

IBM

@server

iSeries

実行管理機能





@server

iSeries

実行管理機能

© Copyright International Business Machines Corporation 1998, 2002. All rights reserved.

© Copyright IBM Japan 2002

目次

| | |
|-----------------------------------|----|
| 実行管理機能 | 1 |
| 関連情報 | 1 |
| V5R2 の新機能 | 2 |
| トピックの印刷 | 3 |
| 日次作業の管理 | 4 |
| システム活動のモニター | 5 |
| システム状況の処理 | 6 |
| ジョブおよびスレッドの管理 | 7 |
| iSeries サーバー上でのジョブの検索 | 7 |
| ジョブの状況の判別 | 10 |
| ジョブのパフォーマンス統計の表示 | 11 |
| ジョブの終了 | 13 |
| ジョブ・アクション | 14 |
| 特定のジョブの下で実行しているスレッドの表示 | 16 |
| スレッド・プロパティの表示 | 16 |
| スレッドの削除または終了 | 16 |
| ジョブ待ち行列の管理 | 17 |
| ジョブ待ち行列上のジョブの表示 | 17 |
| ジョブ待ち行列内でのジョブの優先順位の変更 | 17 |
| 他のジョブ待ち行列へのジョブの移動 | 19 |
| サブシステムの管理 | 21 |
| メモリー・プール内のジョブ数のモニター | 21 |
| サブシステム内のジョブの表示 | 22 |
| サブシステムの開始 | 23 |
| サブシステムの停止 | 23 |
| メモリー・プールの管理 | 24 |
| メモリー・プールを使用したサブシステム数のモニター | 24 |
| メモリー・プールの使用状況の検査 | 25 |
| メモリー・プールのサイズの変更 | 26 |
| ジョブ・ログの管理 | 28 |
| サーバー・ジョブを含むアクティブ・ジョブのジョブ・ログへのアクセス | 28 |
| プリンター出力へのアクセス | 29 |
| 出力待ち行列の管理 | 30 |
| システム上の出力待ち行列の表示 | 30 |
| 出力待ち行列間および出力待ち行列内での出力の移動 | 31 |
| 出力待ち行列の消去 | 31 |
| システムの構造 | 32 |
| ジョブ | 33 |
| アクティブ・ジョブおよび非アクティブ・ジョブ | 33 |
| アクティブ・ジョブ | 33 |
| 非アクティブ・ジョブ | 33 |
| ジョブ・タイプ | 34 |
| 自動開始ジョブ | 34 |
| バッチ・ジョブ | 35 |
| 通信ジョブ | 35 |
| 対話式ジョブ | 35 |
| 事前開始ジョブ | 36 |
| 読み取りプログラムおよび書き出しプログラムのジョブ | 36 |

| | |
|---------------------------------|----|
| サブシステム・ジョブ | 37 |
| システム・ジョブ | 37 |
| ジョブ・プロパティ | 40 |
| プリンター出力の切り離し | 42 |
| 経過パフォーマンス統計 | 42 |
| 詳細状況 | 42 |
| ジョブの終了 | 43 |
| 詳細: アクティブ・ジョブ・アクション | 43 |
| ジョブ・ログ | 44 |
| スレッド | 44 |
| スレッド・アクション | 45 |
| スレッド・タイプ | 46 |
| スレッド状況 | 46 |
| ジョブ待ち行列 | 47 |
| ジョブ待ち行列はどのように機能するか | 47 |
| サブシステム | 48 |
| サブシステム記述 | 50 |
| システム出荷時のサブシステム | 59 |
| ユーザー定義サブシステム | 59 |
| サブシステムのプロパティ | 60 |
| サブシステムのライフ・サイクル | 62 |
| サブシステム開始時に何が行われるか | 62 |
| メモリー・プール | 63 |
| メモリー・プールの活動レベル | 63 |
| メモリー・プールの種類 | 64 |
| 出力待ち行列 | 65 |
| 出力待ち行列の属性 | 66 |
| ファイルの順序 | 67 |
| プリンター出力の状況 | 67 |
| 作業はどのように処理されるか | 68 |
| 作業とは何か | 68 |
| 作業がシステムに入れられる前に何が起こるか | 69 |
| 作業はどのようにシステムに入れられるか | 69 |
| 作業はどのように処理されるか | 70 |
| 作業はどのように終了するか | 70 |
| 実行管理機能のトラブルシューティング | 71 |
| ジョブのハングアップ | 73 |
| ジョブのパフォーマンスが悪い | 74 |

実行管理機能

実行管理機能は、iSeries サーバー・オペレーティング・システムにおける重要なコンポーネントです。すべての作業はこの機能によってシステムに入れられ、iSeries サーバーで処理され、実行され、完了されるため、実行管理機能はシステムの基盤と言えます。実行管理機能は、単純なバッチ・ジョブを週ごとに実行するか、(ロータス ノーツのような) アプリケーションを毎日呼び出すかに関係なく、システムで実行されるオブジェクトの管理を可能にします。さらに、システム操作を制御したり、必要なときにアプリケーションに資源を割り振ったりするのに必要なコマンドおよび内部機能をサポートします。

iSeries サーバーはすでにセットアップされ、すぐに使用できる状態になっています。ほとんどのユーザーはデフォルト設定を変更する必要がないでしょう。しかし、実行管理機能の一部を貴社の状況に合わせて調整する必要がある場合は、この機能に関連した用語や概念、さらには iSeries サーバーのパフォーマンスを最善のものとするための調整方法について理解しておく必要があります。

iSeries の経験者であれ、初心者であれ、このトピックは実行管理機能を分かりやすく概観するのに役立ちます。このトピックにはさまざまな入り口点が含まれているので、実行管理機能についてどこから学び始めるかを選択することができます。

ジョブの開始から終了まで

実行管理機能の中におけるジョブをその開始から終了までを解説します。対話式グラフィックスを使い、実行管理機能についてさらに知りたい点をクリックしながら学ぶことができます。

日次作業の管理

作業を効率良く管理するために iSeries ナビゲーターから実行できる日次タスク、およびそれらのタスクを実行するタイミングについて調べることができます。ジョブ・ログの検査からシステム活動のモニターまで、実行管理に関する重要な日次タスクを知ることができます。

システムの構造

iSeries サーバーで作業管理するために使用する、実行管理機能に関する用語と概念 (ジョブ、ジョブ待ち行列、サブシステム、およびメモリー・プールなど) を説明します。

作業はどのように実行されるか

作業を iSeries サーバーで実行するために行わなければならない事柄を説明します。ジョブ待ち行列のセットアップや、サブシステムへのメモリーの割り振り、さらにジョブの実行が終了したらどうなるかについて学ぶことができます。

実行管理機能のトラブルシューティング

iSeries ナビゲーターを使用してジョブの問題を解決する方法を説明しています。

新規および変更された情報については『新機能』のトピック、このトピック全体を印刷したい場合には『印刷』のトピックを参照してください。

関連情報

IBM の資料では、技術情報、専門的な情報、および実用的な方法についての情報が記載されています。

V5R2 の新機能

V5R2 では、iSeries ナビゲーターの実行管理機能コンポーネントに多くの新機能が追加されました。これらの新機能は、実行管理機能構造に統合されたので、実行管理機能のコンポーネント、つまりジョブの開始から終了まで (対話式グラフィックス)、日々の実行管理、iSeries サーバーの構造、および作業の処理方法のどこからでも学び始めることができます。これらの分野は、それぞれ実行管理機能をさまざまな観点から捕らえたものです。iSeries の経験者であれ、初心者であれ、この情報は実行管理機能を分かりやすく概観するのに役立ちます。

新しい iSeries ナビゲーター GUI 機能

ユーザーと管理者が文字ベースのインターフェースで実行していた実行管理機能やタスクの多くが、iSeries ナビゲーターを通して行えるようになりました。以下は新機能のリストです。

システム状況

- このダイアログは、システム接続および iSeries ナビゲーターの「実行管理機能」フォルダーの両方からアクセスできます。
- このダイアログは、ユーザーが、アクティブ・ジョブ、論理区画、メモリー・プール、およびディスク・プールなどのさまざまな iSeries ナビゲーター機能にアクセスするだけでなく、潜在的な問題を識別して解決するための単一ロケーションを提供します。

ジョブ

- 次のジョブ・リスト・ウィンドウが追加されました。

- サブシステムで実行されるジョブ
- トランザクションのジョブ
- 統合ファイル・システム (IFS) オブジェクトを使用するジョブ
- 磁気テープ装置を使用するジョブ

- ロック要求を出すプログラムまたはプロシージャーの識別
- 特定のロック・オブジェクトのロック・メンバーの処理
- 特定のロック・オブジェクトのロック行の処理
- オブジェクト、メンバー、または行にロックを設定しているジョブおよびロック・スペースの処理
- スレッド管理:

- 特定のジョブの下で実行しているスレッドの表示
- スレッドの終了
- スレッド・プロパティの表示 (経過パフォーマンス統計を含む)
- スレッドの実行優先順位の変更
- スレッドの呼び出しスタックの処理
- スレッドのライブラリー・リストの処理
- スレッドのロックの処理
- スレッドに接続されたトランザクションの処理

- 追加のジョブ・アクション:

- 特定のアクティブ・ジョブに接続されたトランザクションの処理
- 特定のアクティブ・ジョブが実行した最後の SQL ステートメントの処理

経過パフォーマンス統計ウィンドウに追加された日時タイム・スタンプ
アクティブ・ジョブ、スレッド、トランザクション、ロック・スペースのロック・オブジェクトの処理

- 追加のジョブ・プロパティ:

プリンター出力の切り離しオプション

ジョブ・プロパティ・シートの他のページ上の新規ディスク・プール・グループ・プロパティ

ジョブがロックを待機している場合、待ち行列の解除を待機している場合、または待機されている項目を識別するロック・スペースを待機している場合の詳細な状況値

共通ジョブ・リスト中のジョブがシステム上にもう存在しないことを示す詳細な状況値

プロパティ・ページからの「プリンター出力の立ち上げ (Launch Printer Output)」、「ジョブ・ログ」、および「スレッド」

ジョブ待ち行列



- ジョブを他のジョブ待ち行列の最上部に移動する
- ジョブ・ログを作成しないでジョブ待ち行列を消去する


出力待ち行列

- 出力待ち行列でのプリンター出力の表示
- 出力待ち行列内および出力待ち行列間でのプリンター出力の移動
- 変更済みプール・ファイルからプリンター出力ファイル


新機能または変更点を確認する方法

技術上の変更が加えられた箇所を確認するために、この情報では次のアイコンを使用しています。

-  は新規情報または変更情報の開始位置を示すアイコン。
-  は新規情報または変更情報の終了位置を示すアイコン。

このリリースの新機能または変更に関する他の情報については、プログラム資料説明書  を参照してください。

トピックの印刷


この文書の PDF 版を参照用または印刷用にダウンロードし、表示することができます。PDF ファイルを表示したり印刷したりするには、Adobe(R) Acrobat(R) Reader が必要です。これは Adobe Web サイト  からコピーできます。

PDF 版を表示またはダウンロードするには、以下のものを選択します。

- 実行管理機能 (約 896 KB または 82 ページ)
- システム値 (約 1290 KB または 178 ページ)

その他の情報

V4R5 「実行管理機能」資料 PDF を表示または印刷することもできます。

- V4R5 実行管理の手引き  (約 5833 KB または 633 ページ)

表示用または印刷用の PDF ファイルをワークステーションに保存するには、次のようにします。

1. ブラウザーで PDF を開く (上記のリンクをクリックする)。
2. ブラウザーのメニューから「ファイル」をクリックする。
3. 「名前を付けて保存...」をクリックする。
4. PDF を保存したいディレクトリーに進む。
5. 「保存」をクリックする。

日次作業の管理

システム・オペレーターまたは管理者としての作業の 1 つは、サーバーの稼働状況を良好に保つということです。これは、ジョブ、ジョブ待ち行列、サブシステム、メモリー・プール、ジョブ・ログ、および出力待ち行列機能をモニターし、管理し、適正な状態に保つということを意味します。

このセクションでは、さまざまな種類の日次実行管理タスクと、iSeries サーバーで実行する必要があるかもしれない他のタスクについて、トピックとして取り上げます。それぞれのサブトピックでは、それらのタスクを実行することが重要である理由、およびそれらのタスクを完了させる方法について説明します。

システム活動のモニター

システムのモニターは重要な日次活動です。これは数種類の方法で行うことができます。たとえば、iSeries ナビゲーターや iSeries ナビゲーター・マネージメント・セントラルを使用することができます。これらのサブトピックに含まれるタスクは、以下のとおりです。

- システム状況の処理
- システム・パフォーマンスのモニター
- モニターによる作業

ジョブおよびスレッドの管理

特定のジョブまたはスレッドの状況を報告する場合でも、ジョブまたはスレッドのパフォーマンスをモニターする場合でも、必要な情報のほとんどを iSeries ナビゲーターで簡単に得ることができます。これらのサブトピックに含まれるタスクは、以下のとおりです。

- iSeries サーバー上でのジョブの検索
- ジョブの状況の判別
- ジョブのパフォーマンス統計の表示
- ジョブの終了
- ジョブに実行されるアクション
- 特定のジョブの下で実行しているスレッドの表示
- スレッド・プロパティーの表示
- スレッドの終了

ジョブ待ち行列の管理

ジョブ待ち行列は、バッチ・ジョブのライフ・サイクルにおける重要な要素です。ジョブ待ち行列は、バッチ・ジョブがサブシステムに入れられる割合を制御できるようにします。これらのサブトピックに含まれるタスクは、以下のとおりです。

- ジョブ待ち行列上のジョブの表示

- ジョブ待ち行列内でのジョブの優先順位の変更
- 他のジョブ待ち行列へのジョブの移動

サブシステムの管理

ジョブはサブシステム上で実行されるため、ジョブの実行機能に影響する潜在的な問題について、サブシステム活動をモニターする必要があります。これらのサブトピックに含まれるタスクは、以下のとおりです。

- サブシステムのモニター
- サブシステム内のジョブの表示
- サブシステムの開始
- サブシステムの終了

メモリー・プールの管理

メモリー・プールは、ジョブを実行するためのメモリーをサブシステムに割り振ります。ジョブの実行時には、それを有効に完了させるために必要なメモリーを取得することは重要です。これらのサブトピックに含まれるタスクは、以下のとおりです。

- メモリー・プール内のジョブ数のモニター
- メモリー・プール内のサブシステム数のモニター
- メモリーの使用状況の検査
- メモリー・プールのサイズの変更

ジョブ・ログの管理

ジョブ・ログには、ジョブに対して入力された要求に関する情報、たとえばジョブ内のコマンド、プログラム内のコマンド、およびメッセージなどが入っています。これらのサブトピックに含まれるタスクは、以下のとおりです。

- サーバー・ジョブを含むアクティブ・ジョブのジョブ・ログへのアクセス
- プリンター出力へのアクセス

出力待ち行列の管理

出力待ち行列は、ジョブの終了時にプリンター出力を管理するために役立ちます。出力待ち行列を効率的に保守して印刷出力を順調に処理するための方法を理解することは大切です。これらのサブトピックに含まれるタスクは、以下のとおりです。

- システム上の出力待ち行列の表示
- 出力待ち行列の消去
- 出力待ち行列間および出力待ち行列内での出力の移動

システム活動のモニター

管理者が一日に行うさまざまな重要な仕事の一つに、システム活動のモニターがあります。システムを通過する作業のフローのモニターは、毎日モニターしなければならない情報のほんの一部に過ぎません。弊社では、システムの使用状況をチェックする基本システムからマネージメント・セントラルを使用してモニターする拡張システムに至るまで、ご使用のシステム・パフォーマンスのモニターに役立つ多種多様なツールを用意しています。

システム状況の処理

iSeries ナビゲーターでは、「システム状況」ウィンドウによって、システム上の都合のよいあるロケーションで種々のシステム機能を表示したり、システム機能にアクセスしたりできます。

iSeries パフォーマンスの管理

iSeries ナビゲーターのマネージメント・セントラル機能には、リアルタイムのパフォーマンス・データを収集して表示し、そこからシステム・パフォーマンス上の問題をトラッキングしてトラブルシューティングできるシステム・モニターがいくつか付属しています。

モニターによる作業

ジョブ、サーバー、メッセージ待ち行列、選択したフィールドの変更、および企業間取引活動をモニターします。

システム状況の処理

文字ベース・インターフェースのシステム状況の処理 (WRKSYSSTS) 画面の上半分をモデルにした「システム状況」ダイアログで、システムの状態を迅速に、かつ簡単に検査できます。マネージメント・セントラルを使用すると、システム・モニターを介して、より包括的な機能をモニターすることができます。

「システム状況」ウィンドウから実行できるさまざまな機能を以下に示します。

- CPU 使用量の表示
- ジョブの合計数、アクティブ・ジョブ、およびシステムで許可されるジョブの最大数の表示
- システム上の活動スレッドの数の表示
- システムで使用されるアドレス (永続的および一時的なもの) のパーセンテージの表示
- 合計ディスク・スペースの表示
- システム・ディスク・プール容量および使用量の表示
- システム上のプロセッサ数の表示

注: iSeries システムのタイプによって、3種類の「プロセッサ」ページがあります。システムの構成によっては、追加の処理装置の関連情報が表示されることもあります。

区画のないシステム
区画があるシステム、専用プロセッサ
区画があるシステム、共用プロセッサ

iSeries システム上の論理区画の詳細については、『論理区画』を参照してください。

- システム上の合計メモリの表示
- 使用される一時記憶域の表示
- 使用される一時記憶域の現行の量および最後のシステム再始動後に使用された最大量
- アクティブ・ジョブへのアクセス
- ジョブおよび記憶域システム値へのアクセス
- ディスク・プールへのアクセス
- 活動メモリ・プールへのアクセス
- 「論理区画の構成」ダイアログへのアクセス

「システム」フォルダーまたは iSeries ナビゲーター内の「実行管理機能」フォルダーから「システム状況」ダイアログにアクセスできます。

「システム」フォルダーからシステム状況を調べるには、次のようにします。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 処理したい接続を右マウス・ボタン・クリックし、「システム状況」を選択します。

システム状況を使って実行できるさまざまなタスクの詳細については、iSeries ナビゲーターのヘルプを参照してください。

ジョブおよびスレッドの管理

システムで実行される作業がジョブおよびスレッドの形をとるため、システム内のジョブおよびスレッドを検索し、トラッキングし、管理できることは重要です。

以下のサブトピックでは、特定のジョブの検索方法、ジョブ状況の判別方法、ジョブのパフォーマンスのモニター方法、ジョブの終了方法、ジョブに実行できるアクション、スレッドおよびそのプロパティーの表示方法、およびスレッドの終了方法について説明しています。

- iSeries サーバー上のジョブの検出
- ジョブの状況の判別
- ジョブのパフォーマンス統計の表示
- ジョブの終了
- ジョブ・アクション
- 特定のジョブの下で実行しているスレッドの表示
- スレッド・プロパティーの表示
- スレッドの終了

ジョブおよびスレッドに対して実行できる他のタスクについては、iSeries ナビゲーターのヘルプを参照してください。

iSeries でのジョブとジョブ・タイプに関する詳細については、『ジョブ』を参照してください。スレッドの詳細については、『スレッド』を参照してください。

iSeries サーバー上でのジョブの検索

iSeries サーバー上のジョブを検索する方法を理解しておくことは重要です。理由はどうあれ、ある時点で特定のジョブの特定の情報が必要になる場合があります。iSeries ナビゲーターで、すべてのジョブに対する「検索」を実行するか、「検索」の前に「組み込み...」機能を使って検索対象を絞ることができます。

「組み込み...」機能を使用すると、iSeries ナビゲーターに表示されるものを制限することができます。たとえば、何百というジョブに対して検索を実行する代わりに、「組み込み...」を実行して、特定のタイプのジョブだけが表示されるようにできます。あるいは、特定のジョブ・ユーザー ID を持つジョブだけを表示することができます。

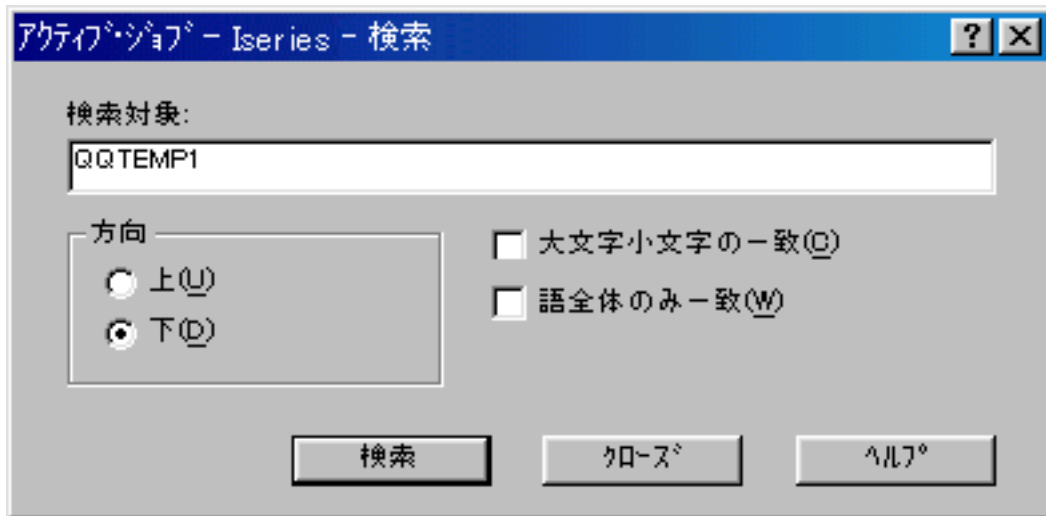
パフォーマンスの観点からすると、システムに多くのジョブが存在する場合は、「組み込み...」機能を使って検索対象の数を絞っておくことをお勧めします。システム上に多くのジョブが存在する場合にすべてを対象に検索をかけると、システム・パフォーマンスが低下してしまいます。

注: ジョブを検索する場合、実行管理機能のどこからでも、メニュー・バーの「検索」および「組み込み...」を使用できます。さらに、同じ方法でこれらのツールを使用して、ジョブ待ち行列、サブシステム、およびメモリー・プールを検索することもできます。これらのツールを使用する前に、検索を行う領域をクリックする必要があることに注意してください。

「**検索 (Find (Ctrl+F))**」オプションを使用してジョブを検索する場合、以下を行います。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「**実行管理機能**」を展開します。

4. 「アクティブ・ジョブ」をクリックして、「編集」を選択します。
5. 「編集」メニューから、「検索 (Find (Ctrl+F))」を選択します。
6. 「検索対象:」テキスト・フィールドに、検索するジョブ ID (たとえば、Qqtemp1) を入力します。ジョブの列はすべて検索対象です。



7. 「検索」をクリックします。iSeries ナビゲーターでは、検出されたジョブが強調表示になります。
注: ジョブ名は、引用符で囲む (たとえば、"MyJob") と大文字小文字が区別されます。ジョブ名を引用符で囲まない場合は、大文字小文字は区別されません。

「組み込み...」機能を使用して表示される情報を制限するには、以下を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「アクティブ・ジョブ」または「サーバー・ジョブ」のいずれかをクリックします。

5. 「表示」メニューから、「ビューのカスタマイズ (Customize this View)」を選択してから、「組み込み」を選択します。「アクティブ・ジョブ - 組み込み」ダイアログが表示されます。

アクティブ・ジョブ - 組み込み

組み込み

ジョブ:

名前:

ユーザー: 参照...

番号:

タイプ

自動開始 読み取り機能

バッチ サブシステム

通信 システム

対話式 書き込み機能

事前開始

サブシステム: 参照...

現行ユーザー: 参照...

?

6. 「アクティブ・ジョブ - 組み込み」ダイアログで、ジョブの検索に使用するオプションを選択します。
7. 「OK」を選択します。この時点から、「検索」を使用して特定のジョブを表示します。

ジョブの詳細については、『ジョブ』を参照してください。

ジョブの状況の判別

ジョブをモニターすることによって、ジョブの状況を理解することができます。ジョブ状況は、ジョブが何を行っているのかを調べるのに使用できる情報の重要な部分です。ジョブ状況は、iSeries ナビゲーターで簡単に調べることができます。

アクティブ・ジョブまたはサーバー・ジョブの状況を調べるには、以下を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している iSeries サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「アクティブ・ジョブ」または「サーバー・ジョブ」をクリックします。
注: 「実行管理機能」フォルダーのどの場所からでも、ジョブにアクセスしてジョブ状況を確認することができます。
5. 「詳細状況」列を調べて、ジョブの状況 (たとえば、イベントを待機中、時間間隔待機中、または待ち行列解除を待機中など) を判別します。

| ジョブ名 | サブシス... | 詳細状況 | タイプ | 実行優先順... |
|-------------|---------|----------|------------------|----------|
| Qbatch | Qbatch | デキューを待機中 | サブシステム | 0 |
| Qcmn | Qcmn | デキューを待機中 | サブシステム | 0 |
| Qacsotp | Qcmn | 要求を待機中 | 事前開始通信 | 20 |
| Qlzperv | Qcmn | 要求を待機中 | 事前開始通信 | 20 |
| Qnmapind | Qcmn | 要求を待機中 | 事前開始通信 | 25 |
| Qnmarexe... | Qcmn | 要求を待機中 | 事前開始通信 | 25 |
| Qnpsevr | Qcmn | 要求を待機中 | 事前開始通信 | 20 |
| Qzrcsvr | Qcmn | 要求を待機中 | 事前開始通信 | 20 |
| Qzscsvr | Qcmn | 要求を待機中 | 事前開始通信 | 20 |
| Qctl | Qctl | デキューを待機中 | サブシステム | 0 |
| Qsyscd | Qctl | イベントを待機中 | バッチ | 10 |
| Qinter | Qinter | デキューを待機中 | サブシステム | 0 |
| Qserver | Qserver | デキューを待機中 | サブシステム | 0 |
| Qpwserv | Qserver | 要求を待機中 | 事前開始バッチ | 20 |
| Qpwserv | Qserver | 要求を待機中 | 事前開始バッチ | 20 |
| Qpwserv | Qserver | 要求を待機中 | 事前開始バッチ | 20 |
| Qpwservsd | Qserver | 選択を待機中 | バッチ - サーバ... | 20 |
| Qpwservso | Qserver | デキューを待機中 | 事前開始バッチ - サーバ... | 20 |
| Qpwservso | Qserver | 要求を待機中 | 事前開始バッチ - サーバ... | 20 |
| Qpwservso | Qserver | 要求を待機中 | 事前開始バッチ - サーバ... | 20 |
| Qpwservss | Qserver | 要求を待機中 | 事前開始バッチ - サーバ... | 20 |
| Qpwservs2 | Qserver | 要求を待機中 | 事前開始バッチ - サーバ... | 20 |
| Qserver | Qserver | イベントを待機中 | 自動開始 | 20 |
| Qtftpjtc | Qserver | 要求を待機中 | 事前開始バッチ - サーバ... | 20 |
| Qzdainit | Qserver | 要求を待機中 | 事前開始通信 | 20 |
| Qzdasrvsd | Qserver | 選択を待機中 | バッチ - サーバ... | 20 |
| Qzlsfile | Qserver | 要求を待機中 | 事前開始バッチ - サーバ... | 20 |

詳細については、『ジョブの状況』を参照してください。

ジョブのパフォーマンス統計の表示

あるジョブの実行状況が極端に悪いと同じシステムの他のジョブに影響することがあるため、iSeries サーバーを使用するあらゆるユーザーにとって、ジョブのパフォーマンスは重要な問題となります。問題となりうるジョブを表示すると、パフォーマンス上の問題を未然に防ぐことができます。

「経過パフォーマンス統計」ウィンドウを使用すると、ジョブの CPU 使用、ディスク入出力 (ハード・ディスク入出力)、ページ不在率、平均応答時間、および対話式トランザクションの数をモニターできます。このウィンドウのオプションを選択して、手動でまたはスケジュール設定によりこれらの統計を最新表示することができます。

経過パフォーマンス統計を表示するには、以下のようにします。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「アクティブ・ジョブ」をクリックします。

注: 実行管理機能内で、ジョブを表示できる場所であればどこからでも、ジョブのパフォーマンスを表示できます。「経過パフォーマンス統計」ダイアログは、「ジョブ」プロパティ・シートの「パフォーマンス」タブから表示できます。

5. パフォーマンス統計を表示したいジョブの上で右マウス・ボタン・クリックし、「詳細...」を選択します。

6. 「詳細...」リストから、「経過パフォーマンス統計 (Elapsed Performance Statistics)」を選択します。

| 経過時間: 00:00:00 | | |
|-----------------------|-----|-----|
| 経過時間の中に計算されたパフォーマンス統計 | | |
| CPU: | | |
| パーセンテージ: | 0.0 | % |
| 時刻: | 0 | ミリ秒 |
| データベース・パーセンテージ: | 0.0 | % |
| データベース時間: | 0 | ミリ秒 |
| ディスク入出力速度: | | |
| 同期: | 0.0 | 兆 |
| 非同期: | 0.0 | 兆 |
| ディスク入出力カウント: | | |
| 同期: | 0 | |
| 非同期: | 0 | |
| ページ不在率: | 0.0 | 兆 |
| 平均応答時間: | 0.0 | 秒 |
| 対話式トランザクション: | 0 | |

パフォーマンス統計を最新表示およびリセットしたり、自動的に最新表示されるようにスケジュール設定することもできます。

注: 複数のウィンドウを開いて、一度に複数のジョブの経過パフォーマンス統計を見ることもできます。このようにして、問題のある複数のジョブを一度に表示できます。一つのウィンドウに表示される情報は、一つのジョブに限られます。

経過パフォーマンス統計は、システム内で変動するジョブのパフォーマンスを表示できる一つの方法です。システム内のジョブを表示できる別の方法としては、「マネージメント・セントラル」フォルダーの使用なども挙げられます。マネージメント・セントラルでは、ジョブだけでなく、システム・パフォーマンスやメッセージもモニターできます。ジョブ・モニターに関する追加情報については、『マネージメント・セントラル・モニター』を参照してください。

ジョブの終了

ジョブの実行に長い時間がかかったり、多くのメモリーが使われたりする場合、システム上の他のジョブのパフォーマンスに影響を与えるため、そのようなジョブを終了する必要があります。

ジョブを終了するには、以下を行います。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries サーバー**の接続を展開します。
3. 「**実行管理機能**」を展開します。
4. 「**アクティブ・ジョブ**」をクリックします。

注: 実行管理機能内で、ジョブを表示できる場所であればどこからでも、ジョブを削除または終了することができます。

5. 終了したいジョブ (たとえば、Qdftjobd) を右クリックして、「削除 / 終了」をクリックします。

| ジョブ名 | ユーザー | 番号 | タイプ |
|----------|------|--------|-----|
| Qdftjobd | Timr | 012131 | バッチ |

終了方法: 制御付き

制御付き終了の時間制限: 30 秒

プリンター出力の削除: いいえ

最大ジョブ・ログ項目数: ジョブ値を使用

関連した対話式ジョブのアクション: 終了しない

削除 キャンセル ヘルプ ?

- 「終了方法」フィールドで、「制御付き」か「即時終了」を選択します。
- 「制御付き終了の時間制限」フィールドに、ジョブが制御された終了から即時終了に切り替わるまでの秒数を入力します。(このパラメーターは制御された削除または終了にのみ適用されます。)
- 「プリンター出力の削除」フィールドで、「Yes」または「No」を選択します。
- 「最大ジョブ・ログ項目数」フィールドで、「ジョブ値を使用」または「最大値なし」を選択します。
- 「関連した対話式ジョブのアクション (Action for related interactive jobs)」フィールドで、「終了しない (Do not end)」、「グループ・ジョブの終了 (End for group jobs)」、または「すべて終了 (End all)」を選択します。
- 「削除」をクリックして、ジョブを削除します。

ジョブに対して実行できるアクションについては、『ジョブ・アクション』を参照してください。

ジョブ・アクション

ジョブおよびスレッドの管理は、実行管理機能で利用できるアクションと関連づけることによってさらに効率的になります。管理したいジョブを検出したら、そのジョブを右クリックすることによって以下に示すアクションを使用できます。

統計のリセット

表示しているリスト情報をリセットして、経過時間を 00:00:00 にセットします。

プリンター出力

可能であれば、プリンター出力を別のウィンドウに表示します。

ジョブ・ログ

選択したジョブに関するジョブ・ログを、別のウィンドウに表示します。

詳細

アクティブ・ジョブの以下のアクションに関する詳細情報が入っています。

- 呼び出しスタック
- ライブラリー・リスト
- **》ロック・オブジェクト《**
- オープンされているファイル
- **》スレッド《**
- **》トランザクション《**
- 経過パフォーマンス統計
- **》最後の SQL ステートメント《**

応答

メッセージを待っているジョブがある場合、メッセージに応答できるようにします。

保留

ジョブを保留できるようにします。ジョブを保留すると、ジョブ内のすべてのスレッドが保留されます。これは、システム・ジョブではない解放されたジョブに使用できます。ジョブを保留すると、ジョブを処理できなくなります。アクティブ・ジョブの処理を一時的に停止するために、それを保留することができます。

解放

保留されているジョブを解放します。ジョブを解放すると、**保留ジョブ・アクション**によって保留されていたすべてのスレッドが解放されます。ジョブは処理できるようになります。

移動

選択したジョブを別のジョブ待ち行列に移動できるようにします。移動できるのは、ジョブ待ち行列にあるジョブだけです。

削除 / 終了

選択したジョブを終了できるようにします。ジョブの終了方法には、制御された終了と即時終了の 2 つがあります。

モニター

1 つまたは複数のジョブのためのジョブ・モニターを作成できるようにします。

ジョブ・プロパティー

選択したジョブのジョブ・プロパティーを表示したり、変更したりできます。

特定のジョブの下で実行しているスレッドの表示

iSeries システムで実行中のすべてのアクティブ・ジョブでは、最低 1 つのスレッドが実行されています。スレッドは、ジョブと同じ資源を使用するジョブ内で実行する独立した作業単位です。ジョブはスレッドが実行する作業に依存するので、特定のジョブ内で実行するスレッドを検索する方法を理解することは重要です。

特定のジョブの下で実行しているスレッドを表示する手順は、次のとおりです。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「アクティブ・ジョブ」をクリックします。
5. 処理したいジョブを右クリックし、「詳細」>「スレッド (Threads)」を選択します。

詳細については、『スレッド』または iSeries ナビゲーターのヘルプを参照してください。

スレッド・プロパティの表示

スレッドによって、ジョブは一度に複数の処理を行うことができます。スレッドが処理を停止すると、ジョブの実行も停止することがあります。「スレッド・プロパティ (Thread Properties)」ページでは、スレッドが実行していない理由を理解するために役立つ、さまざまなスレッド、およびスレッド・パフォーマンスのプロパティを表示できます。

スレッドのパフォーマンスを表示するには、以下の手順を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「アクティブ・ジョブ」または「サーバー・ジョブ」をクリックします。
5. 処理したいジョブを右クリックし、「詳細」>「スレッド (Threads)」を選択します。
6. 処理したいスレッドを右クリックし、「プロパティ」を選択します。

詳細については、『スレッド』または iSeries ナビゲーターのヘルプを参照してください。

スレッドの削除または終了

ジョブの開始時に作成される初期スレッドは、削除または終了することができません。しかし、ジョブが実行を続けられるように 2 次スレッドを削除しなければならないことがあります。終了しようとするスレッドには注意してください。その中で実行しているジョブはそのスレッドの機能がないと完了できないことがあるからです。

重要: スレッドの終了を日次の実行管理ルーチンの一部としないでください。スレッドの終了は他のスレッド内の作業を停止させたり停止させなかったりするもので、ジョブの終了よりも重大な事柄です。ジョブを終了すると、すべての作業が停止します。しかし、スレッドを終了すると、作業の一部だけが停止します。他のスレッドは実行を継続する場合もあれば、継続しない場合もあります。終了したスレッドなしで他のスレッドが実行を継続した場合、予期しない結果となることがあります。

2 次スレッドを削除または終了するには、サービス (*SERVICE) 特殊権限またはスレッド制御権限が必要です。

スレッドを削除または終了するには、以下のようにします。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「**実行管理機能**」を展開します。
4. 「**アクティブ・ジョブ**」または「**サーバー・ジョブ**」をクリックします。
5. 作業したいジョブを右マウス・ボタンでクリックして、「**詳細**」を選択してから「**スレッド (Threads)**」を選択します。
6. 終了したいスレッドを右マウス・ボタンでクリックしてから、「**削除 / 終了**」を選択します。

詳細については、『スレッド』または iSeries ナビゲーターのヘルプを参照してください。

ジョブ待ち行列の管理

バッチ・ジョブのライフ・サイクルにおいて、ジョブ待ち行列はサブシステムへの入り口と言えます。ジョブ待ち行列は、指定された時間でサブシステムに投入できるジョブの数、およびサブシステムに入れられる順番を管理します。

次のサブトピックでは、以下のタスクについて説明しています。

- ジョブ待ち行列上のジョブの表示
- ジョブ待ち行列内でのジョブの優先順位の変更
- 他のジョブ待ち行列へのジョブの移動

詳細については、『ジョブ待ち行列』を参照してください。

ジョブ待ち行列上のジョブの表示

ジョブ待ち行列は実行管理機能で処理される作業の一部 (たとえば、一部のバッチ・ジョブ) にフィルターをかけます。ジョブ待ち行列内のジョブを表示できると、サブシステムに送信されるのを待っているのがどのジョブが表示することができます。

ジョブ待ち行列上のジョブを表示するには、以下を行います。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「**実行管理機能**」を展開します。
4. 「**ジョブ待ち行列**」を展開します。
5. 「**アクティブ・ジョブ待ち行列**」を展開します。「**すべてのジョブ待ち行列**」を展開することもできます。
6. ジョブを表示したいジョブ待ち行列 (たとえば、Jobqueue1) を選択します。そのジョブ待ち行列内のジョブが表示されます。

詳細については、『ジョブ待ち行列』を参照してください。

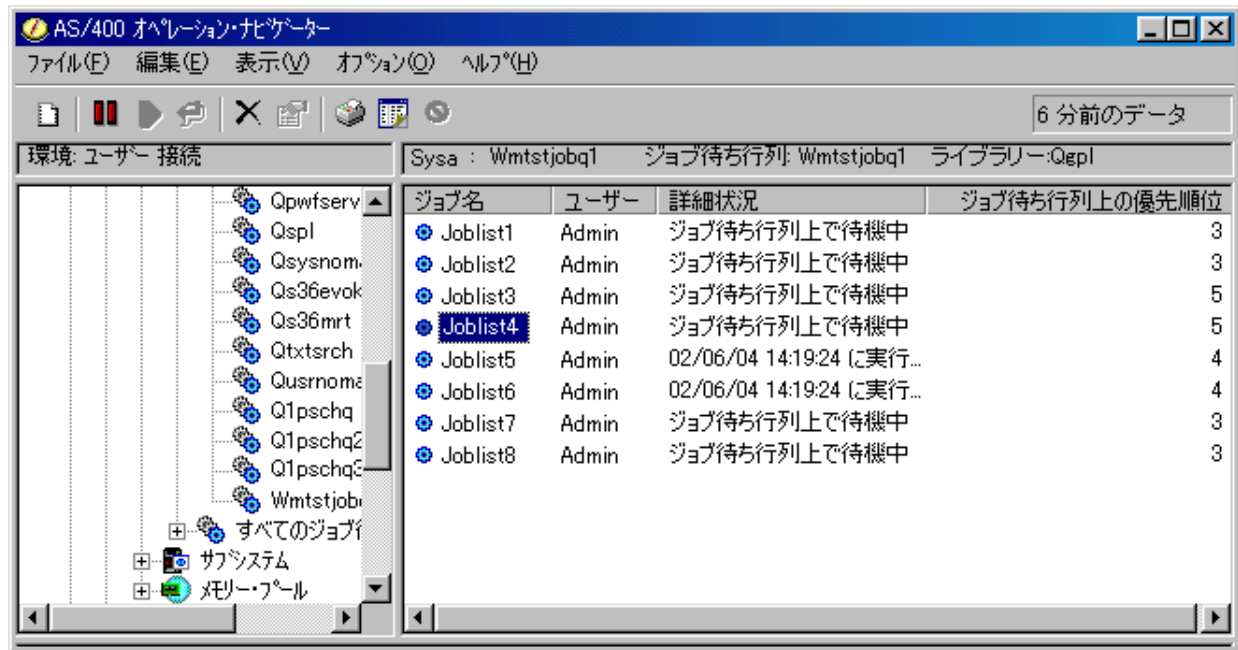
ジョブ待ち行列内でのジョブの優先順位の変更

ジョブの重要性は、その処理が進むにつれて変化します。他のジョブとの関係で、優先順位が高まったり、低くなったりする場合があります。それで、ジョブ待ち行列内でのジョブの優先順位の変更の方法を知っておく必要があります。ジョブ待ち行列でのジョブの優先順位は、そのジョブがいつごろサブシステムに入れられて実行されるかを判別するのに役立ちます。0 ~ 9 の範囲 (0 が最も重要) でジョブ待ち行列におけるジョブの優先順位を決定します。

iSeries ナビゲーターで、ジョブをドラッグ・アンド・ドロップするか、優先順位ページを使用するか、優先順位ページを使用するかして、ジョブの優先順位を上げ下げすることができます。

ドラッグ・アンド・ドロップを使って、ジョブ待ち行列内のジョブの優先順位を変更する場合、以下を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「ジョブ待ち行列」を展開します。
5. 「アクティブ・ジョブ待ち行列」または「すべてのジョブ待ち行列」を展開します。右側のペインにジョブ待ち行列のリストが表示されます。
6. 対象となるジョブ待ち行列 (たとえば、Qbatch) を選択します。ジョブ待ち行列にあるジョブのリストが表示されます。
7. 移動させたいジョブをクリックして、新しい優先順位の位置までドラッグします (たとえば、優先順位 5 の joblist4 を、優先順位 3 の joblist1 の次に移動させます)。



優先順位ページを使って、ジョブ待ち行列内のジョブのジョブ待ち行列優先順位を変更する場合、以下を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「ジョブ待ち行列」を展開します。
5. 「アクティブ・ジョブ待ち行列」または「すべてのジョブ待ち行列」を展開します。右側のペインにジョブ待ち行列のリストが表示されます。
6. 対象となるジョブ待ち行列 (たとえば、Qbatch) を選択します。ジョブ待ち行列にあるジョブのリストが表示されます。

7. 優先順位を変更したいジョブの上で右マウス・ボタン・クリックして、「プロパティ」を選択します。「プロパティ」ダイアログが表示されます。
8. 「ジョブ待ち行列」タブをクリックします。
9. 「ジョブ待ち行列上の優先順位」リストから、今よりも高い (または低い) 優先順位番号を選択します。ジョブ待ち行列の優先順位は 0 ~ 9 であり、0 が最高の優先順位です。
10. 「OK」を選択します。ジョブのジョブ待ち行列優先順位が変更されます。たとえば、優先順位 4 のジョブを優先順位 3 に変更すると、ジョブの優先順位は 3 となり、リストの一番下に移動します。
11. **F5** を押すと、「ジョブ待ち行列」ウィンドウが最新表示されます。

詳細については、『ジョブ待ち行列』を参照してください。

他のジョブ待ち行列へのジョブの移動

あるジョブ待ち行列が非常に混雑しているためにジョブが速やかにサブシステムに移動しないとき、または重要なジョブのために特別のジョブ待ち行列を作成したときに、あるジョブ待ち行列から別のジョブ待ち行列にジョブを移動することが必要になる場合があります。iSeries ナビゲーターを使用すると、ジョブ待ち行列間でのジョブの移動を速やかにかつ簡単に行えます。

あるジョブ待ち行列から別のジョブ待ち行列へのジョブの移動は、ドラッグ・アンド・ドロップおよび「ジョブの移動」ダイアログの二通りの方法で行えます。

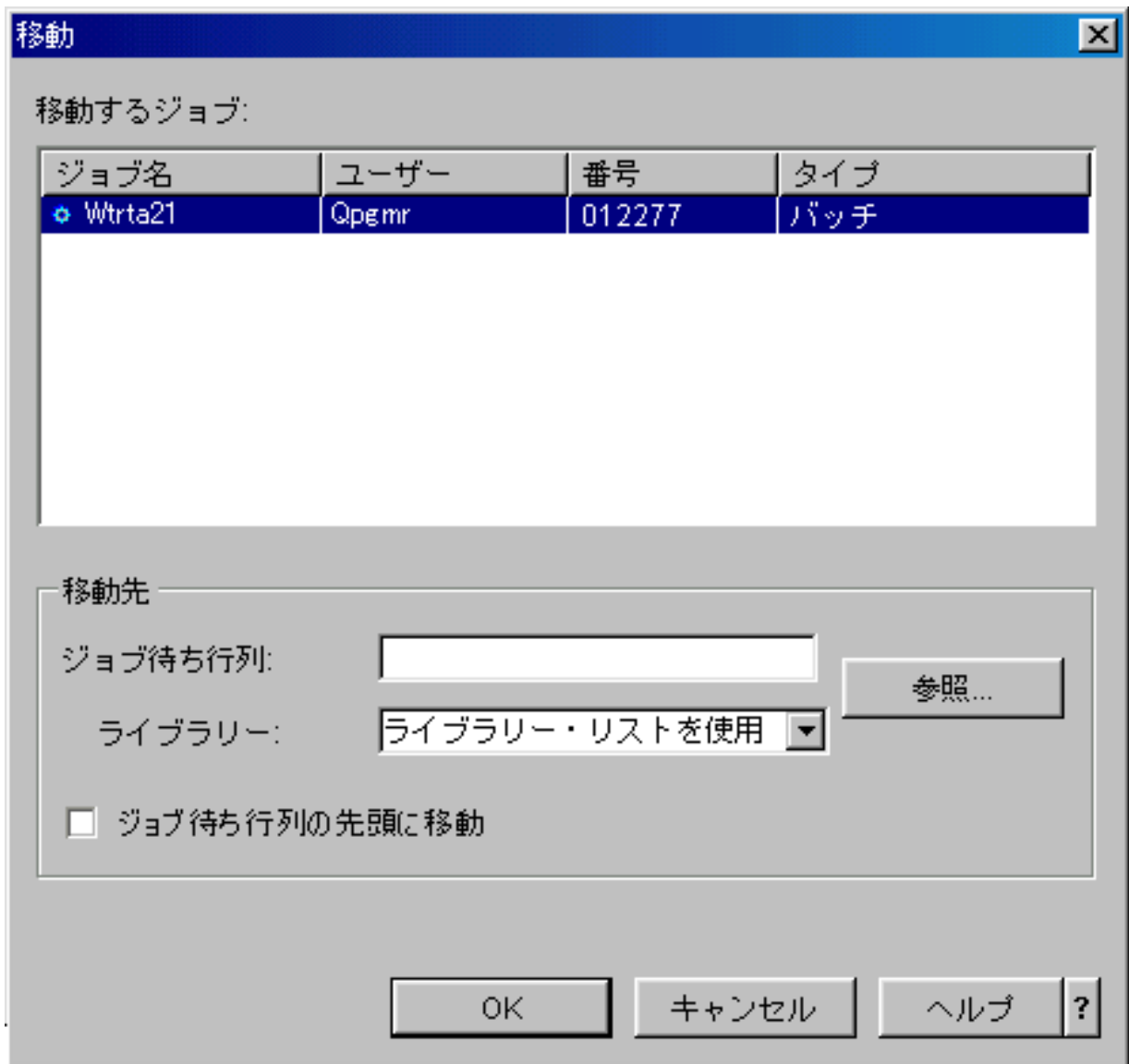
あるジョブ待ち行列から別のジョブ待ち行列へのジョブをドラッグ・アンド・ドロップするには、以下のようになります。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「ジョブ待ち行列」を選択します。
5. 「アクティブ・ジョブ待ち行列」または「すべてのジョブ待ち行列」をクリックします。
6. 処理に使用するジョブ待ち行列をダブルクリックします。
7. 移動するジョブを選択します。
注: Ctrl と Shift の両方のキーを同時に押しながら移動するジョブを選択すると、別のジョブ待ち行列へ移動するジョブを複数選択できます。
8. ジョブを目的のジョブ待ち行列にドラッグします。ジョブを新しいジョブ待ち行列にドロップすると、ジョブは直前のジョブ待ち行列で位置していたのと同じ位置になるように配置されます。たとえば、優先順位 3 のジョブが新しいジョブ待ち行列に移動すると、新しいジョブ待ち行列の優先順位 3 のジョブの末尾に配置されます。
注: 右マウス・ボタンを使用してドラッグする場合は、「移動」、「トップへ移動 (Move to Top)」、および「取消 (Cancel)」コマンドを伴ったメニューが表示されます。目的のコマンドをクリックします。

「移動...」ダイアログを使用してあるジョブを別のジョブ待ち行列に移動するには、以下のようになります。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「ジョブ待ち行列」を選択します。
5. 「アクティブ・ジョブ待ち行列」または「すべてのジョブ待ち行列」をクリックします。

6. 処理に使用するジョブ待ち行列をクリックします。
7. 別のジョブ待ち行列に移動するジョブ (たとえば、Qdftjobd) を右クリックして、「移動...」を選択します。
注: あるジョブ待ち行列から別のジョブ待ち行列に移動するジョブを複数選択することもできます。



8. 「移動するジョブ (Jobs to move)」フィールドで、選択したジョブが強調表示されていることを確認します。選択したジョブを削除するには、Ctrl を押しながらそのジョブをクリックします。
9. 「移動先のジョブ待ち行列 (Where to move Job Queue)」フィールドに、ジョブの移動先にするジョブ待ち行列 (たとえば、Qusrnomax) をタイプまたはブラウズします。
10. 「ライブラリー」フィールドに、ジョブ待ち行列ライブラリーの名前をタイプするか、または利用可能なリストからライブラリーを選択します。
11. 「OK」を選択します。

ジョブを新しいジョブ待ち行列に移動すると、ジョブは直前のジョブ待ち行列で位置していたのと同じ位置になるように配置されます。たとえば、優先順位 3 のジョブが新しいジョブ待ち行列に移動する

と、新しいジョブ待ち行列の優先順位 3 のジョブの末尾に配置されます。保留中のジョブを移動すると、ジョブは保留中のまま、新しいジョブ待ち行列でも移動前と同じ位置になるように配置されます。

「先頭に移動 (Move to Top)」ボックスをチェックすると、ジョブの現行状況およびプロパティにかかわらず、目的の待ち行列のトップに移動します。(しかし、目的の待ち行列のトップにあるジョブが、そのユーザーに与えられているよりも高い優先順位を持っている場合には、エラー・メッセージが表示され、ジョブは移動しません。) 実行を待っているジョブは、他の待ち行列のトップに移動できます。たとえば、選択したジョブがジョブ待ち行列優先順位 5 を持ち、目的の待ち行列の最初のジョブの優先順位が 3 の場合、選択したジョブの優先順位は 3 に変更され、目的の待ち行列上の他のジョブより前に配置されます。

保留中のジョブは開放され、目的の待ち行列のトップに移動します。実行のためのスケジュールが設定されているジョブは、他の待ち行列のトップに移動できません。選択したジョブの移動はできないという趣旨のエラー・メッセージが表示されます。

詳細については、『ジョブ待ち行列』を参照してください。

サブシステムの管理

サブシステムは、iSeries サーバーでのジョブの作業場所です。すべてのユーザー作業はサブシステム内のジョブを実行することによって行われるため、作業パフォーマンスが低下した場合この領域をモニターすることが重要です。iSeries ナビゲーターで、サブシステムに関連したジョブとジョブ待ち行列を表示できます。さらに、他のどの領域からでも、ジョブとジョブ待ち行列を表示して、ジョブとジョブ待ち行列に対して同じ機能性を持つことになります。

サブシステムについてさらに詳しくお知りになりたい場合は、以下のトピックを参照してください。

- サブシステムのモニター
- サブシステム内のジョブの表示
- サブシステムの開始
- サブシステムの停止

メモリー・プール内のジョブ数のモニター

サブシステムにジョブを実行するメモリーを提供するのがメモリー・プールの役目なので、メモリー・プール内で実行するジョブの数をチェックすることも重要です。一つのメモリー・プールにあるジョブが多すぎると、システム・パフォーマンスが低下します。

メモリー・プール内のジョブの数をモニターするには、以下のようになります。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「メモリー・プール」を展開してから、「活動プール」または「共用プール」をクリックします。
5. 使用するメモリー・プール (たとえば「基本」) を右マウス・ボタン・クリックして、「ジョブ」を選択します。メモリー・プールにあるジョブのリストを示したダイアログが表示されます。

「スレッド・カウント (Thread Count)」の欄を表示すると、メモリー・プール内のスレッドの数も分かります。スレッド・カウントから、メモリー・プール内の活動の量についてさらに知ることができます。

| ジョブ名 | 詳細状況 | タイプ | 実行優先順位 | スレッド・カウント |
|-------------|-----------|----------------|--------|-----------|
| Qbatch | デキューを待機中 | サブシステム | 0 | 1 |
| Qcmn | デキューを待機中 | サブシステム | 0 | 1 |
| Qacstcp | 要求を待機中 | 事前開始通信 | 20 | 1 |
| Qlzpssrv | 要求を待機中 | 事前開始通信 | 20 | 1 |
| Qnmapingd | 要求を待機中 | 事前開始通信 | 25 | 1 |
| Qnmarexe... | 要求を待機中 | 事前開始通信 | 25 | 1 |
| Qnpservr | 要求を待機中 | 事前開始通信 | 20 | 1 |
| Qzrcsrvr | 要求を待機中 | 事前開始通信 | 20 | 1 |
| Qzscsrvr | 要求を待機中 | 事前開始通信 | 20 | 1 |
| Qctl | デキューを待機中 | サブシステム | 0 | 1 |
| Qsysscd | イベントを待機中 | バッチ | 10 | 1 |
| Qinter | デキューを待機中 | サブシステム | 0 | 1 |
| Qserver | デキューを待機中 | サブシステム | 0 | 1 |
| Qpwfserv | 要求を待機中 | 事前開始バッチ | 20 | 1 |
| Qpwfserv | 要求を待機中 | 事前開始バッチ | 20 | 1 |
| Qpwfserv | 要求を待機中 | 事前開始バッチ | 20 | 1 |
| Qpwfservsd | 選択を待機中 | バッチ - サーバー | 20 | 1 |
| Qpwfservso | デキューを待機中 | 事前開始バッチ - サーバー | 20 | 1 |
| Qpwfservso | 要求を待機中 | 事前開始バッチ - サーバー | 20 | 1 |
| Qpwfservso | 要求を待機中 | 事前開始バッチ - サーバー | 20 | 1 |
| Qpwfservss | 要求を待機中 | 事前開始バッチ - サーバー | 20 | 1 |
| Qpwfservs2 | 要求を待機中 | 事前開始バッチ - サーバー | 20 | 1 |
| Qserver | イベントを待機中 | 自動開始 | 20 | 1 |
| Qtfpjtcp | 要求を待機中 | 事前開始バッチ - サーバー | 20 | 1 |
| Qzdainit | 要求を待機中 | 事前開始通信 | 20 | 1 |
| Qzdasrvsd | 選択を待機中 | バッチ - サーバー | 20 | 1 |
| Qzlsfile | 要求を待機中 | 事前開始バッチ - サーバー | 20 | 1 |
| Qzlsserver | イベントを待機中 | バッチ - サーバー | 20 | 1 |
| Qspl | デキューを待機中 | サブシステム | 0 | 1 |
| Qsyswrk | デキューを待機中 | サブシステム | 0 | 1 |
| Qcqpmon | メッセージを待機中 | バッチ | 50 | 1 |
| Qcqrsvds | メッセージを待機中 | バッチ | 20 | 1 |
| Qdftjobd | デキューを待機中 | バッチ | 50 | 1 |

(158 の内の) 1 - 32 オブジェクト

この時点から、アクティブ・ジョブ区域またはサーバー・ジョブ区域にいるときと同じ機能を実行することができます。

詳細については、『メモリー・プール』を参照してください。

サブシステム内のジョブの表示

サブシステムはジョブの実行に使用されるワークフローと資源を調整します。 iSeries ナビゲーターを使用すると、サブシステム内で現在活動中のジョブ (必ずしも実行中であるとは限らない) が何かを見ることができます。

サブシステム内のジョブを表示するには、以下の手順を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「サブシステム」を展開します。
5. 「活動サブシステム」を展開してから、その中のジョブを表示したいサブシステムを選択します。

詳細については、『サブシステム』を参照してください。

サブシステムの開始

サブシステムが開始されると、システムはサブシステム記述に定義されているサブシステムに利用可能な資源（メモリー・プール、ワークステーション、ジョブ待ち行列など）を割り振ります。これらの資源はサブシステムを使用可能な状態にします。

サブシステムが開始するときにトリガーとされる一連のイベントについては、『サブシステムが開始すると何が起こるか』を参照してください。

サブシステムを開始するには、以下の手順を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「サブシステム」を右クリックして、「サブシステムの開始」を選択します。
5. 開始するサブシステムの名前とライブラリーを選択するか、「ブラウズ...」をクリックしてサブシステムのリストから選択します。
6. 「OK」を選択します。

サブシステムの停止

iSeries ナビゲーターを使用して、1 つまたは複数の活動サブシステムを停止して、処理されている活動作業をどうするか指定することができます。サブシステムが停止すると、そのサブシステムでは新しいジョブまたは経路指定ステップは開始されません。

サブシステムが停止する場合に、システムによって処理されていた活動作業をどうするかを指定できます。たとえば、サブシステム内のすべてのジョブはすぐに終了する（即時）ように指定するか、サブシステムが終了する前にジョブは処理を終了できる（制御された）ように指定できます。

重要: 可能な場合はいつでも**制御された終了**オプションを使用して、サブシステムを停止することをお勧めします。こうすることによって、アクティブ・ジョブがそれ自身で終了できます。このオプションを使用すると、サブシステムの終了の前にジョブが確実に終了するようにできます。こうすると、実行中のプログラムは終結処置（ジョブ終了処理）を行えます。「即時」を指定すると、望まない結果（たとえば、データが部分的に更新されるなど）を招く場合があります。

サブシステムを停止するときに使用できる追加オプションがあります。これらのオプションについては、iSeries ナビゲーターの「サブシステムの停止」ダイアログに関連したヘルプで詳しく説明されています。

サブシステムを停止するには、以下の手順を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。

4. 「活動サブシステム」を展開します。
5. 停止したいサブシステム (複数も可) を右クリックして、「停止...」を選択します。
6. サブシステムが停止するとき使用されるオプションを指定します。
7. 「停止」をクリックします。

メモリー・プールの管理

メモリー・プールは、サブシステムがジョブの実行に使用するメモリーを割り振ります。あるサブシステムに多くのメモリーが与えられて、別のサブシステムではメモリーが不足するような場合、サブシステムのジョブの実行は効率よくありません。iSeries サーバーは、多くのユーザーのニーズに合わせたデフォルトの調整機能を備えています。しかし、システムの調整機能の能力を超えた調整を必要とする場合、メモリー・プールを管理する方法を知りたいと思うことでしょうか。共用メモリー・プールのプロパティーの中の「調整 (Tuning)」ページによって、iSeries ナビゲーターでのパフォーマンス調整値にアクセスすることができます。詳細については、『パフォーマンス』を参照してください。システムのパフォーマンスを調整する方法の詳細については、『パフォーマンスの調整』を参照してください。

メモリー・プールを管理するには、以下のトピックを参照してください。

- メモリー・プール内のジョブ数のモニター
- メモリー・プールを使用したサブシステム数のモニター
- メモリー・プールの使用状況の検査
- メモリー・プールのサイズの変更

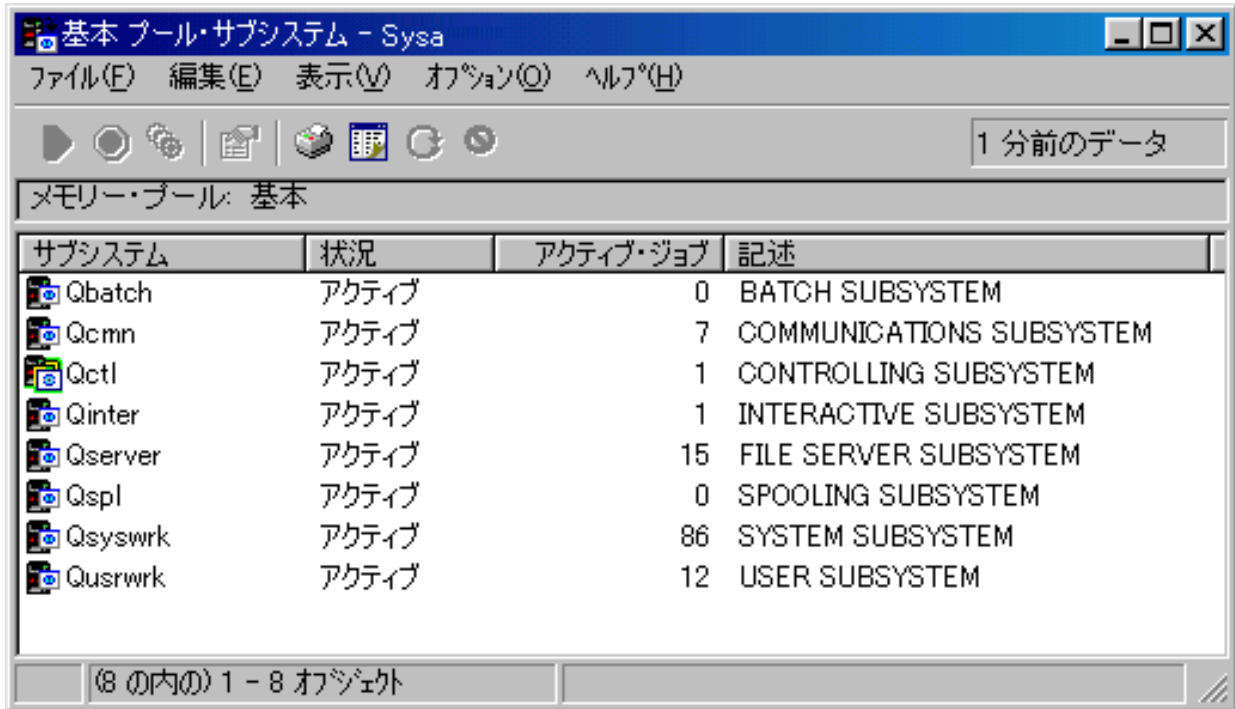
メモリー・プールを使用したサブシステム数のモニター

複数のジョブを実行するために、サブシステムには特定の比率でメモリーが割り振られます。パフォーマンスに関して言えば、いくつの異なるサブシステムが同じメモリー・プールからプルしているかを知っていることは重要です。いくつのサブシステムがプールにジョブを送り出しているか、またいくつのジョブがプールで実行されているかを知ると、資源の競合を減らすためにプールのサイズと活動レベルを調整する必要に気付くかもしれません。

メモリー・プールを使用するサブシステムの数モニターするには、以下のようになります。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「メモリー・プール」を展開します。
5. 「活動プール」または「共用プール」をクリックします。

6. 処理するメモリー・プール (たとえば、「基本」) の上で右マウス・ボタン・クリックして、「サブシステム」を選択します。



このウィンドウから、同一のメモリーを使用してジョブを実行しているサブシステムの数を判断することができます。

詳細については、『メモリー・プール活動レベル』を参照してください。

メモリー・プールの使用状況の検査

メモリー・プールが使用するメモリーのサイズを定期的には検査することは重要です。これらのレベルをモニターすることによって、プールが効率よく実行するように調整でき、作業の実行サイクルを円滑に保つことができます。iSeries ナビゲーターを使用して、プールが使用しているメモリーのサイズを簡単にモニターすることができます。

メモリーの使用状況を調べるには、以下を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している iSeries サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「メモリー・プール」を展開してから、「活動プール」または「共用プール」をクリックします。
5. 処理するメモリー・プール (たとえば「対話式」) を右マウス・ボタン・クリックして、「プロパティ」を選択します。
6. 「構成」タブをクリックします。「サイズ」の下の「現行」フィールドには、プールが現在使用しているメモリーのサイズが表示されます。

注: メモリー・プールの現行サイズは、「活動プール」または「共用プール」をクリックすることによっても表示できます。「現行」サイズ (メガバイト) は、メモリー・プールのリストを iSeries ナビゲーターの右側のペインに表示する場合のデフォルト列です。

詳細については、『メモリー・プール』を参照してください。

メモリー・プールのサイズの変更

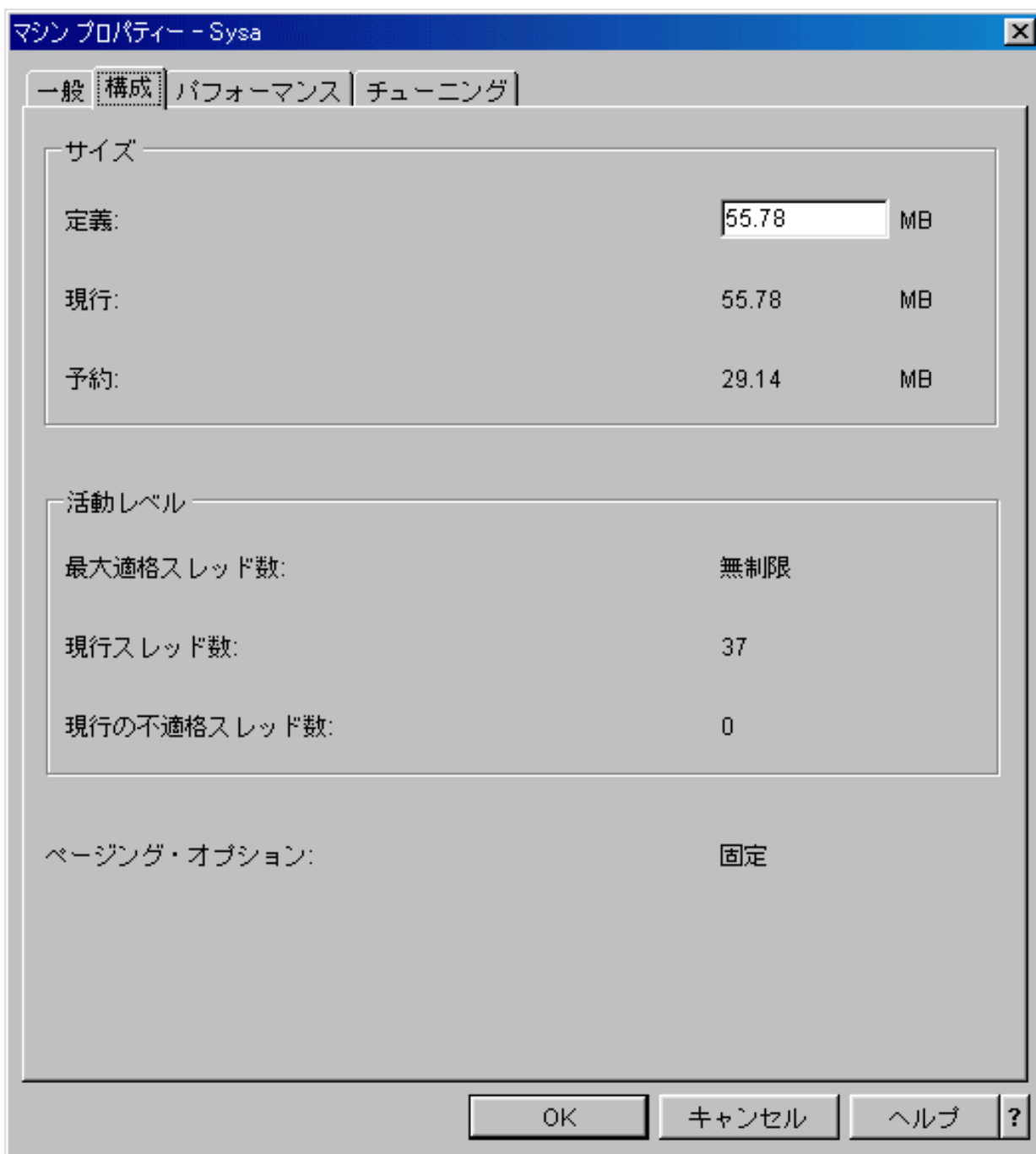
メモリー・プールのサイズ変更は、サブシステムが処理できる作業の量に直接影響します。多くのメモリーが与えられれば、サブシステムはさらに多くの作業を行えるようになります。iSeries ナビゲーターで、プール内のメモリーの定義されているサイズ (または利用可能なサイズ) を変更することができます。しかし、メモリー・プールのパラメーターを変更する前には、システムを注意深くモニターすることは大切です。さらに、再調整が必要になる場合もあるため、調整後もこれらのレベルを定期的に確認してください。

注: 手動でメモリー・プールのサイズを変更する場合、その前に必ずシステムの調整機能をオフにしてください。システムの調整機能は、共用メモリー・プールのサイズを、システムが実行している作業の量に自動的に調整してしまうためです。システムの調整機能をオフにしていると、手動で行った変更は調整機能によってさらに自動的に変更されてしまいます。

メモリー・プールのサイズを変更するには、以下を行います。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries サーバー**の接続を展開します。
3. 「**実行管理機能**」を展開します。
4. 「**メモリー・プール**」を展開してから、「**活動プール**」または「**共用プール**」をクリックします。
5. 作業を行うメモリー・プール (たとえば、「**対話式**」) 上で右マウス・ボタン・クリックして、「**プロパティ**」を選択します。「メモリー・プールのプロパティ (Memory Pool Properties)」ウィンドウが表示されます。

6. 「構成」タブをクリックします。



「プロパティ」ウィンドウの「構成」タブで、定義済みのメモリー・サイズを変更できます。定義済みのメモリーは、プールが使用できるメモリーの最大サイズです。ここに指定される数は、そのプールが機能を果たすサブシステムをサポートするのに必要なメモリー・サイズを反映しているべきです。

基本プールの特別な考慮事項: 基本プールは、メモリー・サイズが定義されていない唯一のメモリー・プールです。実行のために必要なメモリーの最小サイズが指定されています。基本プールには、他に割り振られないすべてのメモリーが含まれます。たとえば、システムのメモリーの総量が 1000 MB とします。そのうちの 250 MB がマシン・プールに割り振られ、さらに 250

MB が対話式プールに割り振られているとします。すると、500 MB はどこにも割り振られていません。この未割り振りのメモリーは、必要が生じるまで基本プールに保管されています。メモリーを移動させるときには注意が必要です。あるプールから別のプールへメモリーを移動する場合、1 つのサブシステムを修正するだけですが、他のサブシステムに問題を引き起こすことがあります。システム・パフォーマンスを低下させることもあります。

詳細については、『メモリー・プール』を参照してください。

ジョブ・ログの管理

iSeries 上のほとんどのジョブには、それに関連したジョブ・ログがあります。ジョブ・ログには、いつジョブが開始したか、いつジョブが終了したか、どのコマンドが実行中か、障害通知およびエラー・メッセージなど、さまざまなことが記録されています。この情報を見ると、ジョブ・サイクルがどのように行われているかがよく分かります。

アクティブ・ジョブのログにアクセスする方法、およびジョブ・ログ・プリンター出力にアクセスする方法を調べてください。

- サーバー・ジョブを含むアクティブ・ジョブのジョブ・ログへのアクセス
- ジョブ・ログ・プリンター出力へのアクセス

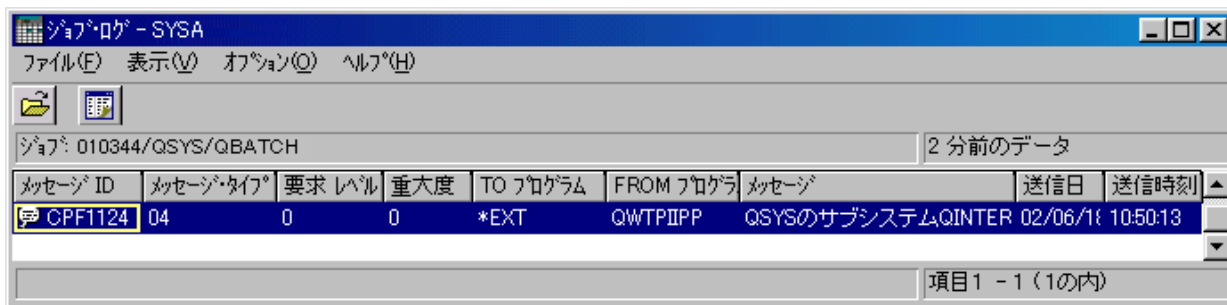
ジョブ・ログの詳細については、実行管理の手引き  の第 5 章にある『ジョブ・ログ』を参照してください。

サーバー・ジョブを含むアクティブ・ジョブのジョブ・ログへのアクセス

ジョブは実行中にジョブに関するログ情報を記録するため、そのログにアクセスする方法を知っておくことは重要です。

アクティブ・ジョブまたはサーバー・ジョブのジョブ・ログにアクセスするには、以下のようにします。

1. 「iSeries ナビゲーター」で、「ユーザー接続」を展開します。
2. 使用している iSeries サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「アクティブ・ジョブ」または「サーバー・ジョブ」を選択します。
注: 実行管理機能の中のどの場所からでも、ジョブにアクセスしてジョブ・ログを参照することができます (たとえば、サブシステム・エリアまたはメモリー・プール・エリア)。
5. ジョブ (たとえば、Qbatch) を右クリックして、「ジョブ・ログ」を選択します。以下に示すイメージで、ジョブ・ログから調べることができる情報のタイプを確認してください。詳細については、「ジョブ・ログ」ダイアログでヘルプを開いて参照してください。



メッセージの詳細を表示する場合は、該当するメッセージをダブルクリックします。「詳細メッセージ情報」ダイアログが表示されます。このダイアログには、メッセージの詳細とともにメッセージ・ヘルプが示されます。詳細なメッセージ・ヘルプは、問題を解決するための情報を提供します。

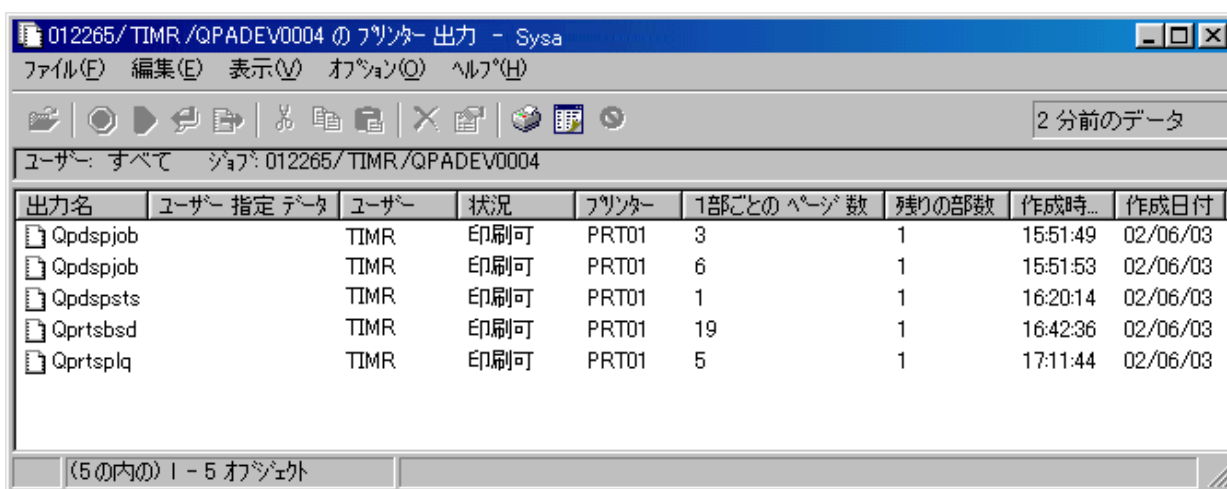
詳細については、『ジョブ・ログ』を参照するか、ヘルプを参照してください。

プリンター出力へのアクセス

ジョブが実行を終了した後でプリンター出力を切り離す (プリンター出力をジョブから完全に分離する) ことができるので、iSeries ナビゲーター内のプリンター出力に基本操作または実行管理機能を介してアクセスすることができます。

基本操作によってジョブのプリンター出力へアクセスするには、以下のようにします。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している iSeries サーバーの接続を展開します。
3. 「基本操作」を展開します。
4. 「ジョブ」を選択します。現行ユーザーのすべてのジョブが表示されます。ジョブを検索する別の方法については、『iSeries サーバー上でのジョブの検索』を参照してください。
5. プリンター出力を表示したいジョブを右マウス・ボタン・クリックし、「プリンター出力」をクリックします。「プリンター出力」ダイアログが表示されます。



「出力待ち行列」フォルダーを介してプリンター出力にアクセスするには、以下のようにします。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している iSeries サーバーの接続を展開します。

3. 「**実行管理機能**」を展開します。
4. 「**出力待ち行列**」を展開します。
5. プリンター出力を表示したい出力待ち行列 (たとえば、Qprint2) を選択します。出力待ち行列内のプリンター出力が表示されます。

出力待ち行列の管理

プリンター出力は、出力待ち行列に入っています。出力待ち行列は、プリンター出力がプリンターによって処理される順序を判別します。出力待ち行列の管理によって、プリンター出力を円滑に処理できます。

適切な権限によって、「**出力待ち行列**」フォルダーから以下のタスクを実行できます。

- システム上の出力待ち行列の表示
- 出力待ち行列のプロパティの表示
- 出力待ち行列の保有
- 出力待ち行列の解放
- 出力待ち行列の消去
- 出力待ち行列で待機中の出力の表示
- 出力待ち行列間および出力待ち行列内での出力の移動
- 出力待ち行列のプロパティの変更

システム上の出力待ち行列を表示する際、出力待ち行列を消去する際、および出力待ち行列間や出力待ち行列内でプリンター出力を移動する際には、以下のサブトピックを使用してください。

- システム上の出力待ち行列の表示
- 出力待ち行列間および出力待ち行列内での出力の移動
- 出力待ち行列の消去

出力待ち行列で実行できる他のタスクの詳細については、iSeries ナビゲーターのオンライン・ヘルプを参照してください。詳細については、『出力待ち行列』を参照してください。

システム上の出力待ち行列の表示

出力待ち行列は、プリンター出力がプリンターに送信される順序を決定します。

システム上の出力待ち行列を表示するには、次の手順を実行します。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「**実行管理機能**」を展開します。
4. 「**出力待ち行列**」を展開します。

iSeries ナビゲーターでは、「**組み込み...**」ダイアログを使って、表示中の出力待ち行列のリストをカスタマイズすることができます。「**組み込み...**」ダイアログを使用すると、iSeries ナビゲーターに表示されるものを制限することができます。たとえば、「**組み込み...**」を実行して、特定の出力待ち行列だけを表示できます。組み込み機能を使用するには、「**表示**」メニューの「**このビューの最適化**」を使用します。

詳細については、『出力待ち行列』を参照してください。

出力待ち行列間および出力待ち行列内での出力の移動

出力をある待ち行列から別の待ち行列に移動する必要や、出力をプリンターに早めに送信するために優先順位を高くする必要が時折生じます。出力トラフィックが 1 つの出力待ち行列に集中しすぎると、そのようになる可能性があります。

出力をある出力待ち行列から別の待ち行列に移動することも、出力を同一の出力待ち行列内で移動することも可能です。

出力待ち行列間で出力を移動するには、以下の手順を行います。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「**実行管理機能**」を展開します。
4. 「**出力待ち行列**」を展開します。
5. 移動する出力を含む出力待ち行列をダブルクリックします。
6. 移動する出力をクリックし、それを iSeries ナビゲーターの画面左側にある、移動先の出力待ち行列までドラッグします。

注: 出力はターゲットの待ち行列に移動し、優先順位に応じてその待ち行列上に配置されます。

出力待ち行列内で出力を移動するには、以下の手順を行います。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「**実行管理機能**」を展開します。
4. 「**出力待ち行列**」を展開します。
5. 移動する出力を含む出力待ち行列をダブルクリックします。
6. 移動する出力をクリックし、移動先にする待ち行列内の出力までドラッグします (ターゲットの出力のすぐ後に配置されます)。

注: その出力は、ターゲットの出力のすぐ後に移動します。

詳細については、『出力待ち行列』を参照してください。

出力待ち行列の消去

ジョブがプリンター出力を作成すると、それは印刷のための出力待ち行列に送られます。ユーザーが、作成されたプリンター出力のすべてを実際に印刷することは、おそらくありません。iSeries ナビゲーターでは、「**消去**」オプションによって出力待ち行列を消去することが可能です。出力待ち行列を消去すると、すべての出力が待ち行列から削除されます。

出力待ち行列を消去するには、以下の手順を行います。

1. iSeries ナビゲーターで、「**ユーザー接続**」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「**実行管理機能**」を展開します。
4. 「**出力待ち行列**」を展開します。

5. 消去したい出力待ち行列を右マウス・ボタン・クリックして、「消去」を選択します。

詳細については、『出力待ち行列』を参照してください。

システムの構造

実行管理機能は、ジョブ、ジョブ待ち行列、サブシステム、メモリー・プール、出力の 5 種類の機能分野に分けることができます。これらの分野には、それぞれに独自の用語とそれに関連する概念があります。それぞれが生成するデータの種類は異なりますが、それらが互いに統合されると、iSeries サーバーの処理を管理する一つの強力なツールになります。

実行管理機能の各機能分野についてよりよく知るには、以下のトピックを参照してください。

ジョブ

さまざまな種類のジョブとそのプロパティーについて知ることができます。また、ジョブに対して実行できるアクションについても知ることができます。

ジョブ待ち行列

実行管理機能ライフ・サイクルの中でのジョブ待ち行列の役割について知ることができます。

サブシステム

さまざまな種類のサブシステムとそのプロパティーについて知ることができます。

メモリー・プール

さまざまな種類のメモリー・プールとそのプロパティーについて知ることができます。

出力待ち行列

作業の実行が終了すると、作業はどうなるかということについて知ることができます。

注: iSeries ナビゲーターは、iSeries システムから情報を検索するアプリケーション・プログラミング・インターフェース (API) を呼び出します。API は、iSeries サーバー用の、iSeries ナビゲーターの入出力装置です。API の詳細については、『アプリケーション・プログラミング・インターフェース (API)』また

は System API Programming  を参照してください。

ジョブ

システムで実行されるすべての作業はジョブによって実行されます。各アクティブ・ジョブには少なくとも 1 つのスレッド (初期スレッド) があり、追加の 2 次スレッドが含まれている場合もあります。スレッドは独立した作業単位です。ジョブ・プロパティーはジョブのスレッド間で共有されますが、スレッドは、呼び出しスタックなど独自のプロパティーも有しています。ジョブのプロパティーには、作業の処理方法に関する情報が含まれます。同一ジョブ内で複数のスレッドに共用されるプロパティーに対して、ジョブは所有者としての役割を果たします。実行管理機能によって、ジョブ・プロパティーによるシステム上の作業の実行を制御する方法が提供されます。

ジョブの汎用プロパティーは、システムが各ジョブを実行する方法を決定します。いくつかのプロパティーは、複数のジョブの管理を簡単にするために、ジョブ記述ごとにグループ化されています。システムは、ジョブ・プロパティーが指定された方法に基づいて、どのプロパティーがいつ必要になるかを認識しています。iSeries システムは、さまざまな必要に応えるために、異なるタイプのジョブを実行します。ほとんどのジョブはジョブ記述を使用します。

ジョブの詳細については、以下のトピックを参照してください。

アクティブ・ジョブおよび非アクティブ・ジョブ

アクティブ・ジョブおよび非アクティブ・ジョブとは何かを説明します。

ジョブ・タイプ

iSeries で実行されるさまざまなタイプのジョブについて説明します。

ジョブ・プロパティ

ジョブ・プロパティを処理する方法を説明します。

ジョブ・アクション

iSeries ナビゲーターによってジョブを管理する方法について説明します。

スレッド

スレッドとジョブの違いについて説明します。

ジョブ待ち行列

ジョブがジョブ待ち行列での待機状態からどのように実行されるかを説明します。

ジョブの開始から終了まで

ジョブの開始から終了までを説明します。

注: 「ジョブのリストのオープン」(QGYOLJOB) および「ジョブ情報の検索」(QUSRJOBI) などの API を呼び出して、ジョブの情報を取得できます。API の詳細については、『アプリケーション・プログラミング・インターフェース (API)』を参照してください。

アクティブ・ジョブおよび非アクティブ・ジョブ

アクティブ・ジョブ:

アクティブ・ジョブとは、実行が開始されていて、まだ完了していないジョブのことです。アクティブ・ジョブの特性を以下に示します。

- 実行コードを含んでいる
- 呼び出しスタックを持っている
- オブジェクトがロックされている
- アクティブ・ジョブの状況を持っている。たとえば、以下のような状況です。

実行中

(x) を待機中

アクティブ・ジョブのプロパティに関する詳細については、『ジョブ・プロパティ』を参照してください。

アクティブ・ジョブの管理方法を知るには、『ジョブおよびスレッドの管理』を参照してください。

非アクティブ・ジョブ:

非アクティブ・ジョブとは、ジョブ待ち行列に入れられて開始を待っているジョブ、または処理が完了(終了)していてプリンター出力ファイル(スプール・ファイルとも呼ばれる)の印刷を待っているジョブのことです。

ジョブ・タイプ

iSeries サーバーはさまざまなジョブ・タイプを処理します。以下に示すジョブ・タイプから 1 つを選んで、そのタイプについてさらに詳しく学ぶことができます。

サーバー・ジョブは、ジョブ変更 (QWTCHGJB) API を使って設定されたサーバー・タイプのジョブであり、以下のジョブ・タイプに加えて、サーバー (Server) として分類されます。

自動開始

自動開始ジョブは、関連づけられているサブシステムが開始するときに自動的に開始するジョブです。

バッチ

バッチ・ジョブは、システムに投入されるアクションの事前定義された処理グループです。

通信

通信ジョブは、リモート・システムからのプログラム開始要求によって開始されるバッチ・ジョブです。

対話式

対話式ジョブは、サインオンしたユーザーと iSeries サーバーの入力を必要とするジョブです。

事前開始

事前開始ジョブは、作業要求を受け取る前に開始するバッチ・ジョブです。2 種類の事前開始ジョブは以下のとおりです。

- 事前開始通信 - このジョブは、遠隔システムがプログラム開始要求を送信する前に実行を開始する通信バッチ・ジョブです。
- 事前開始バッチ - このジョブは、作業要求を受け取る前に開始するバッチ・ジョブです。

読み取りプログラムおよび書き出しプログラム

読み取りプログラム・ジョブはスプールされた入力ジョブで、書き出しプログラム・ジョブはスプールされた出力ジョブです。

サブシステム

サブシステムのジョブは活動サブシステムを制御できるようにします。

システム


システム・ジョブは、システム・リソースを制御したり、システム機能を実行するために、オペレーティング・システムによって作成されます。

自動開始ジョブ: 自動開始ジョブは、関連づけられているサブシステムが開始するときに自動的に開始するジョブです。この種のジョブは、一般的に特定のサブシステムに関連している初期設定作業を実行します。さらに、自動開始ジョブは繰り返し作業を実行したり、同じサブシステム内の他のジョブに集中的なサービス機能を提供したりします。

サブシステム・ジョブは、ジョブを開始するときに、サブシステム記述にある自動開始ジョブ項目の情報を使用します。

注: 自動開始ジョブはすべてそのサブシステムが開始するときに開始します。サブシステムの最大ジョブ数に値を指定しておくこと、自動開始ジョブが開始しなくなるという事態を防ぐことができます。サブシステ

ムの最大ジョブ数を超過してしまうと、その他のジョブは開始できなくなります。相当数の自動開始ジョブが完了して、実行中のジョブの数がサブシステムの最大ジョブ数より少なくなると、サブシステムの他のジョブを開始できます。

自動開始ジョブの詳細とその開始方法については、実行管理の手引き  の『自動開始ジョブ』(第 9 章) および『自動開始ジョブ項目』(第 4 章) を参照してください。

バッチ・ジョブ: バッチ・ジョブは、システムに投入される前に定義された処理のアクションのグループのことです。バッチ・ジョブはシステムのバックグラウンドで実行され、ジョブを投入したユーザーが他の作業を行うように解放します。そのようなジョブは、一度セットアップされると、ユーザーとの対話を必要としません。バッチ・ジョブは通常低優先順位のジョブです。同時にいくつかのバッチ・ジョブを活動状態にすることができます。

さまざまな種類のバッチ・ジョブを以下に示します。

単純バッチ・ジョブ

ほとんどのユーザーは、ジョブ待ち行列に投入される単純なバッチ・ジョブについては理解しています。単純バッチ・ジョブの開始から終了までについては、ジョブの開始から終了までを参照してください。

バッチ即時ジョブ


バッチ即時ジョブは、その親ジョブの多くの属性とともに開始されるバッチ・ジョブです。このジョブは、親ジョブと同じサブシステムで実行されます。このジョブは親ジョブから属性をコピーして、ジョブ待ち行列には入れられないため、ジョブ待ち行列に入れられるジョブよりも速く開始できます。

バッチ MRT ジョブ

バッチ MRT ジョブとは、複数要求端末 (MRT) ジョブのことです。MRT ジョブはサーバーのような機能を果たす S/36 環境ジョブで、他の S/36 環境ジョブが接続して MRT プロシーチャーを実行できるようにします。

バッチ印刷ジョブ

バッチ印刷ジョブは、現行のユーザー・プロファイルが開始時のユーザー・プロファイルと異なるジョブによって作成されたプリンター出力ファイル (スプール・ファイルとも呼ばれる) をトラッキングします。

詳細については、実行管理の手引き  の第 8 章でバッチ・ジョブの開始方法に関する項を参照してください。

通信ジョブ: 通信ジョブは、リモート・システムからプログラム開始要求を受け取るときに開始されます。パフォーマンス上の理由から、プログラム開始要求を受け取るたびに通信ジョブを開始する代わりに、リモート・システムからのプログラム開始要求を扱うように 事前開始ジョブを構成することができます。

プログラム開始要求の詳細については、ICF Programming  の第 3 章を参照してください。

詳細については、実行管理の手引き  の第 10 章にある『通信ジョブ』を参照してください。

対話式ジョブ: 対話式ジョブは、タスクを実行するために、ユーザーと iSeries サーバーの間で両方向の連続した通信が必要です。対話式ジョブは、ユーザーがシステムにサインオンするときに開始されます。システムはサインオン情報を要求します。サインオン要求がシステムに受け入れられると、システムは対話式ジョブを作成します。その後、システムはユーザーに要求を出すよう依頼します。ユーザーは要求を入力して、システムはその要求を処理することによって応答します。このパターンは、ユーザーがシステムをサイ

ンオフして対話式ジョブを終了するまで繰り返されます。対話式ジョブがジョブ・グループまたはジョブの対の一部の場合、以下に示すジョブ・タイプのいずれかになります。

対話式 - グループ

対話式 - グループ・ジョブは、単一の対話式に関連したジョブ・グループの一部です。**対話式 - システム要求**

対話式 - システム要求ジョブは、システム要求機能によって互いに関連している 1 組のジョブの 1 方です。

事前開始ジョブ: 事前開始ジョブは作業要求を受け取る前、つまりサブシステムが開始したときまたは「事前開始ジョブ開始 (STRPJ)」コマンドの結果として開始します。事前開始ジョブはサブシステム記述の事前開始ジョブ項目 (PJE) から開始します。事前開始ジョブ項目は、事前開始ジョブでどのプログラムを実行するか、事前開始ジョブが実行を開始するときに使うユーザー・プロファイル、ジョブ記述、ジョブの実行時プロパティを指定するときに使うクラス、および事前開始ジョブを実行するメモリー・プールなどのプロパティを指定します。


事前開始ジョブは、作業要求を受け取る前に、開始して事前開始ジョブ自身を初期化します。これにより、要求を処理するのに必要な時間が少なくて済みます。作業要求ごとに新しいジョブが必要とされることはありません。また、事前開始ジョブを使用すると、一度初期化しただけで多数の要求を処理できるようになるため、要求ごとに新しいジョブが必要とされることはありません。クライアント・サーバー・アプリケーションのほとんどが、クライアント・ユーザーの要求を処理するために事前開始ジョブを使用しています。このようにジョブをすぐに実行できるようにしておくなら、事前開始ジョブがユーザーの要求の処理を即時に開始できるので、パフォーマンスが向上します。

注: サブシステムの最大ジョブ数に値を指定しておく、事前開始ジョブが開始されないようにすることができます。サブシステムの最大ジョブ数を超えた場合、事前開始ジョブは開始できなくなります。相当数のジョブが完了して、実行中のジョブの数がサブシステムの最大ジョブ数以下になると、サブシステムの事前開始ジョブを開始することができます。

事前開始ジョブには二つの種類があります。それぞれのタイプが処理する要求は異なります。「Prestart (事前開始)」とだけ表示され、その後ジョブがそれ自身の最初の要求を待機しますが、それはその時点ではどの種類の要求をジョブが処理するのかをシステムが認識していないためです。二種類の事前開始ジョブは以下のとおりです。

事前開始通信ジョブ

事前開始通信ジョブは、遠隔システムがプログラム開始要求を送信する前に実行を開始する通信バッチ・ジョブです。

事前開始通信ジョブの詳細については、実行管理の手引き  の第 11 章にある『事前開始ジョブ』を参照してください。

事前開始バッチ・ジョブ

事前開始バッチ・ジョブは、作業要求を受け取る前に開始するバッチ・ジョブです。

読み取りプログラムおよび書き出しプログラムのジョブ: 読み取りプログラム

読み取りプログラム・ジョブは、データベースやディスク・ファイルからバッチ・ジョブ・ストリームを読み取り、ジョブ待ち行列にジョブを配置します。読み取りプログラム・ジョブは入力スプーリングの一部であり、IBM 提供のプログラムです。

書き出しプログラム

書き出しプログラム・ジョブは、プリンター出力ファイル (スプール・ファイルとも呼ばれる) のレコードをプリンターに書き出します。書き出しプログラム・ジョブは IBM 提供のプログラムであり、スプーリング・サブシステムで開始され、そのサブシステムで、出力待ち行列から印刷するファイルが選択されます。

サブシステム・ジョブ: サブシステム・ジョブ (サブシステム・モニター・ジョブと呼ばれることもある) はオペレーティング・システムによって作成され、資源を管理し、ジョブを開始、制御、および終了します。サブシステムのジョブは活動サブシステムを制御できるようにします。1つのシステム上でいつでも多くのサブシステム・ジョブを実行できます。

詳細については、『サブシステム』を参照してください。

システム・ジョブ: システム・ジョブは、システム・リソースを制御したり、システム機能を実行するために、オペレーティング・システムによって作成されます。iSeries サーバーが起動されると、ユーザー入力なしにシステム・ジョブが開始されます。これらのジョブは、オペレーティング・システムの開始はもとより、またサブシステムの開始と終了、ジョブのスケジューリングにいたるさまざまなタスクを実行します。

システム・ジョブの種類とその機能について、以下に示します。

システム始動ジョブ: Scpf (開始制御プログラム機能)

これが、システムを開始したときに中心的な役割をするジョブです。Scpf は Qlus を除くすべてのシステム・ジョブを開始し、システムを使用可能な状態にします。このジョブはシステムが開始した後も活動状態を継続し、低優先順位および実行時間が長くなる可能性のあるシステム機能を実行する環境を提供します。Scpf は電源遮断 (Pwrdownsys) 処理中も実行し、マシン処理を終了させるジョブでもあります。

Qwcbtclnup (ジョブ・テーブル・クリーンアップ)

このジョブは、システムの始動時にジョブ構造体を使用可能であることを確認するために使用されます。このジョブはシステム始動が終わる前に処理を完了しますが、終結処理 (クリーンアップ) するジョブ構造体がたくさんある場合には、システム開始後も実行を継続することができます。このシステム・ジョブは処理が完了すると終了します。

システム・アービター: Qsysarb (システム・アービター)

システム・アービターは高優先順位機能を実行する環境を提供します。システム資源を取り扱い、システムの状態に関する情報を最新に保ちます。システム・アービターは、ただちに処理する必要があり、しかも複数のジョブよりも単一のジョブで扱う方が効率のよいシステム全体に関連するイベントに対処します。Qsysarb および Qcmnarbxx (通信アービター) は、通信要求、装置ロック、回線、制御装置、および装置構成の処理、および他のシステム全体の資源の処理を行います。

Qsysarb2 (システム・アービター 2)

このジョブは、テープ資源の管理、コマンド処理のためのコマンド分析プログラム・スペースの処理、およびオペレーティング・システムに関する他のシステム全体の処理を行います。

Qsysarb3 (システム・アービター 3)

このジョブはシステム上でのジョブ構造体の作成と保守を行います。ジョブ開始に一時的なまたは永続的なジョブ構造体が必要である場合は必ず、要求が Qsysarb3 によって処理されます。

Qsysarb4 (システム・アービター 4)

このジョブはサブシステムの開始と終了を行います。初期電源遮断 (Pwrdownsys) 処理も含まれます。

Qsysarb5 (システム・アービター 5)

このジョブはマシン・イベントの処理を行います。これには、補助電源、continuous powered mainstore (CPM)、システム補助記憶域プール (ASP) および記憶域しきい値、およびロック・テーブル限界値のサポートが含まれます。通常は、マシン・イベントが処理され、対応する CPF メッセージが Qsysopr および Qhst に送信されます。

通信ジョブ: Qlus (論理装置サービス)

Qlus は論理装置 (通信装置とも呼ばれる) のイベント処理を行います。Qlus は正しい通信サブシステムへの装置の割り振りも行います。

Qcmnarbxx (通信アービター)

通信アービターは Qsysarb (システム・アービター) とともに、通信装置だけでなくすべての種類の装置の作業も処理します。この作業には、通信の接続、切断、装置ロック、およびエラー回復処理が含まれます。すべての装置関連の作業は Qcmnarbxx ジョブおよびシステム・アービターを通してスプレッドされます。

Qcmnarbxx システム値は開始済みの通信アービター・ジョブの数を判別します。シングル・プロセッサ・システムでは最低で 3 つの通信アービターが開始されます。

Qsyscomm1 (システム通信)

このジョブはいくつかの通信および入出力 (I/O) 活動を行います。

Q400filsvr (リモート・ファイル・システム通信)

このジョブはリモート・ファイル・システムに対して共通プログラミング・インターフェース通信 (APPN または APPC) を実行します。

データベース・ジョブ: Qdbfstccol (データベース・ファイル統計収集)

このジョブはデータベース・ファイル統計を収集します。これらの統計は、適切なデータベース QUERY の最適化に対して決定的です。

Qdbsrvxr (データベース相互参照)

このジョブは Qsys にある各ファイル・レベル・システム相互参照ファイルを保守します。これらのファイルには、システム内のデータベース・ファイルおよび SQL 情報に関する相互参照情報が含まれています。これらのファイルはライブラリー Qsys にあり、接頭部 Qadb で始まるものです。保守を必要とする主なファイルは Qadbxref という相互参照ファイルです。このファイルにはシステムにある物理データベース、論理データベース、DDM、別名ファイルのそれぞれに関するレコードが含まれています。Qdbsrvxr は、ファイルが作成、変更、削除、復元、名前変更、またはその所有者が変更されたときに、活動化します。

Qdbsrvxr2 (データベース相互参照 2)

このジョブは 2 つのフィールド・レベル相互参照ファイルを保守します。ライブラリー Qsys にある Qadbifld はフィールド相互参照ファイルです。ライブラリー Qsys にある Qadbfld はキー・フィールド相互参照ファイルです。Qdbsrvxr2 は、ファイルが作成、変更、削除されたときに、活動化します。

Qdbsrv01 (データベース・サーバー)

このジョブは、データベース保守タスク・ディスパッチャーと見ることができます。システム上のデータベース・サーバー・ジョブの数は、1 にプロセッサの数の二倍を加えたもの、または 1 に ASP の数の二倍を加えたもののうち、いずれか大きいほうの数です。開始される最少の数は 5 です。Qdbsrv01 は、作業を他のジョブに割り振るメインのシステム・ジョブです。通常、データベース・ファイルを含むライブラリーが復元された直後に、Qdbsrv01 の活動状態は最も高くなります。このジョブには以下の機能が含まれます。

- 新しいアクセス・パスが復元されたことを、システム管理のアクセス・パス保護 (SMAPP) ライセンス内部コード (LIC) に伝令します。すると、SMAPP はこれらのアクセス・パスを保護する必要があるかどうかを判断します。
- アクセス・パスが復元されなかったため再作成が必要なアクセス・パスのリストを準備します。

残りのデータベース・サーバー・ジョブについて、始めの半分の処理は高優先順位要求であり、残りの半分の処理は低優先順位要求です。Qdbsrv02 ~ Qdbsrv05 は高優先順位、Qdbsrv06 ~ Qdbsrv09 は低優先順位です。

Qdbsrvxx (データベース・サーバー、高優先順位)

これらのジョブはシステムのジャーナルおよびコミットメント制御保守を実行するので、高速または短時間実行作業と見なされます。

Qdbsrvxx (データベース・サーバー、低優先順位)


これらのジョブはユーザー・データ・ファイル上でアクセス・パス保守を実行します。通常これらのジョブは非活動ですが、特定の場合にアクセス・パス再作成の実行が活動化される場合があります。これらのジョブが活動状態になる理由には次のいくつかのがあります。

- アクセス・パスとともに保存されなかったデータベース・ファイルを復元した。
- 元になっている物理ファイルとは別に論理ファイルを復元した。
- Rgzpfm コマンドを処理中に取り消した。
- 索引内に見つかった損傷のために索引が無効になった。
- 相互参照を完了するポスト iSeries インストール活動またはその他の DB アップグレード活動。
- 制約の検証。

Qqqtemp1 および Qqqtemp2 (データベース並列性)

データベース並列性システム・ジョブは、DB2 マルチシステムに対する非同期のデータベース処理を実行します。分散しているファイルを照会する場合、このジョブを用いると、あるいくつかのタスクが並列的に処理されて、照会の速度が上がります。

その他のジョブ: Qalert (警報マネージャー)

このジョブは警報の処理に必要なタスクを実行します (警報については、警報サポート  を参照してください)。これには、他のシステムから受け取った警報の処理、そのシステム自体で生成された警報の処理、および制御の範囲の維持などの活動が含まれます。

Qdcpobjx (圧縮解除システム・オブジェクト)

これらのジョブは、必要に応じて、新たにインストールされたオペレーティング・システム・オブジェクトを圧縮解除します。これらのジョブを実行するにあたって、記憶域要件があります。システムで使用可能な記憶域が一定の限界より下であれば、これらのジョブは終了します。圧縮解除システム・オブジェクト・ジョブの数は、プロセッサの数に 1 を加えたものです。

Qfilesys1 (ファイル・システム)

このジョブは統合ファイル・システムのバックグラウンド・プロセスをサポートします。ファイルに加えられる変更は確実に記憶域に書き出され、一般ファイル・システム・クリーンアップ活動もいくつか実行されます。

Qjobscd (ジョブ・スケジュール)

このジョブはシステムのジョブ・スケジュール機能を制御します。Qjobscd はジョブ・スケジュール項目とスケジュール済みジョブのタイマーをモニターします。

Qlur (LU 6.2 再同期)

Qlur は2 フェーズ・コミット再同期処理を行います。

Qpfradj (パフォーマンス調整)

このジョブは、記憶域プール・サイズおよび活動レベルへの変更を管理します。記憶域プールの変更要求は、すべてこのジョブが処理します。さらに、システム値 Qpfradj が 2 または 3 に設定されている場合には、このジョブは記憶域プールのサイズと活動レベルを動的に変更してシステム・パフォーマンスの向上を計ります。

Qsplmaint (システム・スプール保守)

このジョブはシステム・スプール機能を実行します。

ジョブ・プロパティ

ジョブ・プロパティには、ジョブが処理される方法に関する情報が含まれています。これは、ジョブが作成されたときに指定されます。いくつかのプロパティは、ジョブ記述から渡されます。ジョブの作成後、ジョブ・プロパティは、iSeries ナビゲーターの実行管理機能によって表示および管理することができます。iSeries ナビゲーターの「ジョブ・プロパティ」ページは、ジョブを管理するために効果的で使いやすい機能を提供しているため、システム・オペレーターにとって作業しやすい環境になっています。ジョブ・プロパティはどのユーザーでも表示できますが、変更できるのは適切な権限を持ったユーザーだけです。同様に、権限が与えられたユーザーはジョブ・アクションによってジョブを管理できます。iSeries ナビゲーターで、システム・ジョブのプロパティは変更できません。しかし、文字ベースのインターフェースでは、「システム・ジョブの変更 (CHGSYSJOB)」コマンドを使用して、システム・ジョブの実行優先順位を変更できます。

ジョブ・プロパティの処理 ジョブ・プロパティを表示または変更するには、以下の手順を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 「ユーザー接続」で、**iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 処理するジョブのタイプに応じて、「アクティブ・ジョブ」または「サーバー・ジョブ」をダブルクリックします。
5. プロパティを表示または変更するジョブを検索します。
6. 「ジョブ名」を右クリックします。
7. 「プロパティ」を選択します。

ジョブ・プロパティ・シート



汎用ジョブ・プロパティによって、ジョブに関する一般的な情報を表示することができます。この情報には、ジョブがシステムに入れられている場合はジョブ名とそのジョブ・タイプ、ジョブが開始されている場合はジョブの詳細状況とその他の情報が含まれます。

パフォーマンス・プロパティによって、基本的なパフォーマンス情報を表示し、ジョブのパフォーマンスに影響する変更を行うことができます。さらに、ジョブの存在期間に基づいて計算されたパフォーマンス統計 (CPU およびディスク入出力) を表示することができます。ジョブの実行方法に影響する以下の値を変更できます。

- 実行優先順位
- タイム・スライス
- デフォルトの待ち時間

さらに、アクティブ・ジョブに関して計算される「経過パフォーマンス統計」の表示、最新表示、自動最新表示のセットアップ、またはリセットを行うこともできます。詳細については、『経過パフォーマンス統計』を参照してください。

ジョブ待ち行列プロパティは、ジョブ待ち行列にあるか、または、ジョブ待ち行列から開始されたジョブに関して使用できます。現在ジョブ待ち行列にあるジョブの情報は変更できます。ジョブ待ち行列にあるジョブの優先順位を操作したり、ジョブがジョブ待ち行列に入れられた日時を表示したり、さらにジョブが実行されるタイミングを変更したりすることができます。

プリンター出力プロパティによって、ジョブ出力の印刷に影響するプロパティを表示し変更できます。「プリンター出力」ボタンを使用して、ジョブのプリンター出力を表示することもできます。ジョブからの  プリンター出力の切り離し  の選択、プリンターを選択、出力待ち行列とそのライブラリーを選択、情報の印刷順序 (優先順位) の指定、ページ・フッターの指定、さらに枠や見出しを印刷するかどうかの指定を行うことができます。

メッセージ・プロパティによって、照会メッセージおよび中断メッセージの処理方法を指定できます。ジョブがバッチ・ジョブである場合、ジョブを終了させるメッセージ重大度レベルも表示されます。

ジョブ・ログ・プロパティによって、ジョブ・ログに関係した情報を表示したり、変更したり、ジョブ・ログを表示することができます。ジョブ・ログには、ジョブに出された要求に関する情報 (たとえば、ジョブでのコマンド、CL プログラムからのコマンド、およびメッセージなど) が入っています。このページを使用して、ジョブ・ログにメッセージを保持するかどうか、ジョブ・ログが満杯になるときにジョブが取るべきアクション、保持するメッセージの種類、正常に終了するジョブに関してジョブ・ログ (プリンター出力) を印刷するかどうか、さらには各メッセージに含める詳細情報の量などを指定できます。詳細については、『ジョブ・ログ』を参照してください。

セキュリティー・プロパティによって、現在活動中のジョブに関するセキュリティー・プロパティを表示できます。これには、ジョブ・ユーザー ID、ジョブ・ユーザー ID を設定する方法 (設定方法)、現行ユーザー、およびジョブの初期スレッドに関連しているグループ・プロファイルの名前 (Groups) が含まれています。

国際プロパティによって、ジョブに関係するテキストや文字形式、さらには言語や国および地域の情報を表示したり、変更したりできます。これには、日付、時刻、および小数が表示されるときに使用される形式も含まれています。さらには、ジョブが 2 バイト文字セット (DBCS) を扱えるのかも示されます。

スレッド・プロパティによって、現在活動中か、ジョブ待ち行列に置かれているジョブのスレッドに関係した情報を表示できます。「スレッド」ボタンを使用して、ジョブのスレッドを表示することもできます。このページには、ジョブが複数のユーザー・スレッドで実行可能かどうか、ジョブ内の活動スレッドの数、さらにジョブがいつでも実行できるユーザー・スレッドおよびシステム・スレッドの最大数などの情報が含まれています。

サーバー・プロパティによって、サーバー・ジョブの情報を表示できます。サーバー・ジョブごとに、サーバーのタイプ、ジョブ・ユーザー ID、可能であればクライアント IP アドレスを表示できます。クライアント IP アドレスは、このサーバーが現在サービスを提供しているユーザーのアドレスです。

その他のプロパティによって、会計コード、スイッチの設定値、および DDM 接続を活動状態に保つかどうかなど、ジョブに関連した種々のプロパティを表示し変更できます。さらに、ディスク・プール・グループ、ジョブ日付や、ジョブがシステム/36 特殊環境で実行されているかどうかも表示できます。

詳細については、iSeries ナビゲーターのヘルプを参照してください。

プリンター出力の切り離し: V5R2 より前のリリースでは、プリンター出力はプリンターに送信されて削除されるか、またはユーザーにより明示的に削除されるまで、ジョブに付加されていました。

ジョブの終了時にプリンター出力をジョブから切り離すオプションがあります。ジョブから切り離されたプリンター出力は、システムからは削除されませんが、外部待ち行列に存在するようになります。これにより、ジョブがシステムを離れ、システム構造が解放されて他のユーザーがそれを使用できるようになります。

注: プリンター出力をジョブから切り離すことを選択した場合、ジョブを操作してプリンター出力を表示することはできなくなります。それを表示するには、出力が存在する実際の出力待ち行列を表示しなければなりません。

経過パフォーマンス統計: 「経過パフォーマンス統計」ページを使用すると、経過時間に基づいたアクティブ・ジョブまたはスレッドのパフォーマンス統計を表示することができます。これは、ジョブまたはスレッドをモニターし、潜在的な問題を検出する場合に重要です。このような統計には、CPU、ディスク入出力、ページ不在率、平均応答時間、および対話式トランザクションなどが含まれます。

注: スレッドの経過パフォーマンス統計には、平均応答時間および対話式トランザクションは含まれません。

「経過パフォーマンス統計」ページから、以下に示すボタンの 1 つを選択することによって、これらの統計に関する表示オプションを変更することができます。

- **即時最新表示**

経過パフォーマンス統計を最新表示して、統計が計算される時間枠を拡張します。

- **時限最新表示**

経過パフォーマンス統計の自動最新表示をセットアップできるようにします。これを使用して、ジョブのパフォーマンス情報をモニターできます。

- **統計をリセット**

経過パフォーマンス統計をクリアし、統計が計算される時間枠をリセットします。

詳細状況: ジョブの現在の状況は、「詳細状況」の「ジョブ・プロパティ」にある「汎用 (General)」ページに表示されます。以下は、詳細状況の例です。

実行スケジュール (Scheduled to run at)

ジョブは、スケジュール日付および時刻まで待機します。スケジュール日のスケジュール時刻になると、ジョブはジョブ待ち行列から選択できます。

詳細状況では、関連した状況値 (状況 - x) が表示され、ジョブの現在の状況に基づいた追加の情報が示されます。詳細状況に加えて表示される関連した状況値には、以下のようなものがあります。 **終了 - CPU 限度を超えました (Ended - CPU limit exceeded)**

「終了 (Ended)」はジョブの状況 (ジョブが終了した) を指し、「CPU の限界を超えました (CPU limit exceeded)」はジョブの状況 (終了) の原因を表します。

詳細状況には、ジョブの現在の状況を反映した、さらに関連した状況値 [状況 - x (x)] が表示されます。たとえば、ジョブが終了中である場合、以下のような状況値が表示されます。 **終了中 - CPU 限度を超えました (ロックを待っています) Ending - CPU limit exceeded (Waiting for lock)**
CPU 限度を超えたため (CPU 限度を超えました) にジョブが処理を終了しており (終了中)、ジョブはロックを待っています (ロックを待っています)。


ジョブが制限時間内に終了しない場合、この情報が問題分析に役立ちます。

状況値については、「プロパティ」ページで追加情報を得られます。たとえば、「ロックを待っています」の状況では、どのオブジェクトがロック要求に関連しているかが「プロパティ」ページに表示されません。

ジョブの終了: ジョブの終了方法には、制御された終了と即時終了の 2 つがあります。「制御終了」を選択すると、ジョブ内で実行されているプログラムがジョブ終了終結処置と終了を適切に実行できるため、通常はこちらが優れた方法であると言えます。「即時終了」を選択するとジョブはすぐに終了します。これは、制御された終了が失敗した場合にのみ、ジョブをすぐに終了するために使用することをお勧めします。即時終了を選択すると通常のアプリケーション・クリーンアップ・プロシージャが実行されないため、アプリケーション・データが部分的に更新されるなど、望まない結果となる可能性があります。iSeries ナビゲーターでは、制御された終了の時間制限を指定できます。これは指定した時間よりも長く時間がかかる場合に、即時終了を実行できるようにするためです。

ジョブは、ジョブ情報の検索 (QUSRJOB) API などのジョブ API によってジョブの終了状況を調べることができます。制御された終了が選択されると、ジョブ終了終結処置を実行する必要のあるアプリケーションは制御された終了を検出する必要があります。アプリケーションが検出できるようにする 1 つの方法は、非同期信号 SIGTERM による方法です。制御された方法で終了するジョブが、非同期シグナル SIGTERM 用のシグナル処理プロシージャを有する場合、そのジョブに対して SIGTERM シグナルが生成されます。SIGTERM シグナル用のシグナル処理プロシージャに制御が与えられると、そのプロシージャはアプリケーションが制御された方法で終了できるように適切な処置を行うことができます。

ジョブを終了する場合の手順に関する詳細については、ジョブの終了を参照してください。

ジョブの終了および制御された終了の検出に関する詳細については、実行管理の手引き  の第 5 章にある『ジョブの終了』を参照してください。

詳細: アクティブ・ジョブ・アクション: 実行管理機能の「詳細」メニューには、ジョブまたはジョブの初期スレッドによって使用される以下の資源が含まれています。

呼び出しスタック

ジョブの呼び出しスタックが表示されます。呼び出しスタックは、使用されているプログラムおよびプロシージャです。これは、ジョブが実行しているプログラムおよびジョブが実行しているものを検出するのに役立ちます。

ライブラリー・リスト

選択したジョブまたはスレッドのライブラリー・リストが表示されます。ライブラリー・リストは、検索のためのシステムおよびユーザー作成ライブラリー、および検索の順番のリストです。ライブラリーはオブジェクトのコンテナであり、iSeries サーバーのすべてのオブジェクトはオブジェクト名とライブラリーで構成された参照が必要です。オブジェクトは、ライブラリーの検索によって検出されるため、正しく確立されたライブラリー・リストを持つておくことは重要です。ライブラリー・リストが正しく確立されていない場合、ジョブはオブジェクトを検出できないか、間違ったライブラリーのオブジェクトを検出してしまいます。IBM はいくつかのライブラリーを提供しています (ライブラリー名が Q で始まるもの) が、自分で作成することもできます。このダイアログからライブラリーを選択して右クリックすることによって、そのライブラリーのプロパティを処理できます。

ロック・オブジェクト

▶ ロック・オブジェクト ◀ およびジョブがロックを待っているオブジェクトのリストが表示されます。このリストにより、ジョブが使用しているオブジェクトと、ジョブが使用を試行しているオブジェクトを確認できます。

オープンされているファイル

オープンされているファイルのリストと、そのファイルの使用状況 (たとえば、選択されたジョブに発生している入出力操作の数など) の詳細が表示されます。このリストを参照すると、ジョブの状況のデバッグおよび検査に役立ちます。

▶ スレッド

ジョブ内で実行しているスレッドのリスト。初期スレッドは、デフォルトでウィンドウの最上部にリストされます。スレッドとは、ジョブが同時に複数の事柄を処理するために役立つ、作業の独立した部分のことです。◀

▶ トランザクション

ジョブに関連したトランザクションのリスト。トランザクションとは、iSeries システム上の作業論理単位のことで、通常は、データベース操作に関連した用語です。トランザクションの詳細については、iSeries ナビゲーターのヘルプ、または『トランザクション』を参照してください。◀

経過パフォーマンス統計

ある時間間隔に基づいて計算された経過パフォーマンス統計が表示されます。この情報はジョブのモニターや、問題分析に役立ちます。

▶ 最後の SQL ステートメント (Last SQL statement)

「最後の SQL ステートメント (Last SQL statement)」オプションは、ジョブで実行した最後の SQL ステートメントを表示します。この SQL ステートメントは、「SQL スクリプトの実行」に表示されます。「SQL スクリプトの実行」から、ステートメントの再実行、ステートメントの編集と実行、データベース・ファイルや PC ファイルへのステートメントの保管を行うことができます。◀

ジョブ・ログ: ジョブ・ログには、特定のジョブに関連したメッセージのリストが示されます。送信された日時など、メッセージの追加情報も示されます。日付および時刻がジョブ・ログに記録されるため、エラーが発生した日時を判別することができます。メニュー・バーの「ファイル」オプションから「詳細」を選択すると、メッセージの原因、実行すべきアクションの説明、場合によってはエラーからの回復方法など、メッセージに関する詳細な情報が表示されます。ジョブ・ログ・メッセージについては、「拡張」ボタンをクリックすることによって、メッセージを出したプログラムや、メッセージの宛先であるプログラムに関する情報を参照できます。「ジョブ・プロパティ」ダイアログの「ジョブ・ログ」ページで、ジョブ・ログの処理方法およびジョブ・ログに記録される情報の種類を変更することができます。

ジョブのジョブ・ログを参照する方法については、『ジョブ・ログへのアクセス』を参照してください。

スレッド

スレッドは、作業を完了するのにたくさんのジョブ資源を使用するジョブ内の独立した作業単位です。ジョブとスレッドの違いは、スレッドはジョブ内で実行して、ジョブの作業を完了するのを支援するという点です。すべてのアクティブ・ジョブには、最低 1 つのスレッドがあり、これは初期スレッドと呼ばれます。

す。初期スレッドは、ジョブの開始時に作成されます。ジョブ内のスレッドを使用すると、一度にたくさんの処理を実行できます。たとえば、ジョブの処理中に、スレッドはジョブが処理を終了するのに必要なデータを検索し、計算することができます。

スレッドの詳細については、以下のトピックを参照してください。

- **スレッド・アクション**

iSeries ナビゲーターを介してスレッドを管理します。

- **スレッド・タイプ**

このトピックでは、ジョブ内で実行するスレッドのさまざまなタイプを説明しています。

- **スレッド状況**

スレッドのさまざまな状況が説明されています。◀

スレッド・アクション: スレッドは、ジョブが実行中の一時点で、複数の操作を処理するのを支援します。ジョブを効率的に実行し続けるようにしたければ、1つのジョブ内で実行している複数のスレッドをモニターすることが必要になる場合があります。管理したいスレッドを検出したら、そのスレッドを右クリックすることによって、以下に示すアクションを使用できます。

統計のリセット

表示しているリスト情報をリセットして、経過時間を 00:00:00 にセットします。

詳細

スレッドの機能はジョブの機能に類似しているため、この2つに共通するアクションがいくつかあります。以下のアクションに関する詳細情報があります。

- 呼び出しスタック
- ライブラリー・リスト
- ロック・オブジェクト
- トランザクション
- 経過パフォーマンス統計

保留

スレッドを保留にすることができます。スレッドは、複数回保留にできます。オペレーティング・システムが、スレッドの保留回数を追跡します。

解放

保留にされているスレッドを解放します。スレッドを実行するには、保留になっているスレッドを毎回解放する必要があります。

削除 / 終了

選択したスレッド (複数可) を終了できます。詳細については、『スレッドの終了』を参照してください。

スレッド・プロパティ

スレッドのさまざまなプロパティを表示します。

スレッドに実行できるアクションについては、iSeries ナビゲーターのヘルプを参照してください。◀

スレッド・タイプ: スレッド・タイプは、スレッドがシステム上で作成された方法を判別します。


スレッドのタイプは、次のとおりです。

ユーザー

カスタマー・アプリケーションによって作成されたスレッドです。ジョブの初期スレッドは、常にユーザー・スレッドです。複数のユーザー・スレッドを使用する場合は、「複数のスレッドの許可 (Allow multiple threads)」フィールドを「yes」に設定することが必要です。

システム

ユーザーの代わりにシステムによって作成されたスレッドです。一部のシステム機能は、システム・スレッドを使って処理を実行します。カスタマー・アプリケーションが、スレッドを使用するシステム機能を使う場合、システム・スレッドが使用されます。

- 注:** iSeries ナビゲーターでスレッドを表示すると、デフォルトでは、最初のスレッドのタイプとして、リストに「初期」と表示されます。初期スレッドは、ジョブの開始時にジョブの中で作成される最初のスレッドです。iSeries ナビゲーターでは、初期スレッドが  アイコンで表示されます。初期スレッドを削除 / 終了することはできません。

スレッド状況: スレッドの現在の状況は、「詳細状況」の「スレッド・プロパティ (Thread properties)」にある「汎用 (General)」ページに表示されます。詳細状況には以下のような情報が示されます。

待ち行列解除の待機 (Waiting for dequeue)

ジョブのスレッドは、待ち行列解除操作の完了を待機しています。待ち行列解除は、待ち行列からメッセージを除去する操作です。メッセージは、1 人の人物または 1 つのプログラムから別の人物またはプログラムに送信される通信内容です。特にメッセージは、1 つのスレッドによって待ち行列システム・オブジェクトの待ち行列に入れられ (配置され)、別のスレッドによって待ち行列から解除 (除去) されます。

- 注:** 待ち行列解除の待機がプロパティ・ページに表示されている場合、待機中の待ち行列を識別する追加情報が表示されます。ジョブまたはスレッドが、OS/400 オブジェクトに対する待ち行列解除操作が完了するのを待機している場合、10 文字のオブジェクト名、そのライブラリー、およびオブジェクト・タイプが表示されます。ジョブまたはスレッドが内部オブジェクトに対する待ち行列解除操作が完了するのを待機している場合、30 文字のオブジェクト名が表示されます。内部オブジェクトの場合、30 文字の名前を表示するにはジョブ制御特殊権限 (*JOBCTL) が必要です。

詳細状況では、関連した状況値 (状況 - x) が表示され、スレッドの現在の状況に基づいた追加の情報が示されます。詳細状況に加えて表示される関連した状況値には、以下のようなものがあります。

保留 (n) (Held (n))

個々のスレッドが保留にされています。ジョブとは異なり、スレッドは一度に複数回を保留にできません。スレッド状況に続く数値 (たとえば、(3)) は、スレッドが解放されずに保留された回数をユーザーに通知します。たとえば、スレッドで 3 回の保留が行われ、1 回解放があった場合は、まだ 2 回の保留があるということになります。数値は、状況が「プロパティ」ページに表示される場合のみ表示され、リストに表示される場合は表示されません。スレッド処理を再開するには、スレッドの解放アクションを選択します。

さまざまなスレッド状況の詳細については、iSeries ナビゲーターのヘルプを参照してください。

ジョブ待ち行列

ジョブ待ち行列には、サブシステムによる処理を待っているジョブの番号付きリストがあります。ジョブ待ち行列は、バッチ・ジョブがサブシステムで活動状態になる前に、最初に送られる場所です。ジョブは、番号が一致するまでそこに入れられています。ジョブ待ち行列のジョブが処理されるためには、ジョブ待ち行列からそのジョブを受け入れる活動サブシステムがなければなりません。サブシステムが開始すると、作業が入れられているジョブ待ち行列の割り振りが試行されます。ジョブ待ち行列のジョブが処理されるためには、その待ち行列が正しく割り振られる必要があります。そのため、サブシステムが複数のジョブ待ち行列のジョブを処理するとしても、特定のジョブ待ち行列のジョブを処理するのは一度に 1 つのサブシステムだけです。

サブシステムは優先順位に基づき、その各優先順位に構成されている制限内でジョブ待ち行列からジョブを選択します。ジョブにはそれぞれジョブ待ち行列の優先順位があり、ジョブがそのジョブ待ち行列に入れているときにはそのジョブ優先順位によって管理されます。ジョブ待ち行列の基本セットがシステムとともに提供されています。さらに、必要に応じて追加のジョブ待ち行列を作成することができます。

注: 「ジョブ待ち行列のリストのオープン (QSPOLJBQ)」や「ジョブ待ち行列情報の検索 (QSPRJOBQ)」などの API を呼び出して、ジョブ待ち行列についての情報を取得することができます。API の詳細については、『アプリケーション・プログラミング・インターフェース (API)』を参照してください。

ジョブ待ち行列に入っているジョブの詳細については、以下に示すトピックを参照してください。

- **作業はどのようにシステムに入れられるか**
作業がどのようにジョブ待ち行列に入れられるかを説明します。
- **ジョブ待ち行列はどのように機能するか**
ジョブがどのようにジョブ待ち行列からサブシステムに渡されるかを説明します。
- **ジョブ待ち行列の作成**

実行管理の手引き  の第 8 章に記載されている情報を使用して、ジョブ待ち行列を作成します。

ジョブ待ち行列はどのように機能するか

ジョブ待ち行列が活動サブシステムによって割り振られると、ジョブはジョブ待ち行列から取り出されて、サブシステム下で実行されます。ジョブ待ち行列からジョブが選択される方法を決めるさまざまな要素があります。ジョブ待ち行列から出されないジョブは、効率のためにあるジョブ待ち行列から別のジョブ待ち行列へ移動する場合があります。

ジョブ待ち行列からジョブが取り出される場合に関する要素を以下に示します。

サブシステムごとのアクティブ・ジョブの最大数 (Maximum active jobs for subsystems)

これは、サブシステムで実行できるジョブの最大数を表します。この制限に達すると、そのサブシステムではそれ以上ジョブは開始されません。

ジョブ待ち行列ごとのアクティブ・ジョブの最大数 (Maximum active jobs for job queues)

これは、サブシステム内で同時に実行できるジョブ待ち行列のジョブの最大数を表します。この制限に達すると、そのジョブ待ち行列からそれ以上ジョブは開始されません。

ジョブ待ち行列の優先順位 (Priority on job queue)

実行を待っているジョブは、ジョブ待ち行列優先順位に基づいて選択されます。サブシステムは優先順位の高いジョブ (ジョブ待ち行列の優先順位は 0~9 までで、0 が最も高い優先順位です) から先に実行しようとしませんが、優先順位に基づいて実行されているジョブの数が優先順位ごとの「最大アクティブ・ジョブ

数」に達すると、次の優先順位のもの処理されます。(同じ優先順位のジョブがジョブ待ち行列に入れられる場合、最初のジョブがまず実行され、次に 2 番目のものというように順番に処理されます。)

詳細については、『ジョブ待ち行列でのジョブの優先順位の変更』を参照してください。

順序

サブシステム記述のジョブ待ち行列項目に順序を指定します。順序番号は、サブシステムがジョブ待ち行列を処理する順番を定義します。サブシステムは、最初に順序番号が最小のジョブ待ち行列のジョブを処理します。ジョブ待ち行列にジョブがなくなる場合、ジョブ待ち行列に関連した最大値の 1 つに達する場合、サブシステムは次に高い順序番号のジョブ待ち行列を処理します。

ジョブの移動については、『他のジョブ待ち行列へのジョブの移動』を参照してください。

サブシステム

サブシステムとは、iSeries サーバーで作業が処理される場所のことです。システム・ジョブを除くすべてのジョブは、サブシステムの中で実行されます。

さらに技術的な意味において、サブシステムとは、システムがワークフローと資源使用を調整するために用いる単一の事前定義された操作環境のことをいいます。システムには、それぞれ独立して作動する複数のサブシステムを含めることができます。サブシステムは資源を管理します。各サブシステムは、固有の操作を実行できます。たとえば、あるサブシステムは対話式ジョブだけを処理し、他のサブシステムはバッチ・ジョブだけを処理するように設定できます。また、複数のサブシステムで種々の作業を処理できるように設計することもできます。サブシステムの数および各サブシステムで処理する作業のタイプは、ユーザーが設定できます。

サブシステムは活動状態または非活動状態にすることができます。活動サブシステムは、すでに開始されているサブシステムのことです(詳細については、『サブシステムの開始』を参照してください)。非活動状態のサブシステムとは、まだ開始しておらず、停止していないサブシステムのことです(詳細については、『サブシステムの停止』を参照してください)。

制御サブシステムは、システムの開始時に自動的に開始される対話式サブシステムであり、システム・オペレーターがシステムの制御に使用するサブシステムです。

サブシステム・ジョブは、オペレーティング・システムによって作成され、資源を管理し、ジョブを開始、制御、および終了します。

注: 「サブシステム情報の検索 (QWDRSBSD)」や「システム状況の検索 (QWCRSSTS)」などの API を呼び出して、サブシステムについての情報を取得することができます。API の詳細については、『アプリケーション・プログラミング・インターフェース (API)』を参照してください。

サブシステムの詳細については、以下を参照してください。

サブシステム記述

サブシステムの実行時の特性は、サブシステム記述に定義されています。

システム出荷時のサブシステム

IBM 提供のサブシステム構成が 2 つあります。

ユーザー定義サブシステム

独自にサブシステム記述を作成することが可能です。

サブシステムのプロパティ

サブシステムの属性が示されています。

サブシステムのライフ・サイクル



これは、作業が iSeries サーバーでどのように処理されるかを説明します。

サブシステム記述

サブシステムの実行時特性は、**サブシステム記述**と呼ばれるオブジェクトに定義されています。サブシステム記述は一まとまりの指示のような働きをし、どれだけの作業が、どこから、どのようにサブシステムに入ってくるか、およびそれらの作業を実行するためにサブシステムがどの資源を使用するかを指定しています。サブシステム記述が定義または作成されると、サブシステムが作成されます。活動サブシステムの名前には、サブシステム記述の単純名が使用されます。


サブシステム記述にどのような情報が含まれているかについて詳細は、次の表を参照してください。

| サブシステム記述にある情報 | | 記述 | 追加情報 (実行管理の手引き) |
|---------------|-----------|--|--|
| サブシステム属性 | | 全体的なシステム特性を指定します。 <ul style="list-style-type: none">運用属性 (サブシステム内で同時に活動状態にできるジョブの数やサインオン画面など)。サブシステムが使用するメモリー・プール。サブシステム記述に対する権限。サブシステム記述のテキスト記述。 | 実行管理の手引き  の第 4 章『サインオン画面ファイルの変更』。 |
| 実行処理項目 | | サブシステム記述の実行処理項目は、ジョブが送られてきた場合にサブシステムで処理することを受け入れる送信元を指定します。言い換えれば、処理すべき作業がどこからサブシステムに入れるかを指定します。 | 実行管理の手引き  の第 4 章『実行処理項目』。 |
| | 自動開始ジョブ項目 | サブシステムの開始とともに開始される自動開始ジョブを指定します。 | 実行管理の手引き  の第 9 章『自動開始ジョブ』。 |
| | 通信項目 | 他のシステムが作業を実行依頼するために使用する通信装置を指定します。 | 実行管理の手引き  の第 10 章『通信ジョブ』。 |
| | ジョブ待ち行列項目 | 作業をどのジョブ待ち行列から引き受けるかを指定し、受け入れる作業の量を決定します。 | 実行管理の手引き  の第 8 章『バッチ・ジョブ』。 |
| | 事前開始ジョブ項目 | 事前開始ジョブを開始するときに使用する情報を指定します。 | 実行管理の手引き  の第 11 章『事前開始ジョブ』。 |

| サブシステム記述にある情報 | | 記述 | 追加情報 (実行管理の手引き) |
|---------------|-------------|---|---|
| | ワークステーション項目 | 作業をどのワークステーションから引き受けるかを指定します。 | 実行管理の手引き の第 6 章『対話式ジョブ』。  |
| 経路指定項目 | | 使用するサブシステム・メモリー・プール、実行する制御プログラム、および実行時情報を指定します。 | 実行管理の手引き の第 4 章『経路指定項目』。  |

サブシステム記述オブジェクトは、すべてのシステムに用意されています。以下に、iSeries サーバーの出荷時のサブシステム記述に加えられる更新を示します。この表では、オブジェクトごとに次の項目をまとめています。

オブジェクト名
 オブジェクトの更新に使用されるコマンド
 デフォルト以外のコマンド・パラメーター

この表および実行管理の手引き  の付録 C で、iSeries に用意されているサブシステム記述のほとんどを調べることができます。

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|--------|----------------------|---|
| QBASE | 通信項目追加 (ADDCMNE) | SBSD (QSYS/QBASE) DEV (Q1PLOC) DFTUSR (*NONE) MODE (Q1PMOD) MAXACT (0) |
| QBASE | 通信項目追加 (ADDCMNE) | SBSD (QSYS/QBASE) REMLOCNAME (Q1PLOC) DFTUSR (*NONE) MODE (Q1PMOD) MAXACT (0) |
| QBASE | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QBASE) PGM (QSYS/QZCSRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOBID (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (2) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|--------|----------------------|--|
| QBASE | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QBASE) PGM (QSYS/QNPSERVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOB (*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |
| QBASE | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QBASE) PGM (QSYS/QZRCSEVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOB (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (2) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |
| QCMN | 通信項目追加 (ADDCMNE) | SBSD (QSYS/QCMN) REMLOCNAME (Q1PLOC) DFTUSR (*NONE) MODE (Q1PMOD) MAXACT (0) |
| QCMN | 通信項目追加 (ADDCMNE) | SBSD (QSYS/QCMN) DEV (Q1PLOC) DFTUSR (*NONE) MODE (Q1PMOD) MAXACT (0) |

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|--------|----------------------|--|
| QCMN | 事前開始ジョブ項目追加 (ADDPJE) | SBSDB (QSYS/QCMN) PGM (QSYS/QZRCRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOBD (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |
| QCMN | 事前開始ジョブ項目追加 (ADDPJE) | SBSDB (QSYS/QZSCSRVR) PGM (QSYS/QZSCSRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOBD (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |
| QCMN | 事前開始ジョブ項目追加 (ADDPJE) | SBSDB (QSYS/QCMN) PGM (QSYS/QNPSEVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOBD (*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|---------|------------------------|--|
| QSERVER | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QSERVER) PGM (QSYS/QZDAINIT) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOB (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (1) CLS (QGPL/QPWSERVER *CALC *NONE *CALC) |
| QSERVER | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QSERVER) PGM (QSYS/QPWFSEVSO) USER (QUSER) STRJOBS (*NO) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*USRPRF) JOB (*PGM) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QPWFSEVSERVER *CALC *NONE *CALC) |
| QSYSWRK | ジョブ待ち行列項目追加 (ADDJOBQE) | SBSD (QSYS/QSYSWRK) JOBQ (QSYS/Q1PSCHQ) MAXACT (1) SEQNBR (70) |
| QSYSWRK | ジョブ待ち行列項目追加 (ADDJOBQE) | SBSD (QSYS/QSYSWRK) JOBQ (QSYS/Q1PSCHQ2) MAXACT (1) SEQNBR (80) |
| QSYSWRK | ジョブ待ち行列項目追加 (ADDJOBQE) | SBSD (QSYS/QSYSWRK) JOBQ (QSYS/Q1PSCHQ3) MAXACT (1) SEQNBR (90) |
| QSYSWRK | 自動開始ジョブ項目追加 (ADDAJE) | SBSD (QSYS/QSYSWRK) JOB (QGLDPUBA) JOB(QSYS/QGLDPUBA) |
| QSYSWRK | 自動開始ジョブ項目追加 (ADDAJE) | SBSD (QSYS/QSYSWRK) JOB (QGLDPUBE) JOB(QSYS/QGLDPUBE) |

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|---------|----------------------|--|
| QSYSWRK | 自動開始ジョブ項目追加 (ADDAJE) | SBSD (QSYS/QSYSWRK) JOB (QPM400) JOB (QSYS/Q1PJOB) |
| QSYSWRK | 通信項目追加 (ADDCMNE) | SBSD (QSYS/QSYSWRK) DEV (Q1PDEV) JOB (*USRPRF) DFTUSR (QUSER) MODE (Q1PMOD) MAXACT (*NOMAX) |
| QSYSWRK | 通信項目追加 (ADDCMNE) | SBSD (QSYS/QSYSWRK) DEV (Q1PLOC) JOB (*USRPRF) DFTUSR (QPM400) MODE (Q1PMOD) MAXACT (*NOMAX) |
| QSYSWRK | 通信項目追加 (ADDCMNE) | SBSD (QSYS/QSYSWRK) RMTLOCNAME (Q1PLOC) JOB (*USRPRF) DFTUSR (QPM400) MODE (Q1PMOD) MAXACT (*NOMAX) |
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2150) CMPVAL (TOTNTP) PGM (QSYS/QTOTSNT) CLS (QSYS/QSYSCLS10) |
| QSYSWRK | 経路指定項目追加 (ADDRTE) | SBSD (QSYSWRK) SEQNBR (300) CMPVAL (PGMEVOKE 29) PGM (*RTGDTA) CLS (QSYS/QSYSCLS50) MAXACT (*NOMAX) POOLID (1) |
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2536) CMPVAL ('QZSCSRVSD') PGM (QSYS/QZSCSRVSD) CLS (QGPL/QCASERVR) |
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2537) CMPVAL ('QZHQSRVD') PGM (QSYS/QZHQSRVSD) CLS (QGPL/QCASERVR) |

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|---------|--------------------|---|
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2538) CMPVAL ('QNPSERVD') PGM (QSYS/QNPSERVD) CLS (QGPL/QCASERVR) |
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2539) CMPVAL ('QZRCRVSD') PGM (QSYS/QZRCRVSD) CLS (QGPL/QCASERVR) |
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2540) CMPVAL ('QZSOSGND') PGM (QSYS/QZSOSGND) CLS (QGPL/QCASERVR) |
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2541) CMPVAL ('QZSOSMAPD') PGM (QSYS/QZSOSMAPD) CLS (QGPL/QCASERVR) |
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2170) CMPVAL ('QSYEIMMON') PGM (QSYS/QSYEIMMON) CLS (QSYS/QSYSCLS20) MAXACT (*NOMAX) POOLID (1) |
| QSYSWRK | 経路指定項目追加 (ADDRTGE) | SBSD (QSYS/QSYSWRK) SEQNBR (2200) CMPVAL ('QYASPPGM') PGM (QSYS/QYASPPGM) CLS (QSYS/QSYSCLS20) MAXACT (*NOMAX) POOLID (1) |

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|---------|----------------------|---|
| QUSRWRK | 事前開始ジョブ項目追加 (ADDPJE) | SBSDB (QSYS/QSYSWRK) PGM (QSYS/QZSOSIGN) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBDB (QSYS/QZBSJOBDB) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |
| QUSRWRK | 事前開始ジョブ項目追加 (ADDPJE) | SBSDB (QSYS/QUSRWRK) PGM (QSYS/QZSCSRVS) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBDB (QSYS/QZBSJOBDB) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |
| QUSRWRK | 事前開始ジョブ項目追加 (ADDPJE) | SBSDB (QSYS/QUSRWRK) PGM (QSYS/QNPSESRVS) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBDB (QSYS/QZBSJOBDB) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC) |

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|---------|----------------------|---|
| QUSRWRK | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QUSRWRK) PGM (QSYS/QZRC SRVS) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOB (QSYS/QZBSJOB D) MAXUSE (1) WAIT (*YES) POOLID (1) CLS (QG PL/QC ASERVR *CALC *NONE *CALC) |
| QUSRWRK | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QUSRWRK) PGM (QSYS/QZD ASOINIT) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOB (*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QG PL/QPWFSE RVER *CALC *NONE *CALC) |
| QUSRWRK | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QUSRWRK) PGM (QSYS/QZH QSSRV) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOB (QSYS/QZBSJOB D) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QG PL/QC ASERVR *CALC *NONE *CALC) |

| オブジェクト | 追加、削除、または更新 | デフォルト以外のパラメーター |
|----------------------------------|----------------------|---|
| QUSRWRK | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QUSRWRK) PGM (QSYS/QZDASSINIT) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOB (QSYS/*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QSYS/QPWFSEVER *CALC *NONE *CALC) |
| QUSRWRK (QSYSWRK から QUSRWRK に移動) | 事前開始ジョブ項目追加 (ADDPJE) | SBSD (QSYS/QUSRWRK) PGM(QSYS/QRWTSRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOB (*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QSYS/QSYSCLS20 *CALC *NONE *CALC) |

システム出荷時のサブシステム

変更を行わずにそのまま使用できる IBM 提供のサブシステム構成は 2 つあります。

システム起動時にシステムが使用する構成は、「制御サブシステム記述」システム値 (QCTLSBSD) によって制御されます。デフォルト構成は以下のサブシステム記述から成っています。

Qbase (制御サブシステム)

Qbase は対話式ジョブ、バッチ・ジョブ、および通信ジョブをサポートします。このサブシステムには自動開始ジョブがあり、Qusrwrk、Qserver、および Qspl サブシステムを自動的に開始します。

Qsyswrk

これはシステム作業サブシステムです。このサブシステムには、システム起動時およびシステムが制限状態から回復したときに自動的に開始するシステム機能をサポートするジョブが含まれています。

| | |
|-------------------------------|--|
| Qusrwrk | これはユーザー作業サブシステムです。このサブシステムには、ユーザーの代行処理を行うためにサーバーによって開始されるジョブが含まれています。 |
| Qserver Qspl | これはファイル・サーバー・サブシステムです。これはスプール・サブシステムです。このサブシステムは読み取りプログラムおよび書き出しプログラムのジョブをサポートします。 |

IBM 提供のもう 1 つの構成は、次のサブシステム記述から成っています。

| | |
|----------------------------|--|
| Qctl (制御サブシステム) | Qctl には、Qinter、Qbatch、Qcmn、Qusrwrk、Qserver および Qspl サブシステムを自動的に開始する自動開始ジョブがあります。 |
| Qinter | これはシステム作業サブシステムです。このサブシステムには、システム起動時およびシステムが制限状態から回復したときに自動的に開始するシステム機能をサポートするジョブが含まれています。 |
| Qbatch | これはユーザー作業サブシステムです。このサブシステムには、ユーザーの代行処理を行うためにサーバーによって開始されるジョブが含まれています。 |
| Qcmn Qspl | これはファイル・サーバー・サブシステムです。これはスプール・サブシステムです。このサブシステムは読み取りプログラムおよび書き出しプログラムのジョブをサポートします。 |
| Qsyswrk | これはシステム作業サブシステムです。このサブシステムには、システム起動時およびシステムが制限状態から回復したときに自動的に開始するシステム機能をサポートするジョブが含まれています。 |
| Qusrwrk | これはユーザー作業サブシステムです。このサブシステムには、ユーザーの代行処理を行うためにサーバーによって開始されるジョブが含まれています。 |
| Qserver | これはファイル・サーバー・サブシステムです。 |


Qbase 構成では、Qctl 構成で実行可能なすべての機能を実行することができ、しかもより少ないサブシステムで構成されているので管理が容易です。

Qctl デフォルト構成では、システム活動をそれぞれのタイプに応じて種々のサブシステムに類別することによって、システム操作をより個別化して制御を行うことができます。たとえば、週末または夜間にバッチ・ジョブを実行する必要があり、しかもその間はコンソール以外からはだれもサインオンできないようにしたい場合、Qctl 構成では Qinter サブシステムを終了するだけでこれを行うことができます。

ユーザー独自のサブシステム構成を作成することを検討しているのであれば、Qbase 構成ではなく Qctl 構成を出発点として使用したほうが作業が簡単になります。

ユーザー定義サブシステム

IBM では、サブシステム記述をシステムに標準添付しています。独自にサブシステム記述を作成することも可能です。既存のサブシステム記述をコピーしたものを変更することができますし、全く新しい記述を作成することもできます。

詳細については、実行管理の手引き  の第 4 章にある『サブシステム記述の作成』を参照してください。

サブシステムのプロパティ

サブシステムには属性、すなわちプロパティがあります。これらのプロパティは、サブシステムの現在の状況に関する情報、つまりサブシステム記述に指定されている値に関する情報を提供します。iSeries ナビゲーターを使用して、活動サブシステムに関する以下のプロパティを表示することができます。

| | |
|---------------|--|
| サブシステム | サブシステムの名前、およびサブシステム記述を含むライブラリー。 |
| 説明 | サブシステムの説明。 |
| 状況 | サブシステムの現在の状況。ヘルプには、起こり得る状況の詳細が含まれています。 |
| アクティブ・ジョブ | サブシステム内で実行中または実行待機中である現在活動状態にあるジョブの数。この数にはサブシステム・ジョブは含まれません。 |
| アクティブ・ジョブの最大数 | サブシステム内で実行中または実行待機中の活動状態になり得るジョブの数。 |
| サブシステム・ジョブ | サブシステム・ジョブの名前 (ユーザーおよび番号を含む)。 |



サブシステムのプロパティを表示するには、以下の手順を行います。

1. iSeries ナビゲーターで、「ユーザー接続」を展開します。
2. 使用している **iSeries** サーバーの接続を展開します。
3. 「実行管理機能」を展開します。
4. 「サブシステム」を展開します。
5. 「活動サブシステム」を展開します。
6. 表示するサブシステムを右マウス・ボタン・クリックしてから、「プロパティ」を選択します。

サブシステムのライフ・サイクル

サブシステムのライフ・サイクルは、サブシステムの開始とともに始まり、サブシステムの停止とともに終了します。開始から停止までの間、作業がサブシステム内で処理されます。詳細については、以下を参照してください。

- サブシステムの開始
- サブシステム開始時に何が行われるか
- サブシステムの停止


サブシステム開始時に何が行われるか: サブシステムの開始時に、システムはいくつかの項目を割り振り、そのサブシステムが作動可能になるのに先立って、自動開始ジョブおよび事前開始ジョブを開始します。項目がどのように割り振られるかを判別するために、サブシステム記述が使用されます。

次のリストは、システムが開始したときに生じるイベントの連続を表します。

1. サブシステム開始要求が発行されます。

2. メモリー・プールが割り振られます。

メモリーはサブシステム記述に定義されたプールに割り振られます。各定義済みプールに割り振られるメモリーは、「基本」メモリー・プールから取られます。「基本」メモリー・プールに使用できるメモリーの量が、基本メモリー・プール最小サイズ (Qbaspool) システム値より小さい場合は、システムはプールに記憶域を割り振りません。システムが必要なメモリーすべてを割り振ることができない場合には、システムはその時点で使用可能なメモリーを割り振り、残りはメモリーが使用可能になった時点で割り振ります。

実行管理の手引き  の第 4 章にある『プール割り振り』を参照してください。

3. ディスプレイ装置が割り振られます。

- ワークステーション項目があり、装置がオンに構成変更されていて、しかもまだ他のサブシステムによって割り振られていない場合には、サブシステムはその装置を割り振り、サインオン画面を表示することができます。
- 装置がオンに構成変更されていて、しかもすでに他のサブシステムによって割り振られ、現在サインオン画面が表示されている (そのサインオン画面は、2 番目のサブシステムが開始される前に表示されていた) 場合には、2 番目のサブシステムは最初のサブシステムからその装置を割り振り、サインオン画面を表示できます。
- 装置がオンに構成変更されていない場合、サブシステムはその装置を割り振ることができません。システム・アービター (Qsysarb) および Qcmnarbxx のジョブは、オフに構成変更されているすべての装置に対してロックのままです。

実行管理の手引き  の第 4 章にある『ワークステーション装置割り振り』を参照してください。

4. 通信装置が割り振られます。

要求は、Qlus (LU サービス) システム・ジョブに送られます。このシステム・ジョブは、すべての通信装置の装置割り振りを行います。

実行管理の手引き  の『通信装置』および『モード割り振り』を参照してください。

5. ジョブ待ち行列が割り振られます。

サブシステムがすでに別の活動サブシステムに割り振られている場合は、サブシステムはジョブ待ち行列を割り振ることはできません。

6. 事前開始ジョブが開始します。

7. 自動開始ジョブが開始します。

8. 環境が作業可能になります。

メモリー・プール

メモリー・プールとは、メイン・メモリーまたは記憶域の論理的な区画のことで、ジョブまたはジョブ・グループの処理のために予約されています。iSeries システムでは、すべての主記憶域はメモリー・プールと呼ばれる論理的な割り振りに区分されます。デフォルトでは、システムはメモリー・プールを管理します。必要に応じて、システムはメモリー・プールへのデータおよびプログラムの転送も管理します。

メモリー・プールの数やサイズを制御することによって、サブシステムで実行できる作業の量を制御することができます。サブシステムに割り当てられるメモリー・プールのサイズが大きくなればなるほど、より多くの作業をそのサブシステムで行うことが可能になります。

注: システムのチューニングや管理は、iSeries サーバーでのワークフローの効率を高めるのに役立ちますが、不適切なハードウェア資源には対処できません。求められる作業負荷の量が著しく大きい場合、ハードウェアのアップグレードを考慮してください。

ジョブがメモリーを取得するメモリー・プールは常に同じプールであり、活動レベルを制限しています。システム・ジョブ (Scpf、Qsysarb、および Qlus など) は、基本プールからメモリーを取得しますが、マシン・プール活動レベルを使用します。サブシステム・モニターはそのメモリーをサブシステム記述の最初のプールから入手しますが、活動レベルは使用されません。したがって、サブシステム・モニターは、活動レベルの設定には関係なく、いつでも実行が可能です。

注: 「システム状況の検索 (QWCRSSTS)」などの API を呼び出して、メモリー・プールについての情報を取得することができます。詳細については、『アプリケーション・プログラミング・インターフェース (API)』を参照してください。

メモリー・プールの詳細については、以下を参照してください。

- メモリー・プールの活動レベル
- メモリー・プールの種類

メモリー・プールの活動レベル

メモリー・プールの活動レベルによって、メモリー・プールで同時に活動状態にできるスレッドを制限して、システムの資源の使用効率を高めることができます。

メモリー・プールの活動レベルは、メモリー・プール内で CPU を同時に使用できるスレッドの数です。このレベルの制御はシステムが管理します。スレッドの処理過程で、システム資源やワークステーション・ユーザーからの応答をプログラムが待機する状態になることがしばしばあります。このような待機の過程では、スレッドはメモリー・プールの活動レベルの使用を放棄し、処理可能状態にある他のスレッドがそれに代わります。

同時に実行できるスレッドの数を超える多くのスレッドが開始された場合、活動レベルの制御によって、過剰なスレッドは処理装置が使用できるまで待機しなければなりません (通常、これは短時間です)。メモリー・プールの活動レベルによって、サブシステムの種々のメモリー・プールにおけるメイン・メモリーの競合の度合いを制限することができます。

実行中のスレッド (または活動スレッド) の数は、プロセッサの競合に適していて、メモリー・プールの活動レベルに対してカウントされるスレッドの数を指します。したがって、入力、メッセージ、装置の割り振り、またはファイルのオープンを待機しているスレッドは、活動スレッドには含めません。不適格スレッド (スレッドが実行可能であっても、メモリー・プールの活動レベルが最大になっている場合) は、活動スレッドには含めません。

活動レベルはどのように機能するか


スレッドの処理は補助記憶域から必要なデータを取り出すときにわずかに中断されるため、メモリー・プールで同時に複数のスレッドを活動状態にすることができます。この遅延時間は通常は短時間ですが、この間に他のスレッドを実行することができます。活動レベルを使用することによってマシンは、1つのメモリー・プールで多くのスレッドを処理し、同時に、競合のレベルをユーザーが指定する範囲に保つことができます。


最大活動レベル

メモリー・プールの最大活動レベルに達すると、メモリー・プールを必要とする追加のスレッドは不適格状態になり、メモリー・プールの活動スレッドの数が最大活動レベルより低くなるか、あるいはスレッドがそのタイム・スライスの終了に至るまで待ちます。あるスレッドがメモリー・プールの使用を放棄すると、活動状態でない別のスレッドがその優先順位にしたがって実行適格になります。たとえば、実行中のスレッドがワークステーションからの応答を待機する状態になると、そのスレッドは活動レベルを放棄し、活動レベルは最大のレベルではなくなります。

メモリー・プールの活動レベルの定義

メモリー・プールおよび活動レベルを正しく定義できるかどうかは、メモリー・プールのサイズ、CPU の数、ディスク装置アームの数、およびアプリケーションの特性にかかっています。

活動レベルを正しく設定する方法の詳細については、実行管理の手引き  の第 14 章にある『パフォーマンスの調整』を参照してください。

詳細については、実行管理の手引き  の第 4 章にある『システム活動のレベルの制御』を参照してください。

メモリー・プールの種類

メモリー・プールとは、主記憶域または補助記憶域の分割された一区域です。iSeries システムでは、すべての主記憶域は記憶域プールと呼ばれる論理的な割り振りに区分されます。システムには私用と共用の二種類のメモリー・プールがあります。私用プールと共用プールを合わせて最大 64 個のメモリー・プールまで、同時に活動状態にすることができます。

私用メモリー・プール

iSeries ナビゲーターにあるサブシステム名で識別されるプールで、この中では一つのサブシステムしかジョブを実行できません。私用プールは、複数のサブシステムで共用することができない主記憶域プールです。私用プールには指定されたサイズの記憶域が含まれており、この記憶域は 1 つのサブシステムしか使用できません。62 個まで私用プールを割り振って、活動サブシステムで使用することができます。プログラムが入ってしまうほど、私用プールを大きくする必要はありません。

共用メモリー・プール


共用メモリーとは、複数のサブシステムがジョブを実行できるプールです。共用記憶域プールを使用することによって、システムは種類の似たジョブを複数のサブシステムに分散することができます。しかもそれらのジョブを同じ記憶域プールで実行することができます。システムで定義されている共用メモリー・プールは 64 個あり、そのうちの 63 個についてはサブシステム記述の作成時点でその使用を指定することができます。マシン・プールはシステム使用のために予約済みです。共用プールは特

殊または汎用のいずれかです。マシン・プールおよび基本プールは特殊共用プールと見なされ、それ以外のすべてのプールは汎用共用プールと見なされます。

出力待ち行列

出力待ち行列とは、プリンター出力ファイル（スプールされるファイルとも呼ばれる）が処理され、プリンターに送信されるのを待機する領域です。プリンター出力は、システム、または印刷ファイルを使用するユーザーのいずれかによって作成されます。印刷ファイルは、プリンター出力の属性のデフォルト値が設定されているテンプレートまたはガイドラインのようなものです。これは、プリンター出力のライフ・サイクルの始まりです。

印刷ファイルには、プリンター出力を送信する方法を指定する、出力待ち行列 (OUTQ) 属性およびプリンター (DEV) 属性が含まれます。デフォルト設定は通常 *JOB です。この設定では、プリンター出力を送信する方法は、出力待ち行列およびプリンターのジョブ属性によって判別されます。設定されている出力待ち行列およびプリンターのジョブ属性は、ジョブの作成時に入手される情報に基づいています。これは、ジョブを実行しているユーザーのユーザー・プロファイル、ジョブ記述、ワークステーション装置記述、およびデフォルトのプリンター・システム値 (QPRTDEV) からの情報に基づいています。

プリンター出力が作成可能な状態になると、システムは印刷ファイルおよびジョブ属性を（この順序で）検査して、プリンター出力を処理する出力待ち行列、およびシステムが使用するプリンターを確認します。ジョブを投入する際、または拡張処理をう回するためにジョブを実行する際に、出力待ち行列 (OUTQ) およびプリンター (DEV) のパラメーターを変更できます。たとえば、ジョブの変更を開始してその効果がすぐに現れるようにする場合、印刷ファイル出力待ち行列を特定の待ち行列に設定し、プリンターをその印刷ファイルの特定のプリンターに設定できます。そのようにする際、プリンター出力は出力待ち行列および使用するプリンターを検出するために、ジョブ属性に入れられる必要はありません。特定の出力待ち行列が検出できない場合、プリンター待ち行列は QGPL/QPRINT に送られます。プリンター出力を作成する方法の詳細については、印刷装置プログラミング  の第 1 章を参照してください。

プリンター出力ファイルは、印刷および処理を待っている情報が入っているファイルです。プリンター出力ファイルは、他のプリンター出力と関連する、待ち行列上のプリンター出力の位置を定義する重要な属性を持っています。位置は、優先順位、状況、およびスケジュール属性によって定義されます。

出力待ち行列

出力待ち行列は、出力装置に書き出されるプリンター出力ファイルのリストを含むオブジェクトです。出力待ち行列は、プリンター出力が処理される順序、およびプリンター出力ファイルを変更するのに必要な権限を判別する重要な属性を持っています。

優先順位

処理を待っているプリンター出力は、優先順位に基づいて出力待ち行列に移動します（優先順位は 1～9 までで、1 が最も高い優先順位です）。

状況

現在のプリンター出力の状況。「出力」プロパティの「汎用 (General)」ページから、この状況を表示できます。

スケジュール

スケジュール属性は、出力データの物理的な印刷をファイルが開始すべき時を通知します。

即時

プリンター出力ファイルがクローズしていない場合でさえ、即時に印刷します。

ファイル終了 (デフォルト)

プリンター出力ファイルがクローズするとすぐに印刷を開始します。

ジョブ終了

ジョブが終了すると、印刷を開始します。

プリンター出力ファイルが印刷可能な状態になると、書き出しプログラム・ジョブ (出力待ち行列からプリンターに対してプリンター出力を処理するジョブ) は、プリンター出力ファイルからデータを取り出し、指定されたプリンターに送信します。

出力待ち行列の属性


出力待ち行列は、プリンター出力ファイル (スプールされたファイルとも呼ばれる) の処理方法と、出力待ち行列とその関連プリンター出力にアクションを実行する権限を持つユーザーを制御します。

ファイルの順序の属性は、プリンター出力が出力待ち行列から出て処理を受ける順序を判別します。出力待ち行列を構成するには 2 通りの方法があり、ジョブ番号または先入れ先出し法 (FIFO) のいずれかを使用します。

iSeries システムで印刷する情報のほとんどはプリンター出力として作成されるので、権限のないユーザーが重要な機密データにアクセスしないようにするには、セキュリティーが必要となります。出力待ち行列やプリンター出力ファイルにアクセスして変更するには、検査の権限、データ権限、オペレーター制御、スプール制御、所有者権限のいずれかが必要です。出力待ち行列またはプリンター出力に任意のアクションを実行するには、以下の権限のいずれかが必要です。

検査の権限。 待ち行列の所有者であるかデータ権限がなければなりません。


データ表示。 この権限が *YES に設定されている場合、出力の表示、別のシステムへの移動や送信、およびプリンター出力のコピーなどのアクションを実行できます。

オペレーター制御。 この属性が *YES に設定されていると、*JOBCTL 特殊権限を持つユーザーは、保留、解放、および出力待ち行列からのプリンター出力の削除などのアクションを実行する権限が与えられます。プリンター出力、出力待ち行列、および書き出しプログラムに対する他のアクションも可能です。詳細については、機密保護解説書  を参照してください。

スプール制御。 ユーザーは、プリンター出力のすべての操作を実行できます。ユーザーが出力待ち行列に任意のアクションを実行するには、出力待ち行列が配置されているライブラリーに対する *EXECUTE 権限が必要です。

所有者。 出力待ち行列を所有するユーザーは、プリンター出力を変更または削除できます。

注: 出力待ち行列に対するデフォルト権限は、*USE 共通権限です。データ表示権限は *NO に設定されています (だれもプリンター出力を表示できないことを意味します)。検査の権限は *OWNER です (したがって、出力待ち行列の所有者はプリンター出力を操作できます)。オペレーター制御は *YES に設定されています (*JOBCTL のユーザーはプリンター出力の保留、解放、および削除が可能です)。

出力待ち行列を処理するのに必要な権限の詳細については、機密保護解説書  の付録 D を参照してください。

ファイルの順序: ファイルの順序属性は、プリンター出力ファイル (スプール・ファイルとも呼ばれる) が配置され、出力待ち行列で処理される順序を決定します。出力待ち行列を構成するには、**ジョブ番号**および**先入れ先出し法 (FIFO)** の 2 つの方法があります。

ジョブ番号

プリンター出力ファイルの待ち行列項目は、そのプリンター出力ファイルを作成したジョブのジョブ番号を使って優先度順にソートされます。

先入れ先出し法

待ち行列に入る新規のプリンター出力ファイル (スプール・ファイルとも呼ばれる) は、同じ優先度の他のすべてのプリンター出力ファイルの後に配置されます。

注: 出力待ち行列のファイルの順序属性を変更できるのは、待ち行列上にプリンター出力ファイルがない場合だけです。

プリンター出力の状況

プリンター出力ファイル (スプール・ファイルとも呼ばれる) の状況は、そのファイルが出力待ち行列のどこに表示されるかを判別します。次の状況が、出力待ち行列の最下部から最上部までリストされます。

まだ作成中

プリンター出力ファイルは作成中です。

印刷され保管された

プリンター出力ファイル中のデータが印刷されましたが、後で使用するために保管されました。

保留

プリンター出力ファイルは保留中で、書き出しプログラム・ジョブにより処理できなくなっています。

印刷はまだスケジュールされていない

プリンター出力ファイルの作成は完了しましたが、印刷に適格ではありません。これは、プリンター出力ファイルのスケジュール属性が *JOBEND に設定されている場合にのみ見られます。要するに、書き出しプログラム・ジョブがプリンター出力ファイルへの処理を許可される前に、そのプリンター出力ファイルを所有するジョブが終了しなければならないということです。

ページ限界を超えた

書き出しプログラム・ジョブが印刷できるファイルの最大ページ数を超えました。この状況は、出力待ち行列が書き出しプログラム・ジョブに対して活動中である場合にのみ見られます。

印刷可

プリンター出力ファイルが、書き出しプログラム・ジョブによる処理を待っています。

次の状況は、出力待ち行列が書き出しプログラム・ジョブに対して活動中である (書き出しプログラム・ジョブにより処理中である) 場合に、出力待ち行列の最初に見られます。

プリンター用に変換中

プリンター出力ファイルが、プリンター用の変換処理 (印刷のための準備) を受けています。

印刷中

プリンター出力ファイルの内容が、プリンターに送信されています。

プリンターに送信

プリンター出力ファイルの内容が、印刷されています。オペレーティング・システムは、プリンター出力ファイルの印刷実行の確認を待っています。

送信中

プリンター出力ファイルは、システムから別のシステムに転送中です。

メッセージ待機中

書き出しプログラム・ジョブに、用紙切れまたは紙詰まりなどの問題が発生し、印刷を続行することができない可能性があります。この条件が発生した場合、オペレーターの介入が必要になることがあります。

印刷完了

プリンター出力ファイルは削除されました。プリンター出力ファイルは印刷された場合も、印刷されていない場合もあることに注意してください。

作業はどのように処理されるか

この情報を通して、作業とは何か、作業を開始する前に何をセットアップする必要があるか、作業はシステム内でどのように処理されるか、さらに作業の実行が終了するとどうなるかについて学びます。

- 作業とは何か
- 作業がシステムに入れられる前に何が起こるか
- 作業はどのようにシステムに入れられるか
- 作業はどのように処理されるか
- 作業はどのように終了するか

実行管理機能の概念の詳細については、『システムの構造』を参照してください。

作業とは何か

iSeries サーバー上では、ユーザーが開始したものであれ、システムが開始したものであれ、作業は常に実行されています。システム電源をオンにする、ファイルをオープンする、またはデータベースを照会すると、作業が実行されます。iSeries サーバー上で実行されるアクションにはすべて、実行して完了されるいくつかの種類があります。

システム上の各作業はジョブによって実行されます。ジョブは、ユーザーの呼び出しを待つアプリケーションのように単純な場合もあれば、1 時間ごとにシステム上のユーザー数をモニターする常時実行のシステム照会のように複雑な場合もあります。一部のジョブ (とりわけバッチ・ジョブと対話式ジョブ) にはジョブ記述が関係しており、このジョブ記述がジョブを実行する時間と場所を設定します。

ジョブは特定の機能を実行するプログラムから成っています。ジョブが実行する機能の量に限界はありません。ジョブには、作業の実行を完了しなければならないステップバイステップの命令が含まれています。ジョブを構成するプログラムは決まった順序で実行されます。たとえば、プログラム A はプログラム B が始まる前に実行する必要があります。

▶ スレッド ◀ は、ジョブがその処理を完了するのを支援します。アクティブ・ジョブには少なくとも 1 つのスレッドが含まれています。1 つのジョブに複数のスレッドが含まれている場合、そのジョブは一度に 2 つ以上のことを行うことができます。たとえば、あるスレッドがたくさんのデータの処理を待っている間、別のスレッドが呼び出されて計算を行うことも可能です。


iSeries サーバーのジョブとジョブ・タイプの詳細については、『ジョブ』を参照してください。

作業がシステムに入れられる前に何が起こるか

システム・ジョブを除くすべてのジョブは、サブシステムの中で実行されます。作業を活動サブシステムで開始するには、メモリー・プールと、少なくとも 1 つの実行処理項目のソースを確立する必要があります。作業ソースの一例としてジョブ待ち行列があげられます。iSeries サーバーには、ジョブ待ち行列、サブシステム、そしてメモリー・プールのデフォルト・セットが用意されており、システムを起動したらずぐに作業を開始できるようになっています。

サブシステムとメモリー・プールの構成を調整して、iSeries サーバーの機能およびパフォーマンスを最適化することができます。たとえば、ビジネスの成功にバッチ・ジョブが欠かせないようなケースでは、バッチ・ジョブにさらに多くのメモリーを割り振りたいと考えるかもしれません。あるいは、Qbatch サブシステムで一度に実行されるジョブの数を少なくして、バッチ・ジョブが最大限に資源を使って実行できるようにする必要があると判断するかもしれません。さらに、特殊な作業を実行するための特別なジョブ待ち行列、サブシステム、およびメモリー・プールを作成できます。たとえば、Nightreps と呼ばれるジョブ待ち行列を作成して、夜間にバッチ・レポートを Nightrep と呼ばれるサブシステムに送るようにし、これらのバッチ・ジョブの実行専用のメモリーを割り振るようにすることができます。

ジョブ待ち行列、サブシステム、およびメモリー・プールの詳細については、システムの構造を参照して


ください。実行管理機能のための IBM サポートの詳細については、実行管理の手引き  の付録 C 『IBM 提供オブジェクトの内容』および V5R2 の新機能を参照してください。

作業はどのようにシステムに入れられるか

実行処理項目は、ジョブをどのソースからサブシステムに入れて、実行可能にできるかを示します。iSeries 上のジョブのタイプごとに、異なるタイプの実行処理項目が使用されます。

ほとんどのバッチ・ジョブは、ジョブ待ち行列を使用してサブシステムに入れられます。ジョブ待ち行列項目のメカニズムによって、ジョブ待ち行列はサブシステムへの作業のソースとして定義されます。

実行処理項目は、サブシステム記述に保持されます。サブシステム記述に現在実行中の作業タイプの実行処理項目がない場合、そのジョブはサブシステムで実行できません。IBM 提供のサブシステムのサブシステム記述にはデフォルトの実行処理項目が用意されています。これらサブシステムとともに提供されているデフォルトの実行処理項目には、すでに特定のジョブを実行するために割り当てられているものがあるということに注意してください。たとえば、QCMN サブシステムのコミュニケーション実行処理項目の 1 つは、iSeries Access サーバーを実行するためにセットアップされています。

作業がシステムにどのように入れられるかの詳細については、実行管理の手引き  の第 4 章にある『実行処理項目』を参照してください。

作業はどのように処理されるか


iSeries サーバーが起動すると、サブシステム・モニター・ジョブが開始されます。サブシステム・モニター・ジョブは、サブシステム内のジョブを制御します。さらに、作業の開始と終了、サブシステム内の作業に関する資源の管理も行います。作業（またはジョブ）は実行処理項目によってサブシステムに入れられ、活動状態になって実行可能な状態になります。作業は、実行のためのメモリーがサブシステムに割り振られないと完了しません。メモリーはメモリー・プールによってサブシステムに割り振られます。

サブシステム記述はどのように作業の処理に役立つか

ジョブと同じように、サブシステムにもサブシステム記述と呼ばれる記述があります。サブシステム記述には、サブシステム内で一度に活動状態にできる作業に関する方法、場所、数などの重要な情報や、作業を実行するのに使用できる資源に関する情報が含まれています。

経路指定項目

経路指定項目は、ジョブをどのメモリー・プールで実行するか、そのジョブにどのプログラムを実行するか、さらにジョブの実行にどのクラス・オブジェクトを使用するかをサブシステムに通知するサブシステム

内に存在します。経路指定項目の詳細については、実行管理の手引き  の第 4 章を参照してください。


クラス・オブジェクト

クラス・オブジェクトは、実行優先順位、デフォルトの待ち時間、タイム・スライス、その他の属性を定義します。実行優先順位は、ジョブがいつ実行のためにプロセッサ時間を取得するかを決定するため、重要です。実行優先順位は 0 から 99 までで、0 が最高優先順位です。（iSeries サーバーを実行するのがシステム・ジョブであるため、優先順位 0 はシステム・ジョブだけに与えられます。）

ジョブがサブシステムに入れられると、サブシステムは経路指定項目にある比較値と経路指定データの突き合わせに追われます。経路指定データと経路指定項目内の比較値が一致すると、その経路指定項目がジョブに割り当てられます。一致しない場合、ジョブは終了します。

ジョブがサブシステム内で実行される場合に影響を及ぼす別の要素は、サブシステム内で一度に活動状態にすることができるジョブの数（サブシステム内の最大アクティブ・ジョブとも呼ばれます）です。サブシステム内でのアクティブ・ジョブの最大数に達すると、既存の活動状態にあるジョブの実行が完了するまで、追加のジョブはサブシステムに入れられません。ジョブの実行のためには、メモリーがサブシステムに割り振られなければなりません。メモリー・プール活動レベルは、メモリー・プール内でアクティブにできるスレッドの数を iSeries サーバーに通知します。アクティブ・ジョブには少なくとも 1 つのスレッドが含まれていることを覚えておいてください。メモリー・プール活動レベルに達すると、ジョブは別のスレッドが活動レベルの使用をやめるのを待つ必要があります。ジョブはサブシステム内で活動状態になりますが、実行はされません。

注: 最大アクティブ・ジョブをメモリー・プール活動レベルと混同しないようにしてください。

ジョブ、サブシステム、およびメモリー・プールの詳細については、実行管理の手引き  を参照してください。


作業はどのように終了するか

▶ 出力待ち行列は、ジョブ待ち行列と同様の働きをして、印刷される出力をスケジュールします。プリンター出力および出力待ち行列の両方が、情報の印刷に使用される属性を保持しています。

プリンター出力には、印刷を待機中の情報など、処理を待機中の出力データが保持されています。プリンター出力には、いつ印刷するかをスケジュールする際に使用される重要な情報も保持されています。プリンター出力属性には、プリンター出力が存在する 出力待ち行列、優先順位、状況、およびプリンター出力のスケジュールが含まれます。

出力待ち行列には、プリンター出力ファイルを処理する順序を判別するための独自の属性が含まれています。さらに、プリンター出力および出力待ち行列に変更を加えるために必要な権限も含まれています。

プリンター出力をプリンターに送信する準備が完了すると、それは書き出しプログラム・ジョブによって処理されます。書き出しプログラム・ジョブはプリンター出力からデータを取り出して、それを印刷できるように準備します。

出力待ち行列がどのように選択されるかの詳細については、印刷装置プログラミング  の『印刷活動の制御』を参照してください。

特定の出力待ち行列を作成するか、システムに支給されている出力待ち行列をそのまま使用することができます。詳細については、『出力待ち行列の作成』を参照してください。 <<

実行管理機能のトラブルシューティング

iSeries サーバー上でジョブが効率的に処理されていないようである場合、それはジョブがハングしているか、または単にパフォーマンスが悪いかのいずれかです。いずれの場合も、その問題のトラブルシューティングに役立つ診断および回復アクションがあります。詳細については、以下のトピックを参照してください。

- ジョブのハングアップ
- ジョブのパフォーマンスが悪い

ジョブのハングアップ

ジョブがハングアップする場合の理由としては、以下の事柄が考えられます。

ジョブはオブジェクトのロックを待っている

| | |
|------|---|
| 診断方法 | iSeries ナビゲーターで、ジョブの状況を参照します。『ジョブの状況の判別』を参照してください。ロックを待っているジョブの状況は、ロックの待機中 (<i>Waiting for lock</i>) です。 |
| 回復 | そのジョブに関するロック・オブジェクトのリストを参照して、ジョブがロックを待機しているオブジェクトを判別します。『詳細: アクティブ・ジョブ・アクション』を参照してください。そのオブジェクトに対してホルダー・ロック・アクションを使用して、すでにロックを保持しているジョブを判別します。そして、そのジョブがロックを保持している理由と、そのロックを解放するためには何を行えばよいかを判別します。V5R2 では、「プロパティ」ページの状況値に追加情報が含まれていることがあります。たとえば、「プロパティ」ページで、ロック待機中の状況にはロック要求に関連したオブジェクトが表示されます。 |

ジョブが保留中

| | |
|------|--|
| 診断方法 | iSeries ナビゲーターで、ジョブの状況を参照します。『ジョブの状況の判別』を参照してください。 |
| 回復 | ジョブをクリックして、「解放」を選択します。 |

ジョブがハングアップする場合の理由としては、以下の事柄が考えられます。

ジョブ待ち行列が保留中

診断方法
回復

iSeries ナビゲーターで、ジョブの状況を参照します。

1. 保留されていないジョブ待ち行列にジョブを移動します。『他のジョブ待ち行列へのジョブの移動』を参照してください。
2. ジョブ待ち行列を解放します。これを行うには、ジョブをクリックして、「解放」を選択します。

活動サブシステムによってジョブ待ち行列が割り振られていない

診断方法
回復

iSeries ナビゲーターで、ジョブの状況を参照します。

1. 活動サブシステムによって割り振られているジョブ待ち行列にジョブを移動します。『他のジョブ待ち行列へのジョブの移動』を参照してください。
2. このジョブ待ち行列に関するジョブ待ち行列項目が入っているサブシステムを開始します。『サブシステムの開始』を参照してください。
3. 「ジョブ待ち行列項目の追加 (Add Job Queue Entry (ADDJOBQE))」コマンドを使用して、このジョブ待ち行列に関するジョブ待ち行列項目を活動サブシステムに追加します。

サブシステムの最大数に達した

診断方法
回復

iSeries ナビゲーターで、サブシステムの「最大アクティブ・ジョブ」の値を参照します。そのためには、サブシステムを右クリックして、「プロパティ」を選択します。

1. ジョブを別のジョブ待ち行列に移動します。『他のジョブ待ち行列へのジョブの移動』を参照してください。
2. 最大値を増やします。そのために、サブシステム記述変更 (CHGSBSD) コマンドを使用します。

ジョブ待ち行列の最大値に達した

診断方法
回復

iSeries ナビゲーターで、ジョブの「最大アクティブ・ジョブ」の値を参照します。そのためには、ジョブ待ち行列を右クリックして、「プロパティ」を選択します。その後、「活動」タブを選択します。

1. ジョブを別のジョブ待ち行列に移動します。『他のジョブ待ち行列へのジョブの移動』を参照してください。
2. 最大値を増やします。そのために、ジョブ待ち行列項目変更 (CHGJOBQE) コマンドを使用します。

優先順位レベルの最大値に達した

診断方法

プロパティを調べることによって、そのジョブのジョブ待ち行列優先順位を判別します。iSeries ナビゲーターで、ジョブ待ち行列のジョブ優先順位値ごとの「最大アクティブ・ジョブ」を参照します。そのためには、ジョブ待ち行列を右クリックして、「プロパティ」を選択します。その後、「活動」タブを選択して、「拡張」ボタンをクリックします。

回復

1. ジョブを別のジョブ待ち行列に移動します。『他のジョブ待ち行列へのジョブの移動』を参照してください。
2. ジョブのジョブ待ち行列優先順位を変更します。『ジョブ待ち行列でのジョブの優先順位の変更』を参照してください。
3. 最大値を増やします。そのために、ジョブ待ち行列項目変更 (CHGJOBQE) コマンドを使用します。

ジョブのパフォーマンスが悪い

ジョブのパフォーマンスが悪い理由としては、以下の事柄が考えられます。

メモリーが足りない

診断方法

ジョブのプロパティを表示して、ジョブがどのメモリー・プールで実行されているかを判別します。その後、iSeries ナビゲーターでメモリー・プールのプロパティを表示します。『メモリー・プールの使用状況の検査』を参照してください。プールで障害が起きる比率が高い場合、プールのメモリーが十分ではないか、プールにあるジョブが多すぎてメモリーを競合していることを示しています。

回復

1. システム・チューナーを使用していない場合は、オンにします。システム値 QPFRADJ は、メモリー・プールと活動レベルを自動的に調整します。
2. 可能であれば、処理しているプール内のメモリーの量を増やすか、またはメモリー・プールの活動レベルを減らすかして、手でそのプールを調整します。さらにマシン・プールを検査して、システム上のすべてのジョブが、使用中のメモリー・サイズに影響されていないことを確認することもできます。

活動レベルが低すぎる

診断方法

ジョブのプロパティを表示して、ジョブの状況およびどのメモリー・プールでジョブが実行されているかを判別します。ジョブの状況が活動レベルを待機中 (*Waiting for activity level*) であった場合は、iSeries ナビゲーターでメモリー・プールのプロパティを表示します。『メモリー・プールの使用状況の検査』を参照してください。プールで不適格状態に推移する比率が高い場合、プールにあるジョブが多すぎてメモリーを競合していることを示しています。

回復

1. システム・チューナーを使用していない場合は、オンにします。システム値 QPFRADJ は、メモリー・プールと活動レベルを自動的に調整します。
2. メモリー・プールの活動レベルを増やして、手でプールを調整します。

CPU 資源が不足している

診断方法

iSeries ナビゲーターの「アクティブ・ジョブ」リストでジョブの「CPU %」欄を確認します。システムの使用状況が高い場合、ユーザーのジョブの処理を完了させるだけの十分な CPU 資源を得ることができない可能性があります。

回復

1. 可能であれば、システムの不要な作業を終了させるかまたは保留にします。
2. 少数のジョブが CPU を占有しているのであれば、それらのジョブの実行優先順位を変更します (実行優先順位の値が大きいのことは、そのジョブの実行優先順位が低いということです)。

メモリー・プール・ページング・オプション

診断方法

アプリケーションがディスク処理の多いものであり、CPU 稼働率が低く、メモリーが十分にある場合は、エキスパート・キャッシュを使用すると効果的です。

回復

エキスパート・キャッシュは、iSeries ナビゲーターで共用メモリー・プールの「ページング」オプションを「計算済み (Calculated)」に変更して、オンにすることができます。「ページング」オプションは、メモリー・プールの「プロパティ」ページの「構成」タブにあり、共用プール (私用プールではなく) でのみ使用可能です。

ジョブの実行優先順位が低い

診断方法

ジョブのプロパティを表示して、システム上の他のジョブと比較してジョブの実行優先順位を判別します。

回復

他のジョブと比べて実行優先順位が低く (数字が大きい)、かつ優先順位の高い (数字が小さい) ジョブが CPU 資源のほとんどを使用しているために使用できる CPU があまり残されていないという場合、ジョブの実行優先順位を高くする必要があるかもしれません。『ジョブ・プロパティ』を参照してください。さらに、システムの CPU 稼働率が高く、ジョブの実行優先順位が低い場合は、「Dynamically adjust job priorities within priority bands (優先順位帯域内のジョブ優先順位の動的調整) (QDYNPTYSCD) および「Dynamically adjust job priorities of interactive jobs (対話式ジョブのジョブ優先順位の動的調整) (QDYNPTYADJ) システム値を設定すると便利です。

パフォーマンスの詳細については、『パフォーマンス』を参照してください。システムのパフォーマンスを調整する方法の詳細については、『パフォーマンスの調整』を参照してください。



Printed in Japan