (送信されるデータグラム要求の合計)**

(構成要素) ローカル IP ユーザー・プロトコルが送信を要求する際に IP に提供した IP データグラムの合計数

**ELAPSED SECONDS (経過秒数)**

(トランザクション、構成要素) 経過時間 (秒数)。トランザクション報告書のバッチ・ジョブ分析セッションでは、ジョブの開始時から終了時まで経過した秒数。トランザクション報告書の並行バッチ・ジョブ統計では、ジョブ・セット内のすべてのジョブの合計経過時間。

**ELAPSED TIME (経過時間)**

(ジョブ間隔) 間隔中にジョブが存在していた時間の長さ (分および秒)。これは間隔の長さと同じですが、ジョブが間隔中に開始または終了した場合は、間隔の長さより短くなります。

**ELAPSED TIME - SECONDS (経過時間 (秒))**

(トランザクション) 以下の欄でジョブによって費やされた時間を示します。

**LONG WAIT (長時間待機)**

次のトランザクションの待機またはロック待機時間などの状態で費やした経過時間

**ACTIVE/RSP (活動/応答)**

トランザクション処理中に活動レベルを保持し (待機または活動状態で)、ジョブが費やした時間。これは、トランザクション終了時 (トランザクション合計行) に、ロックによって起こった長期待機と不適格状態において活動レベルでのトランザクションの処理でジョブが費やした時間です。

**INEL WAIT (不適格待機)**

ジョブが活動レベルを待機していて不適格待機状態で費やした時間

**EM3270 WAIT /TNS (EM3270 待機/TNS)**

(トランザクション) トランザクションごとに、システム・ネットワーク体系 (SNA) ホスト・システム通信および 2 進データ同期通信 (BSC) 3270DE を待機して費やした時間の平均 (秒数)。エミュレーション・プログラムが表示装置またはホスト処理装置と通信しているかどうかを判別するためには、プログラム論理が必要になります。事象待機処理にも要件があるために、移行のすべての組み合わせを検出できるとは限りません。

**ENTRY**

(ジョブ・トレース) プログラムが制御を渡された場所での、プログラムの命令。

**EORn** (トランザクション) 待機欄にリストされ、トランザクション n の応答時間の終わりを示します。これらのコードは、待機コード欄に示されますが、待機コードではありません。これらはトランザクション境界追跡レコードを示します。

**EOTn** (トランザクション) 待機欄にリストされ、タイプ n のトランザクションの移行の終わりを示します。これらのコードは、待機コード欄に示されますが、待機コードではありません。これらはトランザクション境界追跡レコードを示します。

**ESTIMATED EXPOSUR AP NOT JRNLD (アクセス・パスが未処理の場合の見積時間)**

(構成要素) アクセス・パスがシステムによってジャーナル処理されていない場合、システムが見積もったアクセス・パス回復にかかる時間 (分)

**ESTIMATED EXPOSUR CURR SYSTEM (現行システムによる見積時間)**

(構成要素) システムが見積もったアクセス・パスの回復にかかる時間 (分)

**EST OF AWS (AWS の推定値)**

(トランザクション) 追跡期間または間隔中の活動ワークステーション数の推定値。600 秒を超える遅延時間は、600 秒に丸められます。この手法は、偶発的ユーザー (断続的に作業を行うか、長期間ワークステーションを離れるユーザー) が活動ワークステーション数の推定値に及ぼす影響を軽減するために使用されます。

**EVENT WAIT /TNS (事象待機/TNS)**

(トランザクション) トランザクションごとの事象待機時間の平均時間 (秒数) システムで実行されているジョブが行う要求は多くの場合、非同期ジョブになります。これらの非同期ジョブは、事象を用いて要求の完了を要求側に知らせます。事象待機時間は、要求側ジョブがこの信号を待つ時間です。

**EVT** (トランザクション) Wait Code (待機コード) の欄にリストされる事象待機。メッセージ・キューでの待機時に起こる長時間待機です。

#### **EXCEPTION TYPE (例外タイプ)**

(構成要素) 内部マイクロプログラム命令手順で実行中の内部マイクロプログラム命令の結果であるプログラム例外のタイプ。これらの例外はシステムの低レベルでモニターされるため、これを特定のエンド・ユーザー操作と関連付けることは困難です。このカウントは、それらの処理に要する処理装置時間がシステムのパフォーマンスに影響を与える場合に意味をもちます。このカウントに変動がある場合は、パフォーマンスに影響を与えるシステム変更を意味します。たとえば、占有カウントまたはロック・カウントに大きな変動があれば、ジョブ・スケジューリングに問題があるか、または同じリソースを使用する古いアプリケーションと新しいアプリケーションの間に競合があることを示します。

注: 占有カウントおよびロック・カウントを知るには、パフォーマンス追跡開始 (STRPFTRC) コマンドを使用して追跡データを収集する必要があります。トランザクション報告書印刷 (PRTTNSRPT) を実行して、ロックを保持しているオブジェクトおよびジョブをリストします。

#### **EXCEPTIONAL WAIT (例外的待機)**

(システム) トランザクションごとの例外待機時間の平均秒数。例外的待機は、内部応答時間の中で、処理装置およびディスクの使用によるものではない部分です。例外的待機は、システムの内部リソースの競合、たとえばデータベース・レコードのロックの待機などによって起こります。

#### **CONSTANT (定数)**

例外待機時間でスループットが向上しても一定の部分

#### **VARIABLE (変動)**

例外待機時間でスループットが向上すると変動する部分

#### **EXCP (例外)**

(構成要素、トランザクション) 構成要素報告書では、起こったプログラム例外の合計数です。トランザクション報告書では、この欄の Y はトランザクションに例外があったことを意味します。含まれる例外のタイプは、処理アクセス・グループ例外、および 10 進数、2 進数、ならびに浮動小数点数のオーバーフローです。トランザクションにあった例外を調べるには、移行報告書を参照してください。

#### **EXCP WAIT (例外的待機)**

(トランザクション) ジョブ・セット内のジョブの例外的待機時間の合計 (秒数)

#### **EXCP WAIT /TNS**

(トランザクション) トランザクションごとの例外的待機の平均時間 (秒数)。この値は、ジョブ・タイプ別の例外的待機内訳にリストされた待機の合計です。

#### **EXCP WAIT SEC (例外的待機秒数)**

(トランザクション) ジョブの例外的待機時間の合計量 (秒数)

#### **EXCS ACTM /TNS (超過 ACTM /TNS)**

(トランザクション) トランザクションごとの超過活動レベル時間 (たとえば、活動状態だが処理装置を使用していない場合に費やされる時間) の平均時間 (秒数)。使用可能な活動レベルが十分に、優先順位が高い対話式作業を多く行わなければならない場合は、ジョブは処理装置のサイクルの順番になるまでさらに長く待つこととなります。値が 0.3 より大きい場合は、詳細について、特定のアプリケーションに対応するジョブを見てください。これらのジョブを調べることによって、どの

アプリケーションのジョブがこの値に最も関連しているか判別することができます。詳細については、これらのジョブのトランザクション報告書および移行報告書を使用してください。超過活動レベル時間の計算式を以下に示します。

Active Time - [  
(乗数 X CPU X 開始活動レベル) +  
(同期ディスク入出力操作回数 X .010)]

注: 開始活動レベルが 1 より大きい場合、乗数は 0.5 になります。開始活動レベルが 1 以外の値の場合は、乗数は 1 になります。

**EXIT** (ジョブ・トレース) プログラムが制御を失った場所での、プログラムの命令番号。

#### **EXPERT CACHE (エキスパート・キャッシュ)**

(システム、構成要素) システムに対し、オブジェクト内のデータの参照パターンに基づいて、どのオブジェクトまたはオブジェクトの部分を共用主記憶域プールに残すべきかを判別してシステムに指示します。エキスパート・キャッシュは、システムの動的チューナーとは独立して実行される記憶管理チューナーを使用して、全体的なページング特性およびプールのヒストリーを調べます。この欄に現れる値の中には、共用プールの処理 (WRKSHRPOOL) コマンドと次のような関係があるものがあります。

- 0=\*FIXED。システムが記憶域プールのページング特性を動的には調整しないことを示しています。システムはデフォルトを使用します。
- 3=\*CALC。システムが、最適パフォーマンスのために記憶域プールのページング特性を動的に調整することを示しています。

#### **EXPOSED AP SYSTEM JOURNALED (システムでジャーナル処理されているアクセス・パス)**

(構成要素) 現在システムによってジャーナル処理されている公開アクセス・パスの数

#### **EXPOSED AP SYSTEM NOT JOURNALED (システムでジャーナル処理されていないアクセス・パス)**

(構成要素) 現在システムによってジャーナル処理されていない公開アクセス・パスの数

**/F** (システム、リソース間隔) 全二重として報告されたプロトコルの回線速度。この標識は、イーサネット (ELAN)、トークンリング (TRLAN) 回線、または非同期転送モード回線に適用されます。

#### **FAR END CODE VIOLATION (終端端末コード違反)**

(リソース間隔) T 参照点のインターフェースでネットワーク終端装置 1 (NT1) 終点に伝送されたフレームについて、NT1 終点が検出した意図しないコード違反の数。NT1 終点は、保守チャンネル S1 を介して終端装置 (TE) に違反を報告します。

#### **FAULTS (不在)**

(システム) 収集中に各ジョブ・タイプまたはジョブ優先順位ごとに発生したページ不在の合計を表す値。これは、QAPMJOBS または QAPMJOB L ファイルの JBTFLT フィールドで示されている値と同じです。

#### **ファイル**

(トランザクション) オブジェクトが入っているファイル

**Flp** (トランザクション) 浮動小数点オーバーフロー例外の数

#### **FLP OVERFLOW (FLP オーバーフロー)**

(構成要素) 1 秒当りの浮動小数点オーバーフローの回数

#### **FRAME RETRY (フレーム再試行)**

(リソース間隔) 遠隔制御装置にフレームの再送信を試みた回数



**FRAMES RECEIVED PCT ERR (受信フレーム・エラー %)**

(リソース間隔) 受信エラーのあったフレームのパーセント。ホスト・システムにエラーがあるか、あるいは十分に速く受信データを処理できない場合に、エラーが起こる可能性があります。

**FRAMES RECEIVED TOTAL (受信フレーム合計)**

(リソース間隔) 受信したフレームの合計数。エラーのあったフレームおよび有効でないフレームも含まれます。

**FRAMES TRANSMITTED PCT ERR (伝送フレーム・エラー %)**

(リソース間隔) エラーのために再送信されたフレームのパーセント

**FRAMES TRANSMITTED TOTAL (伝送フレーム合計)**

(リソース間隔) 送信されたフレームの合計数

**FULL CLS**

(ジョブ・トレース) すべてのタイプのファイルに対する、完全クローズの回数。

**FULL OPN**

(ジョブ・トレース) すべてのタイプのファイルに対する、完全オープン回数。

**FUNCTION**

(ジョブ・トレース) これは、トレース項目が記録される原因になります。起こりうるトレース項目には、以下のようなものがあります。

表 7.

機能 ID	説明
DATA	データ・トレース・レコード
CALL	外部呼び出し
XCTL	制御権移動
EVENT	イベント・ハンドラー呼び出し
EXTXHINV	外部例外ハンドラーの呼び出し
INTXHINV	内部例外ハンドラーの呼び出し
INTXHRET	内部例外ハンドラーからの戻り
INVEXIT	呼び出しエグジット
RETURN	外部への戻り
ITRMXRSG	再シグナリング例外による呼び出し終了
EXTXHRET	外部へまたはプロシージャ命令からの戻り
PTRMTPP	終了フェーズの終了
PTRMUNX	ハンドルできない例外によるプロセス終了
NOTUSED	このタイプは無効なトレース・タイプ
ITERM	呼び出し終了
CANCLINV	キャンセル呼び出し命令

**FUNCTIONAL AREAS (業務分野)**

(システム、構成要素、トランザクション、ジョブ間隔、プール間隔) 報告書選択基準では、含める (SLTFCNARA パラメーター) または除外する (OMTFCNARA パラメーター) ことにより選択された業務分野のリスト

**/H** (システム、リソース間隔) 半二重として報告されたプロトコルの回線速度。この標識は、イーサネット (ELAN)、トークンリング (TRLAN) 回線、または非同期転送モード回線に適用されます。

**HDW** (トランザクション) 待機コード欄にリストされる保留待機 (延期されたジョブまたはシステム要求)。ジョブは、報告書の次の明細行 (OBJECT --) に示されたオブジェクトに対して持っていたロックを解放しました。オブジェクトを待機していたジョブは、この行 (WAITER --) に名前が示され、そのジョブがロックの解放の待機に費やされた時間も同時に示されます。

**HIGH SRV TIME (高サービス時間)**

(リソース間隔) システムのディスク・アームの最高平均サービス時間 (秒数)

**HIGH SRV UNIT (高サービス装置)**

最高サービス時間のディスク・アーム

**HIGH UTIL (高使用率)**

(リソース間隔) 最高使用率のディスク・アームの使用パーセント

**HIGH UTIL UNIT (高使用率装置)**

(構成要素、リソース間隔) 使用率が最高のディスク・アーム

**HIGH UTILIZATION DISK (高使用率ディスク)**

(構成要素) この間隔中に最も使用率が高かったディスク・アームの使用率のパーセント

**HIGH UTILIZATION UNIT (高使用率装置)**

(構成要素) この間隔中に最も使用率の高かったディスク・アーム

**HOLDER JOB NAME (保持ジョブ名)**

(トランザクション) オブジェクトを保持していたジョブの名前

**HOLDER NUMBER (保持番号)**

(トランザクション) オブジェクトを保持していたジョブの番号

**HOLDER POOL (保持プール)**

(トランザクション) ジョブの実行中にそのジョブが入っていたプール

**HOLDER PTY (保持優先順位)**

(トランザクション) 保持ジョブの優先順位

**HOLDER TYPE (保持タイプ)**

(トランザクション) 保持ジョブのタイプおよびサブタイプ

**HOLDER USER NAME (保持ユーザー名)**

(トランザクション) オブジェクトを保持していたユーザーの名前

**HOLDER'S JOB NAME (保持ジョブ名)**

(ロック) ロックを保持しているジョブの名前

**I FRAMES RECD PER SEC (受信 I フレーム/秒)**

(リソース間隔) 受信された 1 秒当りの情報フレーム数

**I FRAMES TRNSMITD PER SEC (送信 I フレーム/秒)**

(リソース間隔) 送信された 1 秒当りの情報フレーム数

**I/O WAIT (入出力待機)**

(リソース間隔) 入出力要求が処理可能な状態にありながら、ディスク・アームがまだその要求を実行できない状態にある時間の量

**ICMP MESSAGES ERROR (ICMP メッセージ・エラー)**

(構成要素) エンティティが受信したものの、メッセージにエラーがあることを判別したか、問題があるためにエンティティが送信しなかった Internet Control Message Protocol (ICMP) メッセージの数

**ICMP MESSAGES RECEIVED (受信 ICMP メッセージ)**

(構成要素) エンティティーが受信した Internet Control Message Protocol (ICMP) メッセージの合計数

**ICMP MESSAGES SENT (送信 ICMP メッセージ)**

(構成要素) エンティティーが送信した Internet Control Message Protocol (ICMP) メッセージの合計数

**INCOMING CALLS PCT RETRY (着呼再試行 %)**

(リソース間隔) ネットワークに拒否された着呼のパーセント

**INCOMING CALLS TOTAL (着呼の合計)**

(リソース間隔) 着呼を試みた合計回数

**INEL TIME A-I/W-I (INEL 時間 A-I/W-I)**

(トランザクション) タイム・スライス終了から (活動状態から不適格状態へ) または待機状態から (待機状態から不適格状態へ) の不適格状態でジョブが費やした時間の量

**INEL WAIT (不適格待機)**

(トランザクション) 「経過時間 - 秒数」の欄にリストされ、ジョブが不適格状態で活動レベルの待機に費やした時間

**INT FEAT UTIL (対話式機能使用率)**

(構成要素) 全ジョブによって使用される対話式機能のパーセント

**INTER CPU UTILIZATION (INTER CPU 使用率)**

(構成要素) システムが対話式とみなしたジョブが使用した使用可能な処理装置時間のパーセント。

注: 複数処理装置システムの場合、これはすべての処理装置を通算した平均使用率です。

**INV** (ジョブ・トレース) プログラムの呼び出しレベル。

**IOP** (構成要素) 通信 IOP、DASD IOP、ローカル・ワークステーション IOP、および多機能 IOP のそれぞれについての、入出力処理装置 (IOP) のリソース名およびモデル番号。通信 IOP は、その IOP で使用された CPU のパーセント。パーセントは、必ずしも IOP がデータ転送を行っていることを意味するものではありません。パーセントの一部は、活動回線のオーバーヘッドによることがあります。

**IOP NAME/LINE (IOP 名前/行)**

(システム、リソース間隔) 入出力 (IOP) 処理装置のリソース名およびモデル番号行

**IOP NAME(MODEL) (IOP 名前 (モデル))**

(リソース間隔) 入出力処理装置 (IOP) の識別コードおよびモデル番号 (括弧内)

**IOP NAME (IOP 名)**

(システム、構成要素) 入出力処理装置 (IOP) のリソース名

**IOP NAME NETWORK INTERFACE (IOP 名ネットワーク・インターフェース)**

(リソース間隔) ネットワーク・インターフェースの IOP 名

**IOP PROCESSOR UTIL COMM (通信 IOP 処理装置使用率)**

(構成要素、リソース) 通信活動による IOP の使用率

**IOP PROCESSOR UTIL LWSC (LWSC IOP 処理装置使用率)**

(構成要素、リソース) ローカル・ワークステーション活動による IOP の使用率

**IOP PROCESSOR UTIL DASD (DASD IOP 処理装置使用率)**

(構成要素、リソース) DASD 活動による IOP の使用率

### **IOP PROCESSOR UTIL TOTAL (合計 IOP 処理装置使用率)**

(構成要素、リソース間隔) 各ローカル・ワークステーション、ディスク、および通信 IOP の使用率の合計パーセント

### **IOP UTIL (IOP 使用率)**

(システム) システム報告書のディスク使用率セクションでは、各入出力処理装置 (IOP) の使用率のパーセント

注: 多機能入出力処理装置の場合、この使用率はディスク活動だけによるもので、通信活動によるものではありません。システム・モデル・パラメーターのセクションでは、ディスク IOP が入出力操作を行っていた時間間隔の比率です。

### **ITV END (間隔終了)**

(構成要素、トランザクション、ジョブ間隔、プール間隔、リソース間隔) データが収集された時刻 (時および分)。構成要素報告書の例外発生要約および間隔カウントでは、収集サービスが例外を記録したサンプル間隔の終了時刻です。

### **JOB MAXIMUM A-I (ジョブ 最大 A-I)**

(プール間隔) プールまたはサブシステムで選択されたジョブによる活動状態から不適格状態への移行の最大数

### **JOB MAXIMUM A-W (ジョブ 最大 A-W)**

(プール間隔) プールまたはサブシステムで選択されたジョブによる活動状態から待機状態への移行の最大数

### **JOB MAXIMUM CPU UTIL (ジョブ 最大 CPU 使用率)**

(プール間隔) プールまたはサブシステムで選択されたジョブによって使用された使用可能な処理装置時間の最大パーセント

### **JOB MAXIMUM PHY I/O (ジョブ 最大物理入出力)**

(プール間隔) プールまたはサブシステムで選択されたジョブによる物理ディスク入出力操作の最大回数

### **JOB MAXIMUM RSP (ジョブ 最大応答時間)**

(プール間隔) プールまたはサブシステムで選択されたジョブによるトランザクションごとの最大応答時間 (秒数)。応答時間は、リソースの待機および使用に費やした時間の合計を、トランザクション数で割ったものです。

### **JOB MAXIMUM TNS (ジョブ最大トランザクション)**

(プール間隔) プールまたはサブシステムで選択されたジョブによるトランザクションの最大数

### **JOB MAXIMUM W-I (ジョブ 最大 W-I)**

(プール間隔) プールまたはサブシステムで選択されたジョブによる待機状態から不適格状態への移行の最大数

### **JOB NAME (ジョブ名)**

(構成要素、トランザクション、ジョブ間隔、バッチ・ジョブ追跡) ジョブの名前。トランザクション報告書のジョブ要約報告書では、ジョブがシステムのジョブ再経路指定 (RRTJOB) コマンドを使用している場合、そのジョブ (同一のジョブ名、ユーザー名、およびジョブ番号) がこのリストに複数回示されます。

### **JOB NUMBER (ジョブ番号)**

(構成要素、トランザクション、ジョブ間隔、バッチ・ジョブ追跡) 要約行に示されるジョブの番

号。トランザクション報告書では、ジョブ番号の前のアスタリスク (\*) は、そのジョブが測定期間中にサインオンされたことを示します。ジョブ番号の後のアスタリスク (\*) は、そのジョブが測定期間中にサインオフされたことを示します。

### **JOB PTY (ジョブ PTY)**

(バッチ・ジョブ追跡) ジョブの優先順位

### **JOB SET (ジョブ・セット)**

(トランザクション) ジョブ・セットの数とは、追跡期間の任意の時点で活動状態になっていたバッチ・ジョブの数のことです。2 つのジョブが順次実行される場合は、同じジョブ・セット中の 2 つのジョブとして示されます。2 つのジョブが並行して実行される場合は、別のジョブ・セット中にあるものとして示されます。

### **JOB TYPE (ジョブ・タイプ)**

(トランザクション報告書に関する注釈がある場所を除き、全報告書) ジョブ・タイプおよびサブタイプ。ジョブ・タイプの値は、次のとおりです。

- A** 自動開始
- B** バッチ
- BD** バッチ即時 (トランザクションのみ)

注: バッチ即時値は、「活動ジョブの処理」画面では BCI として、また「サブシステム・ジョブの処理」画面では BATCHI として表示されます。

- BE** バッチ呼び出し (トランザクションのみ)
- BJ** バッチ事前開始ジョブ (トランザクションのみ)
- C** APPC 上の 5250 エミュレーション、および APPC または TCP/IP のいずれかを実行している System i アクセス・ホスト・サーバーを含む、プログラム式ワークステーションのアプリケーション・サーバー。次の事項のいずれかが真である場合、ジョブは System i アクセス・サーバーとして報告されます。
  - 着信 APPC 呼び出しが、サーバー・プログラム名のいずれかを要求している場合。これは、指定のプログラムをすでに待機している、QSERVER、QCMN、および QSYSWRK サブシステムの事前開始ジョブに対しても適用されます。
  - 着信 IP ポート番号が、サービスの名前 - 記述 - ポート - 番号のいずれかに対応する場合。これは、割り当てられた IP ポート番号をすでに待機している、QSERVER、QCMN、および QSYSWRK サブシステムの事前開始ジョブに対しても適用されます。
  - 着信 IPX ソケット番号が、サービスの名前 - 記述 - ポート - 番号のいずれかに対応する場合。これは、割り当てられた IPX ポート番号をすでに待機している、QSERVER、QCMN、および QSYSWRK サブシステムの事前開始ジョブに対しても適用されます。
  - OS/2 コミュニケーション・マネージャーまたは WARP 同等機能のもとでの 5250 エミュレーションによって送信された APPC データ・ストリームに由来する、着信 5250 表示装置エミュレーション・ジョブ。
- D** ターゲット分散データ管理 (DDM) サーバー
- I** 対話式。対話式には、平衡型データ・リンク制御 (TDLC)、5250 遠隔ワークステーション、および 3270 遠隔ワークステーションを含みます。トランザクション報告書の場合、これには平衡型データ・リンク制御 (TDLC)、5250 遠隔ワークステーション、3270 遠隔ワークステーション、SNA パススルー、および 5250 Telnet を含みます。

- L** ライセンス内部コード・タスク
- M** サブシステム・モニター
- P** SNA パススルーおよび 5250 Telnet パススルー。トランザクション報告書では、これらのジョブは I (対話式) として表示されます。
- R** スプール読み取りプログラム
- S** システム
- W** スプール書き出しジョブ、および高機能印刷 (AFP) が指定されている場合は印刷ドライバー・ジョブを含む、スプール書き出しプログラム。
- WP** スプール印刷ドライバー (トランザクションのみ)
- X** システム開始ジョブ

ジョブのサブタイプの値は、次のとおりです。

- D** 即時のバッチ・ジョブ
- E** 呼び出し (通信バッチ)
- J** 事前開始ジョブ
- P** 印刷ドライバー・ジョブ
- T** 複数要求端末 (MRT) (System/36 環境のみ)

### 3 System/36

非対話式ジョブは、次のとおりです。

- 自動開始
- バッチ
- 呼び出し
- System i Access-Bch
- サーバー
- スプール
- 分散データ管理 (DDM) サーバー

特殊な対話式ジョブ・カテゴリーには、以下が含まれます。

- 対話式
- 複数要求端末 (MRT)
- パススルー
- System/36

### JOBS (ジョブ)

(システム、構成要素、トランザクション、プール間隔、ジョブ間隔) 指定するジョブ。項目の形式は、ジョブ番号/ユーザー名/ジョブ名です。報告書選択基準報告書では、含める (SLTJOB パラメーター) または除外する (OMTJOB パラメーター) ことにより選択されたジョブのリストです。これには、SLTFCNARA または OMTFCNARA パラメーターを使用して選択されたジョブは含まれません。

### K PER I/O (K/入出力)

(システム、リソース間隔) 各ディスク入出力操作で読み取りまたは書き込みされたキロバイト (1024 バイト) の平均数

**K/T /TNS SEC (K/T /TNS 秒)**

(トランザクション) 秒単位の平均遅延時間、またはジョブのトランザクション間で入力および考慮に費やされた時間。この値は、ジョブの活動状態から待機状態への移行、待機状態から活動状態への移行、または待機状態から不適格状態への移行の時間間隔を示します。

**KB PER I/O READ (KB/入出力読み取り)**

(リソース間隔) 読み取り操作当りの転送された平均キロバイト数 (1KB は 1024 バイト)

**KB PER I/O WRITE (KB/入出力書き込み)**

(リソース間隔) 書き込み操作当りの転送された平均キロバイト数 (1KB は 1024 バイト)

**KB RECEIVED/SECOND (受信 KB/秒)**

(システム、構成要素) 選択された間隔中に指定されたインターフェースが活動状態だった場合に、そのインターフェース上で 1 秒当りに受信されたキロバイト (1024) の合計数。ただし、フレーム文字を含みます。

**KB TRANSMITTED/SECOND (送信 KB/秒)**

(システム、構成要素) 選択された間隔中に指定されたインターフェースが活動状態だった場合に、そのインターフェースから 1 秒当りに送信されたキロバイト (1024) の合計数。ただし、フレーム文字を含みます。

**KBYTE TRANSMITTED IOP (IOP 送信キロバイト)**

(構成要素、リソース間隔) バスを介して IOP からシステムに送信された合計キロバイト数

**KBYTE TRANSMITTED SYSTEM (システム送信キロバイト)**

(構成要素、リソース間隔) バスを介してシステムから IOP に送信された合計キロバイト数

**KEY/THINK (入力/思考)**

(トランザクション) プログラムがワークステーション・ユーザーを待機するのに費やした時間の量

**KEY/THINK /TNS (入力/思考 /TNS)**

(トランザクション) 対話式ジョブの平均の考慮時間と入力時間、またはトランザクション境界間の遅延時間 (秒数)

**L** (ロック) ロック競合と占有競合のどちらであることを示します。この欄には、ロックの場合は L、占有の場合はブランクが入ります。

**LAPD PCT FRAMES RECD IN ERROR (エラーで受信した LAPD フレーム %)**

(リソース間隔) エラーで受信したフレームのパーセント (D チャネルだけに適用)。ホスト・システムにエラーがあるか、あるいは十分に速く受信データを処理できない場合に、エラーが起こる可能性があります。

**LAPD PCT FRAMES TRNSMITD AGAIN (再送信 LAPD フレーム %)**

(リソース間隔) エラーにより再送信されたフレームのパーセント (D チャネルにだけ適用)

**LAPD TOTAL FRAMES RECD (受信 LAPD フレーム合計)**

(リソース間隔) エラーのあるフレームおよび正しくないフレームを含む受信フレームの合計数 (D チャネルにだけ適用)

**LAPD TOTAL FRAMES TRNSMITD (送信 LAPD フレーム合計)**

(リソース間隔) 送信されたフレームの合計数 (D チャネルにだけ適用)

**LAST 4 PROGRAMS IN INVOCATION STACK (呼び出しスタックの最後の 4 つのプログラム)**

(トランザクション) プログラム・スタックの最後の 4 つのプログラム。たとえば、トランザクションの開始時 (ワークステーションの操作員が実行キーを押した場合など) には、名前が QT3REQIO、QWSGET のプログラム、および読み取り操作を行ったプログラムが示されます。またトランザクションの終了時 (プログラムが画面に書き込みを行う場合など) は、QT3REQIO、

QWSPUT、および画面に書き込みを行ったプログラムが示されます。通常、スタックの 3 番目または 4 番目のプログラムがトランザクション要約の PGMNAME データに示されるプログラムです。ただし、待機コード の欄に値が示されている場合は、最終 のラベルがある欄のプログラムが追跡レコードの原因となったプログラムです。欄にプログラム名がない場合は、その欄の前のプログラム名と同じで、名前が省略されています。

#### **LENGTH OF WAIT (待機の長さ)**

(ロック) 要求元がロック・オブジェクトを待機していたミリ秒数

#### **LGL I/O /SEC (論理入出力/秒)**

(ジョブ間隔) 間隔中にジョブによって実行された論理ディスク入出力操作の 1 秒当り平均回数。これは、論理ディスク入出力カウントを経過時間で割って算出されます。

#### **ライブラリー**

(システム、トランザクション) オブジェクトが入っているライブラリー

#### **LIBRARY**

(ジョブ・トレース) トレース項目と関連したプログラムを含むライブラリー名。

#### **LINE COUNT (行カウント)**

(ジョブ間隔) 間隔中に選択された非対話式ジョブによって印刷された行数

#### **LINE DESCRIPTN (回線記述)**

(リソース間隔) 回線記述名

#### **LINE ERRORS (回線エラー)**

(リソース間隔) 検出されたすべてのエラーの合計。この値が長時間にわたって増加する場合は、回線の状態を検査してください。

#### **LINE SPEED (回線速度)**

(システム、リソース間隔) K ビット (Kb) (1 K ビット (Kb) = 1000 ビット) 単位での 1 秒当りの回線速度

#### **LINE TYPE/LINE NAME (回線タイプ/回線名)**

(構成要素、システム) インターフェースによって使用される回線記述のタイプおよび名前。回線記述を使用しないインターフェースの場合、回線名フィールドは \*LOOPBACK、\*OPC、または \*VIRTUALIP と表示され、回線タイプは指定されません。

#### **LINE UTIL (回線使用率)**

(リソース間隔) 送受信操作で使用された使用可能な回線容量のパーセント

#### **伝送/受信回線使用率**

(リソース間隔) 通信回線のデータ伝送キャパシティーに使用されたパーセント。間隔中に、回線速度により除算された、伝送されたビット数および受信されたビット数。

**LKRL** (トランザクション) ロック解放。ジョブは、報告書の次の明細行 (OBJECT --) に示されたオブジェクトに対して持っていたロックを解放しました。オブジェクトを待機していたジョブは、この行 (WAITER --) に名前が示され、そのジョブがロックの解放の待機に費やされた時間も同時に示されます。

**LKW** (トランザクション) 待機コード欄にリストされるロック待機。これが多数ある場合、または ACTIVE/RSP\* の欄に時間の非常に長い項目がある場合は、追加の分析が必要です。この LKW 報告書行の前の LKWT 報告書行が、待機の対象のオブジェクトおよびそのオブジェクトの所有者を示します。

#### **LKWT**

(トランザクション) 待機コード欄にリストされるロック競合待機。ジョブはロック競合で待機して



います。時間 (\* / time /\*) は、ロック競合の期間で、LKW 時間と同じではないにしても、これに非常に近いはずで、ロックの保持者は、報告書行の右に示されます (HOLDER --)。ロックされたオブジェクトは、次の報告書行に示されます (OBJECT --)。

#### **LOCAL END CODE VIOLATION (ローカル端末コード違反)**

(リソース間隔) ISDN S/T 参照点のインターフェースで受信したフレームについて、端末装置 (TE) が検出した、意図しないコード違反の回数

#### **LOCAL NOT READY (ローカル作動不能)**

(リソース間隔) ホスト・システムから送信されたすべての受信不可フレームのパーセント。このパーセントが高いのは、多くの場合、ホストが十分に速くデータを処理できないこと (輻輳 (ふくそう)) を意味しています。

#### **ローカル・ワークステーション IOP 使用率**

ワークステーション入出力処理装置が使用中である時間間隔の比率

#### **LOCAL WORK STATION IOPS (ローカル・ワークステーション IOP)**

(システム) 各ローカル・ワークステーション IOP のリソース名およびモデル番号

#### **LOCK CONFLICT (ロック競合)**

(構成要素) 1 秒当りのロック例外の回数。データベース・レコードの競合がこのカウントに反映されます。詳しくは、パフォーマンス追跡開始 (STRPFTRC) コマンドを出して、トランザクション報告書印刷 (PRTTNSRPT) および ロック報告書印刷 (PRTLCKRPT) コマンドを使用してください。このカウントは、通常のシステム操作の場合でも非常に高くなる場合があります。このカウントをモニターとして使用してください。大きな変動または変化があった場合は、これを詳細に調べてください。

#### **LOCK WAIT /TNS (ロック待機/TNS)**

(トランザクション) トランザクションごとのロック待機時間の平均時間 (秒数)。この値が高い場合は、トランザクション明細演算と ロック報告書印刷 (PRTLCKRPT) コマンドで調べてください。

#### **LOGICAL (論理)**

(ジョブ間隔) 間隔中に選択された対話式ジョブによって実行された論理ディスク入出力操作の回数

#### **LOGICAL DATABASE I/O OTHER (論理データベース入出力その他)**

(システム) トランザクションごとのその他の論理データベース操作の回数。これには、更新や削除などの操作が含まれます。

#### **LOGICAL DATABASE I/O READ (論理データベース入出力読み取り)**

(システム) トランザクションごとの論理データベース読み取り操作回数

#### **LOGICAL DATABASE I/O WRITE (論理データベース入出力書き込み)**

(システム) トランザクションごとの論理データベース書き込み操作回数

#### **LOGICAL DB I/O (論理 DB 入出力)**

(システム) トランザクションごとの論理入出力操作の平均回数

#### **LOGICAL DB I/O COUNT (論理 DB 入出力カウント)**

(システム) 内部データベース入出力読み取り、書き込み、または各種の機能が呼び出された回数。この回数には、読取装置、書込装置への入出力操作、または スプール・ファイル・コピー (CPYSPLF) コマンドまたはスプール・ファイル表示 (DSPSPLF) コマンドによって引き起こされる入出力操作は含まれません。SEQONLY(\*YES) を指定すると、読み書きされた個々のレコード数ではなくて、読み書きされたレコードの各ブロックを示す数が表示されます。各種機能には、更新、削除、データの強制終了、および解放が含まれます。

**LOGICAL DISK I/O (論理ディスク入出力)**

(構成要素) 論理ディスク操作 (GET、PUT、更新、その他) の回数

**LOGICAL I/O /SECOND (論理入出力/秒)**

(システム) 1 秒当りの論理ディスク入出力操作の平均回数

**LOGICAL I/O PER SECOND (論理入出力/秒)**

(ジョブ間隔) 間隔中に選択された非対話式ジョブによって実行された論理ディスク入出力操作の 1 秒当り平均回数

**LONG WAIT (長時間待機)**

(トランザクション) ジョブがシステム・リソースの待機に費やした時間。長時間待機の例としては、レコード・ロック競合があります。「経過時間 - 秒」の欄にもリストされている通り、これは次のトランザクションの待機またはロック待機時間などの状態で費やした経過時間のことです。

**LONG WAIT LCK/OTH (長時間待機 LCK/OTH)**

(トランザクション) ジョブがシステム・リソースの待機に費やした時間の量。長時間待機の例としては、レコード・ロック競合があります。

**LOSS OF FRAME ALIGNMENT (フレーム位置合せの消失)**

(リソース間隔) 2 つの 48 ビット・フレームに相当する期間が経過する間に有効な回線コード違反の対を検出しない回数

**MAC ERROR (MAC エラー)**

(リソース間隔) 媒体アクセス制御 (MAC) エラーの数

**MAIN STORAGE (MB) (主記憶装置 (MB))**

(システム) メガバイト単位の主記憶装置の合計サイズ。これらのコードは待機コード欄に入っていますが、待機コードではありません。これらはトランザクション境界追跡レコードを示します。

**MAX UTIL (最大使用率)**

(システム) 所定のしきい値以上で一貫して使用している場合は、システムのパフォーマンスに影響を生じ、応答時間の遅れや、スループットの低下の原因になります。

**MAXIMUM (最大)**

(トランザクション) この欄の項目の最大値

**MEMBER (メンバー)**

(システム、トランザクション) システム報告書の場合は、パフォーマンス・データ作成 (CRTPFRTA) コマンドの TOMBR パラメーターで指定されたパフォーマンス・データ・メンバーの名前。トランザクション報告書の場合は、競合に関与したメンバー。

**MINIMUM (最小)**

(トランザクション) この欄の項目の最小値

**MRT MAX TIME (MRT 最大時間)**

(システム) MRTMAX に到達した後、複数要求端末の 1 つに経路指定されたジョブによって待機に費やされた時間

注: ジョブ・タイプが MRT 以外の場合、この欄には値は示されません。

**MSGs** (ジョブ・トレース) 各トランザクションの間に、ジョブに送られたメッセージの数。

**MTU SIZE (BYTE) (MTU サイズ (バイト))**

(システム) インターフェース上で送受信できる最大のデータグラムのサイズ。サイズは、オクテッ

ト (バイト) で指定されます。ネットワーク・データグラムの送信に使用されるインターフェースの場合、これは、インターフェース上で送信できる最大のネットワーク・データグラムのサイズです。

#### **NBR A-I (A-I 数)**

(トランザクション) ジョブ別の活動状態から不適格状態への移行の回数。この欄は、システムがトランザクションの処理を開始できるようになる前に、ジョブが割り当てられたタイム・スライスの値を超え、活動レベル・スロットを待機しなければならなかった回数を示します。この欄に値が表示された場合は、ジョブが実行中の作業を調べて、タイム・スライス値の変更が必要かどうかを判別してください。

#### **ディスク装置数**

(システム) 報告された区画に割り当てられたディスク装置の数。

#### **NBR EVT (事象数)**

(トランザクション) ジョブ処理中に起こった事象待機の数

#### **NBR JOBS (ジョブ数)**

(トランザクション) ジョブの数

#### **NBR SIGN OFFS (サインオフ数)**

(トランザクション) 間隔中にサインオフしたジョブの数

#### **NBR SIGN ONS (サインオン数)**

(トランザクション) 間隔中にサインオンしたジョブの数

#### **Nbr Tns (トランザクション数)**

(トランザクション) 所定のカテゴリ内のトランザクションの数

注: トランザクション報告書印刷 (PRTTNSRPT) コマンドを用いて作成する報告書に示されるトランザクション・カウントの値およびその他のトランザクション関連情報は、システム報告書印刷 (PRTSYSRPT) および 構成要素報告書印刷 (PRTCPTRPT) コマンドを用いて作成する報告書に示される値と異なる場合があります。この相違の原因は、PRTTNSRPT コマンドでは追跡データを入力として使用するのに対し、PRTSYSRPT および PRTCPTRPT コマンドではサンプル・データを入力として使用することにあります。

これらの報告書で示されたトランザクション関連情報の値に著しい相違がある場合は、この相違の原因を調べるまではこのデータを使用しないでください。

#### **NBR W-I (W-I 数)**

(トランザクション) ジョブ別の待機状態から不適格状態への移行の数。この欄には、ジョブがトランザクションを待機しなければならなかった回数が示されます。

#### **NDB READ (NDB 読み取り)**

(トランザクション) 物理入出力カウント欄にリストされている場合は、ジョブがその状態にあった間の非データベース読み取り要求の数。同期ディスク入出力要求 /TNS の欄にリストされている場合は、トランザクションごとの同期非データベース読み取り要求の平均回数。

#### **NDB WRITE (NDB 書き込み)**

(トランザクション) 同期ディスク入出力要求 /TNS の欄にリストされ、トランザクションごとの同期非データベース書き込み要求の平均回数

#### **NDB WRT (NDB 書き込み)**

(トランザクション) 物理入出力カウント欄にリストされている場合は、ジョブがその状態にあった間の非データベース書き込み要求の数。同期ディスク入出力カウント欄にリストされている場合は、トランザクションごとの非同期データベース書き込み要求の数。

**NON-DB**

(ジョブ・トレース) 項目に対し発生した、物理的非データベースの読み取り回数。

**NON-DB FAULT (非 DB 不在)**

(システム、構成要素) 非データベースのページ不在の 1 秒当りの平均回数

**NON-DB PAGE (非 DB ページ)**

(システム、構成要素) 読み取られた非データベース・ページの 1 秒当りの平均数

**NON-DB RDS**

(ジョブ・トレース) 物理的非データベースに発生した読み取り回数。

**NON SMAPP (非 SMAPP)**

(構成要素) 直接 SMAPP (System Managed Access Path Protection) に関連しないジャーナル・デポジット。

**NON-SSL INBOUND CONNECT (SSL 以外のインバウンド接続)**

(システム) サーバーによって受け入れられた SSL 以外のインバウンド接続の数。

**NON-UNICAST PACKETS RECEIVED (受信非ユニキャスト・パケット)**

(システム) 指定されたインターフェース上で受信されたパケット用に高位層プロトコルに配信される非ユニキャスト・パケットの合計数

**NON-UNICAST PACKETS SENT (送信非ユニキャスト・パケット)**

(システム) 高水準プロトコルが、非ユニキャスト・アドレスに送信するように要求したパケットの合計数。したがって、この数には、送信されたパケットだけでなく、廃棄されたパケットまたは送信されなかったパケットも含まれています。

**Number (番号)**

(トランザクション) トランザクションが関連するジョブの番号

**NUMBER I/O PER SECOND (秒当り入出力数)**

(システム) この特定の IOP の 1 秒当りの入出力の数

**NUMBER JOBS (ジョブ数)**

(トランザクション) ジョブ・セット内のバッチ・ジョブの数

**NUMBER LCK CFT (ロック競合の数)**

(トランザクション) ジョブ処理中に起こったロック待機 (データベース・レコード・ロックを含む) 状態競合の数。この欄の値が高い場合は、ジョブのトランザクション報告書および移行報告書で、ロック待機状態競合が継続した時間を調べてください。さらに、ロック報告書印刷 (PRTLCKRPT) コマンドを用いて作成される報告書により、さらに詳しく検査することができます。

**NUMBER LCK CONFLICT (ロック競合の数)**

(トランザクション) ジョブがロック競合を持った回数

**NUMBER LOCKS (ロックの数)**

(トランザクション) 対話式または非対話式待機側に帰されるロックの数

**NUMBER OF BATCH JOBS (バッチ・ジョブの数)**

(システム) 活動バッチ・ジョブの平均数。平均して 5 分間に最低 1 回の入出力を行うバッチ・ジョブは、活動状態にあると見なされます。

**NUMBER OF JOBS (ジョブ数)**

(システム) ジョブの数

**NUMBER OF PACKETS RECEIVED WITH ERRORS (エラー受信パケット数)**

(システム) 受信エラーがあったり、その他の理由で廃棄されるパケットの合計数。たとえば、バッファ・スペースを解放するためにパケットが廃棄される場合があります。

**NUMBER SEIZES (占有の数)**

(トランザクション) 対話式または非対話式待機側に帰される占有の数

**NUMBER SIZE CFT (占有競合の数)**

(トランザクション) ジョブ処理中に発生した占有/ロック競合の数。この値が高い場合は、ジョブのトランザクション報告書および移行報告書を参照して、競合の継続時間、オブジェクトを保留していたジョブの修飾名、保留されていたオブジェクトの名前とタイプ、およびジョブが何を待機していたかを調べてください。

**NUMBER SIZE CONFLICT (占有競合の数)**

(トランザクション) ジョブが占有競合を持った回数

**NUMBER TNS (TNS 数)**

(システム、トランザクション) 処理されたトランザクションの合計数。たとえば、システム報告書では、このプール中のジョブによって処理されたトランザクションの合計数です。トランザクション報告書では、プログラムに関連したトランザクションの数です。

**NUMBER TRACES (追跡数)**

(バッチ・ジョブ追跡) 追跡の数

**NUMBER TRANSACTIONS (トランザクション数)**

(システム) 処理されたトランザクションの合計数

**OBJECT FILE (オブジェクト・ファイル)**

(トランザクション) オブジェクトが入っているファイル

**OBJECT LIBRARY (オブジェクト・ライブラリー)**

(トランザクション) オブジェクトが入っているライブラリー

**OBJECT MEMBER (オブジェクト・メンバー)**

(トランザクション) 競合に関与したメンバー

**OBJECT NAME (オブジェクト名)**

(ロック) ロックされたオブジェクトの名前

**OBJECT RRN (オブジェクト RRN)**

(トランザクション) 競合に関与したレコードの相対レコード番号

**OBJECT TYPE (オブジェクト・タイプ)**

(トランザクション、ロック) ロックされたオブジェクトのタイプ。オブジェクト・タイプは、次のとおりです。

**AG** アクセス・グループ

**CB** コミット・ブロック

**CBLK** コミット・ブロック

**CD** 制御装置記述

**CLS** クラス

**CMD** コマンド

**CTLD** 制御装置記述

**CTX** コンテキスト

**CUD** 制御装置記述  
**CUR** カーソル  
**DEVD** 装置記述  
**DS** データ・スペース  
**DSI** データ・スペース索引  
**DTAARA**  
データ域  
**EDTD** 編集記述  
**FILE** ファイル  
**JOB** ジョブ記述  
**JOBQ** ジョブ待ち行列  
**JP** ジャーナル・ポート  
**JRN** ジャーナル  
**JRNRCV**  
ジャーナル・レシーバー  
**JS** ジャーナル・スペース  
**LIB** ライブラリー  
**LIND** 回線記述  
**LUD** 論理装置記述  
**MBR** メンバー  
**MEM** データベース・ファイル・メンバー  
**MSGF** メッセージ・ファイル  
**MSGQ**  
メッセージ・キュー  
**ND** ネットワーク記述  
**OCUR** データベース操作カーソル  
**OUTQ** 出力待ち行列  
**PGM** Program (プログラム)  
**PROG** Program (プログラム)  
**PRTIMG**  
印刷イメージ  
**QDAG** 複合体 - アクセス・グループ  
**QDDS** 複合体 - データ・スペース  
**QDDSI**  
複合体 - データ・スペース索引  
**QTAG** 一時 - アクセス・グループ  
**QTDS** 一時 - データ・スペース

**QTDSI**

一時 - データ・スペース索引

**SBSD** サブシステム記述

**TBL** テーブル

**OMIT PARAMETERES (除外パラメーター)**

(システム、構成要素、トランザクション、ジョブ間隔、プール間隔) 報告書から除外するデータ・レコードの選択に使用された基準。基準は、一般的に、コマンドの OMT<sub>xxx</sub> パラメーターを使用して指定します。デフォルト以外の値 (\*NONE 以外のもの) だけが印刷されます。パラメーターが指定されなかった場合は、報告書には示されません。

**OP PER SECOND (秒当り操作数)**

(システム) 1 秒当りの平均ディスク操作回数

**OTHER WAIT /TNS (その他の待機 /TNS)**

(トランザクション) トランザクションごとに、以前のカテゴリーのいずれにも含まれなかった待機に費やされた平均時間 (秒数)。たとえば、システムが新しい媒体 (テープまたはディスク) を要求した場合に、保管/復元操作中に待機で費やされた時間。

**OUTGOING CALLS PCT RETRY (発呼再試行 %)**

(リソース間隔) ネットワークに拒否された呼び出しのパーセント

**OUTGOING CALLS TOTAL (呼び出しの合計)**

(リソース間隔) 呼び出しを試みた合計回数

**OVER COMMITMENT RATIO (オーバー・コミットメント率)**

(システム) 主記憶装置のオーバー・コミットメント率

**PAG (プロセス・アクセス・グループ)**

(トランザクション) 処理アクセス・グループ不在の回数

**PAG FAULT (PAG 不在)**

(構成要素、ジョブ間隔) 構成要素報告書の例外発生の要約では、プログラム・アクセス・グループ (PAG) が参照されたが主記憶装置になかった合計回数。ライセンス内部コードでは、データをキャッシュに入れるための処理アクセス・グループは使用しなくなりました。このインプリメンテーションのために、現行リリース以降では、この値は常に 0 です。構成要素報告書の例外発生の要約では、処理アクセス・グループと関連した 1 秒当りの不在回数。

**PAGE COUNT (ページ・カウント)**

(ジョブ間隔) 間隔中に選択された非対話式ジョブによって印刷されたページ数

**PCT CPU BY CATEGORIES (カテゴリー別 CPU パーセント)**

(トランザクション) さまざまなカテゴリーに分類されるトランザクションによって使用された使用可能な処理装置時間のパーセント。カテゴリーの説明については、システム要約データ・セクションの対話式トランザクション・カテゴリー別分析の部分を参照してください。

**PCT DATA CHARACTERS RECEIVED IN ERROR (エラーのあった受信データ文字 %)**

(リソース間隔) 受信エラーのあったデータ文字のパーセント

**PCT DATA CHARACTERS TRANSMITTED IN ERROR (エラーのあった送信データ文字 %)**

(リソース間隔) 送信エラーのあったデータ文字のパーセント

**PCT DATAGRAMS ERROR (データグラム・エラー %)**

(構成要素) 次のエラーによって廃棄されたデータグラムのパーセント

- IP 見出しの宛先フィールドの IP アドレスが、このエンティティで受信される有効なアドレスではない
- プロトコルが不明またはサポートされていない
- バッファ・スペース不足

#### **PCT ERROR RESPONSES (エラー応答 %)**

(構成要素) エラーになった応答のパーセント

#### **PCT EX-WT /RSP (例外的待機 /RSP %)**

(トランザクション) 例外的待機による応答時間のパーセント

#### **PCT ICMP MESSAGES ERROR (ICMP メッセージ・エラー %)**

(構成要素) エンティティが受信したものの、メッセージにエラーがあることを判別したか、問題があるためにエンティティが送信しなかった Internet Control Message Protocol (ICMP) メッセージの数

#### **PCT OF TNS CATEGORIES (TNS カテゴリーのパーセント)**

(トランザクション) 種々のカテゴリーに区別されたすべてのトランザクションのパーセント。カテゴリーの説明については、システム要約データ・セクションの対話式トランザクション・カテゴリー別分析の部分を参照してください。

#### **PCT PACKETS RECEIVED ERROR (エラーのあった受信パケット %)**

(システム) 受信エラーがあったり、その他の理由で廃棄されるパケットのパーセント。たとえば、バッファ・スペースを解放するためにパケットが廃棄される場合があります。

#### **PCT PACKETS SENT ERROR (エラーのあった送信パケット %)**

(システム) 受信エラーがあったり、その他の理由で送信されないパケットのパーセント。たとえば、バッファ・スペースを解放するためにパケットが廃棄される場合があります。

#### **PCT PDUS RECEIVED IN ERROR (エラーで受信した PDU %)**

(リソース間隔) 時間間隔中に受信エラーのあったプロトコル・データ単位 (PDU) のパーセント。ホスト・システムにエラーがある場合、または十分に速くデータを受信できない場合 (輻輳) に、これらのエラーが起こる可能性があります。

注: 非同期通信のプロトコル・データ単位 (PDU) は、終わりがプロトコル制御文字またはバッファのサイズになる可変長データ単位です。

#### **PCT POLL RETRY TIME (ポーリング再試行時間 %)**

(リソース間隔) 切断モードにあったワークステーション制御装置 (または遠隔システム) からの応答を IOP が待っていて、回線が使用不能であった時間間隔のパーセント

注: この時間損失を最小限にするには、以下のことを行います。

- 電源がオンになっている制御装置だけをオンに構成変更する。
- すべての制御装置の電源をオンにする。
- 回線記述変更 (SDLC) (CHGLNSDLC) コマンドを使用してポーリング接続タイマーを小さい値に設定する (待機時間を減らす)。
- 制御装置記述変更 (CHGCTLxxxx) コマンド (xxxx は、APPC、FNC、RWS、または RTL) を使用して NDMPOLLTMR の値を大きい値に設定する (ポーリングとポーリングの間の時間を増やす)。

#### **PCT TNS (トランザクション %)**

(トランザクション) 合計トランザクションのパーセント。ジョブ要約報告書のシステム要約セクションでは、トランザクションは所定の除去属性をもつ所定の追跡期間内にあります。ジョブ要約報



告書の対話式プログラム・トランザクション統計セクションでは、プログラムに関連したトランザクションのパーセントです。ジョブ統計セクションでは、このジョブによる合計トランザクション数のパーセントです。対話式プログラム統計セクションでは、プログラムに関連したすべてのトランザクションです。

**PCT UDP DATAGRAMS ERROR (UDP データグラム・エラー %)**

(構成要素) 宛先ポートにアプリケーションがない、または他の理由で配信できなかった User Datagram Protocol (UDP) データグラムのパーセント

**PERCENT ERRORED SECONDS (エラー秒数 %)**

(リソース間隔) 少なくとも 1 つのインまたはアウトの検出アクセス伝送 (DTSE) エラーが起こった秒数のパーセント

**PERCENT FRAMES RECEIVED IN ERROR (エラー受信フレーム %)**

(リソース間隔) 受信エラーのあったすべての受信フレームのパーセント。ホスト・システムにエラーがあるか、十分に速く受信データを処理できない場合 (輻輳) に、エラーが起こる可能性があります。

**PERCENT FULL (% FULL)**

(システム) 使用されているディスク・スペース容量のパーセント

**PERCENT I FRAMES TRNSMITD IN ERROR (エラー送信 I フレーム %)**

(リソース間隔) 再送信を必要とした送信情報フレームのパーセント。再送信は、遠隔装置にエラーがあるか、十分に速く受信データを処理できない (輻輳) 場合に起こる可能性があります。

**PERCENT SEVERELY ERRORED SECONDS (重大エラー秒数 %)**

(リソース間隔) 少なくとも 3 つのインまたはアウトの検出アクセス伝送 (DTSE) エラーが起こった秒数のパーセント

**PERCENT TRANSACTIONS (DYNAMIC NO) (トランザクション % (DYNAMIC NO))**

(システム) システム主記憶装置利用率の尺度。除去属性 DYNAMIC NO のすべての対話式トランザクションのパーセント。

**PERCENT TRANSACTIONS (PURGE NO) (トランザクション % (PURGE NO))**

(システム) システム主記憶装置利用率の尺度。除去属性 PURGE NO のすべての対話式トランザクションのパーセント。

**PERCENT TRANSACTIONS (PURGE YES) (トランザクション % (PURGE YES))**

(システム) システム主記憶装置利用率の尺度。除去属性 PURGE YES のすべての対話式トランザクションのパーセント。

**PERCENT UTIL (使用率 %)**

(システム) 平均ディスク・アーム使用率 (使用中)。使用率が常に、ディスク・アーム使用率として示されたしきい値になっているかこれを超えていると、システムのパフォーマンスに影響があり、応答時間の遅れや、スループットの低下の原因になります。

注: 使用中のパーセントの値は、入出力処理装置で測定されたデータから算出されます。この値を、ディスク状況の処理 (WRKDSKSTS) コマンドで報告された使用パーセントと比較すると、相違がある場合があります。WRKDSKSTS コマンドは、入出力要求数、転送されたデータの量、およびディスク装置のタイプに基づいて使用パーセントを推定します。

システム全体の平均使用率には、測定間隔におけるミラー保護のアームのデータは含まれません。ミラー保護アームの場合、それらの間隔は再開状況か、中断状況のいずれかにあります。

**PERM SIZE (永続サイズ)**

(構成要素) 永続領域内に配置されるキロバイト。これらは、検索および表示が可能な従来のジャーナル項目です。

**PERM WRITE (永続書き込み)**

(構成要素、ジョブ間隔) 間隔中に選択されたジョブで実行された永続書き込み操作の回数

**PERMANENT WRITES PER TRANSACTION (永続書き込み/トランザクション)**

(システム) 対話式トランザクションごとの永続書き込み操作の平均回数

**PHYSICAL I/O COUNT (物理入出力カウント)**

(トランザクション、バッチ・ジョブ追跡) バッチ・ジョブ追跡報告書のジョブ要約セクションでは、同期および非同期ディスク操作 (読み取りおよび書き込み) の回数。移行報告書では、次の 5 つの欄が、ジョブの所定の状態にあった場合の同期および非同期ディスク入出力要求の数についての情報を示します。最初の行は同期ディスク入出力要求で、2 番目の行は非同期ディスク入出力要求です。

**DB READ (DB 読み取り)**

ジョブがこの状態であったときのデータベース読み取り要求の数

**DB WRT (DB 書き込み)**

ジョブがこの状態であったときのデータベース書き込み要求の数

**NDB READ (NDB 読み取り)**

ジョブがこの状態であったときの非データベース読み取り要求の数

**NDB WRT (NDB 書き込み)**

ジョブがこの状態であったときの非データベース書き込み要求の数

**TOT (合計)**

データベース読み取り、データベース書き込み、非データベース読み取り、および非データベース書き込み要求の合計数

**PHYSICAL WRITES (物理書き込み)**

(構成要素) ディスクへの物理ジャーナル書き込み操作。

**PL (プール)**

(構成要素、トランザクション、ジョブ間隔、プール間隔) サブシステムまたはジョブが実行されたプールの番号

**POOL (プール)**

(トランザクション、ジョブ間隔、バッチ・ジョブ追跡) トランザクションが入っているプール (たとえば、その中でジョブが実行された) の番号

**POOL ID (プール ID)**

(システム) プール識別コード

**POOL ID FAULTS (プール ID 不在)**

(構成要素) ページ不在率が最も高かったユーザー・プール

**POOL MCH FAULTS/SEC (プール MCH 不在/秒)**

(構成要素) プール・マシン・ページ不在の 1 秒当りの平均数

**POOL SIZE (KB) (プール・サイズ (KB))**

(構成要素) 構成要素報告書の記憶域プール活動セクションでは、メガバイト単位の初期プール・サイズ。

**POOL USER FAULTS/SEC (プール・ユーザー不在/秒)**

(構成要素) この間隔中に不在率が最も高かったユーザー・プールのユーザー・プール・ページ不在の 1 秒当りの平均数

**POOLS (プール)**

(システム、構成要素、トランザクション、ジョブ間隔、プール間隔) 報告書選択基準セクションで、組み込み (SLTPOOLS パラメーター) または除外 (OMTPOOLS パラメーター) により選択されたプールのリスト。それ以外の場合は、指定するプール。値は、1 から 64 までの範囲です。

**PRG (プログラム)**

(トランザクション) ジョブの除外属性

**PRINTER LINES (印刷装置行数)**

(システム、ジョブ間隔) 間隔中にジョブによって印刷された行数

**PRINTER PAGES (印刷装置ページ)**

(システム、ジョブ間隔) 間隔中にジョブによって印刷されたページ数

**PRIORITY (優先順位)**

(システム、トランザクション) ジョブの優先順位

**Program (プログラム)**

(トランザクション) トランザクションが関連するプログラムの名前

**PROGRAM (プログラム)**

(ジョブ・トレース) エントリーのためのプログラム名。

**PROGRAM CALL (プログラム呼び出し)**

(ジョブ・トレース) ステップの間に呼び出された QSYS 以外のライブラリー・プログラムの数。これは PROGRAM NAME フィールドで名前付けされているプログラムが呼び出された回数ではありません。

**PROGRAM DATABASE I/O (プログラム・データベース I/O)**

(ジョブ・トレース) IBM 提供のデータベース・モジュールがトランザクションの間に使用された回数。データベースのモジュール名は、QDB 接頭部が除去されました (QDBPUT にかわって PUT)。それぞれに実行された論理入出力操作のタイプは、以下のとおりです。

**GETDR**

直接 Get

**GETSQ**

順次 Get

**GETKY**

キーによる Get

**GETM**

複数 Get

**PUT, PUTM**

レコードの追加

**UDR** レコードの更新、削除、または解放

**PROGRAM INIT (プログラム初期設定)**

(ジョブ・トレース) IBM 提供の初期設定プログラムが、トランザクションの間に呼び出された回数。このプログラムは、RPG の場合は QRGXINIT、COBOL の場合は QCRMAIN です。ユーザー・プログラムが LR (RPG) または END (COBOL) で終了するつど、IBM 提供のプログラムも呼

び出されます。これは PROGRAM NAME フィールドで名前付けされているプログラムが初期化された回数ではありません。プログラムの初期化以外の機能 (たとえば、ブロック化レコードの入出力、いくつかのデータ変換) では、QCRMAIN が使用されます。

#### **PROGRAM NAME (プログラム名)**

(トランザクション) トランザクション報告書のジョブ要約セクションでは、トランザクションの開始時に制御状態にあるプログラムの名前。そのトランザクションの処理時には、他のプログラムが使用されている可能性もあります。トランザクション報告書セクションでは、トランザクションの開始時に活動状態にあるプログラムの名前。ADR=UNKNWN (アドレス不明) が欄に示されている場合は、追跡データがデータベース・ファイルにダンプされる前にそのプログラムは削除されています。ADR=000000 が欄に示されている場合は、プログラム名を判別するための十分な追跡データがなかったか、または追跡レコードの作成時にジョブ内でそのレベルで活動状態のプログラムがなかったことを示します。

#### **PROGRAM NAME (プログラム名)**

(ジョブ・トレース) トランザクションの終了前にはライブラリー QSYS に存在しなかったプログラムで、最後に呼び出されたプログラムの名前。

#### **PROTOCOL (プロトコル)**

(システム) 回線プロトコル

- SDLC
- ASYNC
- BSC
- X25
- TRLAN
- ELAN (イーサネット)
- IDLC
- DDI
- FRLY
- PPP

**PTY** (構成要素、トランザクション、ジョブ間隔) ジョブの優先順位。トランザクション報告書の並行バッチ・ジョブ統計では、ジョブ・セット内のジョブの優先順位。

**PU** (システム) 区画キャパシティー。報告された区画に割り当てられた処理装置の数。

#### **PURGE (除去)**

(トランザクション) ジョブの除外属性

#### **PWRT (永続書き込み)**

(トランザクション) 永続書き込み入出力操作の回数

#### **QUEUE LENGTH (待ち行列長さ)**

(リソース間隔) この装置の待ち行列で待機しなければならなかった入出力要求の平均数

#### **RANK (順位)**

(トランザクション) 順序。ジョブ要約セクションでは、トランザクション数によるプログラムの順序。ジョブ統計セクションでは、ジョブの順序。対話式プログラム統計セクションでは、プログラムの順序。個別トランザクション統計セクションでは、重要度の順に置かれているデータによるトランザクションの順序。最長の占有/ロック競合のセクションでは、占有またはロック競合の順序。

**RATIO OF WRITE DISK I/O TO TOTAL DISK I/O (書き込みと合計ディスク入出力の比)**

(システム) ディスクへのデータの書き込みによる合計ディスク活動の比率

**READS PER SECOND (秒当り読み取り)**

(リソース間隔) ディスク・アームによって行われた 1 秒当りのディスク読み取り操作の平均回数

**RECEIVE CRC ERRORS (受信 CRC エラー)**

(リソース間隔) サイクル冗長検査 (CRC) エラーがあった受信フレームの数。これは、データがエラーなしで受信されなかったことを示します。

**RECORD NUMBER (レコード番号)**

(ロック) データベース・ファイル・メンバーでは、データベース・ファイル・メンバー内のレコードの相対レコード番号

**REMOTE LAN PCT FRAMES RECD (遠隔 LAN 受信フレーム %)**

(リソース間隔) ローカル接続 LAN に接続されたローカル・エリア・ネットワーク (LAN) から受信したフレームの数

**REMOTE LAN PCT FRAMES TRNSMITD (遠隔 LAN 送信フレーム %)**

(リソース間隔) ローカル接続 LAN に接続されたローカル・エリア・ネットワーク (LAN) に送信したフレームの数

**REMOTE NOT READY (遠隔作動不能)**

(リソース間隔) ホスト・システムが受信したすべての受信不可フレームのパーセント。このパーセントが高いのは、多くの場合、遠隔装置が十分に速くデータを処理できないこと (輻輳) を意味しています。

**REMOTE SEQ ERROR (遠隔順序エラー)**

(リソース間隔) 遠隔装置またはシステムが正しい順序で受信できなかったフレームのパーセント。これは、遠隔装置またはシステムが十分に速くデータを処理できない場合に発生します。

**REQ TYPE (要求タイプ)**

(構成要素) 報告された要求のタイプ。

**REQUESTS RECEIVED (受信された要求)**

(システム、構成要素) サーバーが受け取った、すべてのタイプの要求の数。

**REQUESTOR'S JOB NAME (要求元ジョブ名)**

(ロック) ロックされたオブジェクト (明細リスト内と同じ) を要求しているジョブの名前

**RESET PACKETS RECD (受信リセット・パケット数)**

(リソース間隔) ネットワークが受信したりリセット・パケットの数。リセット・パケットは、エラーが起こり、再送信されたパケットです。

**RESET PACKETS TRNSMITD (送信リセット・パケット数)**

(リソース間隔) ネットワークから送信されたりリセット・パケットの数

**RESPONSE (応答)**

(システム) 平均システム応答 (サービス) 時間

**RESPONSE SEC AVG AND MAX (応答秒数 - 平均および最大)**

(トランザクション) ジョブの秒単位の平均 (AVG) および最大 (MAX) トランザクション応答時間。平均応答時間は、待機状態から活動状態への移行および活動状態から待機状態への移行の各対の間の時間の合計を、ジョブで発生した移行の対の数で割って算出されます。MAX 応答時間は、ジョブの最大応答時間です。

**RESPONSE SECONDS (応答秒数)**

トランザクションごとの平均応答時間 (秒数)

**RESPONSES SENT (応答送信)**

(システム、構成要素) サーバーが送った、すべてのタイプの応答の数。

**RSP** (構成要素) 対話式トランザクション平均応答時間 (秒数)

**RSP TIME (応答時間)**

(構成要素、リソース間隔) 平均外部応答時間 (秒数)。リソース間隔報告書のローカル・ワークステーション IOP 使用率セクションでは、この制御装置のワークステーションの応答時間。構成要素報告書の遠隔ワークステーション・セクションでは、このワークステーションの応答時間。

**RSP TIMER ENDED (応答タイマー終了)**

(リソース間隔) 応答タイマーが遠隔装置からの応答を待つのを終了した回数

**RSP/TNS**

(構成要素、トランザクション、ジョブ間隔) トランザクションごとの平均応答時間 (秒数)。ジョブ間隔報告書のジョブ要約セクションでは、間隔中に選択された対話式ジョブのトランザクションごとの応答時間 (システム・リソースを待機または使用するのに費やした時間の量を、処理されたトランザクションの数で割った値)。この数は、少なくとも数秒がトランザクションの処理に費やされていない場合は、正確ではありません。

**S/L** (トランザクション) 競合が占有 (S) 競合またはロック (L) 競合のいずれだったかを示します。

**SECONDS (秒)**

(ジョブ・トレース) ジョブが待機中または活動中であった時間。

**SEGMENTS PCT RTRNS (再送セグメント %)**

(構成要素) 再送されるセグメントのパーセント。この数値は、送信され、以前送信されたオクテット (バイト) を 1 つまたは複数含む、TCP セグメントです。

**SEGMENTS RCVD PER SECOND (秒当り受信されるセグメント)**

(構成要素) 1 秒当りに受信されるセグメント数。この数には、受信エラーのあったセグメントや現在、設定されている接続上で受信されたセグメントが含まれています。

**SEGMENTS SENT PER SECOND (秒当り送信されるセグメント)**

(構成要素) 1 秒当りに送信されるセグメント数。この数には、現在、設定されている接続上で送信されたセグメントは含まれていますが、再送されたオクテット (バイト) のみを含むセグメントは除外されています。

**SEIZE AND LOCK CONFLICTS (占有およびロック競合)**

(バッチ・ジョブ追跡) 占有競合およびロック待機の数

**SEIZE CONFLICT (占有競合)**

(構成要素) 1 秒当りの占有例外の数。詳しくは、パフォーマンス追跡開始 (STRPFRTRC) コマンドを発行して、PRTTNSRPT または PRTLCKRPT コマンドを使用します。このカウントは、通常のシステム操作の場合でも非常に高くなる可能性があります。このカウントをモニターとして使用してください。大きな変動または変化があった場合は、これを詳細に調べてください。

**SEIZE HOLD TIME (占有保留時間)**

(トランザクション) オブジェクトに対する占有またはロックによってトランザクションがシステムの他のジョブを保留していた時間の長さ

**SEIZE WAIT /TNS (占有待機/TNS)**

(トランザクション) 平均トランザクション中に発生したすべての占有ロック競合の平均時間 (秒数)。同じジョブの 1 つのトランザクションで複数の占有ロック競合が起こる可能性があります。この値が高い場合は、占有競合のあるジョブを調べてください。トランザクション報告書には、発生したそれぞれの競合、保持者の名前、および保留されたオブジェクトの名前がリストされています。

す。ジョブ要約報告書の 5 分間隔ごとのトランザクション・セクションでは、トランザクションごとの平均占有待機時間 (秒数) です。これは、トランザクションが占有/ロック競合で費やした平均時間です。この数が大きい場合は、待機時間が大きくなる原因となっているジョブのトランザクション報告書および移行報告書を調べてください。

#### **SELECT PARAMETERS (選択パラメーター)**

(システム、構成要素、トランザクション、ジョブ間隔、プール間隔) 報告書に組み込むデータ・レコードの選択に使用された基準。基準は、一般的に、コマンドの SLTxxx パラメーターを使用して指定します。デフォルト以外の値 (\*ALL 以外のもの) だけが印刷されます。パラメーターが指定されない場合は、報告書には表示されません。

#### **SEQNBR**

(ジョブ・トレース) トレース項目の番号。

#### **SEQNCE または SEQUENCE**

(ジョブ・トレース) この要約行が参照している詳細報告書内の、ジョブ・トレースの順序番号。

#### **SEQUENCE ERROR (シーケンス・エラー)**

(リソース間隔) フレームが失われたことを示す順序番号が入った受信フレームの数

#### **SERVER JOB NAME (サーバー・ジョブ名)**

(システム) サーバー・ジョブ番号。サーバーにとって子のジョブを識別します。

#### **SERVER JOB USER (サーバー・ジョブ・ユーザー)**

(システム) サーバー・ジョブ・ユーザー。サーバーにとって子のジョブを識別します。

#### **SERVER NAME (サーバー名)**

(システム) サーバー・ジョブ名。サーバーにとって子のジョブを識別します。

#### **SERVER START DATE/TIME (サーバー開始日/時間)**

(システム) フォーマット mm/dd/yy hh:mm:ss で表示する、最新の始動または再始動時。

#### **SHORT FRAME ERRORS (短フレーム・エラー)**

(リソース間隔) 受信した短フレームの数。短フレームは、開始フラグと終了フラグとの間のオクテットが許容限界より少ないフレームです。

#### **SHORT WAIT /TNS (短時間待機 /TNS)**

(トランザクション) トランザクションごとの短期 (活動) 待機時間の平均時間 (秒数)。対話式プログラム統計セクションでは、値が高い場合は、データ待ち行列の使用、またはプログラム表示装置ファイルでの DFRWRT(\*NO) または RSTDSP(\*YES) の使用が原因と考えられます。

#### **SHORT WAITX /TNS (SHORT WAIT EXTENDED) (短時間待機 X/TNS)**

(トランザクション) 2 秒を超えた短時間 (活動) 待機に由来し、長時間待機移行が起こる原因となったトランザクションごとの待機時間の平均時間 (秒数)。活動レベルは解放されていますが、この時間は合計応答時間にカウントされます。データ待ち行列上での待機、または表示装置ファイルでの DFRWRT(\*NO) または RSTDSP(\*YES) (あるいはその両方) の使用によって、この値が高くなる場合があります。

#### **SIZE (サイズ)**

(構成要素) 1 秒当りの 10 進データ・オーバーフローおよびアンダーフロー例外の回数。数値計算で正しくないフィールド・サイズがあることを示します。

#### **SIZE (MB)**

(システム) プール・サイズ (メガバイト)。

#### **SIZE (GB)**

(プール間隔) プール・サイズ (ギガバイト)。

**SIZE (M) (サイズ (M))**

(システム) 100 万バイト単位のディスク・スペース容量

**SHARE CLS**

(ジョブ・トレース) すべてのタイプのファイルに対する、共有クローズの回数。

**SHARE OPN**

(ジョブ・トレース) すべてのタイプのファイルに対する、共有オープン回数。

**SMAPP RETUNE (SMAPP 再調整)**

(構成要素) システム管理のアクセス・パス保護調整

**SMAPP SYSTEM (SMAPP システム)**

(構成要素) システム提供 (デフォルト) のジャーナルに保管された、SMAPP に起因するジャーナル項目

**SMAPP USER (SMAPP ユーザー)**

(構成要素) ユーザー提供 (デフォルト) のジャーナルに保管された、SMAPP に起因するジャーナル項目

**SOTn** (トランザクション) 待機コード欄にリストされる、トランザクション n の開始。これらのコードは、待機コード欄に示されますが、待機コードではありません。これらはトランザクション境界追跡レコードを示します。

**SPOOL CPU SECONDS PER I/O (スプール CPU 秒/入出力)**

(システム) すべてのスプール・ジョブが使用したシステム処理装置秒数を、1 つのスプール・ジョブが実行した入出力 1 回当たりの値に平均した秒数

**SPOOL DATABASE READS PER SECOND (スプール・データベース読み取り/秒)**

(システム) スプール処理におけるデータベース・ファイル読み取り操作の 1 秒当たりの平均回数

**SPOOL I/O PER SECOND (スプール入出力/秒)**

(システム) スプール処理における 1 秒当たりの物理ディスク入出力操作の平均回数

**SRV TIME (サービス時間)**

(構成要素) 要求当りのディスク・サービス平均時間 (秒数) (ディスク待機時間は含まれていません)。

**SSL INBOUND CONNECTIONS**

(システム) サーバーによって受け入れられた SSL インバウンド接続の数。

**START (開始)**

(トランザクション) ジョブが開始した時刻

**Started (開始)**

(トランザクション) 追跡データの最初のレコードの時刻で、形式は HH.MM.SS (時、分、秒) です。

**状態** (トランザクション) 起こり得る 3 つのジョブ状態は、以下に示されています。

- **W-**(待機状態) 活動レベルを保留していない。
- **A-**(活動または待機状態) アクティビティー・レベルを保留している。
- **I-**(不適格状態) 活動レベルを待機している。

以下の表は、起こり得るジョブ状態の移行を示しています。たとえば、**W** から **A** への移行は、**yes** になっていますが、これはジョブが待機状態から活動状態に変わることが可能であることを意味します。



表 8.

		状態へ		
		A	W	I
状態から	A	はい	はい	はい
	W	はい		はい
	I	はい		

**STATE TRANSITIONS A-A (状態変換 A-A)**

(バッチ・ジョブ追跡) 活動状態から活動状態への移行の数

**STATE TRANSITIONS A-I (状態変換 A-I)**

(バッチ・ジョブ追跡) 活動状態から不適合状態への移行の数

**STOP (停止)**

(トランザクション) ジョブが終了した時刻

**Stopped (停止)**

(トランザクション) 追跡データの最後のレコードの時刻で、形式は HH.MM.SS (時、分、秒) です。

**SUBFILE READS (サブファイル読み取り)**

(ジョブ・トレース) サブファイルの読み取り回数。

**SUBFILE WRITES (サブファイル書きこみ)**

(ジョブ・トレース) サブファイルの書き込み読み取り回数。

**SUBSYSTEM NAME (サブシステム名)**

(プール間隔) サブシステムの名前

**SUBSYSTEMS (サブシステム)**

(システム、構成要素、プール間隔) システム報告書では、指定するサブシステム名。それぞれ 10 桁の名前です。構成要素報告書では、組み込み (SLTSBS パラメーター) または除外 (OMTSBS パラメーター) により選択されたサブシステムのリスト。

**SUM (合計)**

(トランザクション) 同期ディスク入出力要求 /TNSの欄にリストされ、同期 DB 読み取り、DB 書き込み、NDB 読み取り、NDB 書き込み要求の平均数の合計 (ジョブのトランザクションごとの同期入出力要求の平均数)

**SWX**

(トランザクション) 待機コード欄にリストされる拡張短期待機。短期待機は 2 秒の限界を超過し、システムはトランザクションを長期待機に入れました。この長期待機はトランザクション応答時間に入れなければなりません。多くの場合、この活動状態から待機状態に移行したトランザクションは、トランザクション境界を反映しません。

**SYNC (同期)**

(ジョブ間隔) 間隔中に選択された対話式ジョブによって実行された同期ディスク入出力操作の回数

**SYNC DIO /TNS (同期 DIO/TNS)**

(トランザクション) 間隔中のトランザクションごとの同期入出力要求の平均数

**SYNC DISK I/O (同期ディスク入出力)**

(システム、構成要素、トランザクション) 同期ディスク入出力操作の回数

**SYNC DISK I/O PER SECOND (同期ディスク入出力/秒)**

(構成要素) 1 秒当りの同期ディスク入出力操作の平均数

### **SYNC DISK I/O REQUESTS (同期ディスク入出要求)**

(トランザクション) 優先順位、ジョブ・タイプ、およびプールの所定の組み合わせの同期ディスク入出力要求の合計数

### **SYNC DISK I/O RQS/TNS (同期ディスク入出力要求 /TNS)**

(トランザクション) 次の 5 つの欄は、トランザクションごとの同期ディスク入出力要求の数についての情報を示します。

#### **DB READ (DB 読み取り)**

トランザクションごとの同期データベース読み取り要求の平均回数

#### **DB WRITE (DB 書き込み)**

トランザクションごとの同期データベース書き込み要求の平均回数

#### **NDB READ (NDB 読み取り)**

トランザクションごとの同期非データベース読み取り要求の平均回数

#### **NDB WRITE (NDB 書き込み)**

トランザクションごとの同期非データベース書き込み要求の平均回数

#### **SUM (合計)**

同期 DB 読み取り、DB 書き込み、NDB 読み取り、および NDB 書き込み要求の平均回数の合計 (ジョブのトランザクションごとの同期入出力要求の平均数)

### **SYNC I/O /ELP SEC (同期入出力/経過秒数)**

(トランザクション) ジョブが使用した経過時間 1 秒当りの、すべてのジョブに関する同期ディスク入出力要求の平均回数

### **SYNC I/O /SEC (同期入出力/秒)**

(ジョブ間隔) 間隔中にジョブによって実行された同期ディスク入出力操作の 1 秒当り平均回数。これは、同期ディスク入出力カウントを経過時間で割って算出されます。

### **SYNC I/O PER SECOND (同期入出力/秒)**

(ジョブ間隔) 間隔中に選択された非対話式ジョブによって実行された同期ディスク入出力操作の 1 秒当りの平均回数

### **SYNCHRONOUS DBR (同期 DBR)**

(システム、トランザクション、ジョブ間隔、プール間隔) 同期データベース読み取り操作の平均回数。同期データベース読み取り合計数をトランザクション合計数で割ったものです。プール間隔報告書およびジョブ間隔報告書では、間隔中のジョブのトランザクションごとに計算されます。システム報告書では、秒当りで計算されます。トランザクション (ジョブ要約) では、トランザクションごとに計算されます。平均 DIO/トランザクションの下にリストされ、トランザクションごとの同期データベース読み取り要求の平均回数です。このフィールドは、システム中のジョブがトランザクションを処理しなかった場合は、印刷されません。

### **SYNCHRONOUS DBW (同期 DBW)**

(システム、トランザクション、ジョブ間隔、プール間隔) 同期データベース書き込み操作の平均回数。同期データベース書き込み合計数をトランザクション合計数で割ったものです。プール間隔報告書およびジョブ間隔報告書では、間隔中のジョブのトランザクションごとに計算されます。システム報告書では、秒当りで計算されます。トランザクション (ジョブ要約) では、トランザクションごとに計算されます。平均 DIO/トランザクションの下にリストされ、トランザクションごとの同期データベース読み取り要求の平均回数です。このフィールドは、システム中のジョブがトランザクションを処理しなかった場合は、印刷されません。

**SYNCHRONOUS DIO / ACT SEC (同期 DIO/活動秒)**

(システム、トランザクション) 活動秒当たりの同期ディスク入出力操作の回数。活動時間は、経過時間から待機時間を引いた時間です。

**SYNCHRONOUS DIO / DED SEC (同期 DIO/専用秒)**

(トランザクション) ジョブが専用モードで実行されたとした場合の同期ディスク入出力操作の 1 秒当りの見積数。専用モードは、他のジョブが活動していないか、あるいはシステムのリソースの競合状態がないことを意味します。

**SYNCHRONOUS DIO / ELP SEC (同期 DIO/経過秒数)**

(トランザクション) 経過秒当たりの同期ディスク入出力操作の回数

**SYNCHRONOUS DISK I/O COUNTS (同期ディスク入出力カウント)**

(トランザクション) 次の 5 つの欄は、トランザクションごとの同期ディスク入出力要求の数についての情報を示します。

**DB READ (DB 読み取り)**

トランザクションごとの同期データベース読み取り要求の回数

**DB WRT (DB 書き込み)**

トランザクションごとの同期データベース書き込み要求の回数

**NDB READ (NDB 読み取り)**

トランザクションごとの同期非データベース読み取り要求の回数

**NDB WRT (NDB 書き込み)**

トランザクションごとの同期非データベース書き込み要求の回数

**SUM (合計)**

同期 DB 読み取り、DB 書き込み、NDB 読み取り、および NDB 書き込み要求の合計  
(トランザクションごとの同期入出力要求の回数)

**SYNCHRONOUS DISK I/O PER TRANSACTION (トランザクションごとの同期ディスク入出力)**

(システム、トランザクション) 対話式トランザクションごとの同期物理ディスク入出力操作の平均回数

**SYNCHRONOUS MAX (同期最大)**

(トランザクション) 単一トランザクションでそのジョブによって発生した同期 DBR、NDBR、および WRT 入出力要求の最大数。ジョブが対話式または自動開始のジョブ・タイプでなければ、ジョブに対する合計ディスク入出力がここにリストされます。

**SYNCHRONOUS NDBR (同期 NDBR)**

(システム、トランザクション、ジョブ間隔、プール間隔) 間隔中のシステム内のジョブのトランザクションごとの同期非データベース読み取り操作の平均回数。トランザクション報告書では、プール内の選択されたジョブのトランザクションごとのディスクに対する操作の回数。これは、同期非データベース読み取りカウントを、処理されたトランザクション数で割って算出されます。このフィールドは、システム中のジョブがトランザクションを処理しなかった場合は、印刷されません。

**SYNCHRONOUS NDBW (同期 NDBW)**

(システム、ジョブ間隔、プール間隔) プール内の選択されたジョブのトランザクションごとのディスクに対する同期非データベース書き込み操作の平均回数。システム報告書では、間隔中のシステム内のジョブのトランザクションごとの操作の回数です。これは、同期非データベース書き込みカウントを、処理されたトランザクション数で割って算出されます。このフィールドは、システム中のジョブがトランザクションを処理しなかった場合は、印刷されません。

### **SYNCHRONOUS SUM (同期合計)**

(トランザクション) 同期 DBR、NDBR、および WRT 要求の平均回数の合計 (ジョブのトランザクションごとの同期入出力要求の平均回数)

### **SYNCHRONOUS WRT (同期書き込み)**

(トランザクション) トランザクションごとの同期データベースおよび非データベース書き込み要求の平均回数

### **SYSTEM CPU PER TRANSACTION (SECONDS) (システム CPU/トランザクション (秒))**

(システム) 対話式トランザクションごとのシステム処理装置の平均秒数

### **SYSTEM DISK I/O PER TRANSACTION (システム・ディスク入出力/トランザクション)**

(システム) 対話式トランザクションごとの、システムによる物理ディスク入出力操作の合計回数

### **SYSTEM STARTS (システム開始)**

(構成要素) システムによって開始されたジャーナル開始操作の回数

### **SYSTEM STOPS (システム停止)**

(構成要素) システムによって開始されたジャーナル停止操作の回数

### **SYSTEM TOTAL (システム合計)**

(構成要素) システムでジャーナル処理されたオブジェクトからの結果、保管されたジャーナルの合計数。これらは、システム管理のアクセス・パス保護 (SMAPP) によって実行された保管です。

### **SYSTEM TOUSER (システムからユーザー)**

(構成要素) システムによりジャーナル処理されたオブジェクトからユーザー作成ジャーナルへの結果、保管されたジャーナルの数

**SZWG** (トランザクション) 待機コード欄にリストされる認可占有待機。ジョブは占有競合で待機していました。元の保持者は、オブジェクトに対して持っていたロックを解放し、ロックは待機中のジョブに認可されました。オブジェクトを待機していたジョブは、この行 (WAITER --) に名前が示され、そのジョブが占有競合の解放の待機に費やされた時間も同時に示されます。保留されたオブジェクトは、次の報告書行に示されます (OBJECT --)。

**SZWT** (トランザクション) 待機コード欄にリストされる占有/ロック競合待機。ジョブは占有/ロック競合で待機しています。時間 (\* / time / \*) は、占有/ロック競合の時間で、報告書でこれに続く活動時間に含まれます。ロックの保持者は、報告書行の右に示されます (HOLDER --)。保留されたオブジェクトは、次の報告書行に示されます (OBJECT --)。

### **TERASPACE EAO (テラ・スペース EAO)**

(構成要素) 例外発生の要約と間隔カウントにリストされている。16 境界をクロスするテラ・スペース・アドレスを計算する際に発生するテラ・スペース有効アドレス・オーバーフロー (EAO)。簡易見積もりによれば、1 秒当たり 2,300 の EAO があれば、性能が 1% 低下します。

### **THREAD (スレッド)**

(ジョブ要約、トランザクション、移行) スレッドとは、プロセスの中の 1 つの固有な制御の流れのことです。すべてのジョブは、関連した初期スレッドを持ちます。すべてのジョブは、1 つまたは複数の 2 次スレッドを開始することができます。システムは、ジョブに対して次のようにスレッド番号を割り当てます。

- システムはスレッド識別コードを連続して割り当てます。以前に活動状態であったジョブ構造を使用するジョブが開始されると、初期スレッドに対して割り当てられるスレッド識別コードは一連番号の中の次の番号になります。
- ジョブの最初のスレッドに番号が割り当てられます。

- 同一ジョブに由来する他の追加スレッドに割り当てられる番号は、1 ずつ増えてゆきます。たとえば、

ジョブ	ユーザー名/ 名	ジョブ スレッド	番号
QJVACMSRV	SMITH	023416	
QJVACMSRV	00000006	023416	
QJVACMSRV	00000007	023416	
QJVACMSRV	00000008	023416	

1 より大きいスレッド値は、必ずしもそのジョブに同時に活動状態の多数のスレッドが存在するという意味にはなりません。同一ジョブについて現在活動状態となっているスレッドの数を判別するには、WRKACTJOB、WRKSBSJOB、または WRKUSRJOB コマンドを使用して、同一ジョブ名に付いている複数の 3 つの部分から成る識別コードを探します。

#### THREADS ACTIVE (活動状態のスレッド)

(システム) データのサンプリング時に活動していたスレッドの数。

#### THREADS IDLE (活動停止中のスレッド)

(システム) データのサンプリング時に活動停止中のスレッドの数。

#### TIME (時刻)

(トランザクション) トランザクションが完了した時刻、あるいは占有またはロック競合が起こった時刻。また、1 つの状態から別の状態への移行が起こった時刻を示す欄見出しで、HH.MM.SS.mmm の配列。

#### TIME (時刻)

(ジョブ・トレース) トレース項目の時刻。時刻は、時間、分、秒、およびマイクロ秒の順で表示されます。

**TNS** (構成要素、プール間隔) プールまたはサブシステムの中で選択されたジョブによって処理された、トランザクションの合計数

#### TNS COUNT (TNS カウント)

(構成要素、ジョブ間隔) 間隔中に選択された対話式ジョブによって実行されたトランザクションの数

#### TNS/HOUR (TNS/時間)

(構成要素、トランザクション、ジョブ間隔) 間隔中に選択された対話式ジョブによって処理された 1 時間当りのトランザクションの平均数

#### TNS/HOUR RATE (TNS/時間率)

(システム) 1 時間当りのトランザクションの平均数

#### TOD OF WAIT (待機の TOD)

(ロック) 競合の開始時刻

#### TOT (合計)

(トランザクション) 物理入出力カウント欄にリストされ、DB 読み取り、DB 書き込み、NDB 読み取り、および NDB 書き込み要求の合計回数

#### TOT NBR TNS (合計トランザクション数)

(トランザクション) PRTTNSRPT プログラムがジョブに関して達成した入力データから判別したトランザクションの合計数

#### TOTAL (合計)

(構成) 報告期間中の合計例外カウント

**TOTAL (合計)**

(ジョブ・トレース) フィールドの合計。

**TOTAL /JOB (合計/ジョブ)**

(トランザクション) ジョブの欄の項目の合計 (和)

**TOTAL CHARACTERS PER TRANSACTION (トランザクション当りの合計文字数)**

(システム) 1 つの対話式トランザクション当りの表示装置の画面から読み取られたか、画面に書き出された文字の平均数

**TOTAL CPU SEC /SYNC DIO (合計 CPU 秒/同期 DIO)**

(トランザクション) 合計 CPU 秒数を同期ディスク入出力要求の合計で割った率

**TOTAL CPU UTILIZATION (合計 CPU 使用率)**

(システム、構成要素) 区画が使用した、使用可能な処理装置時間のパーセント。複数処理装置システムの場合、これはすべての処理装置を通算した平均使用率です。専用区画の場合は、CPU 使用率合計は、区画内の各処理装置の使用率によって置き換えられます。下に示す例は、2 つの処理装置を持つ専用区画の場合の画面のこの部分です。

平均 CPU 使用率 . . . . .	: 41.9
CPU 1 使用率 . . . . .	: 41.7
CPU 2 使用率 . . . . .	: 42.2

共用プロセッサ区画の場合、個々の CPU 使用率の行は印刷されません。

注: この値はシステム・カウンターから取られます。他の処理装置率は個々のジョブの作業制御ブロック (WCB) から取られます。これらの合計は、わずかながら異なる場合があります。上限のない (uncapped) 区画では、合計 CPU 使用率は 100 パーセントを超えることがあります。

**TOTAL CPU UTILIZATION (DATABASE CAPABILITY) (CPU 使用率合計 (データベース機能))**

(システム) システム上の DB2 Universal Database™ 活動を表示します。このフィールドは、V4R5 以降のリリースを実行するすべてのシステムに適用され、すべての SQL およびデータ入出力操作を含む、すべてのデータベース活動を含みます。

**TOTAL CPU UTILIZATION (INTERACTIVE FEATURE) (CPU 使用率合計 (対話式機構))**

(システム) CPU 使用率 (対話式機構) は、対話式作業に関するシステムの容量に関連して 5250 ワークステーション入出力操作を行うすべてのジョブに対して CPU 使用率を表示します。システムと購入された関連機構に応じて、対話式容量は、システムの合計容量に等しいかそれ以下になります。

**TOTAL DATA CHARACTERS RECEIVED (受信データ文字合計)**

(リソース間隔) 正常に受信されたデータ文字数

**TOTAL DATA CHARACTERS TRANSMITTED (送信データ文字合計)**

(リソース間隔) 正常に送信されたデータ文字数

**TOTAL DATAGRAMS REQUESTED FOR TRANSMISSION (送信用に要求された合計データグラム)**

(構成要素) 次の理由によって廃棄される IP データグラムのパーセント

- データグラムを宛先に送信するための経路が検出されなかった
- バッファ・スペースの不足

**TOTAL FIELD PER TRANSACTION (トランザクション当りの合計フィールド数)**

(システム) 1 つの対話式トランザクション当りの読み取りまたは書き込みが行われた表示装置フィールドの平均数

**TOTAL FRAMES RECD (受信フレーム合計)**

(リソース間隔) 受信したフレームの数。エラーのあったフレームおよび有効でないフレームも含まれます。

**TOTAL I FRAMES TRNSMITD (送信 I フレーム合計)**

(リソース間隔) 送信された情報フレームの合計数

**TOTAL I/O (入出力合計)**

(システム) 読み取り操作および書き込み操作の合計回数

**TOTAL PDUS RECEIVED (受信 PDU 合計)**

(リソース間隔) 時間間隔中に受信したプロトコル・データ単位 (PDU) の数

注: 非同期通信のプロトコル・データ単位 (PDU) は、終わりがプロトコル制御文字またはバッファのサイズになる可変長データ単位です。

**TOTAL PHYSICAL I/O PER SECOND (物理入出力合計/秒)**

(リソース間隔) ディスク・アームによって実行された 1 秒当りの物理ディスク入出力操作の平均回数

**TOTAL RESPONSES (応答合計)**

(構成要素、リソース間隔) 報告期間中にこの制御装置上でカウントされたトランザクションの合計数、ならびにすべての活動ワークステーションおよび装置の平均応答時間

**TOTAL SEIZE/WAIT TIME (占有/待機の合計時間)**

(構成要素) 各ジョブの応答時間 (ミリ秒)

**TOTAL TNS (トランザクション合計)**

(構成要素) このプールで処理されたトランザクションの数

**TRANSACTION RESPONSE TIME (SEC/TNS) (トランザクション応答時間 (秒/TNS))**

(トランザクション) トランザクションごとの応答時間 (秒数)。この値には、通信回線時間は含まれません。ワークステーションで測定された応答時間は、データ伝送時間 (データをワークステーションから処理装置に伝送し、応答データを処理装置からワークステーションに戻すための時間) だけ、この時間より大きくなります。

**TRANSACTIONS PER HOUR (LOCAL) (トランザクション/時間 (ローカル))**

(システム) ローカル表示装置による 1 時間当りの対話式トランザクション数

**TRANSACTIONS PER HOUR (REMOTE) (トランザクション/時間 (遠隔))**

(システム) 遠隔表示装置による 1 時間当りの対話式トランザクション数

**TRANSIENT SIZE**

(構成要素) ジャーナル一時領域に配置されるキロバイト。これらは、システムが作成する隠しジャーナル項目です。

**TRANSMIT/RECEIVE/AVERAGE LINE UTIL (送信/受信/平均回線使用率)**

(リソース間隔) 全二重モードで、使用された送信回線容量のパーセント、使用された受信回線容量のパーセント、および送受信容量の平均

**TSE** (トランザクション) 待機コード欄にリストされるタイム・スライス終了。LAST のラベルのスタック項目に示されているプログラムが、タイム・スライス終了に移ったプログラムです。

**TYP** (構成要素、トランザクション) システム・ジョブ・タイプおよびサブタイプ。構成要素報告書では、この欄には 1 文字しか許されません。トランザクション報告書では 2 文字が許されます。トランザクション報告書では、QAPMJOBS フィールドから直接、ジョブ・タイプおよびジョブ・サブタイプが報告されます。構成要素報告書は、ジョブ・タイプおよびジョブ・サブタイプの値をと

ってから文字に変換しますが、この文字は QAPMJOBS フィールドの値になる場合もそうでない場合もあります。ジョブのタイプは、次のとおりです。

- A** 自動開始
- B** バッチ
- BD** バッチ即時 (トランザクションのみ)  
  
注: バッチ即時値は、「活動ジョブの処理」画面では BCI として、また「サブシステム・ジョブの処理」画面では BATCHI として表示されます。
- BE** バッチ呼び出し (トランザクションのみ)
- BJ** バッチ事前開始ジョブ (トランザクションのみ)
- C** APPC 上の 5250 エミュレーション、および APPC または TCP/IP のいずれかを実行している System i アクセス・ホスト・サーバーを含む、プログラム式ワークステーションのアプリケーション・サーバー。詳しくは、『ホスト・サーバー管理』のトピックを参照してください。次の事項のいずれかが真である場合、ジョブは System i アクセス・サーバーとして報告されます。
  - 着信 APPC 呼び出しが、サーバー・プログラム名のいずれかを要求している場合。これは、指定のプログラムをすでに待機している、QSERVER、QCMN、および QSYSWRK サブシステムの事前開始ジョブに対しても適用されます。
  - 着信 IP ポート番号が、サービスの名前 - 記述 - ポート - 番号のいずれかに対応する場合。これは、割り当てられた IP ポート番号をすでに待機している、QSERVER、QCMN、および QSYSWRK サブシステムの事前開始ジョブに対しても適用されます。
  - 着信 IPX ソケット番号が、サービスの名前 - 記述 - ポート - 番号のいずれかに対応する場合。これは、割り当てられた IPX ポート番号をすでに待機している、QSERVER、QCMN、および QSYSWRK サブシステムの事前開始ジョブに対しても適用されます。
  - OS/2 コミュニケーション・マネージャーまたは WARP 同等機能のもとでの 5250 エミュレーションによって送信された APPC データ・ストリームに由来する、着信 5250 表示装置エミュレーション・ジョブ。
- D** ターゲット分散データ管理 (DDM) サーバー
- I** 対話式。構成要素報告書の場合、これには平衡型データ・リンク制御 (TDLC)、5250 遠隔ワークステーション、および 3270 遠隔ワークステーションを含みます。トランザクション報告書の場合、これには平衡型データ・リンク制御 (TDLC)、5250 遠隔ワークステーション、3270 遠隔ワークステーション、SNA パススルー、および 5250 Telnet を含みません。
- L** ライセンス内部コード・タスク
- M** サブシステム・モニター
- P** SNA パススルーおよび 5250 Telnet パススルー。トランザクション報告書では、これらのジョブは I (対話式) として表示されます。
- R** スプール読み取りプログラム
- S** システム
- W** スプール書き出しジョブ、および高機能印刷 (AFP) が指定されている場合は印刷ドライバー・ジョブを含む、スプール書き出しプログラム。



**WP** スプール印刷ドライバー (トランザクションのみ)

**X** システムの開始

ジョブのサブタイプは、次のとおりです。

**D** 即時のバッチ・ジョブ

**E** 呼び出し (通信バッチ)

**J** 事前開始ジョブ

**P** 印刷ドライバー・ジョブ

**T** 複数要求端末 (MRT) (System/36 環境のみ)

**3** System/36

注:

1. ジョブ・サブタイプは、構成要素報告書では表示されません。
2. ジョブ・タイプがブランクの場合、またはジョブ・タイプの割り当てを変更したい場合は、ジョブ・タイプ変更 (CHGJOB TYP) コマンドを用いて該当のジョブ・タイプを割り当ててください。

**タイプ** (System、Transaction、Job Interval (システム、トランザクション、ジョブ間隔)) の DTNTY フィールドの記述にリストされているトランザクション・タイプの 1 つ

**(System)**

ディスク・タイプ

**(Transaction)**

ジョブのタイプおよびサブタイプ

**(Transaction)**

オブジェクト・セクションによる占有/ロック競合の場合、占有/ロック競合のタイプ

**UDP DATAGRAMS RECEIVED (受信 UDP データグラム)**

(構成要素) UDP ユーザーに配信される User Datagram Protocol (UDP) データグラムの合計数

**DATAGRAMS SENT (送信 UDP データグラム)**

(構成要素) このエンティティから送信される User Datagram Protocol (UDP) データグラムの合計数

**UNCAP CPU AVAIL (使用可能な上限なし CPU)**

(構成要素) 構成済みの CPU に加え、間隔中に共用プロセッサ・プールの区画に対して使用可能な CPU 時間のパーセント。この値は、特定の区画で使用可能な構成済み CPU に対して相対的なものです。

**UNICAST PACKETS RECEIVED (受信ユニキャスト・パケット)**

(システム) 高位層プロトコルに配信されるサブネットワーク・ユニキャスト・パケットの合計数。この数値には、指定されたインターフェース上で受信されたパケットのみが含まれています。

**UNICAST PACKETS SENT (送信ユニキャスト・パケット)**

(システム) 高水準プロトコルが、サブネットワーク・ユニキャスト・アドレスに送信するように要求したパケットの合計数。この数値には、廃棄されたパケットや送信されなかったパケットが含まれています。

**UNIT (装置)**

(システム、構成要素、リソース間隔) 特定のディスク装置またはアームを識別するためにシステム

によって割り当てられた番号。装置番号の後の A または B は、そのディスク装置がミラー保護されていることを示します。(たとえば、0001A および 0001B はミラー保護ディスクの対です。)

#### UNIT NAME (装置名)

ディスク・アームのリソース名

#### USER ID (ユーザー ID)

(システム、構成要素、トランザクション、ジョブ間隔、プール) 含める (SLTUSRID パラメーター) または除外する (OMTUSRID パラメーター) ことにより選択されたユーザーのリスト。

#### USER NAME (ユーザー名)

(構成要素、トランザクション、ジョブ間隔、バッチ・ジョブ追跡) 関与した (ジョブを実行依頼した、競合したなど) ユーザーの名前

#### USER NAME/THREAD (ユーザー名/スレッド)

(コンポーネント、トランザクション) ジョブ情報に 2 次スレッドが含まれる場合は、この欄にはスレッド識別コードが表示されます。ジョブ通知に 2 次スレッドが含まれない場合は、この欄にはユーザー名が表示されます。システムは、ジョブに対して次のようにスレッド番号を割り当てます。

- システムはスレッド識別コードを連続して割り当てます。以前に活動状態であったジョブ構造を使用するジョブが開始されると、初期スレッドに対して割り当てられるスレッド識別コードは一連番号の中の次の番号になります。
- ジョブの最初のスレッドに番号が割り当てられます。
- 同一ジョブに由来する他の追加スレッドに割り当てられる番号は、1 ずつ増えてゆきます。たとえば、

ジョブ	ユーザー名/ ジョブ名	スレッド	番号
QJVACMSRV	SMITH	023416	
QJVACMSRV	00000006	023416	
QJVACMSRV	00000007	023416	
QJVACMSRV	00000008	023416	

1 より大きいスレッド値は、必ずしもそのジョブに同時に活動状態の多数のスレッドが存在するという意味にはなりません。同一ジョブについて現在活動状態となっているスレッドの数を判別するには、WRKACTJOB、WRKSBSJOB、または WRKUSRJOB コマンドを使用して、同一ジョブ名に付いている複数の 3 つの部分から成る識別コードを探します。

#### USER STARTS (ユーザー開始)

(構成要素) ユーザーによって開始されたジャーナル開始操作の回数

#### USER STOPS (ユーザー停止)

(構成要素) ユーザーによって開始されたジャーナル停止操作の回数

#### USER TOTAL (ユーザー合計)

(構成要素) システムでジャーナル処理されたオブジェクトからの結果、保管されたジャーナルの合計数。

#### UTIL (使用率)

(構成要素、リソース間隔) 各ローカル・ワークステーション、ディスク、または通信 IOP、制御装置、または駆動機構の使用率のパーセント

注: システム全体の平均使用率には、測定間隔におけるミラー保護のアームのデータは含まれません。ミラー保護アームの場合、それらの間隔は再開状況か、中断状況のいずれかにあります。

## UTIL 2 (使用率 2)

(構成要素、リソース) コプロセッサの使用率

## VALUE (値)

(トランザクション) ジョブ要約報告書の個別トランザクション統計セクションでは、トランザクションに関して比較されるデータの値。最長の占有/ロック競合セクションでは、占有またはロック競合が起こった秒数。

## VERIFY (検査)

(構成要素) 1 秒当りの検査例外の数。検査例外は、ポインターの分析解決が必要な場合、ブロックされた MI 命令が機密保護レベル 10、20、または 30 で使用された場合、および解決されていない記号名が呼び出された場合に起こります。このカウントは、通常のシステム操作の場合でも非常に高くなる場合があります。このカウントをモニターとして使用してください。大きな変動または変化があった場合は、これを詳細に調べてください。

**VP** (システム) 報告された区画内で活動状態の仮想プロセッサの数。

## 仮想共用プロセッサ・プール ID

(システム) 仮想共用プロセッサ・プール ID。この列は、i5/OS 区画についてのみ印刷されます。

## W-I WAIT/TNS (W-I 待機/TNS)

(トランザクション) トランザクションごとの待機状態から不適格状態への移行の平均時間 (秒数)。この値は、活動レベルが応答時間に与える影響を示します。この値が低い場合は、待機状態から不適格状態への移行の数が応答時間にほとんど影響しないことになります。この値が高い場合は、対話式プールの記憶域を追加し、対話式プールの活動レベルを増やすことによって、応答時間が改善されるはずですが、対話式プールの記憶域を大きくすることができない (使用可能な記憶域が限定されているため) 場合は、活動レベルを大きくすると応答時間が改善される可能性があります。ただし、活動レベルを大きくすると、その記憶域プール内の不在ページが増えることがあります。

## WAIT CODE (待機コード)

(トランザクション) 追跡レコードが作成される原因となるジョブ状態の移行。値は次のとおりです。

**EVT** 事象待機。メッセージ・キューでの待機時に起こる長時間待機。

**EOTn** タイプ n のトランザクションのトランザクション終了。これらのコードは待機コード欄に入っていますが、待機コードではありません。これらはトランザクション境界追跡レコードを示します。

**EORn** トランザクション n の応答時間の終わりを示します。これらのコードは、待機コード欄に示されますが、待機コードではありません。これらはトランザクション境界追跡レコードを示します。

## ERROR RESPONSE (エラー応答)

(構成要素) エラーになった応答の数

**HDW** 保留待機 (延期されたジョブまたはシステム要求)

**LKRL** ロック解放。ジョブは、報告書の次の明細行 (OBJECT --) に示されたオブジェクトに対して持っていたロックを解放しました。オブジェクトを待機していたジョブは、この行 (WAITER --) に名前が示され、そのジョブがロックの解放の待機に費やされた時間も同時に示されます。

**LKW** ロック待機。これが多数ある場合、または ACTIVE/RSP\* の欄に時間の非常に長い項目が

ある場合は、追加の分析が必要です。この LKW 報告書行の前の LKWT 報告書行が、待機の対象のオブジェクトおよびそのオブジェクトの所有者を示します。

#### LKWT

ロック競合待機。ジョブはロック競合で待機しています。時間 (\* / time /\*) は、ロック競合の期間で、LKW 時間と同じではないにしても、これに非常に近いはずですが、ロックの保持者は、報告書行の右に示されます (HOLDER --)。ロックされたオブジェクトは、次の報告書行に示されます (OBJECT --)。

**SOTn** トランザクション n の開始を示します。これらのコードは、待機コード欄に示されますが、待機コードではありません。これらはトランザクション境界追跡レコードを示します。

**SWX** 拡張短時間待機時間。短期待機は 2 秒の限界を超過し、システムはトランザクションを長期待機に入れました。この長期待機はトランザクション応答時間に入れなければなりません。言い換えると、活動状態から待機状態に移行したトランザクションは、トランザクション境界を反映しません。

**SZWG** (トランザクション) 待機コード欄にリストされる認可占有待機。ジョブは占有競合で待機していました。元の保持者は、オブジェクトに対して持っていたロックを解放し、ロックは待機中のジョブに認可されました。オブジェクトを待機していたジョブは、この行 (WAITER --) に名前が示され、そのジョブが占有競合の解放の待機に費やされた時間も同時に示されます。保留されたオブジェクトは、次の報告書行に示されます (OBJECT --)。

**SZWT** 占有/ロック競合待機。ジョブは占有/ロック競合で待機しています。時間 (\* / time /\*) は、占有/ロック競合の時間で、報告書でこれに続く活動時間に含まれます。ロックの保持者は、報告書行の右に示されます (HOLDER --)。保留されたオブジェクトは、次の報告書行に示されます (OBJECT --)。

**TSE** タイム・スライス終了。LAST のラベルのスタック項目に示されているプログラムが、タイム・スライス終了に移ったプログラムです。長期待機の間ジョブが 0.5 秒 (高速処理装置では 0.2 秒) の CPU 時間を使用するごとに、システムは CPU 待ち行列に等しい優先順位をもつジョブがあるかどうかを調べます。ある場合は、等しい優先順位をもつ次のジョブに CPU の使用权を与え、割り込まれたジョブを待ち行列の等しい優先順位の最後の位置に移動します。ただし、そのジョブは活動レベルを維持します。これは内部のタイム・スライス終了です。ジョブが外部タイム・スライス値に達した場合は、別のジョブが活動レベルを待機していれば、活動状態から不適格状態への状態移行が起こることがあります。ジョブが強制的に活動レベルをはずされる場合、そのページは別のジョブによって奪取され、そのジョブが再び活動レベルを入手する時点でさらに別の入出力が発生することになります。対話式ジョブでは 2 秒、バッチ・ジョブでは 5 秒の IBM 提供のデフォルト値は、特にハイエンドの処理装置では長すぎることがあります。初期値として、トランザクションごとの平均 CPU 秒数の 3 倍に、タイム・スライスを設定するようにしてください。

**WTO** 待機タイムアウト。ジョブは、待機 (たとえば、ロック、メッセージ・キュー、またはレコードに対する待機) に定められた待機タイムアウト限度を超過しました。

#### WAITS

(ジョブ・トレース) 発生した待機状態の回数。

#### WAIT-ACT

(ジョブ・トレース) ジョブ追跡分析要約では、これは ENDTNS と STRTNS プログラムの間で、

WAIT-ACT とラベルされている時間です。対話式ジョブを追跡していて、デフォルトの STRTNS および ENDTNS パラメーターを使用している場合は、この値はトランザクションを処理するのにかかった時間です。

ジョブ追跡分析 I/O 要約では、これはジョブが活動状態でなかった (おそらく、ユーザーの入力または考慮時間が原因) 時間です。

**WAIT-INEL (待機 - 不適格)**

(システム、構成要素) 待機状態から不適格状態へのジョブの状態移行の 1 分当りの平均回数

**WORK STATION CONTROLLER (ワークステーション制御装置)**

(リソース間隔) 遠隔ワークステーション制御装置の名前

**WRITES (書き込み)**

(ジョブ・トレース) 発生した物理的書き込み回数。

**WRITES PER SECOND (書き込み/秒)**

(リソース間隔) ディスク・アームによって実行された 1 秒当りのディスク書き込み操作の平均回数

**WRITTEN**

(ジョブ・トレース) 項目に対し発生した、物理的書き込み回数。

**WTO** (トランザクション) 待機コード欄にリストされる待機タイムアウト。ジョブは、待機 (たとえば、ロック、メッセージ・キュー、またはレコードに対する待機) に定められた待機タイムアウト限度を超過しました。

**0.0-1.0** (構成要素、リソース間隔) 応答時間が 0-1 秒であった回数。

**1.0-2.0** (構成要素、リソース間隔) 応答時間が 1-2 秒であった回数。

**2.0-4.0** (構成要素、リソース間隔) 応答時間が 2-4 秒であった回数。

**4.0-8.0** (構成要素、リソース間隔) 応答時間が 4-8 秒であった回数。

**パフォーマンス報告書の見出し:**

各報告書には、そのタイプやセクションには関係なく、データの特性を示す情報が報告書の見出しに含まれます。ここでは見出し情報を説明します。

**報告書のタイトル**

最初の行ではパフォーマンス報告のタイプを識別します。2 行目は、報告書のセクションを示しています。

**Current date and time (現在の日付と時刻)**

この報告書が印刷された日付と時刻を示します。

**Report page number (報告書のページ番号)**

報告書のページを示しています。

**Perf data from time to time at interval (「間隔」での「時間」から「時間」までのパフォーマンス)**

データが収集された時間および間隔を示します。

**User-selected report title (ユーザー選択の報告書タイトル)**

ユーザーにより指定されたこの報告書の名前を示します。

**Member (メンバー)**

この報告書で使用されたパフォーマンス・データ・メンバーを示します。この名前は、パフォーマンス・データ作成 (CRTPFRDTA) コマンドの MBR パラメーターに指定した名前に対応しています。

### Library (ライブラリー)

特定の報告書で使用されるパフォーマンス・データが入っているライブラリーを示します。

### Model/Serial (モデル/シリアル)

報告書用のパフォーマンス・データが収集されたサーバーのモデルおよびシリアル番号です。シリアル番号は 10 文字で示されます。

### Main storage size (主記憶域サイズ)

パフォーマンス・データが収集されたサーバーの主記憶域のサイズを示します。

### Started (開始)

収集サービスが、報告書のパフォーマンス・データの収集を開始した日付と時刻を示します。特定の間隔または特定の開始時刻を選択するか否かによって、以下が示されます。

- 報告書作成を実行する間隔を指定しなかった場合、開始日付と時刻は、データが収集された日付と時刻になります。
- 報告書作成を実行する特定の間隔を指定した場合、開始日付と時刻は、データが収集された日付と時刻になります。

注: システム報告書の場合に限り、報告書選択基準セクションを調べて、選択されている間隔を調べる必要があります。

### Stopped (停止)

収集サービスが、報告書のパフォーマンス・データの収集を停止した日付と時刻を示します。特定の間隔または特定の停止時刻を選択するか否かによって、以下が示されます。

- 報告書を実行する間隔を指定しなかった場合、停止日付と時刻は、データが収集された日付と時刻になります。
- 報告書を実行する特定の間隔を指定した場合、停止日付と時刻は、データが収集された日付と時刻になります。

注: システム報告書の場合に限り、報告書選択基準セクションを調べて、選択されている間隔を調べる必要があります。

### System name (システム名)

報告書で使用されたパフォーマンス・データが収集されたサーバーの名前を示します。

### Version/Release level (バージョン/リリース・レベル)

x/ x.0 は、サーバーがパフォーマンス・データの収集時に稼働していたオペレーティング・システムのバージョンおよびリリース・レベルを示しています。

### Partition ID (区画 ID)

収集が実行された区画の ID を識別します。この変更によって、論理区画を取り扱えるようになりました。表示される値は、次のとおりです。

- 使用中のシステムが区画に分割されていない場合 (これがデフォルトの解釈です)、または収集サービスを使用して論理区画システムの 1 次区画のパフォーマンス・データを収集して印刷した場合は、この値は 00 です。
- 前のリリースにおいてパフォーマンス・モニター開始 (STRPFRMON) コマンドを用いてデータを収集した場合は、区画 ID の値は 00 です。
- 収集サービスを使用して論理区画システムの任意の 2 次区画のパフォーマンス・データを収集して印刷した場合は、この値は、サービス・ツール開始 (STRSST) コマンドの下で「システム区画処理」画面上に表示される区画 ID と同じです。

### Feature Code (フィーチャー・コード)

サーバーの対話式フィーチャー・コード値を識別します。

### Int Threshold (対話式しきい値)

収集期間に使用された対話式作業に使用された、システム CPU 合計のパーセントを示します。値は QAPMCONF ファイル (GKEY IT) から取得され、収集開始時に獲得される構成メトリックを反映します。この値が、論理区画構成での動的変更のために、収集期間内に各間隔ごとに変更することがあるということに注意する必要があります。

### Virtual Processors (仮想プロセッサ)

区画用に構成された仮想プロセッサの数。値は QAPMCONF ファイル (GKEY 13) から取得され、収集開始時に獲得される構成メトリックを反映します。この値が、論理区画構成での動的変更のために、収集期間内に各間隔ごとに変更することがあるということに注意する必要があります。

### Processor Units (処理装置)

区画に割り振られた処理装置の数。値は QAPMCONF ファイル (GKEY PU) から取得され、収集開始時に獲得される構成メトリックを反映します。この値が、論理区画構成での動的変更のために、収集期間内に各間隔ごとに変更することがあるということに注意する必要があります。

処理装置は、1 つ以上の仮想プロセッサすべてにおける、共用処理能力の計測単位です。1 つの仮想プロセッサ上の 1 つの共用処理装置は、専用プロセッサとほぼ同じ作業を実行します。2 つの仮想プロセッサ上の 1 つの共用処理装置は、2 つの専用プロセッサの作業のほぼ半分を実行します。

### Column headings (欄の見出し)

各報告書には、その報告書の情報を構成する多くの欄が示されます。特定の報告書に特有の欄もあれば、どの報告書にも共通する欄もあります。これらの欄の簡単な説明については、パフォーマンス報告書の欄のページ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になることです。

### 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。

#### 状態

最近システム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。

#### 詳細

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。i5/OS および Performance Tools ライセンス・プログラム (5770-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 リソースが 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 System i 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 リソースが消費されている場合は、欠落ファイル呼び出しているか、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、システムおよびコンポーネント報告書を使用して、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・リソースを割り当てる。
  - 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、パフォーマンスを自動的に調整することができます。
3. ジョブ要約報告書を実行して、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。

## 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。

### 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リソースが、確実に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. System i 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) 「i5/OS コマンド (i5/OS command)」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) 「i5/OS コマンド (i5/OS command)」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) 「i5/OS コマンド (i5/OS command)」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: メッセージ・モニター

この例のメッセージ・モニターは、システム上で生じた、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。

### 状態

貴社に複数のサーバーがあり、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. System i 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システム上の 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 i5/OS コマンドを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

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。

### 状態

現在システム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. System i 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) 「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) 「i5/OS トリガー・コマンド (i5/OS 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

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。

## 状態

現在、システム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。

注: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

1. System i 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) 「**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) 「i5/OS トリガー・コマンド (i5/OS 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。

注: このコード例を使用することによって、お客様は 210 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとします。

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

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

DCL &MSGTXT *CHAR 132

RCVMSG 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 のキャパシティーが 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。

---

## パフォーマンスの関連情報

パフォーマンスのトピックと関連のある製品マニュアル および IBM Redbooks (PDF 形式)、Web サイト、および information center トピックをここにリストします。PDF は、どれでも表示または印刷することができます。

### 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。

- System i 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、キャパシティー・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。

### IBM Redbooks

- IBM i でのエンドツーエンド・パフォーマンス管理 (英語) 

この IBM Redbooks 資料に含まれるトピックは、パフォーマンス管理のサイクルをさらに理解する助けになるだけでなく、ヒントやベスト・プラクティスも提供します。収集サービス、Job Watcher、Disk Watcher、および Performance Explorer の各データ・コレクターに関する情報が記載されています。また、IBM Systems Director Navigator for i の一部として 6.1 から提供されようになった新しい Web ベースのグラフィカル・ユーザー・インターフェースを使用してパフォーマンス分析を最大限に高める方法についても説明しています。

- IBM Systems Director Navigator for i (英語) 

この新しい Web ベースのコンソールを使用して IBM i を管理する方法について説明しています。この新しいコンソールを使い始めるうえで役立つこと、およびコンソールのさまざまな部分を使用して作業するためのヒントを提供することを目的に書かれたものです。ネットワーク、データベース、パフォーマンス、ファイル・システム、Advanced Job Scheduler for IBM i、セキュリティ、統合サーバー管理など、組み込まれている個々のタスク (機能) の多くについて詳しく解説しています。

- Performance Management for IBM eServer™ iSeries and pSeries®: A Systems Management Guide 

この IBM Redbooks 資料のトピックでは、IBM Performance Management をシステム管理ストラテジーにおける基礎でありかつ重要な構成要素として扱うための、増大しつつあるニーズと勧告事項をサポートします。これは Performance Management オファリングをシステム管理ストラテジーの一部とし、対話式オファリングを使用し、オファリングの構成要素間でナビゲートする方法を説明しています。

- IBM eServer iSeries Performance Management Tools 

IBM iSeries Performance Management Tools の全ラインアップを取り上げています。この IBM Redpaper では、IBM i5/OS V5R3M0 レベルの入手可能である多様な Performance Management Tools を理解でき、それらをどのような場合に使用すべきかが説明されています。

- 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、System i 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 Web ベース・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その他の情報技術者を対象としています。

- 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>®</sup> UDB/WebSphere Performance Tuning Guide 

この資料は、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。

- 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設定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。

- Java and WebSphere Performance on IBM eserver iSeries Servers 

この資料では、Java および WebSphere Application Server のパフォーマンス関連問題を扱うための、ヒント、テクニック、および方法論を記載しています。

- Lotus<sup>®</sup> Domino for AS/400: 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、リソースの調整、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。


- Managing OS/400 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。

## Web サイト

- **Performance Management for IBM System i Resource Library**  ([www.ibm.com/servers/eserver/iserries/perfmgmt/resource.html](http://www.ibm.com/servers/eserver/iserries/perfmgmt/resource.html))

このライブラリーには、System i のパフォーマンスの専門家が執筆した、パフォーマンス参照資料、ホワイト・ペーパー、ベンチマーク・レポート、および業界誌の記事がまとめられています。

- **Performance Management for IBM System i**  ([www.ibm.com/servers/eserver/iserries/perfmgmt/](http://www.ibm.com/servers/eserver/iserries/perfmgmt/))

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。

## 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. PDF をローカルに保存するオプションをクリックする。
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 Reader がご使用のシステムにインストールされている必要があります。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))  から無償でダウンロードできます。

---

## コードに関するライセンス情報および特記事項

IBM 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権を許諾します。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します。



いかなる場合においても、IBM および IBM のサプライヤーならびに IBM ビジネス・パートナーは、その予見の有無を問わず発生した以下のものについて賠償責任を負いません。

1. データの喪失、または損傷。
2. 直接損害、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。



---

## 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。

〒242-8502  
神奈川県大和市下鶴間1623番14号  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

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 YBWA  
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。

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。サンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

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。

© (お客様の会社名) (西暦年). 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。 © Copyright IBM Corp. \_年を入れる\_.

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。

---

## プログラミング・インターフェース情報

本書「パフォーマンス」には、IBM i のサービスを利用するためのプログラムを、ユーザーが作成できるようにするためのプログラミング・インターフェースが記述されています。

---

## 商標

IBM、IBM ロゴおよび [ibm.com](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名は、IBM または各社の商標です。現時点での IBM の商標リストについては、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) の Copyright and trademark information をご覧ください。

Adobe、Adobe のロゴ、PostScript、および PostScript のロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標のいずれかです。

Intel、Intel Inside (ロゴ)、MMX、および Pentium は Intel Corporation または子会社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。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Java 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です。

Linux 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。

---

## 使用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

**個人使用:** これらの資料は、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、これらの資料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:** これらの資料は、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れらの資料の二次的著作物を作成したり、お客様の企業外で資料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、資料や資料内に含まれる情報、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。

資料の使用が 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、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、第三者の権利の不侵害の保証、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なしで提供されます。







Printed in Japan