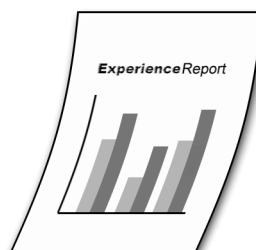


iSeries



サブシステム構成

# Experience Report





iSeries



## サブシステム構成

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

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

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

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

(URL は、変更になる場合があります)

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

原 典： iSeries  
Subsystem configuration  
Experience Report

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2005.8

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

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

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

© Copyright IBM Japan 2005

---

# 目次

サブシステム構成 . . . . .	1
複数のサブシステムを考慮する理由 . . . . .	1
サーバー・サブシステム . . . . .	2
複数サブシステムについての考慮事項 . . . . .	3
サーバー・サブシステム構成 . . . . .	4
サーバー・サブシステムのデフォルト . . . . .	10
ユーザー定義のサーバー・サブシステム . . . . .	13
サーバー・ジョブの管理 . . . . .	17
すべてのサーバー・ジョブ . . . . .	18
サーバー別のサーバー・ジョブ . . . . .	19
ユーザー別のサーバー・ジョブ . . . . .	21
終了したサーバー・ジョブ . . . . .	24
サーバー・ジョブ・テーブル . . . . .	26
サーバー・ジョブの詳細とプロパティ . . . . .	27
対話式サブシステム構成 . . . . .	42
シナリオの詳細: CL プログラムの例 . . . . .	48
対話式ユーザーおよび装置の管理 . . . . .	57
特定のサブシステムへのユーザーの経路指定 . . . . .	57
装置名の割り当て . . . . .	57
非アクティブの対話式セッションの管理 . . . . .	59
装置回復の管理 . . . . .	61
装置記述の IP アドレスの判別 . . . . .	61
参照およびリソース . . . . .	63
特記事項 . . . . .	65



---

## サブシステム構成

OS/400<sup>(R)</sup> に標準で用意されているデフォルトのサブシステム構成は、小規模のシステムでの稼働に適した基本的なサブシステム構成です。ただし、システムのユーザー数が増えるにつれ、システム上の処理をより適切に管理するために、処理を複数のサブシステムに分割することが求められます。

この経験報告には以下の情報が含まれています。

### 『複数のサブシステムを考慮する理由』

この情報では、サブシステム構成を行ってユーザー用に新たにサブシステムを追加することが必要となる、さまざまな理由について説明します。

### 2 ページの『サーバー・サブシステム』

この情報では、サーバー・ジョブに対してサブシステム構成を行う方法を説明します。

### 17 ページの『サーバー・ジョブの管理』

この情報では、サーバー・ジョブを管理する方法、およびサーバー・ジョブを使用しているユーザーを確認する方法を説明します。

### 42 ページの『対話式サブシステム構成』

この情報では、対話式ユーザーに対してサブシステム構成を行う方法を説明します。

### 57 ページの『対話式ユーザーおよび装置の管理』

この情報では、対話式ユーザー・セッションおよび装置を管理する方法を説明します。

---

## 複数のサブシステムを考慮する理由

OS/400<sup>(R)</sup> にはサブシステムの標準セットが付属しています。以下のサブシステムが、ユーザーの作業を支援するために必要です。

- QBASE または QCTL、QINTER、および QCMN
- QSYSWRK
- QSERVER
- QUSRWRK

対話式ユーザーは、通常は QBASE または QINTER で実行します。どちらを使用するかは、制御サブシステムのシステム値 (QCTLSBSD) によって決まります。この経験報告では、対話式サブシステムとして QINTER を使用することを前提としています。

サーバー・ジョブは、QSYSWRK、QUSRWRK、および QSERVER で実行します。通常、デーモン・ジョブ (処理要求を待機中の事前開始ジョブに送るサーバー・ジョブ) が QSYSWRK および QSERVER で実行されるのに対し、ユーザーに代わって処理を行う事前開始サーバー・ジョブは、通常は QUSRWRK で実行されます。ただし、事前開始ジョブのなかには QSERVER で実行されるものもあります。

この経験報告では、QINTER および QUSRWRK で実行するユーザーにフォーカスします。システム上のユーザー数が増えるにつれ、処理のセットに対して単一のサブシステムでは不十分になることがよくあります。

ユーザーを複数のサブシステムに分割することによって、以下のような数多くのメリットを享受できます。

- どの処理がどのサブシステムで実行されるかをより理論的に制御できるため、システム上の処理の管理の容易性が向上します。例えば、サーバー・ジョブの場合、すべてのデータベース・サーバー・ジョブを 1 つのサブシステムに分離し、リモート・コマンド・サーバー・ジョブを別のサブシステムに分離し、DDM サーバー・ジョブをさらに別のサブシステムに分離することをお勧めします。これにより、個々のサブシステムで行われる処理のタイプを把握することができます。
- 特定の時間帯にユーザーのセットがシステムにアクセスできないようにする機能を提供します。例えば、バックアップのために毎週金曜日の午後にシステムを制限状態にする必要がある場合は、一度に 1 つのサブシステムを終了することによって、ユーザーを段階的にオフラインにすることができます。
- スケーラビリティと可用性が向上します。単一のサブシステムに少人数のユーザーの処理を行わせることによって、サブシステムがビジー状態になりにくくなり、対応する処理要求への即応性が向上します。
- 複数のサブシステム間に処理を分散させることによって、エラー許容度が改善されます。対話式サブシステムの場合、これは特に重要です。例えば、ネットワーク障害が発生し、多数のセッションが装置回復状態に入った場合、その装置回復処理のために対話式サブシステムが極めてビジーな状態になることがあります。これは、サブシステムのジョブが、装置回復を管理するための中心のジョブであるためです。ネットワーク障害によって数百の装置が影響を受ける場合、それらの装置はサブシステムによって 1 つずつリカバリーされます。このような処理は、ユーザーによるシステムへのサインオン、またはシステムからのサインオフにマイナスの影響を与えることがあります。ユーザーを複数の対話式サブシステムに分割することによって、装置回復処理を、複数のサブシステムのジョブに管理させることができます。1 つのサブシステム内の装置の数を 250 から 300 に制限すると、対話式サブシステムの可用性とスケーラビリティは大幅に向上します。
- 対話式サブシステムの始動時間が改善されます。対話式サブシステムは、始動時に、サブシステム記述に含まれているワークステーション項目によって定義されている、すべての装置を割り振ろうとします。装置の数が増えるにつれ、サブシステムの起動時間は長くなる可能性があります。処理を複数のサブシステム間に分割することによって、サブシステムの始動時間を短くしておくことができます。
- パフォーマンス調整のための追加のオプションを提供します。実行優先順位、タイム・スライス、デフォルトの待機、最大 CPU などの一部のパフォーマンス属性は、クラス記述で指定されます。記憶域プールと最大ジョブは、サブシステム記述で指定されます。単一のサブシステム内で、適切なクラスを含む複数の経路指定項目を使用することによってパフォーマンスを調整したり、単に経路指定項目の数が少ないサブシステムを複数セットアップすることができます。どちらの方法でも、ビジネス上重要な作業を行うユーザーにはより多くのシステム・リソースを与え、重要度の低い作業を行うユーザーに提供するシステム・リソースは少なめにします。どちらのアプローチを使用するかは、使用する環境での要件によって異なります。この経験報告で後述する詳細情報では、比較的単純な構成のサブシステムを複数使用します。

---

## サーバー・サブシステム

OS/400<sup>(R)</sup> に標準で用意されているデフォルトのサブシステム構成は、小規模のシステムでの稼働に適した基本的なサブシステム構成です。ただし、サブシステム上のユーザーの数や処理量が増えるにつれ、システム上でより適切に処理を管理するために、処理を複数のサブシステムに分割する必要があります。

この経験報告には以下の情報が含まれています。

### 3 ページの『複数サブシステムについての考慮事項』

このセクションでは、複数サブシステムの使用が適している場合の例を示し、サブシステムを作成および構成する際に留意すべき事項について説明します。



#### 4 ページの『サーバー・サブシステム構成』

このセクションでは、サーバー・ジョブに対してサブシステム構成を行う方法を説明します。

#### 10 ページの『サーバー・サブシステムのデフォルト』

このセクションでは、サーバー・サブシステムのデフォルト値を変更する方法を説明します。

#### 13 ページの『ユーザー定義のサーバー・サブシステム』

このセクションでは、ユーザー定義のサーバー・サブシステムの作成方法を説明します。

## 複数サブシステムについての考慮事項

サーバー・ジョブとは、処理を待機中の iSeries<sup>(TM)</sup> サーバー上で、バックグラウンドで継続的に実行されるジョブを指します。作業は、ユーザーの代わりにするネットワーク機能、オペレーティング・システム機能、ネットワーク内の別のシステム、クラスター・サーバー・ジョブなどの汎用システム・サービスから送られてきます。通常、サーバー・ジョブは、システム出荷時に付属の基本サブシステム (QSYSWRK、QSERVER、または QUSRWRK) のいずれかで実行されます。一般にサーバー・ジョブは、HTTP、Lotus Notes<sup>(R)</sup>、および TCP/IP などの機能と関連しています。サーバー・ジョブについての詳細は、iSeries Information Center のトピック『実行管理機能: サーバー・ジョブ』を参照してください。

サーバー・ジョブに対してサーバー・サブシステムを構成することの利点には、いくつかの理由があります。これらの理由の一部は次のとおりです。

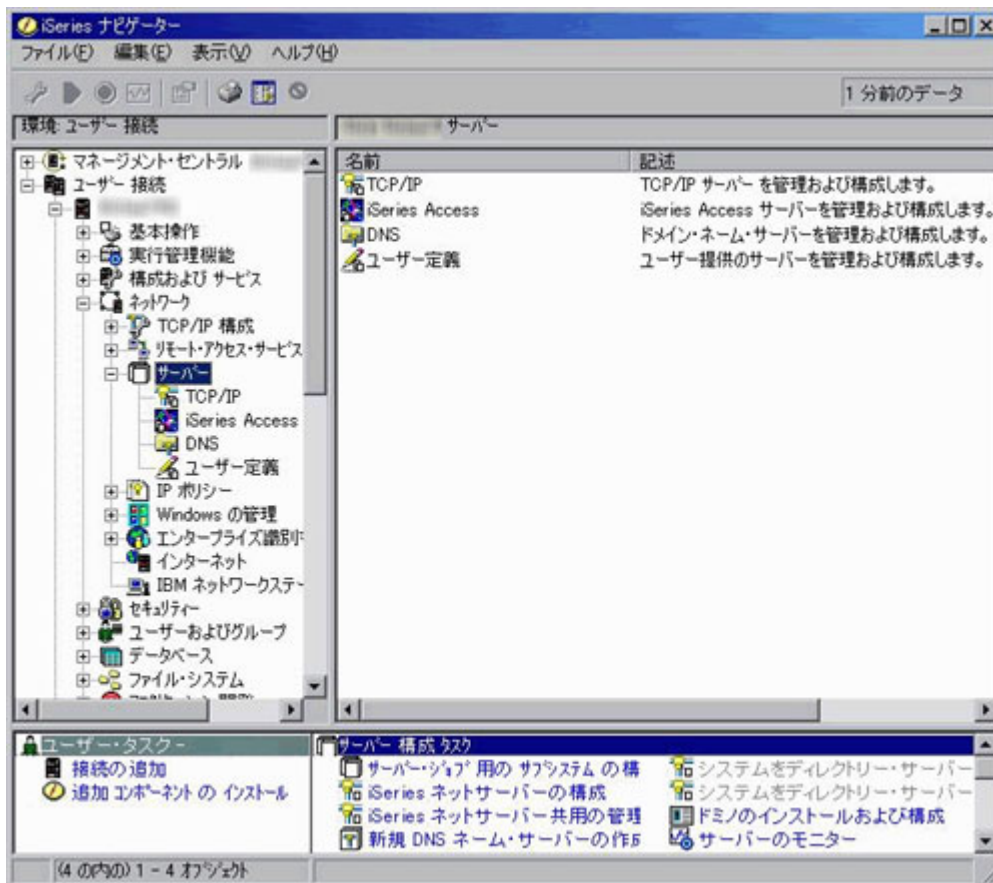
- サブシステム構成を行う理由の 1 つはパフォーマンス調整であり、これはルーティング項目情報を変更することで調整できます。変更可能なルーティング情報には、サーバー・ジョブが実行されるメモリー・プールやクラス情報などがあります。クラス情報は、実行優先順位、タイム・スライス、許可される最大スレッド数、およびデフォルト待機時間などのさまざまなパフォーマンス属性の構成に使用されます。サブシステム記述表示 (DSPSBSD) コマンドは、ルーティング項目情報にアクセスするためのメニューを表示します。パフォーマンス調整に関する追加情報は、iSeries Information Center を参照してください。
- サブシステム構成を行うもう 1 つの理由は、システム管理です。複数サブシステムを使用するようサーバーを構成すると、システムの処理が分散され、システム・リソースをより大幅に制御することができます。例えば、データベース・サーバーの処理専用のサブシステムを作成して、そのサブシステムに大量のシステム・リソースを与えることができます。もう 1 つの例としては、サブシステムに処理を送信できるユーザーを特定のサブネット内のユーザーに限定することによって、サブシステムへのアクセスを制御することができます。また、システム管理をシステム保守に役立てることもできます。その他の例としては、東海岸のすべての従業員用、中西部の従業員用、および西海岸の従業員用に、それぞれ 1 つのサブシステムを持つ、ということが可能になります。1 日の中で、東海岸の従業員が処理を終了したときに、西海岸と中西部の従業員に影響を与えずに、東海岸のサブシステムを終了することができます。それぞれの従業員をそのままにして各サブシステムを徐々に停止することによって、ユーザーに大きな影響を与えずに、使用されていないリソースの保守を実行できます。詳しくは、iSeries Information Center の『実行管理機能』トピックで、ジョブとスレッドの管理およびサブシステムの管理についての情報を参照してください。
- 処理を複数のサブシステムに分散すると、エラー許容度が改善されます。処理を複数のサブシステムに分散することによって、1 つのサブシステムでの障害の影響が抑制され、影響を受ける可能性のあるユーザーすべてに及ぶことがなくなります。サブシステムが同時に 5000 を越えるジョブを処理しているときに、エラー状態になるような何らかの不具合がそのサブシステムで発生した場合、5000 のジョブが影響を受け、失われる可能性があります。仮にこれら 5000 のジョブが複数のサブシステムに分散されていれば、それらの複数のサブシステムのうちの 1 つで障害が発生したとしても、影響を受けるジョブの数ははるかに少なく済みます。

## サーバー・サブシステム構成

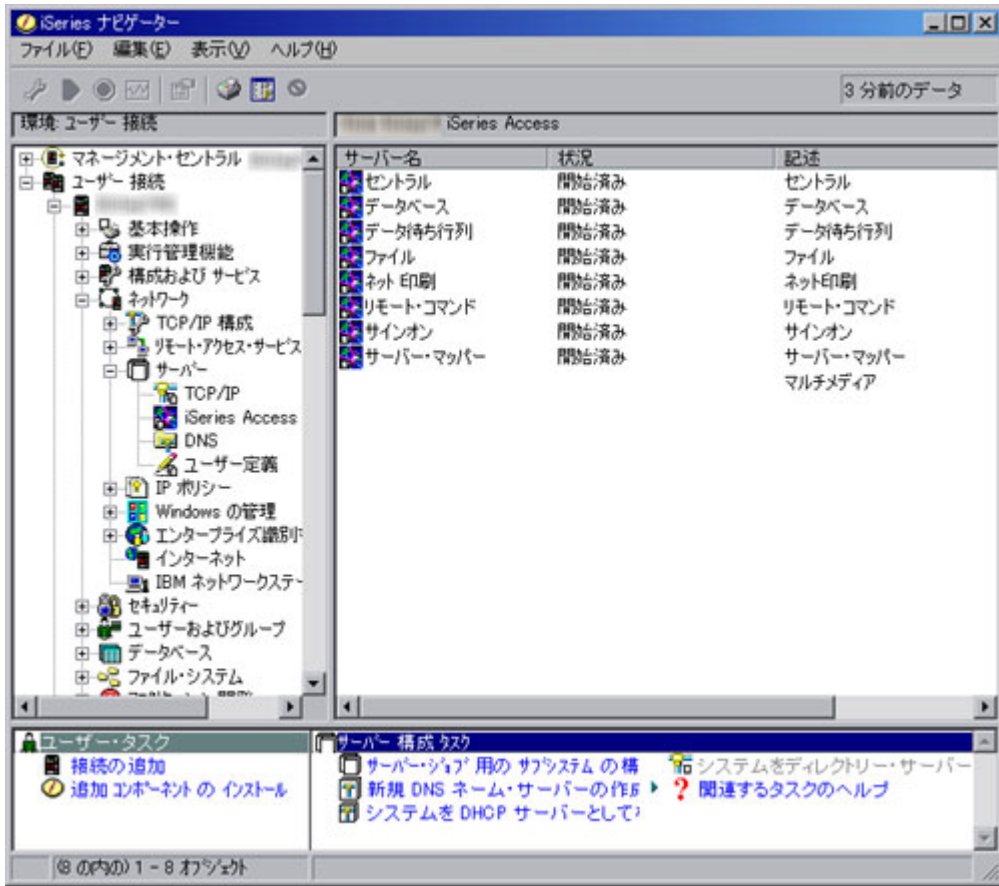
OS/400<sup>(R)</sup> が提供するサーバー・ジョブの一部は、デフォルト・サブシステム以外のサブシステムで実行するように構成できます。サーバーは、システムに標準で用意されているサブシステムか、13 ページの『ユーザー定義のサーバー・サブシステム』を使用することができます。また、サーバーは複数のサブシステムを使用することができます。適切なサブシステムへの経路指定は、IP アドレスによって制御されます。

サーバー・ジョブが使用するサブシステムを指定するには、iSeries<sup>(TM)</sup> ナビゲーターを使用する必要があります。以下のステップでは、iSeries ナビゲーターを使用してサーバー・サブシステムを構成する方法を示します。

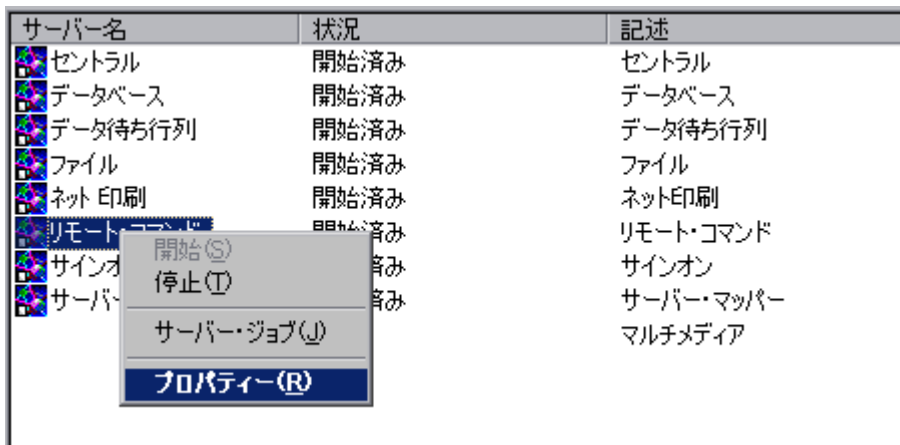
1. 「iSeries ナビゲーター」を開く。
2. 「ユーザー接続」を展開する。
3. 「iSeries サーバー」を展開する。
4. 「ネットワーク」を展開する。
5. 「サーバー」を展開する。



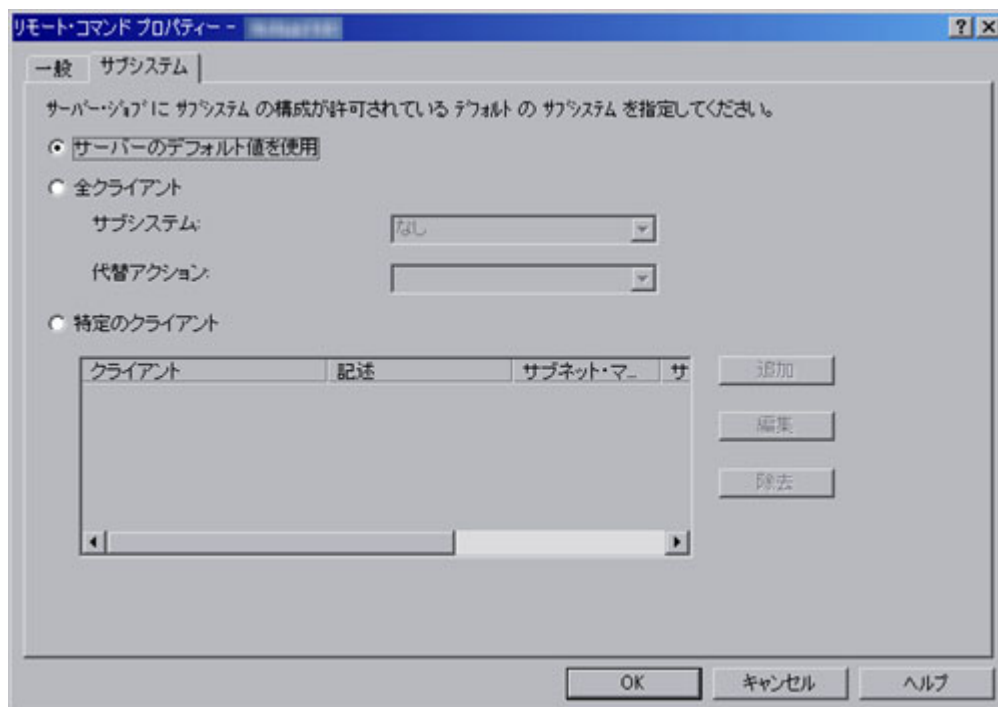
6. サブシステムを構成するサーバーのタイプを選択する。DDM および iSeries ネットサーバーは **TCP/IP** サーバーです。セントラル、データベース、データ待ち行列、ファイル、ネットワーク・プリント、リモート・コマンド、およびサインオンは **iSeries Access** サーバーです。以降のステップでは、iSeries Access からサブシステムを使用します。サブシステム構成ステップは、各サーバーで同一です。



7. 右側のペインで、サブシステムを構成するサーバーを右マウス・ボタン・クリックし、「プロパティ」を選択する。別のウィンドウがオープンし、当該サーバーの一般情報およびサブシステム情報のタブが表示されます。



8. 「サブシステム」タブを選択する。



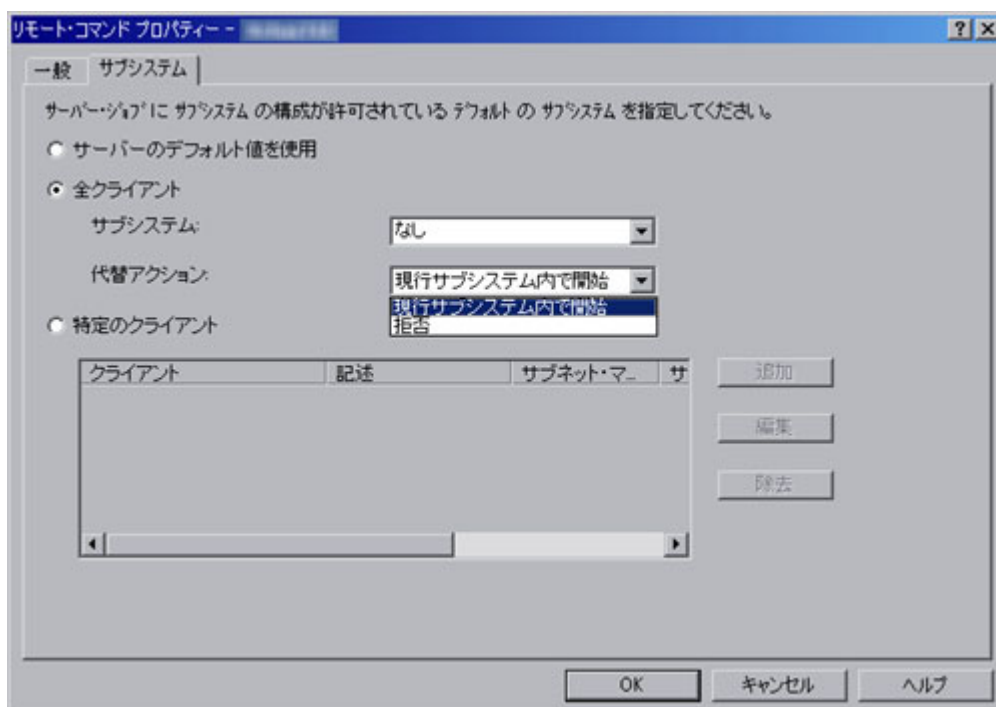
「サブシステム」タブで、このサーバーのジョブを実行するサブシステムを指定することができます。サブシステムの構成には、さまざまな方法があります。適切な構成方法を使用して、以下の指示に従って選択を行ってください。

- **サーバーのデフォルト値を使用**

処理中の特定のサーバーにサーバーのデフォルト値を使用する場合は、このオプションを選択します。「サーバーのデフォルト値」により、すべてのサーバー・ジョブのサブシステムに指定したい一般特性を、1カ所ですべて構成することができます。その後、個々のサーバーでこれらのデフォルトを使用させることも、特定のタイプのサーバーに対してより詳細な構成を行うこともできます。サーバー・デフォルトの使用および設定の方法については、10ページの『サーバー・サブシステムのデフォルト』を参照してください。

- **全クライアント**

このサーバーを使用するすべてのクライアントで同じサブシステムと代替処置を使用する場合は、このオプションを選択します。「全クライアント」を使用すると、さまざまなサーバーが実行する処理を、サーバーのタイプごとにサブシステムに簡単に分割できます。例えば、データベース・サーバーのすべてのクライアントに DATABASE サブシステムを使用させたり、リモート・コマンド・サーバーのすべてのクライアントに RMTCMD サブシステムを使用させたりすることができます。

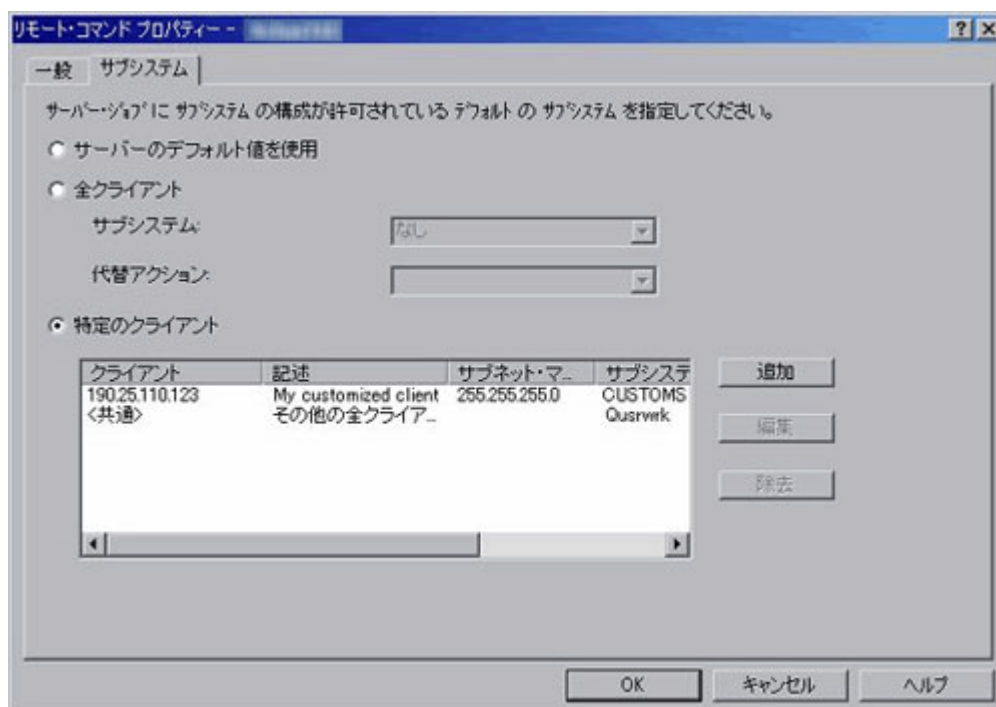


「サブシステム」リストは、このサーバーのジョブを実行するサブシステムを指定します。「なし」を選択すると、サーバー・ジョブは代替処置を実行します。使用するサブシステムがリストに含まれていない場合は、サブシステムの名前を 10 文字で入力できます。サブシステムがシステム提供のサブシステムではない場合は、13 ページの『ユーザー定義のサーバー・サブシステム』を参照して、目的のサブシステムが正しく設定され、稼働していることを確認してください。

「代替アクション」リストは、指定されたサブシステムでサーバー・ジョブが実行できない場合（例えば、サブシステムがアクティブではない場合や、特定のユーザーによる特定のサーバーの使用を禁止したい場合）に行う処置を指定します。代替処置がとられるもう 1 つの例は、特定のサブシステムで必要な事前開始ジョブがアクティブではない場合です。指定可能な値は、「拒否」、および「現行サブシステム内で開始」です。「拒否」を選択すると、指定されたサブシステムで実行できない場合には、要求は拒否されます。「現行サブシステム内で開始」を選択すると、指定されたサブシステムでジョブが実行できない場合、そのジョブは、サーバー・デーモンが実行中の同じサブシステムでの実行が試行されます。データベース・サーバーとファイル・サーバーの場合は、通常、こうすることによって、要求が QSERVER サブシステムで実行されます。その他のサーバーの場合は、要求が QSYSWRK に送信されます。サーバー・ジョブやそれらが実行されるサブシステム（サーバーやデーモン・ジョブなど）などの詳細については、iSeries Information Center のサーバー・テーブルを参照してください。

#### • 特定のクライアント

特定のクライアントに対して固有のサブシステム構成を設定する場合は、このオプションを選択します。特定のクライアント構成を追加すると、リストの終わりに <共通 (Public)> 項目が追加されます。<共通 (Public)> 項目は、追加した特定のクライアント項目に含まれないすべてのクライアントに適用されます。



特定のサーバ・サブシステム・クライアントを構成するために使用できるアクションには、さまざまなものがあります。

#### – 追加

リストにクライアントを追加するには「追加」をクリックします。すると「クライアントの追加」ダイアログが表示され、ここから特定のクライアントやクライアントのグループに、サブシステム構成を指定できるようになります。その結果、新しいウィンドウ、「クライアントの追加」ダイアログが表示されます。

「クライアントの追加」ダイアログから、特定のクライアントまたはクライアントのグループのサブシステム構成を指定できます。

「記述」テキスト・ボックスには、構成するクライアントのテキスト記述を指定します。

「クライアント」ラジオ・ボタン・オプションでは、個々の IP アドレスを使用するか、IP アドレスの範囲を使用するかを指定します。個々の IP アドレスは、単一のクライアントに使用する必要があります。IP アドレスの範囲は、クライアントのグループに使用します。選択したサーバーに対して、重複する IP アドレス範囲を指定することはできません。

「サブネット・マスク (Subnet mask)」は、この IP アドレスのサブネット・マスクを指定します。サブネット・マスクは、インターフェースが接続するネットワークの部分を定義する、固有の 32 ビットの整数です。サブネット・マスクは xxx.xxx.xxx.xxx の形式で表されます。ここで、各フィールドはサブネット・マスクの 1 バイト (8 ビット) の 10 進表記です。例えば、16 進表記が X'FFFFFF00' であるサブネット・マスクは、255.255.255.0 と表記されます。

「サブシステム」リストは、このサーバーのジョブを実行するサブシステムを指定します。「なし」を選択すると、サーバー・ジョブは代替処置を実行します。使用するサブシステムがリストに含まれていない場合は、サブシステムの名前を 10 文字で入力することができます。サブシステムがシステム提供のサブシステムではない場合は、13 ページの『ユーザー定義のサーバー・サブシステム』を参照して、目的のサブシステムが正しく設定され、稼働していることを確認してください。

「代替アクション」リストは、指定されたサブシステムでサーバー・ジョブが実行できない場合 (例えば、サブシステムがアクティブではない場合や、特定のユーザーによる特定のサーバーの使用を禁止したい場合) に行う処置を指定します。代替処置がとられるもう 1 つの例は、特定のサブシステムで必要な事前開始ジョブがアクティブではない場合です。指定可能な値は、「拒否」、および「現行サブシステム内で開始」です。「拒否」を選択すると、指定されたサブシステムで実行できない場合には、要求は拒否されます。「現行サブシステム内で開始」を選択すると、指定されたサブシステムでジョブが実行できない場合、そのジョブは、サーバー・デーモンが実行中の同じサブシステムでの

実行が試行されます。データベース・サーバーとファイル・サーバーの場合は通常、これによって、要求が QSERVER サブシステムで実行されます。その他のサーバーの場合は、要求が QSYSWRK に送信されます。サーバー・ジョブやそれらが実行されるサブシステム (サーバーやデーモン・ジョブなど) などの詳細については、iSeries Information Center のサーバー・テーブルを参照してください。

#### - 編集

「**編集**」をクリックして、リストに追加されたクライアントを編集します。編集する前に、リストからクライアントを選択する必要があります。「クライアントの編集 (Edit Client)」ダイアログの外観は「クライアントの追加」ダイアログと同じであり、構成オプションも同じです。

#### - 除去

「**除去**」をクリックして、リストに追加されているクライアントを削除します。除去する前に、リストからクライアントを選択する必要があります。リストの終わりにある <共通 (Public)> 項目を除去することはできません。

すべての選択を行ったら、指定したサーバー・サブシステムのデフォルトを受け入れて適用するには「**OK**」をクリックします。サーバー・サブシステムのデフォルト設定に対する変更を拒否して無視するには「**キャンセル**」をクリックします。

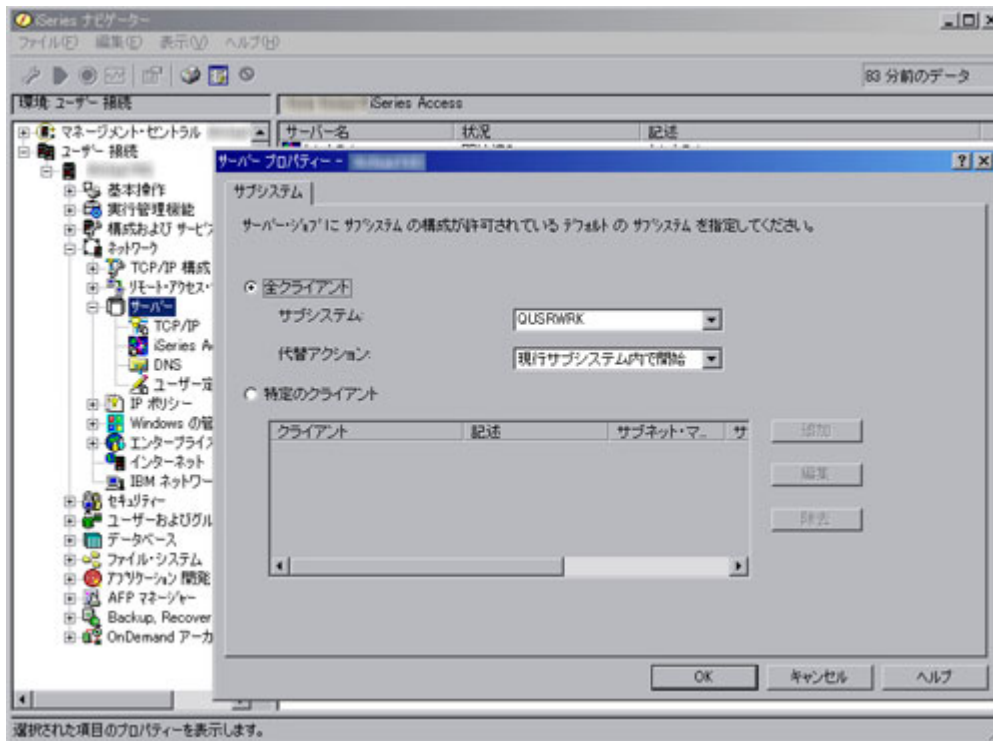
## サーバー・サブシステムのデフォルト

サーバーのデフォルトを設定すると、すべてのサーバー・ジョブ用のサブシステムの一般特性を 1 カ所でまとめて構成することができます。個々のサーバーは、その後でこれらのデフォルトを使用することができます。サーバーのデフォルトを指定するには、iSeries<sup>TM</sup> ナビゲーターを使用する必要があります。この機能にはグリーン・スクリーン・インターフェースはありません。以下のステップでは、iSeries ナビゲーターを使用してサーバー・サブシステムのデフォルトを設定する方法を示します。

1. 「**iSeries ナビゲーター**」を開く。
2. 「**ユーザー接続**」を展開する。
3. 「**iSeries サーバー**」を展開する。
4. 「**ネットワーク**」を展開する。
5. 「**サーバー**」を選択して右クリックし、オプション・メニューを表示する。
6. プロパティ・メニュー・オプションを選択して、「**プロパティ**」を開く。

サーバーのデフォルトは、サーバー・コンテナのサブシステム・プロパティ・ページで指定します。





サブシステム・プロパティ・ページで、すべてのサーバー・ジョブを実行させるデフォルトのサブシステムを指定することができます。サブシステムの構成には、さまざまな方法があります。適切な構成方法を使用して、以下の指示に従って選択を行ってください。

- 全クライアント

すべてのサーバーのすべてのクライアントが同じサブシステムと代替処置を使用することを表します。

「サブシステム」リストは、このサーバーのジョブを実行するサブシステムを指定します。「なし」を選択すると、サーバー・ジョブは代替処置を実行します。使用するサブシステムがリストに含まれていない場合は、サブシステムの名前を 10 文字で入力できます。サブシステムがシステム提供のサブシステムではない場合は、13 ページの『ユーザー定義のサーバー・サブシステム』を参照して、目的のサブシステムが正しく設定され、稼働していることを確認してください。

「代替アクション」リストは、指定されたサブシステムでサーバー・ジョブが実行できない場合 (例えば、サブシステムがアクティブでない場合や、特定のユーザーによる特定のサーバーの使用を禁止したい場合) に行う処置を指定します。代替処置がとられるもう 1 つの例は、特定のサブシステムで必要な事前開始ジョブがアクティブではない場合です。指定可能な値は、「拒否」、および「現行サブシステム内で開始」です。「拒否」を選択すると、指定されたサブシステムで実行できない場合には、要求は拒否されます。「現行サブシステム内で開始」を選択すると、指定されたサブシステムでジョブが実行できない場合、そのジョブは、サーバー・デーモンが実行中の同じサブシステムでの実行が試行されます。データベース・サーバーとファイル・サーバーの場合は通常、これによって、要求が QSERVER サブシステムで実行されます。その他のサーバーの場合は、要求が QSYSWRK に送信されます。サーバー・ジョブやそれらが実行されるサブシステム (サーバーやデーモン・ジョブなど) などの詳細については、iSeries Information Center のサーバー・テーブルを参照してください。

- 特定のクライアント

特定のクライアントに対して、固有のサブシステム構成を設定することを表します。特定のクライアント構成を追加すると、リストの終わりに <共通 (Public)> 項目が追加されます。<共通 (Public)> 項目は、追加した特定のクライアント項目に含まれないすべてのクライアントに適用されます。

特定のデフォルト・サーバー・サブシステム・クライアントを構成するために使用できるアクションには、さまざまなものがあります。

#### - 追加

リストにクライアントを追加するには「追加」をクリックします。すると「クライアントの追加」ダイアログが表示され、ここから特定のクライアントやクライアントのグループに、サブシステム構成を指定できるようになります。その結果、新しいウィンドウ、「クライアントの追加」ダイアログが表示されます。

The screenshot shows a dialog box titled "クライアントの追加" (Add Client). It contains the following fields and controls:

- クライアント情報** (Client Information) section:
  - 記述:** (Description) - An empty text input field.
  - クライアント:** (Client) - Two radio button options:
    - IP アドレス: (IP Address) - A text input field with four dots (". . .").
    - IP アドレス範囲: (IP Address Range) - Two text input fields, each with four dots.
  - サブネット・マスク:** (Subnet Mask) - A text input field containing "255.255.255.0".
- サブシステム:** (Subsystem) - A dropdown menu with "Qusrwrk" selected.
- 代替アクション:** (Alternative Action) - A dropdown menu with "現行サブシステム内で開始" (Start in current subsystem) selected.
- Buttons at the bottom: "OK", "キャンセル" (Cancel), and "ヘルプ" (Help).

「クライアントの追加」ダイアログから、特定のクライアントまたはクライアントのグループのサブシステム構成を指定できます。

「記述」テキスト・ボックスには、構成するクライアントのテキスト記述を指定します。

「クライアント」ラジオ・ボタン・オプションでは、個々の IP アドレスを使用するか、IP アドレスの範囲を使用するかを指定します。個々の IP アドレスは、単一のクライアントに使用する必要があります。IP アドレスの範囲は、クライアントのグループに使用します。選択したサーバーに対して、重複する IP アドレス範囲を指定することはできません。

「サブネット・マスク (Subnet mask)」は、この IP アドレスのサブネット・マスクを指定します。サブネット・マスクは、インターフェースが接続するネットワークの部分を定義する、固有の 32 ビットの整数です。サブネット・マスクは xxx.xxx.xxx.xxx の形式で表されます。ここで、各フィールドはサブネット・マスクの 1 バイト (8 ビット) の 10 進表記です。例えば、16 進表記が X'FFFFFF00' であるサブネット・マスクは、255.255.255.0 と表記されます。

「サブシステム」リストは、このサーバーのジョブを実行するサブシステムを指定します。「なし」を選択すると、サーバー・ジョブは代替処置を実行します。使用するサブシステムがリストに含まれていない場合は、サブシステムの名前を 10 文字で入力することができます。サブシステムがシステム提供のサブシステムではない場合は、『ユーザー定義のサーバー・サブシステム』を参照して、目的のサブシステムが正しく設定され、稼働していることを確認してください。

「代替アクション」リストは、指定されたサブシステムでサーバー・ジョブが実行できない場合 (例えば、サブシステムがアクティブでない場合や、特定のユーザーによる特定のサーバーの使用を禁止したい場合) に行う処置を指定します。代替処置がとられるもう 1 つの例は、特定のサブシステムで必要な事前開始ジョブがアクティブではない場合です。指定可能な値は、「拒否」、および「現行サブシステム内で開始」です。「拒否」を選択すると、指定されたサブシステムで実行できない場合には、要求は拒否されます。「現行サブシステム内で開始」を選択すると、指定されたサブシステムでジョブが実行できない場合、そのジョブは、サーバー・デーモンが実行中の同じサブシステムでの実行が試行されます。データベース・サーバーとファイル・サーバーの場合は通常、これによって、要求が QSERVER サブシステムで実行されます。その他のサーバーの場合は、要求が QSYSWRK に送信されます。サーバー・ジョブやそれらが実行されるサブシステム (サーバーやデーモン・ジョブなど) などの詳細については、iSeries Information Center のサーバー・テーブルを参照してください。

#### - 編集

「編集」をクリックして、リストに追加されたクライアントを編集します。編集する前に、リストからクライアントを選択する必要があります。「クライアントの編集 (Edit Client)」ダイアログの外観は「クライアントの追加」ダイアログと同じであり、構成オプションも同じです。

#### - 除去

「除去」をクリックして、リストに追加されているクライアントを削除します。除去する前に、リストからクライアントを選択する必要があります。リストの終わりにある <共通 (Public)> 項目を除去することはできません。

すべての選択を行ったら、指定したサーバー・サブシステムのデフォルトを受け入れて適用するには「OK」をクリックします。サーバー・サブシステムのデフォルト設定に対する変更を拒否して無視するには「キャンセル」をクリックします。

## ユーザー定義のサーバー・サブシステム

この記事では、QUSRWRK システム提供のサブシステムを基にして、新しいユーザー定義のサーバー・サブシステムを作成する方法について説明します。ユーザー定義サブシステムは、4 ページの『サーバー・サブシステム構成』に使用できます。

最初の計画ステップの 1 つとして、サブシステムに使用する命名規則を決定します。命名規則は、シンプルなもの、ユーザーをより反映したもの (INVENTORY、PGMR、ORDERENT) や、ユーザーの地理的な所在地を反映するもの (EAST、CENTRAL、WEST) などにすることができます。システム管理のために、分かりやすい命名規則を決定してください。

以下のステップは、サブシステムを対話式に作成するものとして説明しています。CL プログラムを使ってサブシステムを作成すると、リカバリーの際に構成を簡単に再作成できるようにすることができます。

1. サブシステム構成オブジェクトを保管するライブラリーを作成する。この例では、SBSLIB を使用します。

```
CRTLIB SBSLIB TEXT('Library to hold subsystem configuration objects')
```

2. クラスを作成する。クラスは、実行優先順位、タイム・スライス、デフォルト待機時間などの、使用するサブシステムの一部のパフォーマンス特性を定義します。

次の例では、優先順位 20、タイム・スライス 2000 ミリ秒、デフォルト待機時間 30 秒の MYSBS クラスが SBSLIB ライブラリー内に作成されます。

```
CRTCLS SBSLIB/MYSBS RUNPTY(20) TIMELICE(2000) DFTWAIT(30)
TEXT('Custom Subsystem Class')
```

クラスの作成についての詳細は、クラス作成 (CRTCLS) のコマンドの説明を参照してください。

**注:**

サブシステムのすべてに使用する単一のクラスを作成することも、また、各サブシステムごとにクラスを作成することもできます。どちらを選ぶかは、特定のサブシステムのパフォーマンス設定の一部をカスタマイズしたいかどうかによって変わります。IBM<sup>(R)</sup> 提供のサブシステムにはサブシステムごとに作成されたクラスが含まれており、そのクラスの名前はサブシステムの名前と同じです。サブシステムごとのクラスをサブシステムと同じ名前で作成しない場合は、経路指定項目追加 (ADDRTGE) コマンドでクラス名を指定する必要があります。これは、クラス (CLS) パラメーターのデフォルトが \*SBSD (クラス名がサブシステム記述と同じであることを示す) であるためです。

各サブシステムで別個のクラスを使用したほうがよい理由の例は、以下のとおりです。重要なユーザーに他のユーザーよりも高い実行優先順位を与えたい場合に、重要なユーザーを別個のサブシステムで実行させ、そのサブシステムに使用するクラスに高い実行優先順位を指定することによって、これを実現できます。実行優先順位では、数値の低いほうが優先順位が高いことに注意してください。

3. サブシステム記述を作成する。定義する必要があるサブシステムごとに、このステップを繰り返します。サブシステム記述の作成方法の詳細については、iSeries Information Center を参照してください。

以下の例では、QUSRWRK と同一の属性を持つサブシステム記述を作成します。ジョブ待ち行列、経路指定項目、および事前開始ジョブの情報は、後で手動で追加する必要があります。

```
CRTSBSDB SBSD(SBSLIB/MYSBS) POOLS((1 *BASE)) TEXT('カスタムのサーバー・サブシステム')
```

サブシステム記述が作成されたら、サブシステム記述表示 (DSPSBSDB) コマンドを使ってサブシステム記述情報を表示します。

サブシステム記述を最初から作成する代わりに、複製オブジェクト作成 (CRTDUPOBJ) コマンドを使って既存のサブシステム記述をコピーすることができます。以下の例では、QUSRWRK と同一の属性を持つサブシステム記述を作成します。これにより、ジョブ待ち行列、経路指定項目、および事前開始ジョブの情報も QUSRWRK からコピーされます。これらの値は、サブシステムの要件によっては変更が必要になることがあります。

```
CRTDUPOBJ OBJ(QUSRWRK) FROMLIB(QSYS) OBJTYPE(*SBSDB) TOLIB(SBSLIB)
NEWOBJ(MYSBS)
```

この報告書では、サブシステム記述を最初から作成したと仮定しています。

**注:**

使用される記憶域プールは、サブシステム記述で指定されます。ユーザーのセットをそれらのユーザー自身のプールに分離したい場合は、\*BASE プールを使用するのではなく、そのサブシステム

の具体的な記憶域サイズと活動レベルと共にプール ID を指定することによってこれを行うことができます。また、 共用記憶域プール変更 (CHGSHRPOOL) コマンドを使って共用プールを定義し、その共用プールをサブシステム記述で指定することもできます。

サブシステム記述では、サブシステムで実行できるジョブの数も定義します。以下のパラメーターは サブシステム記述作成 (CRTSBS) コマンドで使用できますが、複雑さが増し、偶発的にサブシステムへのユーザーのアクセスを妨げる可能性が高くなるため、これらのパラメーターは使用しないことをお勧めします。

**活動レベル (MAXACT):** 活動レベルは、一度にアクティブに実行できるスレッドの最大数を決定します。サブシステム内ではこの数よりも多くのスレッドをアクティブにすることができますが、活動レベルは、特定の時点でアクティブに実行できる数を決定します。

**最大ジョブ数 (MAXJOBS):** 最大ジョブ数パラメーターは、サブシステムで実行するジョブの数を決定します。この値を超えることはできません。既に最大数のジョブが実行しているときに追加のジョブを開始しようとすると、失敗します。

- サブシステム名と同じ名前を使ってサブシステムのジョブ待ち行列を作成し、ジョブ待ち行列項目をサブシステム記述に追加する。一般に、サブシステムにはジョブ待ち行列が必要です。ジョブ待ち行列を持つことで、必要ときにサーバー・ジョブをサブシステムにサブミットすることができます。ジョブ待ち行列にあるサーバー・ジョブは、その後で処理されます。また、ジョブ転送 (TFRJOB) コマンドを使ってジョブをカスタム・サブシステムに転送する場合も、ジョブ待ち行列が必要になります。

```
CRTJOBQ JOBQ(SBSLIB/MYSBS)
```

```
ADDJOBQE SBS(SBSLIB/MYSBS) JOBQ(SBSLIB/MYSBS) MAXACT(*NOMAX)
```

各コマンドについての詳細は、ジョブ待ち行列作成 (CRTJOBQ) コマンドの説明とジョブ待ち行列項目追加 (ADDJOBQE) コマンドの説明を参照してください。 iSeries Information Center には、ジョブ待ち行列の動作に関する情報も記載されています。

- 経路指定項目をサブシステムに追加する。

以下の例は、特定の経路指定項目と一致しないすべての要求を受け取る、汎用項目となるよう設計された経路指定項目を追加します。ユーザー定義サブシステムに追加される経路指定項目は、この例とは異なることがあります。経路指定項目は、サブシステムの特定の要件に応じて変わります。ユーザー定義サブシステム上に経路指定項目を設定するための基本は、それらを QUSRWRK で使用される経路指定項目と同一にすることです。

```
ADDRTGE SBS(SBSLIB/MYSBS) SEQNBR(9999) CMPVAL(*ANY) PGM(QSYS/QCMD)
CLS(SBSLIB/MYSBS)
```

QUSRWRK 用にシステム上に標準で用意されている経路指定項目を見ると、複数の追加の経路指定項目があることが分かります。これらの機能が必要な場合は、カスタマイズしたサブシステム記述にもこれらの経路指定項目を追加します。経路指定項目を、QUSRWRK から新しいユーザー定義サブシステム内にコピーすることをお勧めします。

詳しくは、経路指定項目追加 (ADDRTGE) コマンドの説明を参照してください。

#### 注:

**最大活動レベル (MAXACT):** 最大活動レベルは、この経路指定項目からアクティブにすることができる経路指定ステップの最大数を指します。これには、デフォルト値の \*NOMAX を使用することを強くお勧めします。

ADDRTGE コマンドは、使用するプール ID を指定します。このコマンドのデフォルトは 1 です。専用プールを含むサブシステム記述をセットアップした場合は、経路指定項目に適切なプール ID を指定してください。

クラスの名前がサブシステム記述の名前と異なる場合、 ADDRTGE コマンドでクラスを指定する必要があります。

経路指定項目は非常に複雑になることがあります。使用可能なオプションを調べる必要がある場合は、「AS/400e シリーズ 実行管理の手引き」を参照してください。

6. 事前開始ジョブ項目をサブシステムに追加する。事前開始ジョブは、開始済みで、処理のディスパッチを待っているジョブです。サブシステムへサブミットされる要求の処理に使用できるだけの数の事前開始ジョブを、サブシステムに設定しておくると特に便利です。

以下の例は、事前開始ジョブ項目を追加します。ユーザー定義サブシステムに追加される事前開始ジョブ項目は、この例とは異なることがあります。事前開始ジョブ項目は、サブシステムの特定の要件に応じて変わります。ユーザー定義サブシステム上の事前開始ジョブ項目を、QUSRWRK で使用される事前開始ジョブ項目と同じにすることをお勧めします。

```
ADDPJE SBS(D(SBSLIB/MYSBS) PGM(QSYS/QZSOSIGN) INLJOBS(50) THRESHOLD(4)
JOB(QZSOSIGN) JOB(D(QSYS/QZBSJOB) CLS(QGPL/QCASERVR) STRJOBS(*YES)
```

このコマンドではデフォルトは使用しないでください。ジョブの初期数、しきい値、ジョブ・パラメータの追加の数などのバランス、およびその他の重要な事前開始ジョブの設定についての詳細は、事前開始ジョブ項目の調整に関する情報を参照してください。事前開始ジョブ項目をサブシステム記述に追加する方法についての詳細は、事前開始ジョブ項目追加 (ADDPJE) コマンドの説明を参照してください。

#### 注:

作業が到着する前にジョブを初期化することによって、初期化の実行によるオーバーヘッドと時間が回避され、処理のスループットをさらに高めることができます。必要とされる処理量、および送信されてくることが予想される要求の量に対処できるよう、事前開始ジョブ項目を適切に構成することが非常に重要です。構成が適切でない場合、ジョブでは代替処置がとられることになります。また、事前開始ジョブの現在のプールが使い果たされた場合には、追加の事前開始ジョブが作成されるときにパフォーマンスが低下します。サブシステムの代替処置についての詳細は、4 ページの『サーバー・サブシステム構成』を参照してください。

以下のパラメータは、最初に行う事前開始ジョブの数、および、最初の事前開始ジョブの現行プールが使用中の場合に実行する内容を決定するために使用されます。

- ジョブの初期数 (INLJOBS): サブシステム記述 (SBSD) パラメータに指定されたサブシステムの開始時に開始される、事前開始ジョブの初期数を指定します。このパラメータの値は、最大ジョブ数 (MAXJOBS) パラメータの値以下でなければなりません。このパラメータの値は、しきい値 (THRESHOLD) パラメータの値以上でなければなりません。現行ワークロードは、ジョブの初期数と同じではありません。経験上、ジョブの初期数を、求められるワークロードにしきい値を足した数に設定する、という方法が適しています。
- しきい値 (THRESHOLD): 追加の事前開始ジョブがいつ開始されるかを指定します。使用可能なジョブのプール (サービス要求に使用できるジョブ) がこの値より低くなると、追加のジョブ (追加ジョブ数 (ADLJOBS) パラメータに指定されている) が開始され、使用可能なプールに追加されます。このパラメータの値は、ジョブの初期数 (INLJOBS) パラメータに指定され

た値以下でなければなりません。しきい値は、システムのワークロードと、使用可能な事前開始ジョブがすべて使用不可になった後で投入される新しい処理の量を反映した値とする必要があります。

- 追加ジョブ数 (ADLJOBS): 事前開始ジョブの数がしきい値 (THRESHOLD) パラメーターで指定された値より低くなったときに開始される、追加の事前開始ジョブの数を指定します。このパラメーターに指定される値は、最大ジョブ数 (MAXJOBS) に指定される値より低い値でなければなりません。

事前開始ジョブの詳細については、iSeries Information Center の『実行管理機能』セクションで、事前開始ジョブ、事前開始ジョブの使用、および事前開始ジョブの使用についての情報を参照してください。

サブシステム記述の作成が完了したら、サブシステム記述表示 (DSPSBSD) コマンドを使ってサブシステムのさまざまな属性を表示し、正しく設定されているかどうかを検査します。

サブシステム開始 (STRSBS) コマンドを使ってサブシステムを開始します。処理がサブシステムによって処理されていることを確認するには、ユーザー定義サブシステムを使用するように 4 ページの『サーバー・サブシステム構成』を行い、サブシステムで適切なクライアントが実行中であることを確認します。サブシステム・ジョブ処理 (WRKSBSJOB) コマンドは、ジョブがユーザー定義サブシステムで稼働していることを確認する際に役立ちます。

ユーザー定義サブシステムはすべて、ユーザー定義サブシステムを使用するサーバーが開始される前に開始する必要があります。サーバーに対してユーザー定義サブシステムを確実に使用可能な状態にするためには、TCP/IP が開始する前にサブシステムを開始することが最良の方法です。これを行うには、2 つの方法があります。1 つ目の方法は、IPL開始プログラムを変更することです。これには、IPL 時にサブシステムを自動的に開始するという利点があります。これにより、制限状態からシステムを再始動または始動するたびに、手動で始動する必要がなくなります。サブシステム開始を制御する 2 つ目の方法は、STRTCP IPL 属性を \*NO に変更してから、システム開始プログラムで、サブシステム、TCP/IP、およびサーバー・ジョブの開始を適切な順序で管理することです。IPL 自動始動などの詳細については、TCP/IP 開始 (STRTCP) コマンドの説明を参照してください。サーバーの始動方法についての詳細は、iSeries Information Center に記載されています。

---

## サーバー・ジョブの管理

サーバー・サブシステムをカスタマイズできるようになると、サブシステムで実行するサーバー・ジョブをより適切に管理したいという要求が生じます。iSeries<sup>TM</sup> ナビゲーターには、サーバー・ジョブを検索および管理するための、多数のさまざまなインターフェースが備えられています。サーバー・ジョブを見つけたら、さまざまな機能を使用してそれらのジョブを制御および管理できます。

この経験報告には以下の情報が含まれています。

### 18 ページの『すべてのサーバー・ジョブ』

このセクションでは、iSeries ナビゲーターを使用してすべてのサーバー・ジョブを検索する方法を説明します。

### 19 ページの『サーバー別のサーバー・ジョブ』

このセクションでは、iSeries ナビゲーターを使用して、特定のサーバーのサーバー・ジョブを検索する方法を説明します。

## 21 ページの『ユーザー別のサーバー・ジョブ』

このセクションでは、iSeries ナビゲーターを使用し、ユーザーを基にしてサーバー・ジョブを検索する方法を説明します。

## 24 ページの『終了したサーバー・ジョブ』

このセクションでは、iSeries ナビゲーターを使用して、既に終了したサーバー・ジョブを検索する方法を説明します。

## 26 ページの『サーバー・ジョブ・テーブル』

このセクションでは、QSYSWRK で実行するサーバー・ジョブのリストを示します。

## 27 ページの『サーバー・ジョブの詳細とプロパティ』

このセクションでは、iSeries ナビゲーターを使用して管理することができる、サーバー・ジョブの詳細とプロパティを説明します。

## すべてのサーバー・ジョブ

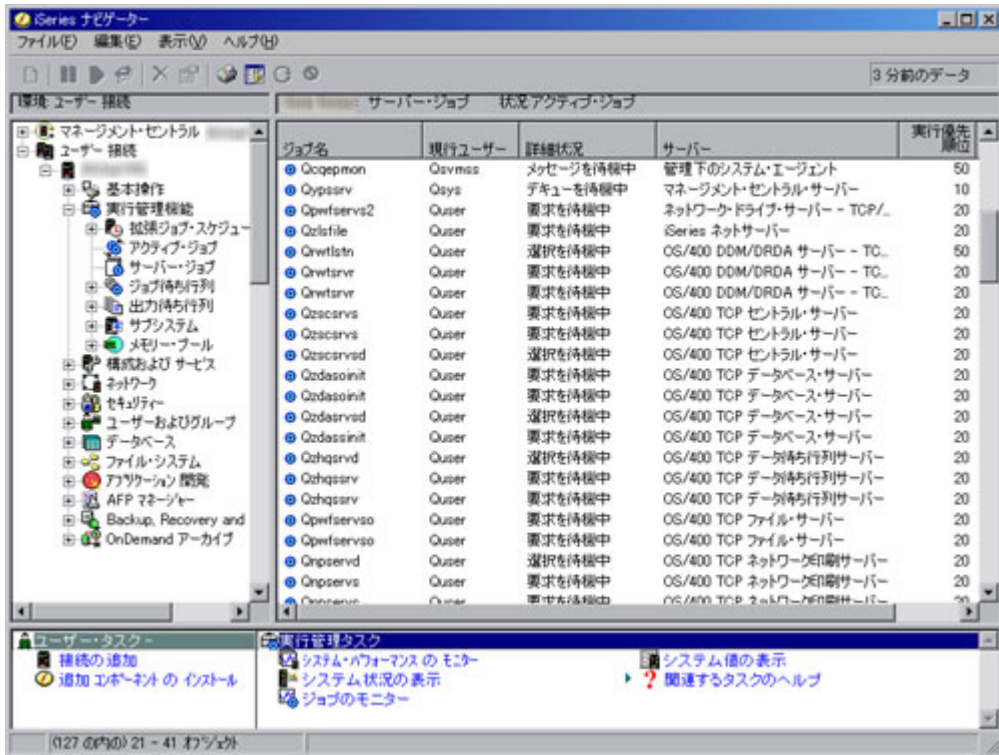
iSeries<sup>(TM)</sup> ナビゲーターを使用すると、出力待ち行列に含まれていないすべてのサーバー・ジョブを確認することができます。これは、所属するサーバーに関係なくすべてのサーバー・ジョブを確認するときに便利です。これは、所有しているサーバーが不明な場合に、特定のサーバー・ジョブを検索する際に役立ちます。

サーバー・ジョブ用にはグリーン・スクリーン・インターフェースもありますが、以下の情報では、iSeries ナビゲーターを使用します。サーバー・ジョブとグリーン・スクリーン・インターフェースについての詳細は、iSeries Information Center の『iSeries サーバー上のサーバー・ジョブの識別』の情報を参照してください。

以下のステップでは、iSeries ナビゲーターを使用してすべてのサーバー・ジョブを表示する方法を示します。

1. 「**iSeries ナビゲーター**」を開く。
2. 「**ユーザー接続**」を展開する。
3. 「**iSeries サーバー**」を展開する。
4. 「**実行管理機能**」を展開する。
5. 「**サーバー・ジョブ**」を選択する。右側のペインに、サーバー・ジョブがすべてリスト表示されます。





6. 管理するサーバー・ジョブを検索する。
7. ジョブ名を右マウス・ボタンでクリックする。
8. メニュー・オプションを選択する。

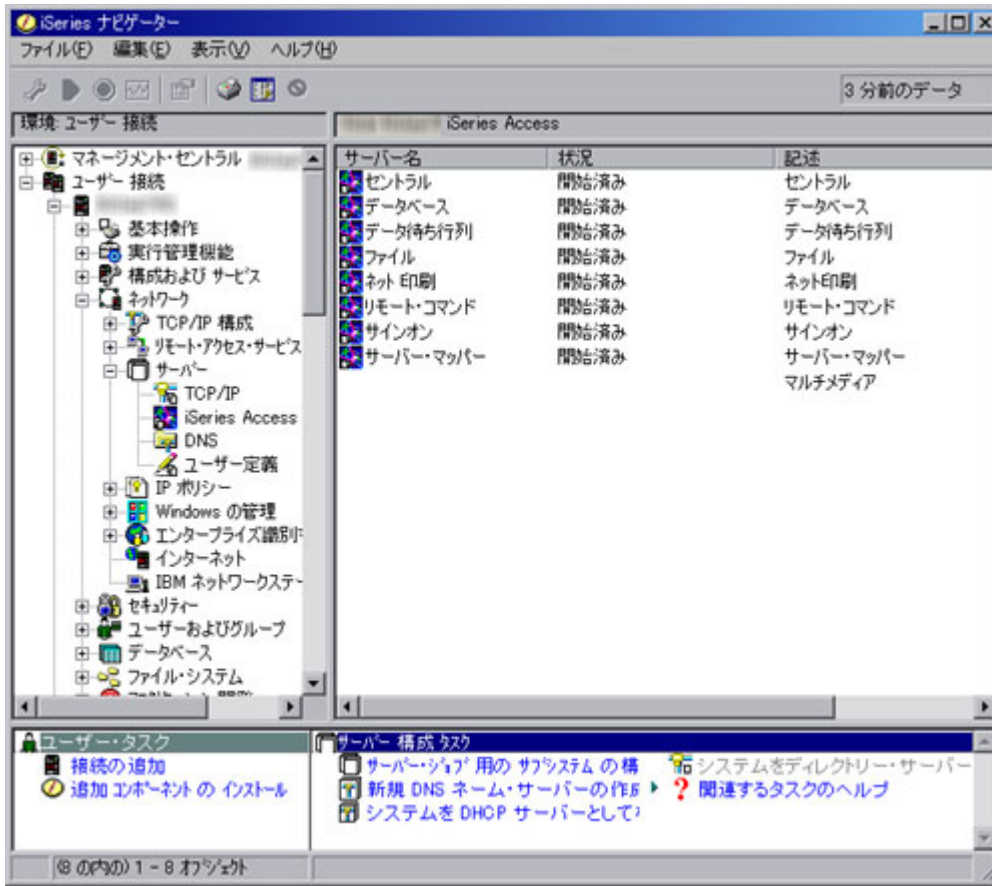
サーバー・ジョブの管理に使用できる各メニュー・オプションの詳細については、27 ページの『サーバー・ジョブの詳細とプロパティ』を参照してください。

## サーバー別のサーバー・ジョブ

以下のステップでは、iSeries<sup>TM</sup> ナビゲーターを使用して、特定のサーバーのすべてのサーバー・ジョブを表示する方法を示します。

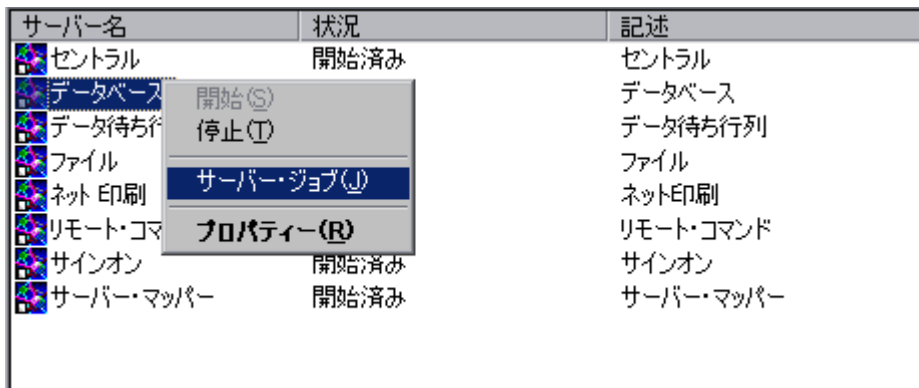
1. 「iSeries ナビゲーター」を開く。
2. 「ユーザー接続」を展開する。
3. 「iSeries サーバー」を展開する。
4. 「ネットワーク」を展開する。
5. 「サーバー」を展開する。
6. 「TCP/IP」または「iSeries Access」を選択する。右側ペインに該当するサーバーが表示されます。

選択するオプションは、サーバー・ジョブの管理を行いたいサーバーによって決定されます。TCP/IP サーバーには、DDM サーバーおよび iSeries NetServer サーバーなどがあります。iSeries Access サーバーには、セントラル・サーバー、データベース・サーバー、データ待ち行列サーバー、ファイル・サーバー、ネットワーク・プリント・サーバー、リモート・コマンド・サーバー、およびサインオン・サーバーなどがあります。



以降のステップでは、iSeries Access サーバーを使用します。

- 「データベース」を右マウス・ボタンでクリックしてから、「サーバー・ジョブ」を選択する。



- サーバー・ジョブの管理に使用できるもう 1 つのウィンドウがオープンする。

The screenshot shows a window titled 'データベース 個のサーバー・ジョブ' with a menu bar (File, Edit, View, Help) and a toolbar. Below the toolbar is a table with the following data:

ジョブ名	現行ユーザー	詳細状況	サーバー	実行優先順位	スレッド・カウント
Qzdsasinit	Quser	要求を待機中	OS/400 TCP データベース・サーバー	20	1
Qzdassinit	Quser	要求を待機中	OS/400 TCP データベース・サーバー	20	1
Qzdsarvsd	Quser	選択を待機中	OS/400 TCP データベース・サーバー	20	1
Qzdsasinit	Quser	要求を待機中	OS/400 TCP データベース・サーバー	20	1

At the bottom of the window, it says '(4 の内の) 1 - 4 オブジェクト'.

このウィンドウから、iSeries ナビゲーターのメイン・ウィンドウの右ペインと同じように、サーバー・ジョブを管理し、情報をカスタマイズできます。

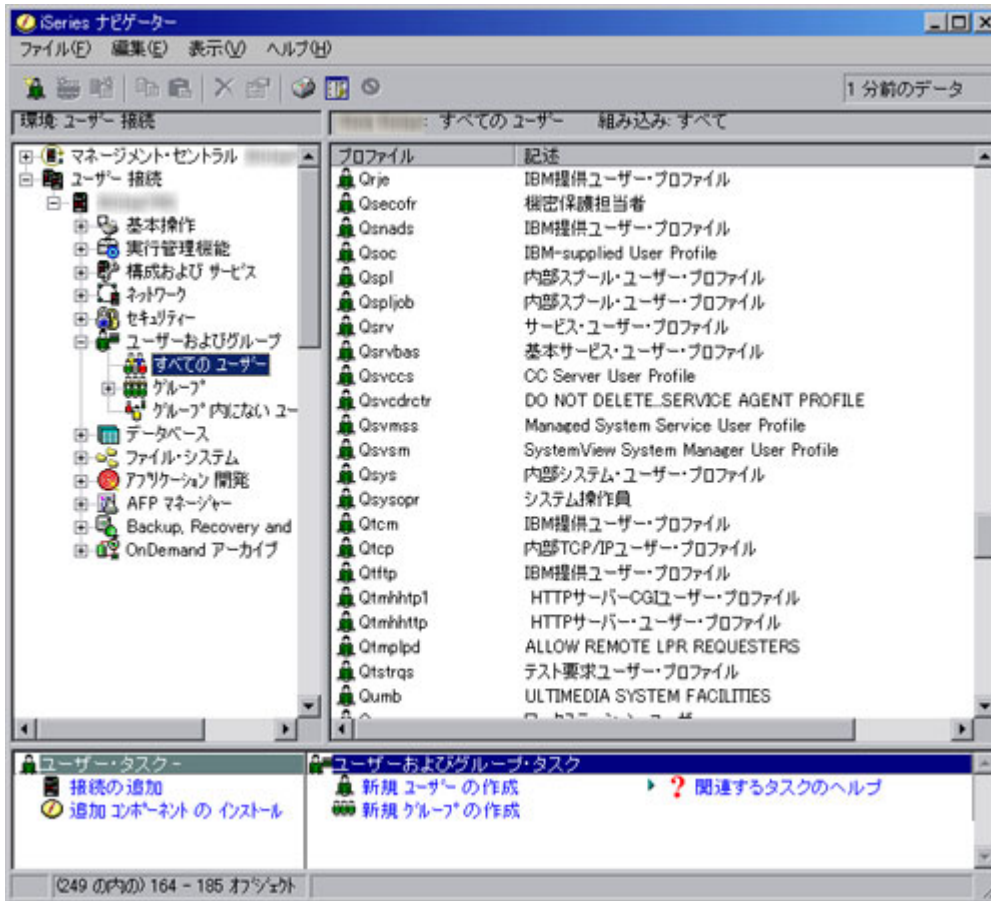
9. 管理するサーバー・ジョブを検索する。
10. ジョブ名を右マウス・ボタンでクリックする。
11. メニュー・オプションを選択する。

サーバー・ジョブの管理に使用できる各メニュー・オプションの詳細については、27 ページの『サーバー・ジョブの詳細とプロパティ』を参照してください。

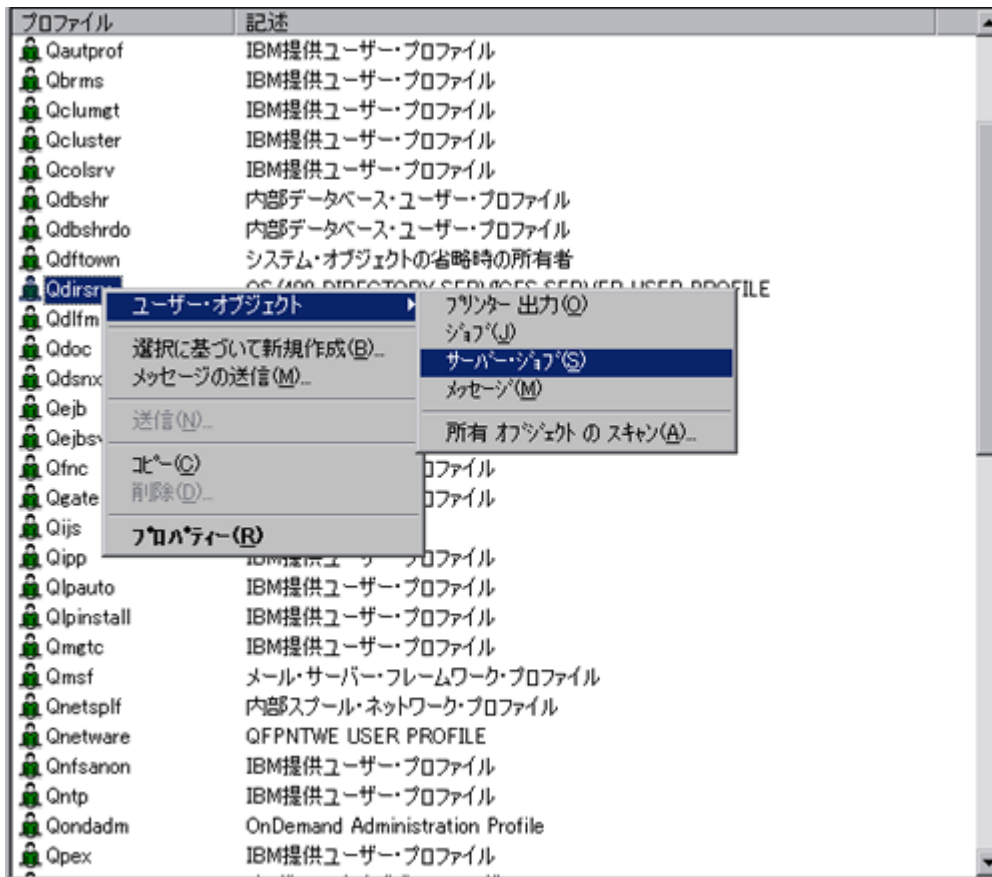
## ユーザー別のサーバー・ジョブ

以下のステップでは、iSeries<sup>TM</sup> ナビゲーターを使用して、特定のユーザーのすべてのサーバー・ジョブを表示する方法を示します。特定のユーザー・プロファイルについて検索されたサーバー・ジョブは、その特定のユーザーのプロファイルの下で実際に実行されているサーバー・ジョブ、ということになります。これは、そのサーバー・ジョブが開始されたユーザー・プロファイルと同じユーザー・プロファイルであるとは限りません。そのユーザーが実行しているサーバー・ジョブの確認のために選択するユーザー・プロファイルを決定する際には、この点に注意してください。

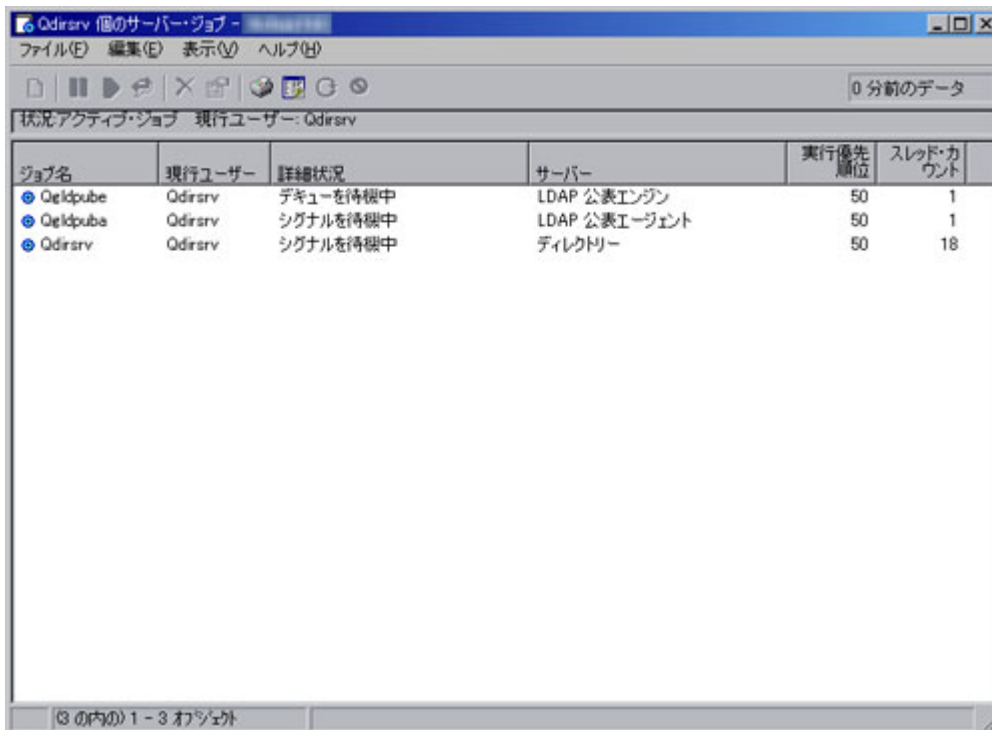
1. 「iSeries ナビゲーター」を開く。
2. 「ユーザー接続」を展開する。
3. 「iSeries サーバー」を展開する。
4. 「ユーザーおよびグループ」を展開する。
5. 「すべてのユーザー」を選択する。



6. サーバー・ジョブを確認したいユーザーの「プロファイル」を右クリックし、「ユーザー・オブジェクト」を選択してから「サーバー・ジョブ」を選択する。



7. このユーザーのサーバー・ジョブを管理できる、もう 1 つのウィンドウがオープンする。



このウィンドウから、iSeries ナビゲーターのメイン・ウィンドウの右ペインと同じように、サーバー・ジョブを管理し、情報をカスタマイズできます。

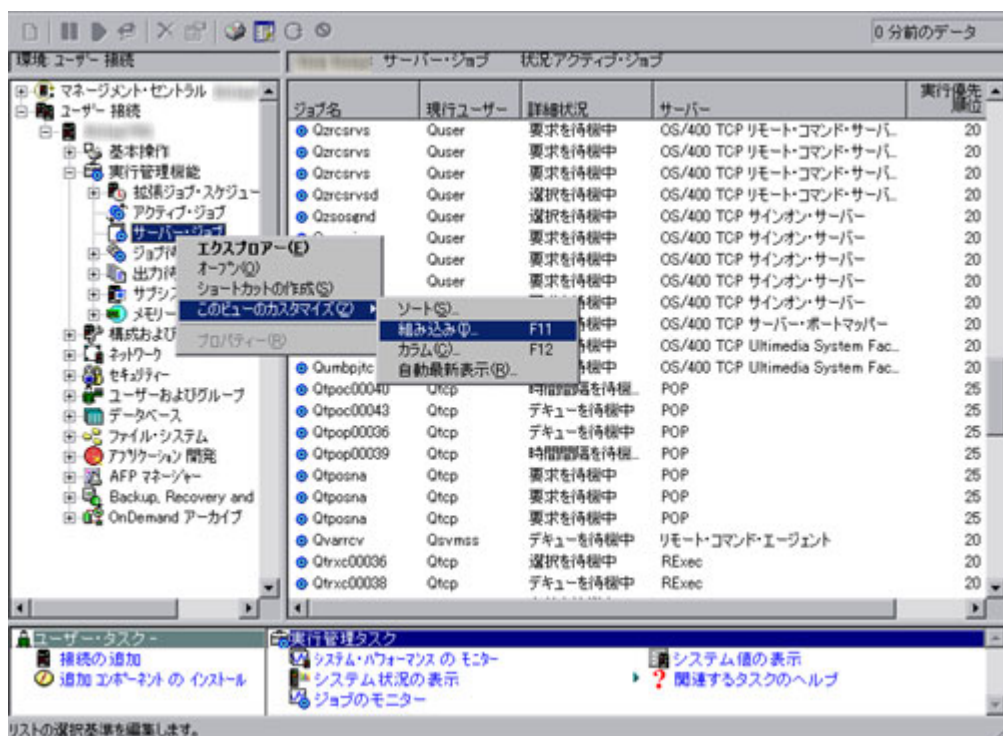
8. 管理するサーバー・ジョブを検索する。
9. ジョブ名を右マウス・ボタンでクリックする。
10. メニュー・オプションを選択する。

サーバー・ジョブの管理に使用できる各メニュー・オプションの詳細については、27 ページの『サーバー・ジョブの詳細とプロパティ』を参照してください。

## 終了したサーバー・ジョブ

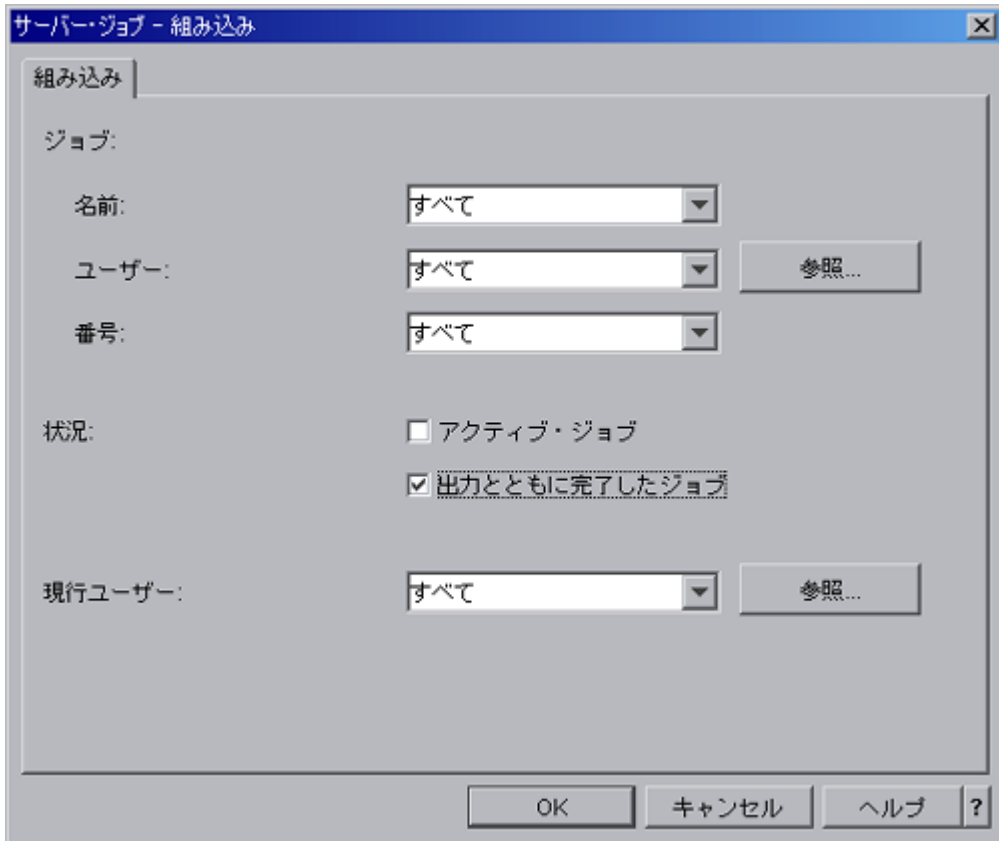
スプール出力を持つサーバー・ジョブが終了すると、そのジョブはシステム上に残り、表示することができます。以下のステップでは、iSeries™ ナビゲーターを使用して、スプール出力を持つ、終了したサーバー・ジョブの検索方法を説明します。

1. 「iSeries ナビゲーター」を開く。
2. 「ユーザー接続」を展開する。
3. 「iSeries サーバー」を展開する。
4. 「実行管理機能」を展開する。
5. 「サーバー・ジョブ」を右クリックし、「このビューのカスタマイズ」を選択し、次に「組み込み」を選択する。



「サーバー・ジョブ - 組み込み (25 ページを参照)」ダイアログ・ボックスが表示されます。

6. 「出力とともに完了したジョブ」を選択し、「OK」をクリックする。



サーバー・ジョブ・リストは、出力を持つ完了ジョブのみを表示するように変更されます。

7. 管理する完了サーバー・ジョブを検索する。
8. ジョブ名を右マウス・ボタンでクリックする。
9. メニュー・オプションをクリックする。

サーバー・ジョブの管理に使用できる各メニュー・オプションの詳細については、27 ページの『サーバー・ジョブの詳細とプロパティ』を参照してください。

### サーバー・ジョブ - 組み込み

「サーバー・ジョブ - 組み込み」ダイアログを使用して、iSeries ナビゲーター・ウィンドウに表示されるサーバー・ジョブのリストを、指定する基準に一致するものに制限します。以下に、このダイアログの各セクションについて説明します。

#### • ジョブ

iSeries ナビゲーター・ウィンドウに表示されるジョブ・リストの制限に使用する、以下に示す基準を指定します。

##### - 名前

リストに含めるジョブの名前。以下の値から選択します。

- **すべて** - すべてのジョブ名が含まれるようにします。ジョブ名は、リストのサブセット化には使用されません。
- **特定の名前** - 特定のジョブ名の下で現在実行中のジョブをリストに含めます。例えば、QZDASOINIT という名前を入力すると、この名前を持つジョブのみが表示されます。

- **総称名** - 指定した文字で始まるジョブをリストに含めます。1 つ以上の文字と、その後にアスタリスク (\*) を入力します。アスタリスクを文字と文字の間に入力することはできません。

#### - ユーザー

リストに含める必要のあるジョブの所有者であるユーザー。指定できる値は以下のとおりです。

- **すべて** - すべてのユーザーが含まれるようにします。ユーザー名は、リストのサブセット化には使用されません。
- **現在サインオン中のユーザー (Current signed on user)** - 現在サインオン中のユーザーに属するジョブをリストに含めます。このオプションは、リストに含めるサーバー・ジョブ名の一部として、現在サインオン中のユーザーのプロファイルと一致するユーザー・プロファイルを持つサーバー・ジョブを検索します。ほとんどの場合、このオプションはサーバー・ジョブには適用されません。
- **特定の名前** - 特定のユーザーの下で現在実行されているジョブが含まれます。有効なユーザー・プロファイル名を入力するか、「参照...」を選択してユーザーのリストから選択します。
- **総称名** - 指定した文字で始まるユーザー名がリストに含まれるようにします。1 つ以上の文字と、その後にアスタリスク (\*) を入力します。アスタリスクを文字と文字の間に入力することはできません。
- **参照** - 「参照」をクリックすると、「ユーザーの参照」ダイアログが表示されます。リストからユーザーを選択して、「OK」をクリックします。

#### - 番号

リストに含めるジョブのシステム生成ジョブ番号。指定できる値は以下のとおりです。

- **すべて** - すべてのジョブ番号が含まれるようにします。ジョブ番号は、リストのサブセット化には使用されません。
- **特定の番号 (Specific number)** - 特定のジョブ番号を持つジョブが表示されます。000000 から 999999 の 6 桁のジョブ番号を入力します。

#### • 状況

指定した状況のジョブがリストに含まれるようにします。ジョブの状況は、ジョブが実行されるときに、そのジョブがシステム内のどこにあるかを示します。指定できるオプションは以下のとおりです。

- **アクティブ・ジョブ** - 実行は開始されているが、まだ完了していないジョブを含めるよう選択します。
- **出力を持つ完了ジョブ** - 実行が完了した、スプール出力を持つジョブを含めるよう選択します。

#### • 現行ユーザー

現行ユーザーの下で実行中のジョブをリストに含めます。指定できるオプションは以下のとおりです。

- **すべて** - すべてのユーザーのジョブが含まれるようにします。
- **特定の名前** - 特定のユーザーの下で現在実行されているジョブが含まれます。有効なユーザー・プロファイル名を選択するか、「参照...」を選択してユーザーのリストから選択します。
- **参照** - 「参照」をクリックすると、「ユーザーの参照」ダイアログが表示されます。リストからユーザーを選択して、「OK」をクリックします。

## サーバー・ジョブ・テーブル

QSYSWRK で実行するサーバー・ジョブなどの、システム上で実行するサーバー・ジョブのジョブ名についての詳細は、iSeries<sup>™</sup> Information Center の『サーバー・テーブル』を参照してください。このテーブ



ルにはサブシステムとジョブ名が示されているため、アクティブ・ジョブとそのジョブ・ログを検索することができます。また、このテーブルには各サーバー・ジョブが使用するジョブ記述も示されています。デフォルトでは、ほとんどのサーバー・ジョブはジョブの終了時にジョブ・ログを生成しません (LOG パラメーターが 40 \*NOLIST に設定されています)。つまり、ジョブ・ログは作成されません。ジョブ・ログを生成して、すべてのメッセージがジョブ・ログに送信されるようにする場合は、LOG パラメーターを 40 \*SECLVL と指定する必要があります。

## サーバー・ジョブの詳細とプロパティ

iSeries<sup>TM</sup> サーバーでのジョブの検索方法に関係なく、ジョブが見つければ、そのジョブの管理 (ジョブの制御またはジョブ情報の表示のいずれか) のために実行できる作業がいくつかあります。以下のセクションでは、iSeries ナビゲーターを使って管理することができるサーバー・ジョブの詳細とプロパティについてクローズアップしています。

iSeries ナビゲーターを使用してサーバー・ジョブの詳細とプロパティを表示する方法を、以下のステップに示します。

1. 管理するサーバー・ジョブを検索する。
2. ジョブ名を右マウス・ボタンでクリックする。
3. メニュー・オプションを選択する。

ジョブ名	現行ユーザー	詳細状況	サーバー	実行優先 順位
Qsqsvr	Qsys	条件を待機中	SQL	10
Qsqsvr	Qdirsv	条件を待機中	SQL	20
Qsqsvr	Qsys	条件を待機中	SQL	10
Qsqsvr	Qdirsv	条件を待機中	SQL	20
Qsqsvr	Qdirsv	条件を待機中	SQL	20
Qsqsvr	Qdirsv	条件を待機中	SQL	20
Qsqsvr	Qdirsv	条件を待機中	SQL	20
Qsqsvr	Qdirsv	条件を待機中	SQL	20
Qsqsvr	Qsys	条件を待機中	SQL	10
Qecs	Qsvsm	デキューを待機中	システム・マネージャー	50
Qtvdevic	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvdevic	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvdevic	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvdevic	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvtelne	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvtelne	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvtelne	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvtelne	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvtelne	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtvtelne	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtcpip	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qiwvppjt	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qqfwfsv	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Server1	Qsys	時間間隔を待機...	TELNET 装置マネージャー	20
Qtfpjtcp	Quser	要求を待機中	転送機能サーバー - TCP/IP	20

各メニュー・オプションは、さらに以下のように展開されます。

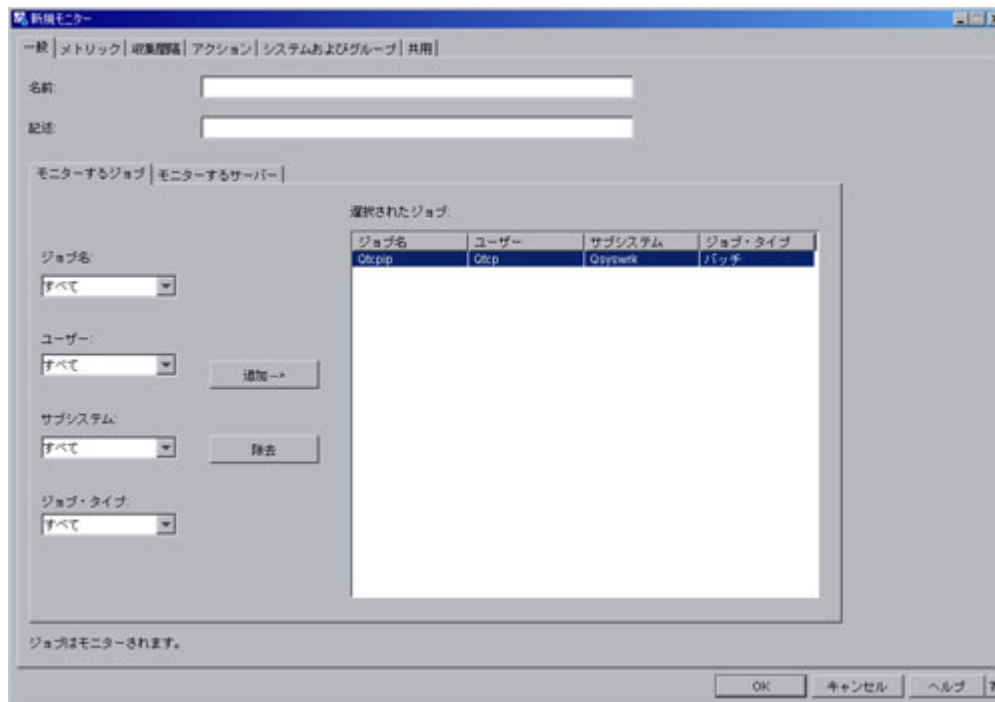
- モニター (28 ページを参照)
- プリンター出力 (28 ページを参照)
- ジョブ・ログ (29 ページを参照)
- 呼び出しスタック (30 ページを参照)
- ライブラリー・リスト (30 ページを参照)

- ロックされたオブジェクト (31 ページを参照)
- オープン・ファイル (32 ページを参照)
- スレッド (32 ページを参照)
- トランザクション (33 ページを参照)
- 経過パフォーマンス統計 (34 ページを参照)
- 最後の SQL ステートメント (37 ページを参照)
- 保留 (38 ページを参照)
- 削除 / 終了 (39 ページを参照)
- プロパティ (42 ページを参照)

## モニター

「モニター」を選択します。「モニター」ダイアログが新しいウィンドウでオープンします。

ジョブ・モニターを使用すると、ジョブ名、ジョブのユーザー、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することができます。例えば、ジョブの CPU 使用率、ジョブの状況、またはジョブ・ログのメッセージなどをモニターする必要があります。ジョブ・モニターを使用して、所定のしきい値を越えた場合に実行されるコマンドを定義することができます。iSeries Information Center の『シナリオ: CPU 使用率のジョブ・モニター (scenario: job monitor for CPU utilization)』に、ジョブ・モニターを適切に設定して構成する方法の例が記載されています。



「新規モニターまたはモニター・プロパティ」の「一般」ページを使用すると、モニターに関する一般情報を表示および変更することができます。一般情報には、モニターの名前、モニターの要旨、およびモニターするジョブとサーバーが含まれています。新規モニターの作成方法についての詳細は、iSeries Information Center の『新しいモニターの作成』を参照してください。さらに、詳細については、iSeries Information Center のトピック『マネージメント・セントラル』も参照してください。

## プリンター出力

「プリンター出力」を選択します。ジョブのプリンター出力を表示できる、新しいウィンドウがオープンします。

出力名	ユーザー	指定データ	ユーザー	状況	プリンター	1部ごとのページ数	残りの部数	作成時刻
Qpsrvdnp			QUSER	作動可能	未割り当て	2	1	09:41:37
Qpjoblog	QZRCRSRV5		QUSER	作動可能	未割り当て	3	1	09:41:38
Qpdsplib			QUSER	作動可能	PRT01	6	1	09:41:38
Qpsrvdnp			QUSER	作動可能	未割り当て	2	1	10:41:00
Qpjoblog	QZRCRSRV5		QUSER	作動可能	未割り当て	6	1	10:41:00
Qpdsplib			QUSER	作動可能	PRT01	6	1	10:41:01

表示される情報は、以下の方法を使用してアクセスおよびカスタマイズできます。

- プリンター出力の「表示」、「オープン」、「保留」、「解放」、「次を印刷」、「送信」、「カット」、「コピー」、「移動」、または「削除」
- プリンター出力プロパティの変更
- プリンター出力表示の調整
- プリンター出力のドラッグ・アンド・ドロップ

## ジョブ・ログ

「ジョブ・ログ」を選択します。ジョブのジョブ・ログを表示できる、新しいウィンドウがオープンします。

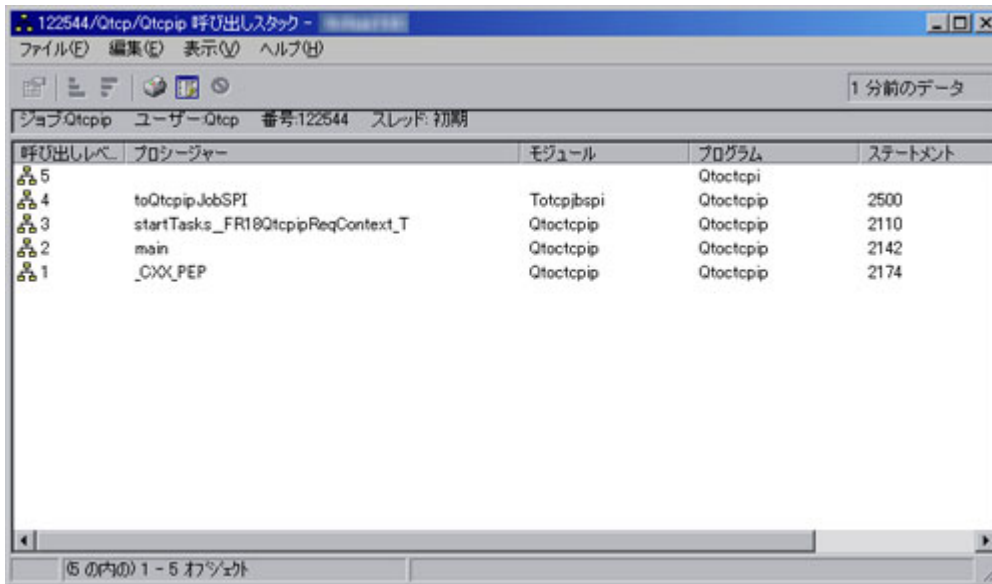
メッセージ ID	メッセージ・テキスト	送信	メッセージ・タイ...	重大度
CPFI124	OSYSのサブシステムOSYSWRKのジョブ122644/QTCP/QTCPPIPが...	04/01/14 12:12:05	情報	0
CPN125	ジョブ122644/QTCP/QTCPPIPが投入された。	04/01/14 12:12:05	情報	0
CPC1221	ジョブ122644/QTCP/QTCPPIPがライブアラームOSYSのジョブ待...	04/01/14 12:12:08	情報	0
TCP1A95	ジョブ122644/QTCP/QTCPPIPが開始された。	04/01/14 12:12:10	完了	0
TCP1B07	ジョブ122644/QTCP/QTCPPIPのユーザーQTCPがTCP/IP *LOOP...	04/01/14 12:12:10	情報	0
TCP1B07	ジョブ122644/QTCP/QTCPPIPのユーザーQTCPがTCP/IP *OPC...	04/01/14 12:12:10	情報	0
TCP1B31	*LIN TEH01の状況はオンに構成変更である。	04/01/14 12:12:11	情報	0
CPD2645	回線TEH01はすでにオンに構成変更されている。	04/01/14 12:12:11	診断	40
CPC2609	制御装置TEHNETのオンへの構成変更が完了した。	04/01/14 12:12:11	完了	0
CPC2605	装置TEHTCPのオンへの構成変更が完了した。	04/01/14 12:12:11	完了	0
TCP1B31	*CTL TEHNETの状況は活動状態である。	04/01/14 12:12:11	情報	0
TCP1B31	*DEV TEHTCPの状況はオンに構成変更である。	04/01/14 12:12:11	情報	0
TCP1B33	活動化要求3DEDC99BA4BDTEHTCPが出された。	04/01/14 12:12:11	情報	0
TCP1B34	活動化要求3DEDC99BA4BDTEHTCPは完了した。コードは0で...	04/01/14 12:12:11	情報	0
TCP1B03	ユーザーQTCPがTEH01のIPインターフェース9.170.152.27を開始しまし...	04/01/14 12:12:11	完了	0
TCP1B31	*LIN VETH01の状況はオンに構成変更である。	04/01/14 12:12:11	情報	0
CPD2645	回線VETH01はすでにオンに構成変更されている。	04/01/14 12:12:11	診断	40
CPC2609	制御装置VETHNETのオンへの構成変更が完了した。	04/01/14 12:12:11	完了	0
CPC2605	装置VETH0TCPのオンへの構成変更が完了した。	04/01/14 12:12:11	完了	0
TCP1B31	*CTL VETHNETの状況は活動状態である。	04/01/14 12:12:11	情報	0
TCP1B31	*DEV VETH0TCPの状況はオンに構成変更である。	04/01/14 12:12:11	情報	0
TCP1B33	活動化要求3DEDCA1586C7VETH0TCPが出された。	04/01/14 12:12:11	情報	0
TCP1B34	活動化要求3DEDCA1586C7VETH0TCPは完了した。コードは0...	04/01/14 12:12:11	情報	0

表示される情報は、以下の方法を使用してアクセスおよびカスタマイズできます。

- ジョブ・ログ画面の調整
- ジョブ・ログ画面のソート

### 呼び出しスタック

「詳細」を選択してから「呼び出しスタック」を選択します。ジョブの呼び出しスタックを表示できる、新しいウィンドウがオープンします。



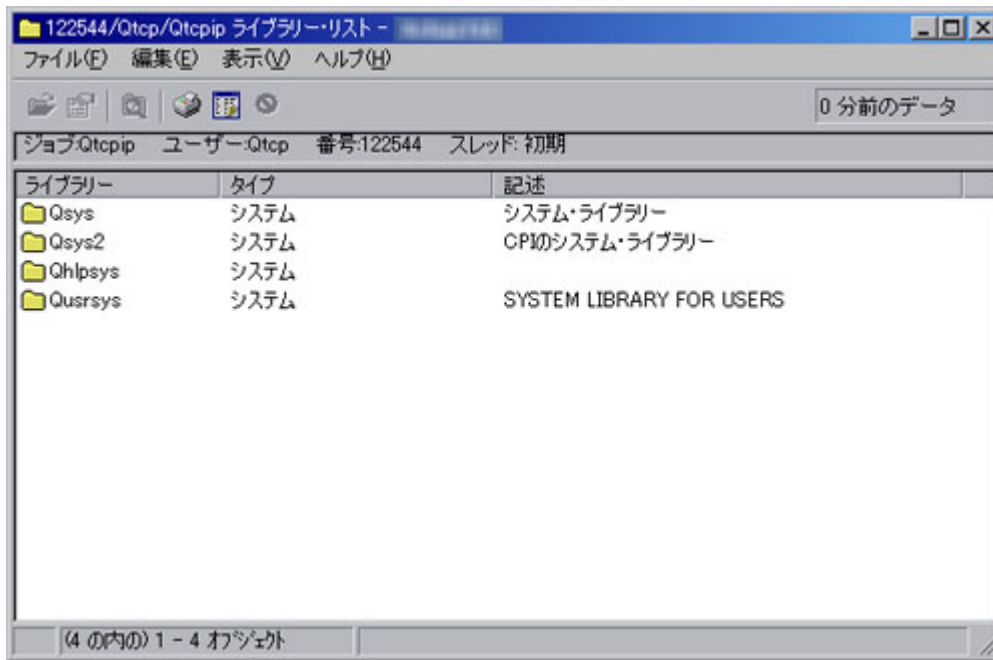
The screenshot shows a window titled "122544/Qtcp/Qtcp 呼び出しスタック". The window contains a table with the following data:

呼び出しレベル	プロシージャ	モジュール	プログラム	ステートメント
5			Qtoctpi	
4	toQtcpJobSPI	Totcpjobspi	Qtoctpip	2500
3	startTasks_FR18QtcpReqContext_T	Qtoctpip	Qtoctpip	2110
2	main	Qtoctpip	Qtoctpip	2142
1	_CXX_PEP	Qtoctpip	Qtoctpip	2174

呼び出しスタックは、表示することはできますが、変更はできません。

### ライブラリー・リスト

「詳細」を選択してから「ライブラリー・リスト」を選択します。ジョブのライブラリー・リストを表示できる、新しいウィンドウがオープンします。

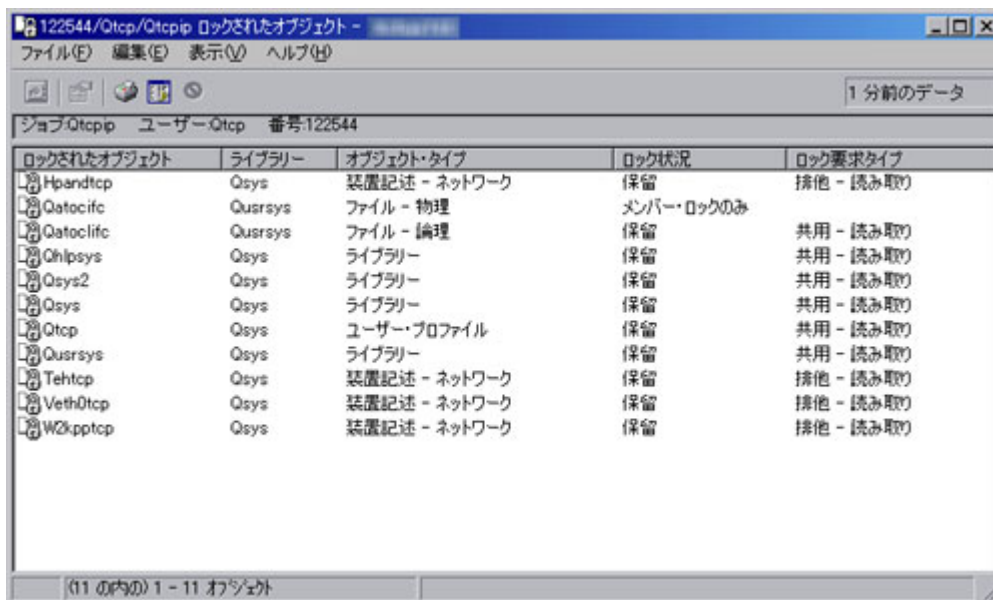


ライブラリー・リストからライブラリーを選択します。ライブラリーを選択した後、以下の操作を実行できます。

- ライブラリーのオープン
- ライブラリーのプロパティの表示
- ライブラリー・リストの検索

### ロックされたオブジェクト

「詳細」を選択してから「ロックされたオブジェクト」を選択します。ジョブによってロックされたオブジェクトを表示できる、新しいウィンドウがオープンします。



ロックされたオブジェクトをリストから選択します。ロックされたオブジェクトを選択した後、以下の操作を実行できます。

- ロック・ホルダーの表示
- ロックされたメンバーの表示
- ロックのプロパティの表示

## オープン・ファイル

「詳細」を選択してから「オープン・ファイル」を選択します。ジョブによってオープンされているオブジェクトを表示できる、新しいウィンドウがオープンします。



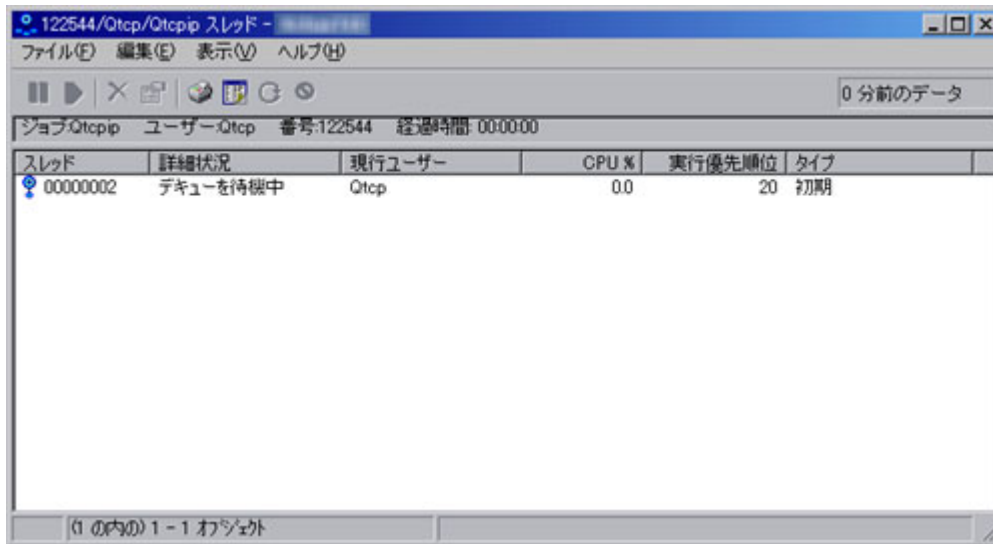
オープンされているファイルをリストから選択します。オープンされているファイルを選択した後、以下の操作を実行できます。

- ファイルのプロパティの表示

## スレッド

「詳細」を選択してから「スレッド」を選択します。ジョブ・スレッドを表示できる、新しいウィンドウがオープンします。

スレッドは、ジョブ内にあるディスパッチ可能な独立した作業単位、とみなすことができます。詳細については、iSeries Information Center のトピック『実行管理機能』でスレッドの情報を参照してください。



リストからスレッドを選択します。スレッドを選択した後、以下の操作を実行できます。

- スレッドの保留
- スレッドの開放
- スレッドの削除/終了
- スレッドのプロパティの表示
- スレッドの呼び出しスタック、ライブラリー・リスト、ロックされたオブジェクト、トランザクション、および経過パフォーマンス統計の表示
- スレッド統計のリセット

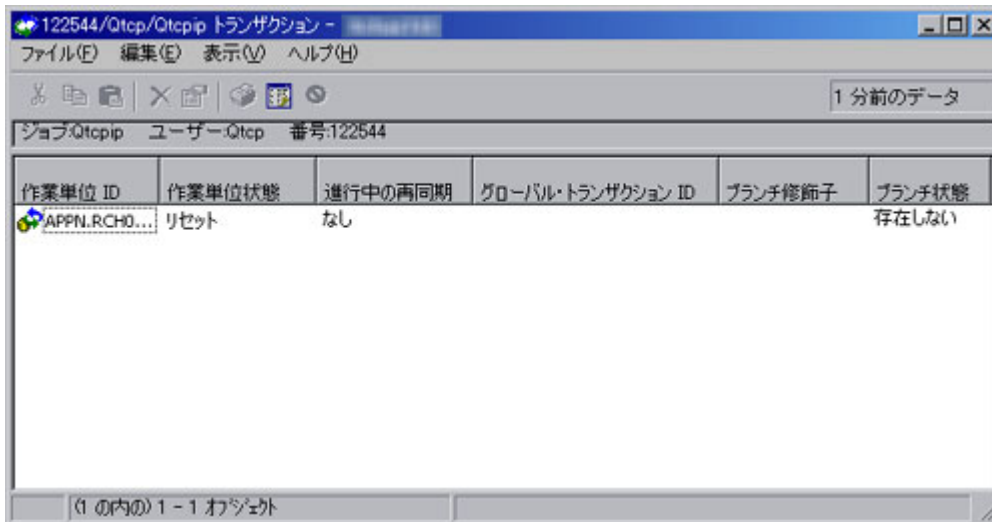
### トランザクション

「詳細」を選択してから「トランザクション」を選択します。ジョブ・トランザクションを表示できる、新しいウィンドウがオープンします。

トランザクションは、システム上のオブジェクトに対する個々の変更のグループとして定義されるもので、ユーザーに対しては単一の微細な変更のように見えます。トランザクションは、作業論理単位 (LUW) と呼ばれる場合もあります。コミットメント制御が使用され、個々の変更をまとめたグループ全体の変更が、関係するすべてのシステム上で生じるか、あるいはその変更が何も生じないか、のいずれかであることが保証されます。

iSeries サーバーは、グローバル・トランザクションとデータベース・トランザクションという、2 つの異なるタイプのトランザクションをサポートしています。グローバル・トランザクションには、DB2 UDB for iSeries の外部と内部の両方の変更が含まれることがあります。データベース・トランザクションには、DB2<sup>(R)</sup> UDB for iSeries<sup>(TM)</sup> の変更のみが含まれます。

コミットメント制御とトランザクションについての詳細は、iSeries Information Center の『コミットメント制御』を参照してください。



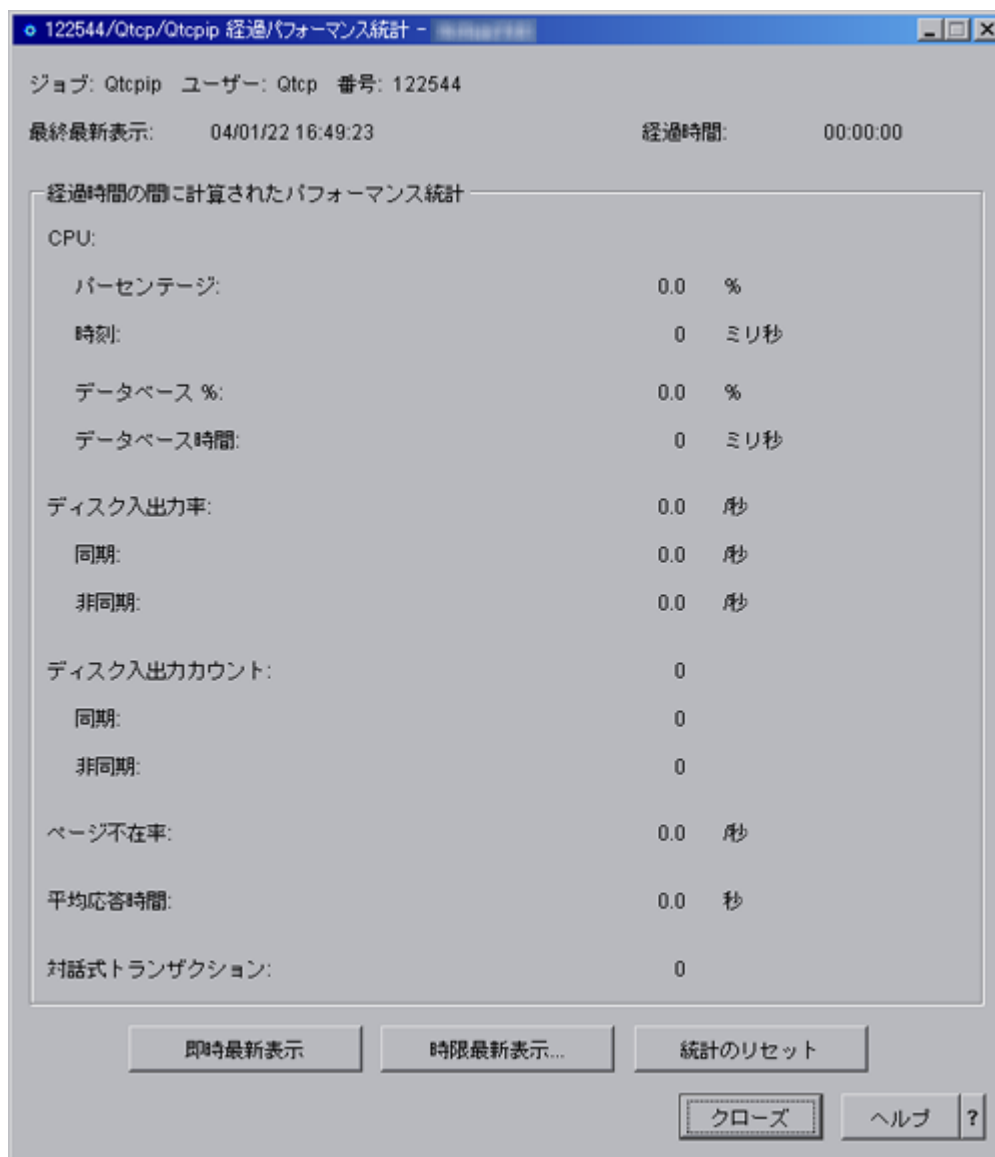
「作業単位識別コード (トランザクション) (Unit of Work ID (transaction))」を選択します。トランザクションを選択した後、以下の操作を実行できます。

- トランザクション・プロパティの表示
- トランザクションのジョブの表示
- トランザクションのリソース統計の表示
- トランザクションのコミットの強制
- トランザクションのロールバックの強制
- トランザクションの再同期のキャンセル

#### 経過パフォーマンス統計

「詳細」を選択してから「経過パフォーマンス統計」を選択します。経過パフォーマンス統計のダイアログが、新しいウィンドウでオープンします。





「経過パフォーマンス統計」ページを使って、ジョブ名、CPU 統計、ディスク入出力のレートと回数、およびページ不在率などの、ジョブまたはスレッドの詳細なパフォーマンス情報を表示します。経過パフォーマンス統計は、「経過時間」フィールドに指定された時間間隔で計算されます。「即時最新表示」ボタンまたは「時限最新表示」ボタンを使って、データを更新します。このウィンドウの以下のエレメントについては、より詳細なヘルプが用意されています。

- ジョブ

スレッドが実行されているジョブの名前。

- ユーザー

このスレッドが実行されているジョブのユーザーの名前。

- 番号

このスレッドが実行されているジョブにシステムが割り当てた番号。

- スレッド

ジョブ内の特定のスレッドを示す。注: 「スレッド (Thread)」 フィールドは、スレッドの経過パフォーマンス統計が表示される場合にのみ表示されます。

- **最終最新表示**

経過パフォーマンス統計が最後に最新表示された日時。

- **経過時間**

経過パフォーマンス統計が計算されている時間間隔。

- **経過時間の中に計算されたパフォーマンス統計**

- **CPU**

CPU の使用率に関する、以下のパフォーマンス統計が表示されます。

- **パーセンテージ**

経過時間中にジョブまたはスレッドによって使用された、使用可能な処理装置時間の割合。マルチプロセッサ・システムの場合は、すべてのプロセッサの平均値となります。

- **時刻**

経過時間中にジョブまたはスレッドによって使用された、処理装置時間の合計値 (ミリ秒)。

- **データベース %**

経過時間中に、データベース処理に使用された処理装置時間の合計の割合。マルチプロセッサ・システムの場合は、すべてのプロセッサの平均値となります。

- **データベース時間**

経過時間中にデータベース処理に使用された処理装置時間の合計 (ミリ秒)。

- **ディスク入出力率**

経過時間中、1 秒間にジョブまたはスレッドによって行われたディスク入出力操作の平均回数。この値は同期ディスク入出力操作および非同期ディスク入出力操作の合計です。

- **同期**

経過時間中、1 秒間にジョブまたはスレッドによって行われた同期 (物理) ディスク入出力操作の平均回数。この値はデータベースとデータベース以外に対する同期読み書きの合計です。

- **非同期**

経過時間中、1 秒間にジョブまたはスレッドによって行われた非同期ディスク入出力操作の平均回数。この値はデータベースとデータベース以外に対する非同期読み書きの合計です。

- **ディスク入出力カウント**

経過時間中に、ジョブまたはスレッドによって行われたディスク入出力操作の回数。この値は同期ディスク入出力操作および非同期ディスク入出力操作の合計です。

- **同期**

経過時間中に、ジョブまたはスレッドによって行われた同期 (物理) ディスク入出力操作の回数。この値はデータベースとデータベース以外に対する同期読み書きの合計です。

- **非同期**

経過時間中に、ジョブまたはスレッドによって行われた非同期 (物理) ディスク入出力操作の回数。この値はデータベースとデータベース以外に対する非同期読み書きの合計です。

- ページ不在率

経過時間中、1 秒間にアクティブ・プログラムが主記憶域に存在しないアドレスを参照した回数の平均値。

- 平均応答時間

このジョブに対して、経過時間中に行われた対話式トランザクションの応答時間の平均値を秒単位で表したものの。

- 対話式トランザクション

このジョブに対して、経過時間中に行われたユーザー対話の回数。例えば、ユーザーがファンクション・キーを押した回数です。注: スレッドに対する経過パフォーマンス統計が表示されていない場合、「平均応答時間」フィールドと「対話式トランザクション数」フィールドは表示されません。

- 即時最新表示

これをクリックすると、最新の経過パフォーマンス統計が表示され、統計計算の対象となる期間が延長されます。

- 時限最新表示...

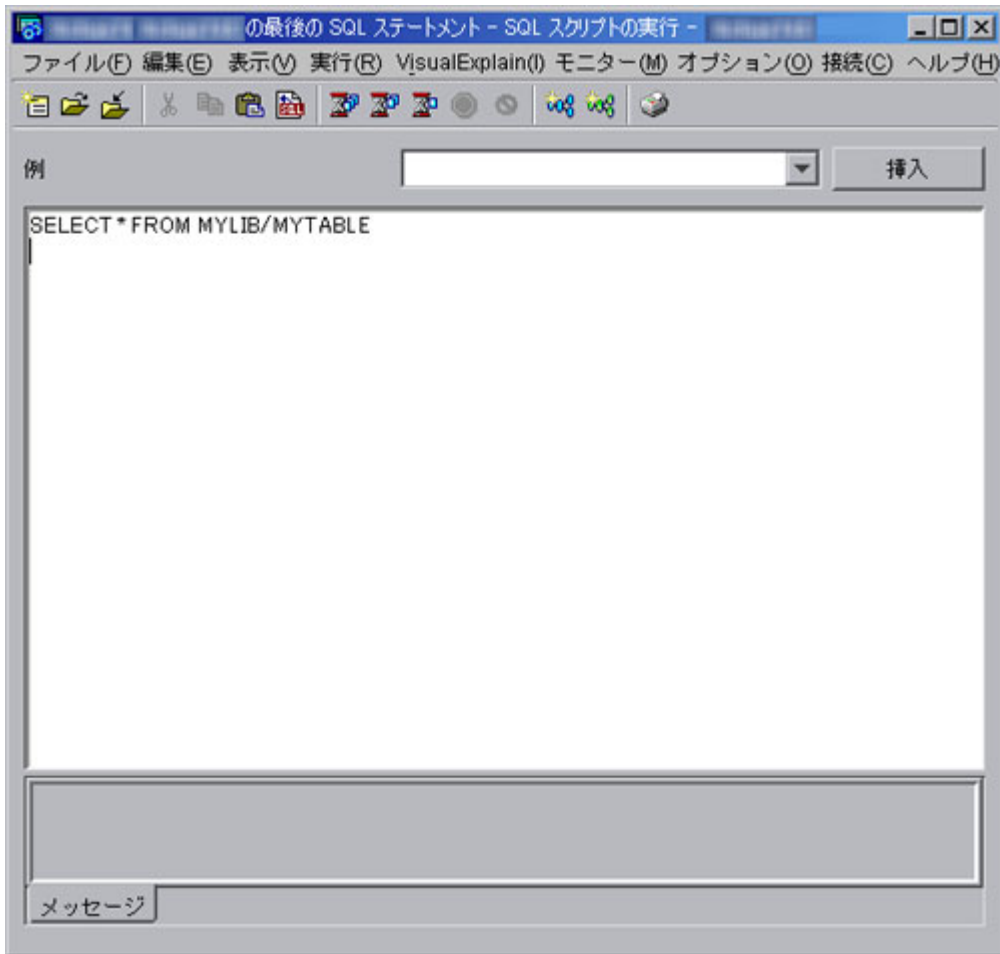
経過パフォーマンス統計の自動更新をセットアップするためのダイアログを表示します。

- 統計のリセット

経過パフォーマンス統計をクリアし、統計の計算対象となる期間をリセットします。経過パフォーマンス統計をリセットしても、スレッドのコンテナには影響はありません。現在のダイアログのみリセットされます。

## 最後の SQL ステートメント

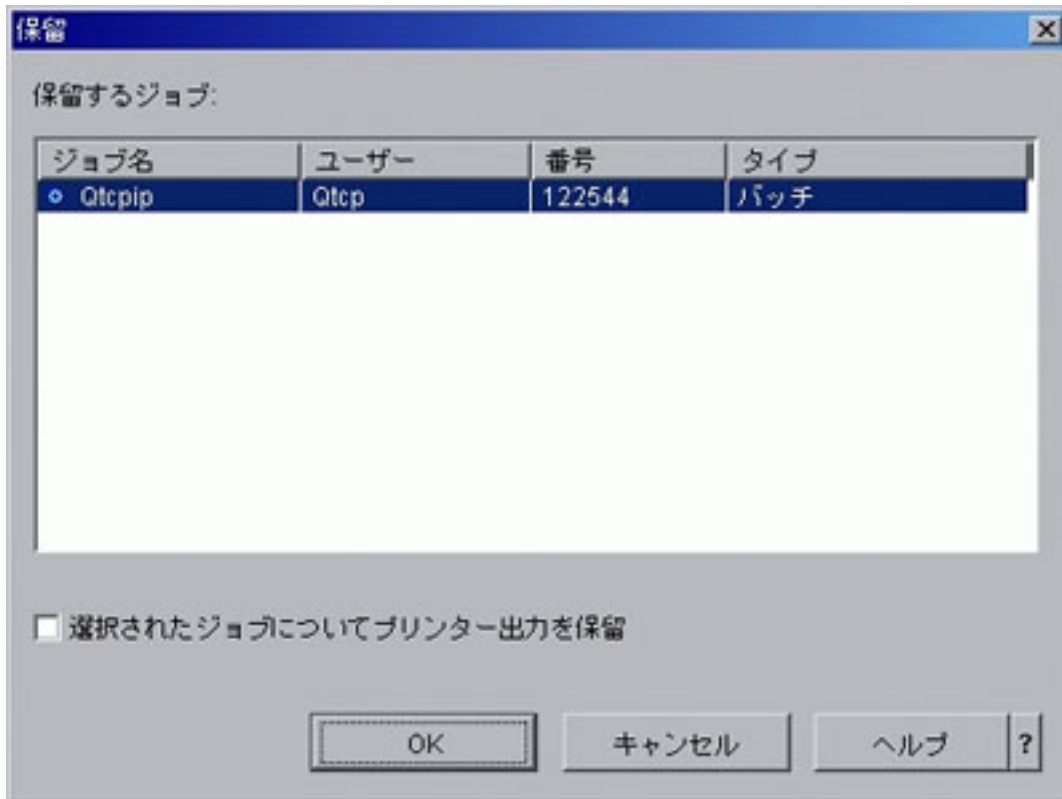
「詳細」を選択してから「最後の SQL ステートメント」を選択します。ジョブが SQL ステートメントを実行している場合は、「SQL スクリプトの実行」ウィンドウがオープンします。



「SQL スクリプトの実行」ウィンドウを使って、SQL ステートメントのスクリプトを作成、編集、実行、およびトラブルシューティングします。スクリプトを使った作業が終了したら、それらのスクリプトを PC に保管することができます。詳細については、iSeries ナビゲーターで「SQL スクリプトの実行」のヘルプ・トピックを参照してください。

## 保留

「保留」を選択します。「保留」ダイアログが新しいウィンドウで開きます。



「保留」ダイアログを使って、選択したジョブを保留します。ジョブを保留すると、ジョブは中断され、それ以降の処理が実行されなくなります。このウィンドウの以下のエレメントについては、より詳細なヘルプが用意されています。

- 保留するジョブ

保留が選択されているジョブが表示されます。選択内容が正しくない場合は、「キャンセル」をクリックするか、表示されている項目を選択解除して選択内容を変更します。リスト内の項目を選択解除するには、Ctrl キーを押しながら左マウス・ボタンをクリックします。

- 選択したジョブについてプリンター出力を保留

保留中のジョブによって作成されたプリンター出力も保留にするかどうかを指定します。プリンター出力を保留するとは、出力待ち行列にファイルを保留し、印刷しないという意味です。指定できるオプションは以下のとおりです。

- はい

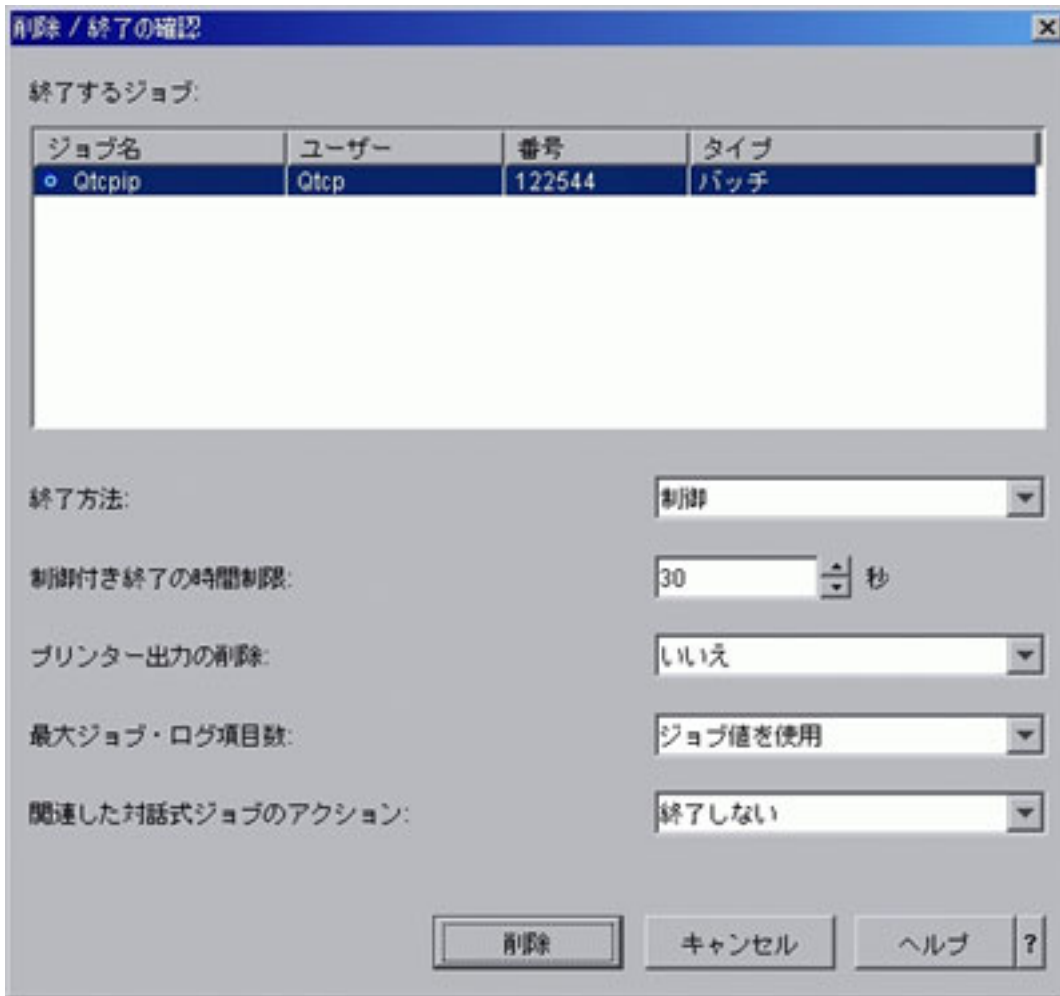
保留されているジョブのプリンター出力も保留されます。

- いいえ

保留中のジョブに対するプリンター出力は保留されず、印刷できます。

### 削除 / 終了

「削除 / 終了」を選択します。「削除 / 終了の確認」ダイアログが新しいウィンドウで開きます。



「削除 / 終了の確認」ダイアログを使って、選択したジョブを削除します。ジョブを削除すると、ジョブの処理は終了されます。「削除」をクリックすると、ジョブが終了します。このウィンドウの以下のエレメントについては、より詳細なヘルプが用意されています。

- 終了するジョブ

終了または削除するよう選択したジョブが表示されます。選択内容が正しくない場合は、「キャンセル」をクリックするか、表示されている項目を選択解除して選択内容を変更します。リスト内の項目を選択解除するには、Ctrl キーを押しながら左マウス・ボタンをクリックします。

- 終了方法

ジョブを終了する方法を指定します。指定できるオプションは以下のとおりです。

- 制御

指定した時間制限内に、制御された方法でジョブを終了することができます。ジョブのアプリケーション・プログラムは、ジョブ終了処理を実行できます。

- 即時

ジョブは遅延時間なしで即時に終了します。アプリケーション・プログラムは、終了方法で「制御付き」を指定した場合のようにジョブ終了処理を実行することはできません。注: データが部分的に更

新されている場合、このオプションは好ましくない結果を引き起こす可能性があります。このため、このオプションは、制御付き終了が正常に行われなかった場合のみ使用してください。

- **制御付き終了の時間制限**

アプリケーション・プログラムがジョブ終了処理を実行しなければならない制限時間を指定します。指定可能な値は 1 から 999999 秒です。

- **プリンター出力の削除**

ジョブによって作成され、まだ印刷されていないプリンター出力も削除するかどうかを指定します。指定できるオプションは以下のとおりです。

- はい

終了するジョブのプリンター出力は削除され、印刷はされません。

- いいえ

終了するジョブのプリンター出力は保持され、印刷できます。

- **最大ジョブ・ログ項目数**

終了するジョブのジョブ・ログに書き込む項目の最大数を選択します。指定できるオプションは以下のとおりです。

- 番号

ジョブ・ログに書き込むメッセージの最大数を指定します。ジョブ・ログに含まれるメッセージの数が最大数よりも少ないときにこの値を入力した場合のみ、この値が最大数になります。それ以外の場合、この制限により、ジョブ・ログに対してさらにメッセージを書き込む処理だけが停止されます。ジョブ・ログにメッセージを書き込む前に 0 を指定した場合、ジョブ・ログは作成されません。指定可能な値は 0 から 2147483647 までです。

- ジョブ値を使用

ジョブ・ログの項目の最大値は変更されません。すべてのジョブは値が「無限大」で開始され、後で変更 (例えば、直前の削除など) されない限り、これが現行のジョブ値となります。

- 無限大

ジョブ・ログに書き込まれるメッセージの数に制限はありません。

- **関連する対話式ジョブのアクション**

選択したジョブが終了したときに、その他の関連する対話式ジョブをどのように処理するかを指定します。指定できる選択肢は以下のとおりです。

- 終了しない

選択したジョブのみ終了します。

- **グループ・ジョブを終了 (End for group jobs)**

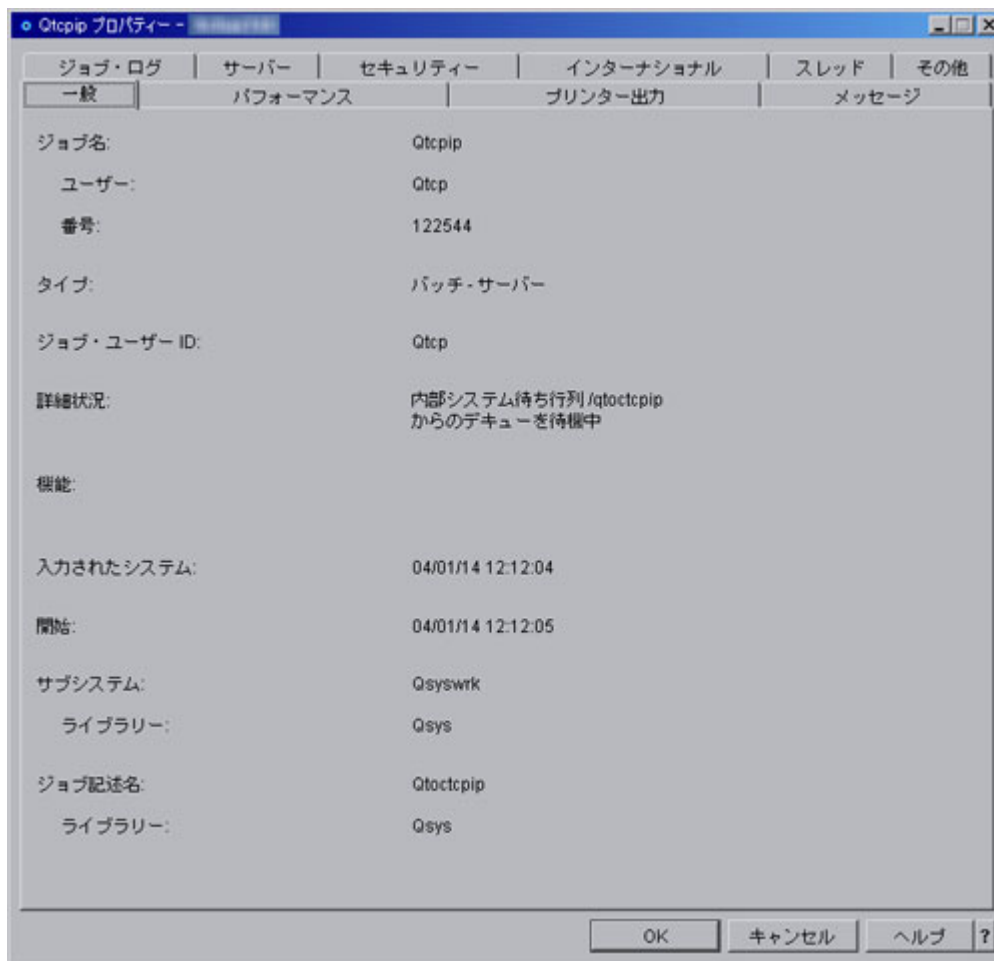
選択したジョブがグループ・ジョブである場合、このグループに対するすべてのジョブを終了します。ジョブがグループ・ジョブではない場合、選択したジョブが終了します。

- **すべて終了 (End all)**

選択したジョブについて、グループ・ジョブおよび 2 次ジョブを含む、ワークステーション上で実行中の対話式ジョブをすべて終了します。

## プロパティ

「プロパティ」を選択します。「ジョブ・プロパティ」ダイアログが新しいウィンドウで開きます。



「ジョブ・プロパティ」ダイアログについての詳細は、iSeries Information Center の『ジョブ・プロパティ』を参照してください。

## 対話式サブシステム構成

対話式ユーザーとは、5250 ディスプレイ装置セッションを実行するユーザーを指します。これらのセッションは、平衡型、リモート・ワークステーション、Telnet、5250 ディスプレイ装置パススルー、仮想端末 API ベース・アプリケーション、およびさまざまな Web ベースの接続 (WebFacing、IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Host Access Transformation Server (HATS)、iSeries<sup>(TM)</sup> Access for the Web 5250 サポートなど) から接続されます。

新しい対話式サブシステムのセットアップ方法を、以下のステップで説明します。単一の対話式サブシステムに割り当てる装置の数は、250 から 300 に制限することをお勧めします。このため、システム上のユーザーの数を考慮して、適切な数の対話式サブシステムを作成する必要があります。

最初の計画ステップの 1 つとして、サブシステムに使用する命名規則を決定します。命名規則は、INTER1、INTER2、INTER3 といった単純なもの、処理のタイプを反映するもの



(INVENTORY、ORDERENT、PGMR)、またはユーザーの地理的な所在地を反映するもの (EAST、CENTRAL、WEST) などに行うことができます。システム管理のために、分かりやすい命名規則を決定してください。

以下のステップでは、コマンドを手動で入力するかのようにして説明しています。ただし、リカバリーの際に構成を簡単に再作成できるよう、CL プログラムを使ってサブシステムを作成する必要があります。

1. サブシステム構成オブジェクトを保管するライブラリーを作成する。この例では、SBSLIB を使用します。

```
CRTLIB SBSLIB TEXT('Library to hold subsystem configuration objects')
```

2. クラスを作成する。クラスは、使用する対話式サブシステムの一部のパフォーマンス特性を定義します。QINTER クラスと同一のクラスを作成するには、以下のコマンドを入力します。

```
CRTCLS SBSLIB/INTER1 RUNPTY(20) TIMESLICE(2000) PURGE(*YES) DFTWAIT(30)  
TEXT('Custom Interactive Subsystem Class')
```

カスタムの対話式サブシステムに QGPL 内の QINTER クラスを使用するか、対話式サブシステムのすべてに使用する単一のクラスを作成するか、各対話式サブシステムごとにクラスを作成することができます。どの方法を選択するかは、特定のサブシステムのパフォーマンス設定の一部をカスタマイズしたいかどうかによって変わります。IBM 提供のサブシステムは、各サブシステムごとにクラスが作成された状態で出荷されており、そのクラスの名前はサブシステムの名前と同じです。このため、各サブシステムごとのクラスをサブシステムと同じ名前で作成しない場合は、経路指定項目追加 (ADDRTGE) コマンドにクラス名を指定する必要があります。これは、CLS パラメーターのデフォルトが \*SBSD である (つまり、クラス名がサブシステム記述と同じ名前になる) ためです。

例えば、ビジネス上重要な処理を行うユーザーには、他のユーザーよりも高い実行優先順位を与える必要がある場合があります。これは、ビジネス上重要な処理を実行するユーザーを、高い実行優先順位 (RUNPTY(15) など) を指定するクラスを使用して構成された別個のサブシステムで実行させることによって行うことができます (実行優先順位では、数値の低いほうが優先順位が高いことに注意してください)。

3. サブシステム記述を作成する。定義する必要のある各サブシステムごとに、このステップを繰り返します。

```
CRTSBSD SBSDB(SBSLIB/INTER1) POOLS((1 *BASE) (2 *INTERACT)) SGNDSPF(*QDSIGNON)
```

これにより、QINTER と同一の属性を持つサブシステム記述が作成されます。

使用される記憶域プールは、サブシステム記述で指定されます。ユーザーのセットをそれらのユーザー自身のプールに分離したい場合には、共用 \*INTERACT プールを使用するのではなく、そのサブシステムの具体的な記憶域サイズと活動レベルと共にプール ID を指定することによって、これを行うことができます。オプションで、共用記憶域プール変更 (CHGSHRPOOL) コマンドを使って共用プールを定義し、その共用プールをサブシステム記述に指定することもできます。

サブシステム記述では、サブシステムで実行するジョブの数も定義します。次に示す、考慮すべきいくつかの異なるオプションがあります。

- CRTSBSD コマンドの POOLS パラメーターの活動レベル。活動レベルは、プール内で一度にアクティブに実行できるスレッドの最大数を決定します。プール内ではこの数よりも多くのスレッドをアクティブにすることができますが、活動レベルは、任意の一時点においてアクティブに実行できる数を決定します。
- CRTSBSD コマンドの最大ジョブ数 (MAXJOBS) パラメーター。最大ジョブ数は、サブシステムで実行するユーザー・ジョブの数を決定します。この値を超えることはできません。既に最大数のジョブが実行しているときに追加のジョブを開始しようとすると、失敗します。

- 経路指定項目追加 (ADDRTGE) コマンドの最大アクティブ経路指定ステップ (MAXACT) パラメーター。これは、この経路指定項目からアクティブにすることができるジョブの最大数です。デフォルト値の \*NOMAX を使用することをお勧めします。
- ワークステーション項目追加 (ADDWSE) コマンドの最大アクティブ・ジョブ数 (MAXACT) パラメーター。これは、ワークステーション項目から同時にアクティブにすることができるアクティブ・ジョブの最大数です。デフォルト値の \*NOMAX を使用することをお勧めします。
- ジョブ待ち行列項目追加 (ADDJOBQE) コマンドの最大アクティブ・ジョブ数 (MAXACT) パラメーター。これは、ジョブ待ち行列から同時にアクティブにすることができるジョブの最大数です。このパラメーターのデフォルトは 1 であるため、この値は適切に構成する必要があります。

経路指定項目とワークステーション項目の最大アクティブ (MAXACT) パラメーターは使用可能ですが、複雑さが増し、偶発的にシステムへのユーザーのアクセスを妨げる可能性が高くなるため、これらのパラメーターは使用しないことをお勧めします。

各サブシステムごとにカスタムのサインオン・ディスプレイ・ファイルを使用することもできます。追加のパラメーター SGNDSPF(\*QDSIGNON) は、サブシステムがシステム提供のサインオン・ディスプレイ・ファイルである QDSIGNON を使用することを指定します。独自のカスタム・サインオン・ディスプレイ・ファイルを作成し、サブシステムの作成時にここでそのファイル名を指定することができます。独自のサインオン・ディスプレイ・ファイルを作成する場合は、「AS/400e シリーズ 実行管理の手引き」で詳細を参照してください。

4. サブシステム名と同じ名前を使ってサブシステムのジョブ待ち行列を作成し、ジョブ待ち行列項目をサブシステム記述に追加する。ジョブ転送 (TFRJOB) コマンドを使ってカスタム・サブシステム内にジョブを転送する必要がある場合、このステップは必須です。

ただし、ジョブ待ち行列項目を対話式サブシステムに追加することによって、バッチ処理をサブミットする能力も対話式サブシステムに追加されます。対話式サブシステム内でバッチ処理が実行されないようにする必要がある場合は、ジョブ待ち行列を作成せず、ジョブ待ち行列項目を追加しないでください。ジョブ待ち行列項目がない場合、ジョブ転送 (TFRJOB) コマンドは使用できなくなるため、後述するワークステーション項目を定義して、ユーザーを正しいサブシステムに入れる必要があります。

```
CRTJOBQ JOBQ(SBSLIB/INTER1)
ADDJOBQE SBSL(SBSLIB/INTER1) JOBQ(SBSLIB/INTER1) MAXACT(*NOMAX)
```

5. 経路指定項目をサブシステムに追加する。次の経路指定項目を追加します。QINTER 用にシステム上に標準で用意されている経路指定項目を見ると、いくつかの追加の経路指定項目が含まれていることが分かります。これらの機能が必要な場合は、カスタマイズしたサブシステム記述にもこれらの経路指定項目を追加します。

```
ADDRTGE SBSL(SBSLIB/INTER1) SEQNBR(9999) CMPVAL(*ANY) PGM(QSYS/QCMD) POOLID(2)
```

**注:** ADDRTGE コマンドはどのプール ID (POOLID) を使用するかを指定します。このコマンドのデフォルトは 1 です。この例では、\*INTERACT プールである 2 を指定します。専用プールを含むサブシステム記述をセットアップした場合は、経路指定項目に適切なプール ID を指定してください。

また、クラスの名前がサブシステム記述の名前と異なる場合は、ADDRTGE コマンドでクラスを指定する必要があります。

サインオン時にユーザーをアプリケーションに直接送れるように経路指定項目を設定すると、経路指定項目が非常に複雑になることがあります。さらに、複数の経路指定項目を使用して、サブシステム内の

複数のユーザーにさまざまなパフォーマンス特性をセットアップすることができます。経路指定項目によって使用できるオプションを調べる必要がある場合は、「AS/400e シリーズ 実行管理の手引き」を参照してください。

- ワークステーション項目をサブシステム記述に追加する。これは、サブシステムに割り当てる装置を割り振るための重要なステップです。

対話式サブシステムが割り振る装置は、サブシステム記述に追加されたワークステーション項目によって決定されます。ワークステーション項目は、ワークステーション名またはタイプのいずれかと、それらのワークステーションに対して行う必要がある割り振りを示します。ここでの鍵は、割り振りパラメーター (AT) です。AT(\*SIGNON) は、これらの装置の割り振りを試みて、可能な場合はサインオン表示画面を表示するよう、サブシステムに指示します。AT(\*ENTER) は、装置を割り振らず、サインオン表示画面を表示しないよう、サブシステムに指示します。ただし、AT(\*ENTER) では、指定されたワークステーション装置を、ジョブ転送 (TFRJOB) コマンドを使ってそのサブシステムに転送することができます。

どのサブシステムがどの装置を割り振るかを決定する必要があります。さらに、あるサブシステムから別のサブシステムへの TFRJOB の使用を許可する必要があるかどうかを決定します。

以下に、処理を分割するための手法をいくつか例示します。これらの例では、サブシステムによって割り振りを試行する装置の指定にフォーカスしています (この後の次のステップは、サブシステムが割り振らない装置のワークステーション項目を設定する方法を示しています)。

- 最初の例では、システムにアクセスする主な方法として Telnet またはパススルーのいずれかを使用していること、および、システム・デフォルトの装置命名規則が使われることを前提としています。このシナリオでは、すべての装置が QPADEVxxxx 命名規則を使用します。この例では、単純に、装置名に基づいて装置をサブシステムに割り振ります。

QPADEVxxxx 命名規則では、装置名の最後の 4 文字として数字 (0 から 9) と文字 (B から Z の子音を含み、母音 A、E、I、O、U、Y は含まない) が使用されるため、0001 から 00B0 で 300 個の異なる装置名を指定できます。3 つのサブシステムに 900 人のユーザーを分割すること、また、このシナリオでは、ユーザーが実行するサブシステムに関する優先度はないと仮定します。

自動的に作成されたすべての装置に QPADEVxxxx という名前が付けられるため、処理を分割する簡単な方法の 1 つとしては、QPADEV0001 から QPADEV00B0 を最初のサブシステム、QPADEV00B1 から QPADEV00M0 を 2 番目のサブシステム、QPADEV00M1 から QPADEV00Z0 を 3 番目のサブシステムに移動する、という方法があります。この単純な方法は問題なく機能しますが、システムは常に、QPADEV0001 から上位の番号の装置を選択します。つまり、最初の 300 人のユーザーはサブシステム INTER1、次の 300 人は INTER2、最後の 300 人は INTER3 に移動されます。しかし、多数のユーザー全員が同時にサインオンしようとするシナリオの場合、処理はサブシステム間で均等に分散されません。

複数のサブシステム間で適切にワークロードを分散させるためには、以下の例で示すように装置をサブシステムに割り当てることができます。この方法では複数のサブシステム間で処理が適切に分散されますが、ワークステーション項目の定義がより複雑になります。

- INTER1 は以下のように割り振ります。

```
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV000*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV003*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV006*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV009*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV00D*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV00H*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV00L*) AT(*SIGNON)
```

```
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV00P*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV00S*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER1) WRKSTN(QPADEV00W*) AT(*SIGNON)
```

- INTER2 は以下のように割り振ります。

```
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV001*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV004*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV007*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV00B*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV00F*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV00J*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV00M*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV00Q*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV00T*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER2) WRKSTN(QPADEV00X*) AT(*SIGNON)
```

- INTER3 は以下のように割り振ります。

```
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV002*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV005*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV008*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV00C*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV00G*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV00K*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV00N*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV00R*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV00V*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/INTER3) WRKSTN(QPADEV00Z*) AT(*SIGNON)
```

- b. 2 番目の例では、地域によって業務が異なると仮定し、各地域のユーザーがそれぞれ別の装置名を使用して、各地域ごとに定義された個別のサブシステムで実行するように、サブシステムと装置の命名規則が設定されていると仮定しています。例えば、東海岸のユーザーはすべて、EAST で始まる名前の装置を使用し、サブシステム EAST で実行します。西海岸のユーザーはすべて、WEST で始まる名前の装置を使用し、サブシステム WEST で実行します。同様に、中部のユーザーはすべて、CENTRAL で始まる名前の装置を使用し、サブシステム CENTRAL で実行します。このシナリオでは、それぞれのサブシステムがその地域的エリアの装置を割り振ります。

```
ADDWSE SBSDB(SBSLIB/EAST) WRKSTN(EAST*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/CENTRAL) WRKSTN(CENTRAL*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/WEST) WRKSTN(WEST*) AT(*SIGNON)
```

- c. 3 番目の例では、サブシステムと装置の命名規則が、ユーザーの行う処理のタイプに基づいていると仮定しています。すべてのプログラマーの装置には PGMR を含む名前が付いており、PGMR サブシステムで実行します。すべての受注処理スタッフの装置には、ORDERENT を含む名前が付いており、ORDERENT サブシステムで実行します。その他のすべてのユーザーは、システム・デフォルトの命名規則 QPADEVxxxx を使用し、IBM 提供のサブシステム QINTER で実行します。

```
ADDWSE SBSDB(SBSLIB/PGMR) WRKSTN(PGMR*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/ORDERENT) WRKSTN(ORDERENT*) AT(*SIGNON)
ADDWSE SBSDB(QGPL/QINTER) WRKSTN(QPADEV*) AT(*SIGNON)
```

7. 独自のサブシステムのセットを使い始めると、QINTER を使用する必要がなくなる場合があります。ただし、引き続き QINTER を使用する理由がある場合は、QINTER が、他のサブシステムの下で実行させたいワークステーションを割り振らないように設定する必要があります。これを行うためには、以下の 2 つの方法を使用できます。

- a. \*ALL ワークステーション項目を QINTER から除去し、次に、QINTER で割り振りたい装置を指定する特定のワークステーション項目を追加する。

ワークステーション・タイプ項目 \*ALL を除去すると、QINTER はすべてのワークステーションの割り振りを試行できなくなります。

```
RMVWSE SBSDB(QGPL/QINTER) WRKSTNTYPE(*ALL)
```

DSP\* という名前の装置のワークステーション項目を追加して、平衡型接続のすべてのディスプレイ装置が引き続き QINTER に割り振られるようにします。この例では、平衡型接続のディスプレイ装置は引き続き QINTER で実行され、また、QINTER はその他の装置の割り振りを試行しません。

```
ADDWSE SBS(D(SBSLIB/QINTER) WRKSTN(DSP*))
```

AT(\*ENTER) キーワードを含むワークステーション項目の使用が回避されるため、これは最良の解決策です。AT(\*ENTER) は装置でサインオン表示画面を表示しないようにサブシステムに指示しますが、その装置上のユーザーは、ジョブ転送 (TFRJOB) コマンドを使ってそのサブシステムでジョブを実行できます。TFRJOB の必要がない場合、最良の解決策は、AT(\*ENTER) を含むワークステーション項目を使用しないことです。さらに、AT(\*ENTER) を含むワークステーション項目を使用すると、交換回線の切断を使用する交換回線装置に対するシステムのオーバーヘッドが増すことがあります。

- b. ワークステーション項目を追加して、他のサブシステムに割り当てられている装置を割り振らないよう、QINTER に指示する。ただし、QINTER は、サブシステムに割り振られていないその他の装置を、引き続き割り振ります。この方法では、ワークステーション・タイプ項目 \*ALL が QINTER サブシステム内に保持され、別のサブシステムに割り振られている装置に、AT パラメーターを持つワークステーション名項目が追加されます。

上記のステップ 6 では、処理を再分割する方法のさまざまな例を示すため、カスタマイズしたサブシステムにワークステーション名項目を追加しました。3 つの例がありましたが、以下に、特定のサブシステムに割り振られた装置を除き、装置が QINTER によって割り振られるようにするために、QINTER に追加する必要があるワークステーション名項目を示します。

- 1) 最初の例の場合は、次のワークステーション項目が、いかなる QPADEVxxxx 装置も割り振らないよう QINTER に通知します。

```
ADDWSE SBS(D(QGPL/QINTER) WRKSTN(QPADEV*)) AT(*ENTER)
```

- 2) 同様に、上記の 2 番目の例の場合は、次のコマンドを使用します。

```
ADDWSE SBS(D(QGPL/QINTER) WRKSTN(EAST) AT(*ENTER)
ADDWSE SBS(D(QGPL/QINTER) WRKSTN(WEST) AT(*ENTER)
ADDWSE SBS(D(QGPL/QINTER) WRKSTN(CENTRAL) AT(*ENTER)
```

- 3) 3 番目の例の場合は、以下のとおりです。

```
ADDWSE SBS(D(QGPL/QINTER) WRKSTN(PGMR) AT(*ENTER)
ADDWSE SBS(D(QGPL/QINTER) WRKSTN(ORDERENT) AT(*ENTER)
```

8. QINTER に関する最後かつ重要な考慮事項として、コンソール用のワークステーション・タイプ項目 \*CONS があります。コンソールでは、誰かが誤ってサインオンしないようにしてください。

システムには、コンソール用のワークステーション項目 AT(\*SIGNON) (\*CONS ワークステーション・タイプ項目) が含まれた制御サブシステムが標準で用意されています。QINTER には、コンソール用の AT(\*ENTER) ワークステーション・タイプ項目があります。

常に制御サブシステムでコンソールを実行し、コンソール・ジョブを他の対話式サブシステムに転送しないようにすることをお勧めします。これにより、コンソールにいるユーザーが、意図せずに自分のジョブを終了してしまうことを防止できます。例えば、コンソールにいるユーザーが自分のジョブを INTER1 に転送した後にそれを忘れてしまい、後でシステム終了 (ENDSYS) コマンドを実行してバックアップ処理の準備を開始しようとする、オペレーターが意図していない場合でも、コンソール・ジョブも終了してしまいます。こうしたことを防ぐために、コンソール用のワークステーション項目をカスタムの対話式サブシステムに追加しないようにします。

独自の対話式サブシステムを構成する場合の開始点として、48 ページの『シナリオの詳細: CL プログラムの例』が提供されています。

サブシステム記述の作成が完了したら、サブシステム記述表示 (DSPSBSD) コマンドを使ってサブシステムのさまざまな属性を表示し、正しく設定されているかどうかを検査します。

サブシステム開始 (STRSBS) コマンドを使ってサブシステムを開始します。さまざまな名前を持つ装置を使ってシステムにサインオンし、ジョブが適切なサブシステムで開始することを検査します。この検査は、サブシステム・ジョブ処理 (WRKSBSJOB) コマンドを使って実行できます。

システムが制限状態になったり再始動されるたびにサブシステムを手動で始動しなくてもすむよう、システム始動プログラムを変更して、これらのサブシステムが自動的に開始されるようにすることをお勧めします。IPL 始動プログラムの変更方法についての詳細は、『IPL 始動プログラムを変更する』を参照してください。

## シナリオの詳細: CL プログラムの例

以下のサンプル・ソース・コードは、カスタマイズ・サブシステムをセットアップするための開始点として使用できます。

**注:** 重要なリーガル情報 (法律的事柄に関する情報) については、『コード例の特記事項』をお読みください。

ソースを QGPL/QCLSRC 内に配置し、以下のコマンドを実行してコマンドと CPP を作成します。

```
CRTCMD CMD(QGPL/CRTMLTSBS) PGM(QGPL/QCRTMLTSBS) SRCFILE(QGPL/QCLSRC)
CRTCLPGM PGM(QGPL/QCRTMLTSBS) SRCFILE(QGPL/QCLSRC) LOG(*NO)
```

プログラムを実行するには、コマンド CRTMLTSBS を入力し、F4 (プロンプト) キーを押します。以下のプロンプトが表示されます。

```
-----
                Create multiple subsystems (CRTMLTSBS)
-----
```

選択項目を入力して、実行キーを押してください。

```
Subsystem name prefix . . . . . QINTER____ QINTER, name
Library name . . . . . *CURLIB____ *CURLIB, name
Number of subsystems . . . . . 1____ 1-999
Devices per subsystem . . . . . 200____ 1-500
-----
```

このプログラムが実行する内容は、以下のとおりです。

- aaaaaaaNNN という命名規則を使って、最大 999 のサブシステムを作成します。
  - aaaaaaa は、ユーザーが指定する 7 文字までの名前です。
  - NNN は、作成されるサブシステムの 001 から 999 の連番です。
  - プログラムはサブシステムに対して、作成された順に番号を付けます。
- 装置名項目は、QPADEV.... 命名規則に基づいてサブシステムに追加されます。プログラムは、サブシステムが作成されたときに、要求された数の装置名を各サブシステムに追加します。装置名は、サブシステムに“ラウンドロビン”スタイルで追加されます。つまり、サブシステム 001 は QPADEV0001、サブシステム 002 は QPADEV0002 のようになります。
- CRTMLTSBS コマンドで指定したライブラリーは、このコマンドを実行する前に存在している必要があります。
- コマンドの実行時にサブシステム記述が存在している場合、そのサブシステムは、装置名配列が引き続き大きくなる場合でも、変更されることはありません。つまり、サブシステムが存在し、要求が各サブシステムに 200 の装置を追加するということであっても、その 200 の装置名 (アルファベット順) が再

度使用されることはありません。このような状況では、プログラムは完了するまで実行され、「コマンドの処理中に 1 つ以上のエラーが発生しました (One or more errors occurred during processing of command)」という CPF2227 メッセージが返されます。実際の原因についてはジョブ・ログをチェックしてください。

- 各サブシステムには、それぞれ固有のジョブ待ち行列項目、ジョブ待ち行列名、およびクラスがあります。
- 各サブシステムには、IBM 提供の QINTER サブシステム記述と同じ、以下のような基本的な構成があります。

```
-----
サブシステム記述 . . . . . : QINTER001
ライブラリー . . . . . : SBSLIB
サブシステム内の最大ジョブ数 . . . . . : *NOMAX
サインオン表示ファイル . . . . . : QDSIGNON
ライブラリー . . . . . : QSYS
システム・ライブラリー・リスト項目 . . . : *NONE
-----
```

```
-----
プール          記憶域          活動
ID             サイズ (K)       レベル
1              *BASE
2              *INTERACT
-----
```

オプションを入力して、実行キーを押してください。  
5= ワークステーション名の明細表示

```
-----
OPT  名前          OPT  名前          OPT  名前          OPT  名前
QPADEV0000
QPADEV0001
QPADEV0002 *** 装置数は要求される数によって異なります。
QPADEV0003
QPADEV0004
QPADEV.....
-----
```

```
-----
SEQ  ジョブ          最大          ----- 優先順位による最大数 -----
NBR  待ち行列 ライブラリー 活動          1  2  3  4  5  6  7  8  9
10  QINTER001  SBSLIB        *NOMAX      *  *  *  *  *  *  *  *  *
-----
```

経路指定項目の表示

システム : TESTSYS

サブシステム記述 : QINTER001 状況 : 活動

オプションを入力して、実行キーを押してください。  
5= 明細の表示

```
-----
OPT  順序          プログラム          ライブラリー          比較値          開始
番号          番号          ライブラリー          値          桁
10  10  QCMD          QSYS          'QCMDI'          1
20  20  QCMD          QSYS          'QS36MRT'        1
40  40  QARDRIVE       QSYS          '525XTEST'       1
700 700  QCL           QSYS          'QCMD38'         1
-----
```

```
-----
経路指定項目の順序番号 . . . . . : 10
プログラム . . . . . : QCMD
ライブラリー . . . . . : QSYS
-----
```

```

クラス . . . . . : QINTER001
ライブラリー . . . . . : SBSLIB
活動経路指定ステップの最大数 . . . . . : *NOMAX
プール識別コード . . . . . : 2
比較値 . . . . . : 'QCMDI'

比較開始桁 . . . . . : 1

```

```

-----
経路指定項目の順序番号 . . . . . : 20
プログラム . . . . . : QCMD
ライブラリー . . . . . : QSYS
クラス . . . . . : QINTER001
ライブラリー . . . . . : SBSLIB
活動経路指定ステップの最大数 . . . . . : *NOMAX
プール識別コード . . . . . : 2
比較値 . . . . . : 'QS36MRT'

比較開始桁 . . . . . : 1

```

```

-----
経路指定項目の順序番号 . . . . . : 40
プログラム . . . . . : QARDRIVE
ライブラリー . . . . . : QSYS
クラス . . . . . : QINTER001
ライブラリー . . . . . : SBSLIB
活動経路指定ステップの最大数 . . . . . : *NOMAX
プール識別コード . . . . . : 2
比較値 . . . . . : '525XTEST'

比較開始桁 . . . . . : 1

```

```

-----
経路指定項目の順序番号 . . . . . : 700
プログラム . . . . . : QCL
ライブラリー . . . . . : QSYS
クラス . . . . . : QINTER001
ライブラリー . . . . . : SBSLIB
活動経路指定ステップの最大数 . . . . . : *NOMAX
プール識別コード . . . . . : 2
比較値 . . . . . : 'QCMD38'

比較開始桁 . . . . . : 1

```

```

-----
経路指定項目の順序番号 . . . . . : 9999
プログラム . . . . . : QCMD
ライブラリー . . . . . : QSYS
クラス . . . . . : QINTER001
ライブラリー . . . . . : SBSLIB
活動経路指定ステップの最大数 . . . . . : *NOMAX
プール識別コード . . . . . : 2
比較値 . . . . . : *ANY

比較開始桁 . . . . . :

```

このコマンドのソース・コードを以下に示します。

```

/*****
/* 5722-SS1 (C) COPYRIGHT IBM CORP. 2003, 2003 */
/* LICENSED MATERIAL - PROGRAM PROPERTY OF IBM */

```



```

/*                                                                    */
/*****                                                                    */
/* Create this command using the following command:                    */
/*                                                                    */
/* Assuming the source is in file QGPL/QCLSRC                          */
/*                                                                    */
/* CRTCMD CMD(QGPL/CRTMLTSBS) PGM(QGPL/QCRTMLTSBS) SRCFILE(QGPL/QCLSRC) */
/*                                                                    */
/*****                                                                    */
/*                                                                    */
      CMD          PROMPT('Create multiple subsystems')
      PARM         KWD(SBSNAM) TYPE(*CHAR) LEN(7) DFT(QINTER) +
                  CHOICE('QINTER, name') PROMPT('Subsystem +
                  name prefix')
      PARM         KWD(SBSLIB) TYPE(*CHAR) LEN(10) DFT(*CURLIB) +
                  CHOICE(*CURLIB, name') PROMPT('Library +
                  name')
      PARM         KWD(NBRSBS) TYPE(*DEC) LEN(3) DFT(1) RANGE(1 +
                  999) CHOICE(*VALUES) PROMPT('Number of +
                  subsystems')
      PARM         KWD(NBRDEV) TYPE(*DEC) LEN(3) DFT(200) +
                  RANGE(1 300) CHOICE(*VALUES) +
                  PROMPT('Devices per subsystem')
/*                                                                    */
/*                                                                    */

```

**CL プログラムのソース・コードを以下に示します。**

```

/*****                                                                    */
/* Assuming the source member is in QGPL/QCLSRC file.                  */
/* Create this program with the following command:                      */
/*                                                                    */
/* CRTCLPGM PGM(QGPL/QCRTMLTSBS) SRCFILE(QGPL/QCLSRC) LOG(*NO)       */
/*                                                                    */
/* This program must be invoked using the CRTMLTSBS command.          */
/*                                                                    */
/*****                                                                    */
/*                                                                    */
      PGM          PARM(&SBSNAM &SBSLIB &NBRSBS &NBRDEV)
/*                                                                    */
/*****                                                                    */
/* 5722-SS1 (C) COPYRIGHT IBM CORP. 2003, 2003                       */
/* LICENSED MATERIAL - PROGRAM PROPERTY OF IBM                        */
/*****                                                                    */
/*                                                                    */
      QSYS/DCL    VAR(&CPYR) TYPE(*CHAR) LEN(230) +
                  VALUE('LICENSED MATERIALS PROPERTY OF IBM +
                  5722SS1 (C) COPYRIGHT IBM CORP. 2003, +
                  2003 ALL RIGHTS RESERVED. US GOVERNMENT +
                  USERS RESTRICTED RIGHTS - USE, DUPLICATION +
                  OR DISCLOSURE RESTRICTED BY GSA ADP +
                  SCHEDULE CONTRACT WITH IBM CORP.')
```

```

/*                                                                    */
/*****                                                                    */
/*                                                                    */
      DCL         VAR(&SBSNAM) TYPE(*CHAR) LEN(7)
      DCL         VAR(&SBSLIB) TYPE(*CHAR) LEN(10)
      DCL         VAR(&NBRSBS) TYPE(*DEC) LEN(3 0)
      DCL         VAR(&NBRDEV) TYPE(*DEC) LEN(3 0)
      DCL         VAR(&SBSCNT) TYPE(*DEC) LEN(3 0)
      DCL         VAR(&SBSCNTC) TYPE(*CHAR) LEN(3)
      DCL         VAR(&DEVCNT) TYPE(*DEC) LEN(10 0)
      DCL         VAR(&SBSNAME) TYPE(*CHAR) LEN(10)
      DCL         VAR(&DEVNAME) TYPE(*CHAR) LEN(10)
      DCL         VAR(&POS1) TYPE(*DEC) LEN(2 0) VALUE(1)
      DCL         VAR(&POS2) TYPE(*DEC) LEN(2 0) VALUE(1)
      DCL         VAR(&POS3) TYPE(*DEC) LEN(2 0) VALUE(1)
      DCL         VAR(&POS4) TYPE(*DEC) LEN(2 0) VALUE(1)

```

```

DCL      VAR(&TMP4) TYPE(*CHAR) LEN(4)
DCL      VAR(&MSGID) TYPE(*CHAR) LEN(7)
DCL      VAR(&MSGDTA) TYPE(*CHAR) LEN(512)
DCL      VAR(&KEYVAR) TYPE(*CHAR) LEN(4)
DCL      VAR(&MSGFILE) TYPE(*CHAR) LEN(10)
DCL      VAR(&MSGLIB) TYPE(*CHAR) LEN(10)
DCL      VAR(&ERR) TYPE(*DEC) LEN(1 0) VALUE(0)
DCL      VAR(&ARY) TYPE(*CHAR) LEN(31) +
          VALUE('0123456789BCDFGHJKLMNPQRSTVWXYZ')

/*
/*****
/* Monitor for any unexpected condition then resignal the exception to
/* the caller.
/*
/*****
/*
      MONMSG      MSGID(CPF0000) EXEC(GOTO CMDLBL(ERRORXIT))

/*
/*****
/* Insert the copyright.
/*
/*****
/*
      CHGVAR      VAR(&CPYR) VALUE(&CPYR)

/*
/*****
/* Insert the copyright.
/*
/*****
/*
      CHGVAR      VAR(&DEVCNT) VALUE(&NBRDEV * &SBSCNT)

/*
/*****
/* Set the library name if necessary.
/*
/*****
/*
      IF          COND(&SBSLIB *EQ 'CURLIB') THEN(DO)
RTVJOBA      CURLIB(&SBSLIB)
ENDDO
      IF          COND(&SBSLIB *EQ 'NONE') THEN(DO)
CHGVAR      VAR(&SBSLIB) VALUE('QGPL')
ENDDO

/*
/*****
/* Make sure the specified library exists.
/*
/*****
/*
      CHKOBJ      OBJ(QSYS/&SBSLIB) OBJTYPE(*LIB)

/*
/*****
/* Put out a status message.
/*
/*****
/*
      CHGVAR      VAR(&MSGDTA) VALUE('Checking for existing +
          subsystems')

/*
      SNDPGMMSG  MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
          MSGDTA(&MSGDTA) TOPGMQ(*EXT) MSGTYPE(*STATUS)

/*
/*****
/* Make sure none of the subsystems exist.
/*
/*****
/*
CHKFORSBS:

/*
      CHGVAR      VAR(&SBSCNT) VALUE(&SBSCNT + 1)

/*
      IF          COND(&SBSCNT *GT &NBRSBS) THEN(DO)
GOTO          CMDLBL(NOSBS)
ENDDO

/*

```

```

        CHGVAR      VAR(&SBSCNTC) VALUE(&SBSCNT)
        CHGVAR      VAR(&SBSNAME) VALUE(&SBSNAM *TCAT &SBSCNTC)
/*
/*****
/* Check for the existence of the subsystem.
/*****
/*
        CHKOBJ      OBJ(&SBSLIB/&SBSNAME) OBJTYPE(*SBSD)
        MONMSG      MSGID(CPF9801) EXEC(DO)
        RCVMSG      MSGTYPE(*EXCP) RMV(*YES)
        GOTO        CMDLBL(CHKFORSBS)
        ENDDO
/*
/*****
/* If a subsystem already exists, signal the error and return.
/*****
/*
        SNDPGMMSG   MSGID(CPD1411) MSGF(QSYS/QCPFMSG) +
                   MSGDTA(&SBSNAME *CAT &SBSLIB) +
                   TOPGMQ(*SAME) MSGTYPE(*INFO)
/*
        SNDPGMMSG   MSGID(CPF1696) MSGF(QSYS/QCPFMSG) +
                   MSGDTA(&SBSNAME) +
                   TOPGMQ(*SAME) MSGTYPE(*DIAG)
/*
        CHGVAR      VAR(&ERR) VALUE(1)
/*
        GOTO        CMDLBL(END)
NOSBS:
        CHGVAR      VAR(&SBSCNT) VALUE(0)
/*
/*****
/* Main loop of the program.
/*****
/*
NEXTSBS:
        CHGVAR      VAR(&SBSCNT) VALUE(&SBSCNT + 1)
        CHGVAR      VAR(&SBSCNTC) VALUE(&SBSCNT)
        CHGVAR      VAR(&SBSNAME) VALUE(&SBSNAM *TCAT &SBSCNTC)
/*
/*****
/* Put out a status message.
/*****
/*
        CHGVAR      VAR(&MSGDTA) VALUE('Creating subsystem' +
                                     *BCAT &SBSNAME)
/*
        SNDPGMMSG   MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
                   MSGDTA(&MSGDTA) TOPGMQ(*EXT) MSGTYPE(*STATUS)
/*
/*****
/* If the subsystem doesn't exist, create it and the associated entries.
/*****
/*
/* Create the subsystem description.
/*
        CRTSBS      SBSDB(&SBSLIB/&SBSNAME) POOLS((1 *BASE) (2 +
                                     *INTERACT)) TEXT('Created by the +
                                     CRTMLTSBS command')
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
/*
/* Create a jobq.
/*
        CHKOBJ      OBJ(&SBSLIB/&SBSNAME) OBJTYPE(*JOBQ)
        MONMSG      MSGID(CPF9801) EXEC(DO)
        RCVMSG      MSGTYPE(*EXCP) RMV(*YES)

```

```

        CRTJOBQ      JOBQ(&SBSLIB/&SBSNAME) TEXT('Created by the +
                    CRTMLTSBS command')
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
        GOTO        CMDLBL(ADDJOBQE)
        ENDDO

/*                                                    */
        CHGVAR      VAR(&ERR) VALUE(1)
/*                                                    */
        SNDPGMMSG   MSGID(CPF3323) MSGF(QSYS/QCPFMSG) +
                    MSGDTA(&SBSNAME *CAT &SBSLIB) +
                    TOPGMQ(*SAME) MSGTYPE(*DIAG)
/*                                                    */
/* Add the jobq entry.
/*                                                    */
/* AddJOBQE:
        ADDJOBQE    SBSDB(&SBSLIB/&SBSNAME) +
                    JOBQ(&SBSLIB/&SBSNAME) MAXACT(*NOMAX)
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
/*                                                    */
/* Create the class
/*                                                    */
/*
        CHKOBJ      OBJ(&SBSLIB/&SBSNAME) OBJTYPE(*CLS)
        MONMSG      MSGID(CPF9801) EXEC(DO)
        RCVMSG      MSGTYPE(*EXCP) RMV(*YES)
        CRTCLS      CLS(&SBSLIB/&SBSNAME) RUNPTY(20) PURGE(*YES) +
                    DFTWAIT(30) CPUTIME(*NOMAX) +
                    MAXTMPSTG(*NOMAX) MAXTHD(*NOMAX) +
                    TEXT('Created by the CRTMLTSBS command')
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
        GOTO        CMDLBL(ADDRTGE)
        ENDDO

/*                                                    */
        CHGVAR      VAR(&ERR) VALUE(1)
        SNDPGMMSG   MSGID(CPF1064) MSGF(QSYS/QCPFMSG) +
                    MSGDTA(&SBSNAME *CAT &SBSLIB) +
                    TOPGMQ(*SAME) MSGTYPE(*DIAG)
/*                                                    */
/* Add the standard routing entries.
/*                                                    */
/* AddRTGE:
        ADDRTGE     SBSDB(&SBSLIB/&SBSNAME) SEQNBR(10) +
                    CMPVAL('QCMDI' 1) PGM(QSYS/QCMD) +
                    CLS(&SBSLIB/&SBSNAME) POOLID(2)
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
/*                                                    */
        ADDRTGE     SBSDB(&SBSLIB/&SBSNAME) SEQNBR(20) +
                    CMPVAL('QS36MRT' 1) PGM(QSYS/QCMD) +
                    CLS(&SBSLIB/&SBSNAME) POOLID(2)
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
/*                                                    */
        ADDRTGE     SBSDB(&SBSLIB/&SBSNAME) SEQNBR(40) +
                    CMPVAL('525XTEST' 1) PGM(QSYS/QARDRIVE) +
                    CLS(&SBSLIB/&SBSNAME) POOLID(2)
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
/*                                                    */
        ADDRTGE     SBSDB(&SBSLIB/&SBSNAME) SEQNBR(700) +
                    CMPVAL('QCMD38' 1) PGM(QSYS/QCL) +
                    CLS(&SBSLIB/&SBSNAME) POOLID(2)
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
/*                                                    */
        ADDRTGE     SBSDB(&SBSLIB/&SBSNAME) SEQNBR(9999) +
                    CMPVAL(*ANY) PGM(QSYS/QCMD) +
                    CLS(&SBSLIB/&SBSNAME) POOLID(2)
        RCVMSG      MSGTYPE(*COMP) RMV(*YES)
/*                                                    */
/* Go create the next subsystem.
/*                                                    */
/*

```

```

                IF          COND(&SBSCNT *LT &NBRSBS) THEN(DO)
                GOTO      CMDLBL(NEXTSBS)
                ENDDO

/*
/*****
/* Put out a status message.
/*****
/*
                CHGVAR      VAR(&MSGDTA) VALUE('Adding device names to +
                                the' *BCAT &SBSNAM *TCAT '... subsystems')
/*
                SNDPGMMSG  MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
                                MSGDTA(&MSGDTA) TOPGMQ(*EXT) MSGTYPE(*STATUS)
/*
/*****
/* Generate a device name.
/*****
/*
NEXTDEV:
                CHGVAR      VAR(&POS4) VALUE(&POS4 + 1)
                IF          COND(&POS4 *GT 31) THEN(DO)
                CHGVAR      VAR(&POS4) VALUE(1)
                CHGVAR      VAR(&POS3) VALUE(&POS3 + 1)
                ENDDO
                IF          COND(&POS3 *GT 31) THEN(DO)
                CHGVAR      VAR(&POS3) VALUE(1)
                CHGVAR      VAR(&POS2) VALUE(&POS2 + 1)
                ENDDO
                IF          COND(&POS2 *GT 31) THEN(DO)
                CHGVAR      VAR(&POS2) VALUE(1)
                CHGVAR      VAR(&POS1) VALUE(&POS1 + 1)
                ENDDO

/*
/* Check for using up all device names.
/*
                IF          COND(&POS1 *GT 31) THEN(DO)
/*
/* We ran out of device names.
/*
                CHGVAR      VAR(&MSGDTA) VALUE('We used all possible +
                                device names')
                SNDPGMMSG  MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
                                MSGDTA(&MSGDTA) TOPGMQ(*PRV) MSGTYPE(*ESCAPE)
                RETURN
                ENDDO

/*
                CHGVAR      VAR(%SST(&TMP4 1 1)) VALUE(%SST(&ARY &POS1 1))
                CHGVAR      VAR(%SST(&TMP4 2 1)) VALUE(%SST(&ARY &POS2 1))
                CHGVAR      VAR(%SST(&TMP4 3 1)) VALUE(%SST(&ARY &POS3 1))
                CHGVAR      VAR(%SST(&TMP4 4 1)) VALUE(%SST(&ARY &POS4 1))
                CHGVAR      VAR(&DEVNAME) VALUE('QPADEV' *TCAT &TMP4)
/*
/* Set the subsystem name.
/*
                IF          COND(&SBSCNT *GE &NBRSBS) THEN(DO)
                CHGVAR      VAR(&SBSCNT) VALUE(0)
                ENDDO
                CHGVAR      VAR(&SBSCNT) VALUE(&SBSCNT + 1)
                CHGVAR      VAR(&SBSCNTC) VALUE(&SBSCNT)
                CHGVAR      VAR(&SBSNAME) VALUE(&SBSNAM *TCAT &SBSCNTC)
/*
/* Add the device name to the subsystem.
/*
                ADDWSE      SBSDB(&SBSLIB/&SBSNAME) WRKSTN(&DEVNAME) +
                                JOBD(*USRPRF) MAXACT(*NOMAX) AT(*SIGNON)

                RCVMSG      MSGTYPE(*LAST) RMV(*NO) MSGID(&MSGID)

```

```

        IF          COND(&MSGID *EQ 'CPC1602') THEN(DO)
        RCVMSG     MSGTYPE(*LAST) RMV(*YES)
        ENDDO
        ELSE
        CMD(DO)
        IF          COND(&MSGID *EQ 'CPF1698') THEN(DO)
        RCVMSG     MSGTYPE(*LAST) RMV(*YES)
        ENDDO
        RCVMSG     MSGTYPE(*LAST) RMV(*NO) MSGID(&MSGID)
        IF          COND(&MSGID *EQ 'CPD1431') THEN(DO)
        RCVMSG     MSGTYPE(*LAST) RMV(*YES)
        ENDDO
        ENDDO

/*                                                    */
/*                                                    */
/* Keep a count of the number of devices in a subsystem. */
/*                                                    */
        CHGVAR     VAR(&DEVCNT) VALUE(&DEVCNT + 1)
/*                                                    */
/*                                                    */
/* Have we reached the number of devices in a subsystem? */
/*                                                    */
        IF          COND(&DEVCNT *GE (&NBRDEV * &NBR SBS)) THEN(DO)
        GOTO       CMDLBL(END)
        ENDDO
/*                                                    */
/* Go create another device name. */
/*                                                    */
        GOTO       CMDLBL(NEXTDEV)
/*                                                    */
/******
/*                                                    */
END:
/*                                                    */

        IF          COND(&ERR *EQ 1) THEN(DO)
        SNDPGMMSG  MSGID(CPF2227) MSGF(QSYS/QCPFMSG) +
                   TOPGMQ(*PRV) MSGTYPE(*COMP)
        ENDDO
        ELSE
        CMD(DO)
        CHGVAR     VAR(&MSGDTA) VALUE('CRTMLTSBS command +
                                     completed..')
        SNDPGMMSG  MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
                   MSGDTA(&MSGDTA) TOPGMQ(*PRV) MSGTYPE(*COMP)
        ENDDO
        RETURN
/*                                                    */
/******
/*                                                    */
ERRorexIT:
/*                                                    */
/* Receive an exception message and save the variables. */
/*                                                    */
        RCVMSG     PGMQ(*SAME) MSGTYPE(*EXCP) RMV(*YES) +
                   KEYVAR(&KEYVAR) MSGDTA(&MSGDTA) +
                   MSGID(&MSGID) MSGF(&MSGFILE) MSGFLIB(&MSGLIB)
        MONMSG     MSGID(CPF0000) CMPDTA(*NONE) EXEC(RETURN)
/*                                                    */
/* Resignal the message. */
/*                                                    */
        SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGLIB/&MSGFILE) +
                   MSGDTA(&MSGDTA) MSGTYPE(*ESCAPE) +
                   KEYVAR(&KEYVAR)
        MONMSG     MSGID(CPF0000) CMPDTA(*NONE) EXEC(RETURN)
/*                                                    */
/******
/*                                                    */
        RETURN
/*                                                    */

```

```
/*
ENDPGM
*/
```

## 対話式ユーザーおよび装置の管理

このセクションでは、装置名の割り当て、非アクティブ装置のタイムアウト、装置回復処置など、OS/400<sup>(R)</sup> での対話式ジョブ管理に関連するいくつかの追加のトピックについて説明します。

### 特定のサブシステムへのユーザーの経路指定

対話式セッションの場合、OS/400 は装置記述によってジョブをサブシステムに割り当てます。どのユーザーがどのディスプレイ装置と関連付けられているかを識別できるため、この方法は、特にユーザーが平衡型接続の端末を使用してシステムに接続していたときには、問題なく機能しました。ただし、対話式の処理が平衡型端末から仮想ディスプレイ装置に移行したことで、デバイス記述からユーザー・プロファイルへの関連付けが極めて難しくなりつつあります。

装置記述ではなくユーザー・プロファイルに基づいてユーザーをサブシステムへ経路指定できるようにしなければならない、という場合が数多くあります。しかしながら、OS/400 によって提供されるサブシステム構成では、この機能は提供されません。ユーザー・プロファイルに基づいて処理をサブシステムに経路指定するためには、最初にすべてのユーザーを単一のサブシステム (通常は QINTER) を介してシステムに送り、ユーザー・プロファイル内でそのユーザーの最初のプログラムに、ユーザーのジョブを目的のサブシステムに転送するジョブ転送コマンドを指定する、という方法が典型的な方法として使用されます。通常、この方法は問題なく機能しますが、以下のような重要な考慮事項があります。

- ジョブ転送 (TFRJOB) コマンドを使用する際に、システムのオーバーヘッドが発生します。ジョブを単にサブシステムから別のサブシステムに移動するのではなく、ターゲット・サブシステムで新しい経路指定ステップを開始して、その経路指定ステップを現行サブシステムで終了します。(経路指定ステップについての詳細は、「AS/400e シリーズ 実行管理の手引き」を参照してください。) TFRJOB を使わずに目的のサブシステムにユーザーを入れると、システム・リソースをはるかに効率的に使用できます。
- すべての装置が QINTER などの 1 つのサブシステムに割り振られていて、300 を超える多数の装置がある場合は、装置回復処置によってサブシステムがスローダウンする恐れがあります。単一のサブシステムに割り振る装置を一定数にとどめることをお勧めします。

ユーザーを適切なサブシステムに入れるための方法として、ジョブ転送を使用する以外の、より優れた解決策を見つけることを強くお勧めします。次のセクションでは、装置名を割り当てらうえて使用できる技法について説明し、ユーザーと関連付けられた装置名を与えることのできる機能を提供します。このハードルを乗り越えれば、ジョブ転送を使わずに、ワークステーション項目を使って、目的のサブシステムにユーザーを入れることが可能になります。

### 装置名の割り当て

システムには、ディスプレイ・セッションに使用するデフォルトの命名規則があります。現在、5250 セッションへのサインオン方法としては、Telnet の使用が最も一般的です。例えば、iSeries<sup>(TM)</sup> Access を使ってサインオン画面を表示する場合、実際には、基礎となる通信メカニズムとして Telnet を使用する PC5250 を使用しています。Telnet を使用するときには、システムで変更を加えない限り、装置名に対してシステム・デフォルトの命名規則である QPADEV\* が使用されます。これには、以下のようないくつかの大きな欠点があります。

- システムがデフォルトの装置命名規則 (QPADEV\* など) を使って自動的に選択した装置を使用するジョブは、切断することができない。

- すべての装置に QPADEV\* という名前が付けられていると、複数のユーザーの管理や偶発的な問題のデバッグが困難である。
- 処理のタイプや処理を行うユーザーに基づいて処理を再分割することができないため、複数のサブシステムへの装置の分割は、やや無原則なものとなる。
- ワークステーション項目は装置名によって処理を経路指定するが、ユーザーは、ユーザー・プロファイルによって処理を経路指定する必要がある。QPADEV\* と特定ユーザーとの間に関連を作ることはできません。

システム上で変更を加え、独自の装置命名規則を割り当てて管理することによって、システムのデフォルトの動作の欠点を克服することができます。これを行うためには、いくつかの方法があります。次のセクションでは、そのすべての方法について説明します。各アプローチには、それぞれ利点と欠点があります。

1. **Telnet 装置の初期設定出口点と終了出口点。**これらの出口点は、システムにサインオンするクライアントに基づいて装置名を割り当てる機能を提供します。出口点は、クライアント IP アドレスとユーザー・プロファイル名、および追加情報を提供します。これにより、ユーザーは、クライアントに対して使用する必要のある装置記述に対し、独自のクライアント・マッピングを行うことができます。装置初期設定出口点は、サインオン・パネルをバイパスする方法も提供します。これらの出口点についての詳細は、『Telnet 出口点 (Telnet Exit Point)』を参照してください。出口プログラムの例は、『Technical Studio: Telnet Exit Programs』を参照してください。

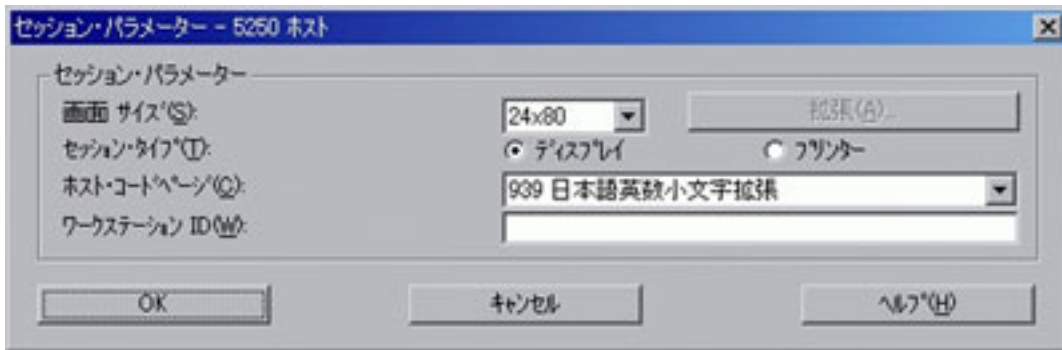
これらの出口点を使って装置命名規則を管理することの主な利点は、すべてのクライアントを iSeries サーバー上で集中管理できるという点です。主な欠点は、プログラミングのスキルが必要であるという点ですが、Technical Studio に、極めて実用的な出口プログラムのソース・コードの例が用意されています。

2. **装置選択出口点。**この出口点を使うと、自動的に作成された仮想装置と仮想コントローラーに対して使用する命名規則の指定や、特定の要求に対して使用する自動作成制限の指定ができます。この出口点を使って、Telnet、5250 ディスプレイ装置のパススルー、および仮想端末 API が使用する、自動作成された装置に対して、さまざまな命名規則を指定することができます。さらに、仮想装置の自動構成 (QAUTOVRT) システム値を、より細かい方法で管理することができます。例えば、ある 1 つの値を Telnet 用に自動作成された装置に許可し、別の値を 5250 ディスプレイ装置のパススルー装置に許可することができます。詳しくは、『装置選択出口点 (Device Selection Exit Point)』を参照してください。

この出口点は、装置に使用するデフォルトの命名規則 (QPADEV\* など) を制御する機能を提供しますが、単独で特定のユーザーに特定の装置を指定することはできません。この出口点は、さまざまなアクセス方法に対してさまざまな命名規則ときめ細かい QAUTOVRT 管理を使用できるため、システムへの接続方法が混在している (Telnet、5250 ディスプレイ装置のパススルー、WebFacing など) 場合に最も便利です。

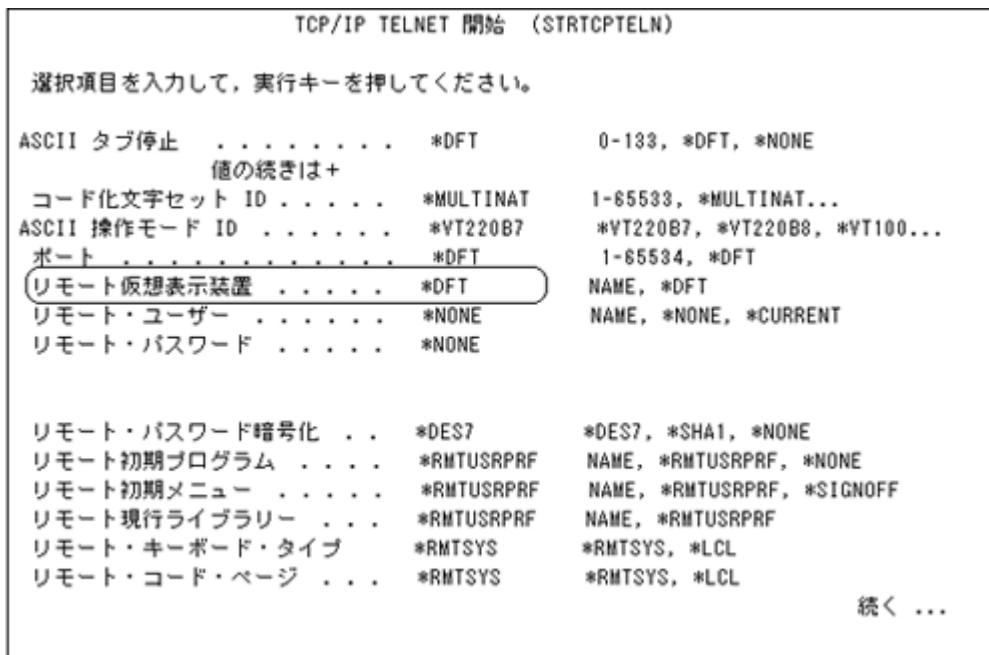
3. **PC5250 (iSeries Access) ワークステーション ID のサポート。**特定のワークステーション名を使って接続するよう、iSeries Access を構成できます。このパネルのヘルプ・ボタンを選択すると、ワークステーション ID を指定するためのさまざまなオプション (指定したものが既に使用中の場合は新しい名前を生成する、など) が表示されます。





この方法の主な欠点は、PC5250 構成設定を、サーバーに接続しているすべてのクライアントでそれぞれ管理する必要があるという点です。

- OS/400 Telnet クライアント。OS/400 Telnet クライアント・コマンド (STRTCPTELN または TELNET) を使用して、サーバー・システムへのサインオンに使用する装置名を指定することができます。



このデフォルトの方法の主な欠点は、STRTCPTELN (TELNET) コマンドを使用するには常に、リモート仮想表示装置の値が正しく指定されていることを確認する必要がある、という点です。この問題を多少でも緩和するため、カスタム版の STRTCPTELN コマンドを作成し、リモート仮想端末表示装置の値を確認してから、IBM<sup>(R)</sup> 提供のコマンドを実行することができます。

- 仮想コントローラーおよび装置は、手動で作成することができます。Telnet 用の仮想装置の作成についての詳細は、iSeries Information Center の『Telnet サーバーの構成』のトピックを参照してください。

この方法では、コントローラーと装置に使用する名前を管理できますが、特定のユーザーに特定の装置をマップする機能は備えられていません。

## 非アクティブの対話式セッションの管理

非アクティブの対話式セッションとは、サインオン状態ではあるものの、端末への I/O を行っておらず、状態が変更されておらず、また、指定された時間フレーム中に CPU 時間を使用していないユーザー・ジョ

ブのことを指します。端末セッションにサインオンしていても不在のままである場合、これらの非アクティブ・セッションは機密漏れ状態であると言えます。幸いにも、OS/400 は、非アクティブな対話式セッションをタイムアウトにする機能を備えています。

非アクティブ・セッションの管理には、以下の 2 つのシステム値を使用します。

- 非アクティブ・ジョブのタイムアウト・インターバル (QINACTITV)。このシステム値は、システムが非アクティブ・セッションの有無をチェックする時間間隔を決定します。
- ジョブがタイムアウトになったとき (QINACTMSGQ)。このシステム値は、非アクティブ・セッションがタイムアウトになった際に行われる処置を指定します。

システムの出荷時には QINACTITV システム値が \*NONE に設定されているため、デフォルトではセッションはタイムアウトになりません。この値を \*NONE から適切な分数に変更する必要があります。最大タイムアウト値は 300 分 (5 時間) です。サーバーの使用に適した時間の値を選択してください。

セッションがいつタイムアウトになるかを理解するには、QINACTITV タイムアウト・インターバルがどのように機能するかを把握する必要があります。この時間間隔が終了すると、すべての対話式サブシステムがそのサブシステムで実行中のジョブをチェックして、その時間フレーム中にアイドル状態であったかどうかを判別します。アイドルになっていた場合は、非アクティブ・ジョブに対して、QINACTMSGQ システム値で定義した処置がとられます。ジョブは、非アクティブ状態になっていたことをシステムが検出するまでに、QINACTITV に指定した値のほぼ 2 倍の時間の間アイドル状態となる可能性があります。これは、非アクティブ・ジョブをチェックするインターバルが終了した直後にジョブが開始した場合に発生します。例えば、QINACTITV が 10 分に設定されていて、9:10 にシステムが非アクティブ・ジョブをチェックしたとします。9:12 に新しい対話式ジョブ NEWJOB が開始し、そのジョブでは何も実行されません。9:20 にシステムはもう一度非アクティブ・ジョブをチェックしますが、NEWJOB が非アクティブ状態となっていたのは 10 分間ではなく 8 分間であるため、タイムアウトにはなりません。9:30 にシステムは再度非アクティブ・ジョブをチェックします。ここで NEWJOB はタイムアウトになりますが、非アクティブ状態となってから 18 分間が経過していることとなります。

上述のとおり、QINACTMSGQ はシステムが非アクティブ・セッションに対して実行する処置を決定します。このシステム値には、以下の 3 つのオプションがあります。

- \*ENDJOB - 非アクティブ・ジョブを終了します。
- \*DSCJOB - 非アクティブ・ジョブを切断します。

**注:** QPADEV\* 装置を使用するジョブを切断することはできません。切断された装置にもう一度サインオンするためには、同じ装置記述とユーザー・プロファイルを使ってシステムにサインオンする必要があります。ただし、システムによって割り当てられた仮想装置では、同じ QPADEV\* 装置にもう一度サインオンするということはほとんどないため、システムはこれらの装置からのジョブの切断を許可しません。ジョブ切断オプションを使用するには、装置名を割り当てます。この資料で前述した『装置名の割り当て』のトピックを参照してください。

- メッセージ待ち行列とライブラリー。メッセージは、非アクティブ状態の各セッションごとに指定されたメッセージ待ち行列に送信されます。このオプションを使用して、プログラムを作成し、ユーザー独自に非アクティブ・ジョブを管理することができます。これは、\*ENDJOB または \*DSCJOB が提供するシステム・ワイドな動作よりも細分度の高い動作が必要な場合に、非常に便利です。例えば、非アクティブになっている JOHN のジョブは常に終了させ、MARY のジョブは切断し、BOSS のジョブは常にアクティブのままにしておきたい、という場合があります。

非アクティブ装置のタイムアウトは、QShell または PASE 端末セッションには適用されません。これらのセッションは表示待ち状態になることは絶対にないため、非アクティブとみなされることはなく、タイムアウトになることはありません。

## 装置回復の管理

ワークステーションで装置エラーが起こったときに行われるアクション (QDEVRCYACN) と装置回復処置のジョブ属性によって、対話式ジョブで装置エラーが発生したときに取られる処置が決まります。以下のオプションがあります。

- \*MSG - 機密漏れの恐れがあるため、このオプションの使用はお勧めできません。
- \*DSCMSG - 可能な場合、このジョブは切断されます (QPADEV\* 装置を持つジョブの場合、ジョブは終了します)。装置に再度サインオンする際に、古いジョブに再接続するか、古いジョブを終了するかを選択できます。古いジョブに再接続すると、実行中のアプリケーションにエラー・メッセージが送信されます。
- \*DSCENDRQS - 可能な場合、このジョブは切断されます (QPADEV\* 装置を持つジョブの場合、ジョブは終了します)。装置に再度サインオンする際に、古いジョブに再接続するか、古いジョブを終了するかを選択できます。古いジョブに再接続すると、キャンセル要求機能が実行されます。
- \*ENDJOB - ジョブは終了し、ジョブ・ログが生成されます。
- \*ENDJOBNOLOG - ジョブは終了し、ジョブ・ログは生成されません。このオプションを選択すると、ジョブ・ログの作成によるシステムのオーバーヘッドが回避されるため、ネットワーク上の問題によって多数のジョブを一度に終了する場合は、特に便利なオプションです。

## 装置記述の IP アドレスの判別

Telnet を使用して iSeries にアクセスする場合は、装置記述のユーザーのリモート IP アドレスを検索することができます。これは、その装置でジョブがアクティブであるときに行うことができます。

これを行うには、装置記述の検索 (QDCRDEVD) API、フォーマット DEVD0600 を使用します。





---

## 参照およびリソース

iSeries<sup>(TM)</sup> Information Center

- 実行管理機能
- OS/400 システム値

マニュアル

- AS/400e シリーズ 実行管理の手引き バージョン 4 
- Job Scheduler for OS/400 



---

## 特記事項

本書の情報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。他社製品への言及および参照は、単に情報提供目的で記載されたものであり、IBM がそれらの製品を推奨するものではありません。

本書に含まれるパフォーマンス・データは、管理環境下で標準の IBM ベンチマークを使用し得られた測定結果と予測に基づくものです。ユーザーが実際に得られるスループットまたはパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、I/O 構成、記憶域構成、および処理されるワークロードなどの考慮事項によって異なります。したがって、個々のユーザーがここで述べる比率と同等のスループットまたはパフォーマンスの向上を得られるという保証はありません。









Printed in Japan